FILE HISTORY

11,936,876

PATENT:	11,936,876
INVENTORS:	Qian XU
	Yunfei ZHENG
	Xiaoan LU
	Peng YIN
	Joel SOLE
TITLE:	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS
FILED:	18 JUN 2021
ISSUED:	19 MAR 2024
COMPILED:	04 AUG 2025

AMAZON-1002
Amazon.com, Inc. v. InterDigital, Inc.
IPR2026-00193

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

Transaction History

Date	Transaction Description
06-18-2021	Information Disclosure Statement (IDS) Filed
06-18-2021	Information Disclosure Statement (IDS) Filed
06-18-2021	PTO-SB-69-Authorize EPO Access to Search Results
06-18-2021	Applicants have given acceptable permission for participating foreign
06-18-2021	Information Disclosure Statement (IDS) Filed
06-18-2021	Information Disclosure Statement (IDS) Filed
06-18-2021	Entity Status Set To Undiscounted (Initial Default Setting or Status Change)
06-18-2021	Initial Exam Team nn
06-30-2021	Filing Receipt
06-30-2021	Corrected Paper
07-01-2021	Application ready for PDX access by participating foreign offices
08-23-2021	Patent Term Adjustment - Ready for Examination
08-23-2021	Additional Application Filing Fees
08-23-2021	Applicant has submitted a new specification to correct Corrected Papers problems
09-01-2021	Application Dispatched from OIPE
09-01-2021	FITF set to NO - revise initial setting
09-02-2021	Application Is Now Complete
09-02-2021	Filing Receipt - Updated
10-15-2021	Case Docketed to Examiner in GAU
10-26-2021	Information Disclosure Statement (IDS) Filed
10-26-2021	Information Disclosure Statement (IDS) Filed
12-09-2021	PG-Pub Issue Notification
07-10-2022	Case Docketed to Examiner in GAU
07-12-2022	Interview Summary - Examiner Initiated - Telephonic
07-18-2022	Information Disclosure Statement considered
07-18-2022	Information Disclosure Statement considered
07-18-2022	Information Disclosure Statement considered
08-16-2022	Non-Final Rejection
08-18-2022	Mail Non-Final Rejection
01-18-2023	Response after Non-Final Action
01-18-2023	Request for Extension of Time - Granted
01-21-2023	Date Forwarded to Examiner
02-07-2023	Final Rejection
02-08-2023	Mail Final Rejection (PTOL - 326)
06-22-2023	Request for Continued Examination (RCE)
06-22-2023	Miscellaneous Incoming Letter
06-22-2023	Workflow - Request for RCE - Begin
06-29-2023	Date Forwarded to Examiner
06-29-2023	Disposal for a RCE - CPA - R129
08-25-2023	Interview Summary - Examiner Initiated - Telephonic
08-25-2023	PARALEGAL OR ELECTRONIC TERMINAL DISCLAIMER APPROVED
08-25-2023	Terminal Disclaimer Filed
08-28-2023	Information Disclosure Statement (IDS) Filed
08-28-2023	Information Disclosure Statement (IDS) Filed
08-28-2023	Information Disclosure Statement (IDS) Filed
08-31-2023	Notice of Allowance Data Verification Completed
09-01-2023	Mail Notice of Allowance
09-06-2023	Pubs Case Remand to TC
09-07-2023	Corrected Notice of Allowability
09-07-2023	Information Disclosure Statement considered
09-11-2023	Mailing Corrected Notice of Allowability

09-13-2023	Pubs Case Remand to TC
09-14-2023	Printer Rush- No mailing
09-27-2023	Interview Summary - Applicant Initiated - Telephonic
09-28-2023	Corrected Notice of Allowability
10-02-2023	Mailing Corrected Notice of Allowability
12-01-2023	Information Disclosure Statement (IDS) Filed
12-01-2023	Request for Continued Examination (RCE)
12-01-2023	Information Disclosure Statement (IDS) Filed
12-01-2023	Information Disclosure Statement (IDS) Filed
12-01-2023	Workflow - Request for RCE - Begin
12-06-2023	Information Disclosure Statement considered
12-06-2023	Information Disclosure Statement considered
12-06-2023	Information Disclosure Statement considered
12-06-2023	Disposal for a RCE - CPA - R129
12-17-2023	Notice of Allowance Data Verification Completed
12-18-2023	Mail Notice of Allowance
02-09-2024	Issue Fee Payment Verified
02-09-2024	Issue Fee Payment Received
02-12-2024	Dispatch to FDC
02-12-2024	Application Is Considered Ready for Issue
02-28-2024	Issue Notification Mailed
03-19-2024	Mail Patent eGrant Notification
03-19-2024	Recordation of Patent Grant Mailed
03-19-2024	Patent eGrant Notification
03-19-2024	Recordation of Patent eGrant
03-19-2024	Patent Issue Date Used in PTA Calculation

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

CPC						
Symbol					**Type**	**Version**
H04N	/	19	/	136	F	2014-11-01
H04N	/	19	/	176	I	2014-11-01
H04N	/	19	/	70	I	2014-11-01
H04N	/	19	/	147	I	2014-11-01
H04N	/	19	/	196	I	2014-11-01
H04N	/	19	/	96	I	2014-11-01
H04N	/	19	/	11	I	2014-11-01
H04N	/	19	/	463	I	2014-11-01
H04N	/	19	/	157	I	2014-11-01
H04N	/	19	/	159	I	2014-11-01
H04N	/	19	/	46	I	2014-11-01
H04N	/	19	/	593	I	2014-11-01

CPC Combination Sets								
Symbol					**Type**	**Set**	**Ranking**	**Version**
	/		/					

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	25 August 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20230825

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

INTERNATIONAL CLASSIFICATION				
CLAIMED				
H04N19/136	/	19	/	136
H04N19/176	/	19	/	176
H04N19/70	/	19	/	70
H04N19/96	/	19	/	96
H04N19/11	/	19	/	11
H04N19/157	/	19	/	157
H04N19/159	/	19	/	159
H04N19/46	/	19	/	46
H04N19/593	/	19	/	593

NON-CLAIMED				
H04N19/147	/	19	/	147
H04N19/196	/	19	/	196
H04N19/463	/	19	/	463

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**
375	240.02

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	25 August 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20230825

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

☑ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIMS

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		10		19		28	16	37						
	2		11	1	20	8	29	17	38						
	3		12	2	21	9	30	18	39						
	4		13	3	22	10	31		40						
	5		14		23	11	32		41						
	6		15	4	24	12	33		42						
	7		16	5	25	13	34		43						
	8		17	6	26	14	35								
	9		18	7	27	15	36								

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	25 August 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20230825

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

CPC						
Symbol					**Type**	**Version**
H04N	/	19	/	136	F	2014-11-01
H04N	/	19	/	176	I	2014-11-01
H04N	/	19	/	70	I	2014-11-01
H04N	/	19	/	147	I	2014-11-01
H04N	/	19	/	196	I	2014-11-01
H04N	/	19	/	96	I	2014-11-01
H04N	/	19	/	11	I	2014-11-01
H04N	/	19	/	463	I	2014-11-01
H04N	/	19	/	157	I	2014-11-01
H04N	/	19	/	159	I	2014-11-01
H04N	/	19	/	46	I	2014-11-01
H04N	/	19	/	593	I	2014-11-01

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
	/		/					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	06 December 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20231206

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

INTERNATIONAL CLASSIFICATION				
CLAIMED				
H04N19/136	/	19	/	136
H04N19/176	/	19	/	176
H04N19/70	/	19	/	70
H04N19/96	/	19	/	96
H04N19/11	/	19	/	11
H04N19/157	/	19	/	157
H04N19/159	/	19	/	159
H04N19/46	/	19	/	46
H04N19/593	/	19	/	593

NON-CLAIMED				
H04N19/147	/	19	/	147
H04N19/196	/	19	/	196
H04N19/463	/	19	/	463

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**
375	240.02

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	06 December 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20231206

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

☑ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIMS

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		10		19		28	16	37						
	2		11	1	20	8	29	17	38						
	3		12	2	21	9	30	18	39						
	4		13	3	22	10	31		40						
	5		14		23	11	32		41						
	6		15	4	24	12	33		42						
	7		16	5	25	13	34		43						
	8		17	6	26	14	35								
	9		18	7	27	15	36								

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	18	
/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	06 December 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	20	7A

U.S. Patent and Trademark Office — Part of Paper No.: 20231206

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☐ Claims renumbered in the same order as presented by applicant						☐ CPA	☐ T.D.		☐ R.1.47	
CLAIM		**DATE**								
Final	Original	08/16/2022								
	1	-								
	2	-								
	3	-								
	4	-								
	5	-								
	6	-								
	7	-								
	8	-								
	9	-								
	10	-								
	11	-								
	12	-								
	13	-								
	14	-								
	15	-								
	16	-								
	17	-								
	18	-								
	19	-								
	20	✓								
	21	✓								
	22	✓								
	23	✓								
	24	✓								
	25	✓								
	26	✓								
	27	✓								
	28	✓								
	29	✓								
	30	✓								
	31	✓								
	32	✓								
	33	✓								
	34	✓								
	35	✓								
	36	✓								
	37	✓								
	38	✓								
	39	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☐ Claims renumbered in the same order as presented by applicant				☐ CPA		☐ T.D.		☐ R.1.47		
CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023							
	1	-	-							
	2	-	-							
	3	-	-							
	4	-	-							
	5	-	-							
	6	-	-							
	7	-	-							
	8	-	-							
	9	-	-							
	10	-	-							
	11	-	-							
	12	-	-							
	13	-	-							
	14	-	-							
	15	-	-							
	16	-	-							
	17	-	-							
	18	-	-							
	19	-	-							
	20	✓	✓							
	21	✓	✓							
	22	✓	✓							
	23	✓	-							
	24	✓	✓							
	25	✓	✓							
	26	✓	✓							
	27	✓	✓							
	28	✓	-							
	29	✓	✓							
	30	✓	✓							
	31	✓	✓							
	32	✓	✓							
	33	✓	✓							
	34	✓	✓							
	35	✓	✓							
	36	✓	✓							
	37	✓	✓							
	38	✓	✓							
	39	✓	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☑ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023						
	1	-	-	-						
	2	-	-	-						
	3	-	-	-						
	4	-	-	-						
	5	-	-	-						
	6	-	-	-						
	7	-	-	-						
	8	-	-	-						
	9	-	-	-						
	10	-	-	-						
	11	-	-	-						
	12	-	-	-						
	13	-	-	-						
	14	-	-	-						
	15	-	-	-						
	16	-	-	-						
	17	-	-	-						
	18	-	-	-						
	19	-	-	-						
1	20	✓	✓	=						
2	21	✓	✓	=						
3	22	✓	✓	=						
	23	✓	-	-						
4	24	✓	✓	=						
5	25	✓	✓	=						
6	26	✓	✓	=						
7	27	✓	✓	=						
	28	✓	-	-						
8	29	✓	✓	=						
9	30	✓	✓	=						
10	31	✓	✓	=						
11	32	✓	✓	=						
12	33	✓	✓	=						
13	34	✓	✓	=						
14	35	✓	✓	=						
15	36	✓	✓	=						
16	37	✓	✓	=						
17	38	✓	✓	=						
18	39	✓	✓	=						
	40			-						
	41			-						
	42			-						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023						
	43			-						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☑ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023						
	1	-	-	-						
	2	-	-	-						
	3	-	-	-						
	4	-	-	-						
	5	-	-	-						
	6	-	-	-						
	7	-	-	-						
	8	-	-	-						
	9	-	-	-						
	10	-	-	-						
	11	-	-	-						
	12	-	-	-						
	13	-	-	-						
	14	-	-	-						
	15	-	-	-						
	16	-	-	-						
	17	-	-	-						
	18	-	-	-						
	19	-	-	-						
1	20	✓	✓	=						
2	21	✓	✓	=						
3	22	✓	✓	=						
	23	✓	-	-						
4	24	✓	✓	=						
5	25	✓	✓	=						
6	26	✓	✓	=						
7	27	✓	✓	=						
	28	✓	-	-						
8	29	✓	✓	=						
9	30	✓	✓	=						
10	31	✓	✓	=						
11	32	✓	✓	=						
12	33	✓	✓	=						
13	34	✓	✓	=						
14	35	✓	✓	=						
15	36	✓	✓	=						
16	37	✓	✓	=						
17	38	✓	✓	=						
18	39	✓	✓	=						
	40			-						
	41			-						
	42			-						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023						
	43			-						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☑ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023	12/06/2023					
	1	-	-	-	-					
	2	-	-	-	-					
	3	-	-	-	-					
	4	-	-	-	-					
	5	-	-	-	-					
	6	-	-	-	-					
	7	-	-	-	-					
	8	-	-	-	-					
	9	-	-	-	-					
	10	-	-	-	-					
	11	-	-	-	-					
	12	-	-	-	-					
	13	-	-	-	-					
	14	-	-	-	-					
	15	-	-	-	-					
	16	-	-	-	-					
	17	-	-	-	-					
	18	-	-	-	-					
	19	-	-	-	-					
1	20	✓	✓	=	=					
2	21	✓	✓	=	=					
3	22	✓	✓	=	=					
	23	✓	-	-	-					
4	24	✓	✓	=	=					
5	25	✓	✓	=	=					
6	26	✓	✓	=	=					
7	27	✓	✓	=	=					
	28	✓	-	-	-					
8	29	✓	✓	=	=					
9	30	✓	✓	=	=					
10	31	✓	✓	=	=					
11	32	✓	✓	=	=					
12	33	✓	✓	=	=					
13	34	✓	✓	=	=					
14	35	✓	✓	=	=					
15	36	✓	✓	=	=					
16	37	✓	✓	=	=					
17	38	✓	✓	=	=					
18	39	✓	✓	=	=					
	40			-	-					
	41			-	-					
	42			-	-					

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	**Art Unit**
	Tyler W Sullivan	2487

CLAIM		DATE								
Final	Original	08/16/2022	02/07/2023	08/25/2023	12/06/2023					
	43			-	-					

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
PE2E SEARCH Notes including Inventor and Assignee Search - See Attached	08/16/2022	TWS

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
PE2E SEARCH Notes including Inventor and Assignee Search	08/16/2022	TWS
PE2E SEARCH Notes Updated - See Attached	02/07/2023	TWS

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
PE2E SEARCH Notes including Inventor and Assignee Search	08/16/2022	TWS
PE2E SEARCH Notes Updated	02/07/2023	TWS
PE2E SEARCH Notes Updated - See Attached	08/25/2023	TWS
Consulted SPE David Czekaj	08/25/2023	TWS

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	PE2E SEARCH Notes - See Attached	08/25/2023	TWS

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
PE2E SEARCH Notes including Inventor and Assignee Search	08/16/2022	TWS
PE2E SEARCH Notes Updated	02/07/2023	TWS
PE2E SEARCH Notes Updated	08/25/2023	TWS
Consulted SPE David Czekaj	08/25/2023	TWS
PE2E SEARCH Notes Updated - See Attached	09/07/2023	TWS

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	PE2E SEARCH Notes	08/25/2023	TWS

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

CPC - Searched*		
Symbol	Date	Examiner

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
PE2E SEARCH Notes including Inventor and Assignee Search	08/16/2022	TWS
PE2E SEARCH Notes Updated	02/07/2023	TWS
PE2E SEARCH Notes Updated	08/25/2023	TWS
Consulted SPE David Czekaj	08/25/2023	TWS
PE2E SEARCH Notes Updated	09/07/2023	TWS
PE2E SEARCH Notes Updated - See Attached	12/06/2023	TWS

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/351,649	Zheng et al.
	Examiner	Art Unit
	Tyler W Sullivan	2487

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	PE2E SEARCH Notes	08/25/2023	TWS
	PE2E SEARCH Notes Updated - See Attached	12/06/2023	TWS

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	1	13/381538.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/19 04:11 PM
L3	56273	Thomson.AS. OR (Thomson WITH licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:34 AM
L4	412	(Zheng WITH Yunfei).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:37 AM
L5	20	(Abbas WITH Adeel).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L6	481	(Joel WITH Sole).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L7	1842	(Peng WITH Yin).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L8	344	(Xiaoan WITH Lu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L9	992	(Qian WITH Xu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L10	59046	L3 OR L4 OR L5 OR L6 OR L7 OR L8 OR L9	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L11	1	L10 AND ((intra WITH predict$4).CLM. SAME ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L12	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L13	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:48 AM
L14	526	((intra WITH predict$4)	(US-PGPUB; USPAT;	OR	ON	ON	2014/09/22

		SAME ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))))	USOCR; FPRS; EPO; JPO; DERWENT)				08:50 AM
L15	16	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((coding OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:55 AM
L16	17	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L17	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L18	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super OR "32" OR "64") NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:14 AM
L19	7540	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) OR ((coding OR predict$4) WITH unit)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:34 AM
L20	4104	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM

		OR macroblock OR (macro NEAR3 block))) OR ((coding AND predict$4) WITH unit)))					
L21	1154	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM
L22	5	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:36 AM
L23	56273	Thomson.AS. OR (Thomson WITH Licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:37 AM
L24	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:38 AM
L25	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND ((quad WITH tree) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:39 AM
L26	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:41 AM

		partit$6 OR struct$4))					
L27	45	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME tabl$4 OR (most WITH probab$4 WITH mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:44 AM
L28	22	("8467450"|"20080123977").PN. OR (WO-2010039731-$ | JP-2004128749-$ | JP-2009118233-$ | WO-2008027192-$ | WO-2007034918-$ | WO-2008088140-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 10:37 AM
L29	75	L26 AND ((H04N19/00278 OR H04N19/00042 OR H04N19/00781 OR H04N19/00884 OR H04N19/00763 OR H04N19/00545 OR H04N19/00969 OR H04N19/00024 OR H04N19/00018 OR H04N7/26244 OR H04N19/00733 OR H04N7/26026 OR G06T9/004).CPC.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:02 AM
L30	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L31	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM

		(divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))					
L32	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L33	18	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header OR syntax))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:32 AM
L34	5	("20090046781" | "20060251330").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:38 AM
L35	17	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:19 PM

		OR syntax) SAME (encod$4 AND decod$4))					
L36	30	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:31 PM
L37	1	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree)) SAME adapt$4) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:33 PM
L38	38	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:34 PM
L39	1	12/827195.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 08:07 AM
L40	6	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:07 PM

L41	34	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:09 PM
L42	4	WO-2009051719-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:04 AM
L44	3	(CN-101207810-$ | KR-20090065401-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:22 AM
L45	554219	(large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L46	244625	(large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L47	133	L46 AND ((MPM OR (most NEAR3 probab$4)) WITH intra)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L48	6	"5060285".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 09:37 AM
L49	5	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:02 AM
L50	8	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:16 AM
L51	870	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM
L52	307	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4))) SAME (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM

		OR mb OR macroblock)					
L53	327	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L54	255	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L55	162	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) WITH (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:07 AM
L56	250	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:11 AM
L57	21	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L58	10	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L59	10	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:21 AM
L60	25	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:22 AM

		split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"					
L61	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:23 AM
L62	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4 OR (top NEAR2 down) OR (bottom NEAR2 up))) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L63	30	intra SAME (((quad WITH tree) OR tree OR (top NEAR2 down) OR (bottom NEAR2 up)) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L64	2	12/310090.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L65	4	(WO-2008027192-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L66	7	("8885727" | "20110243244" | "20110096829").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:20 PM
L67	1160	merg$4 SAME ((intra OR spatial) WITH (mode OR type))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L68	124	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) AND (intra WITH DC)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L69	16	L68 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L70	674	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM

L71	92	L70 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM
L72	396	merg$4 SAME (intra WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L73	81	L72 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L74	4	WO-2009028922-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:46 AM
L75	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L76	2	L75 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L77	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR super OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:51 AM
L78	2	"20030099292".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/20 01:13 PM
L79	26448	(split$4 AND (merg$4 OR group$4)) WITH (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L80	8139	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L81	1826	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:42 AM

		decision) SAME2 (block OR macroblock OR MB)					
L82	976	L81 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:53 AM
L83	631	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:56 AM
L84	976	L82 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L85	222	L83 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L86	33	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) WITH (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:03 PM
L87	0	L86 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L88	0	L86 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L89	0	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) SAME (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L90	491	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:32 PM

		flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))					
L91	606	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:33 PM
L92	224	L91 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:37 PM
L93	434	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:46 PM
L94	152	L93 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L95	266	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L96	48	L95 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:49 PM

L97	378	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:13 PM
L98	72	L97 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:20 PM
L99	328	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:24 PM
L100	2	"20030202602".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 03:05 PM
L102	3	("20090110070").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 09:59 AM
L103	7	("20030099292" | "20030206594" | "20120106629").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 05:29 PM
L104	85	DC SAME (default OR intial$4) SAME (intra WITH predict$4) SAME mode	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:58 AM
L105	7	L104 AND (large OR super) WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:59 AM
L106	55	L104 AND (large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM

L107	8	L104 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM
L108	139	DC SAME (intra WITH predict$4) SAME mode SAME2 (default OR intial$4)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:01 AM
L109	23	L108 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:19 AM
L110	74	DC SAME (intra WITH predict$4) SAME mode SAME2 ((default OR intial$4) WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:20 AM
L112	5	(JP-2007503784-$ | WO-2004104930-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/12/14 09:08 AM
L113	4248	(spatial WITH intra WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:38 PM
L114	239	(spatial WITH intra WITH mode) SAME table	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L115	1	15/806893.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L116	222	(spatial WITH intra WITH mode) SAME table SAME (partition OR prediction)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L117	104	L116 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L118	98	L116 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L119	1008	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:42 PM
L120	162	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 (save OR store) AND((block OR unit)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:53 PM

		SAME (partition$4 OR merg$4 OR split$4))					
L121	14	L120 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:54 PM
L122	112	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 SAME2 (save OR store) AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:58 PM
L123	4	L122 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L124	4	L123 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L125	41	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4)) AND (DC WITH default)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L126	1	L125 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L127	2	L125 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:46 PM
L128	1171	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 (transmit$4 OR receiv$4 OR send OR sent) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:56 PM
L129	46	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:58 PM
L130	15	L129 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:59 PM

L131	65	(((spatial OR intra) NEAR3 mode) SAME table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:11 PM
L132	24	L131 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L133	21	L131 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L134	19	("100208827"|"20030202602"|"20060109913"|"20060251330"|"20070237228"|"20080123947"|"20080123977"|"20080267291"|"20090022221"|"20090046781"|"20090060345"|"20090154567"|"20100086029"|"20100208827"|"20110286520"|"20110292999"|"5060285"|"7302006"|"8467450"|"9288494").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2018/04/01 06:44 PM
L135	19	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO)	OR	ON	ON	2018/04/01 10:58 PM
L136	27	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 10:58 PM
L137	16	(JP-2009118233-$ | CN-101394565-$ | JP-2005519543-$ | JP-2007528675-$ | JP-08280020-$ | JP-H08280020-$ | WO-03075579-$ | KR-	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM

		20050026318-$ | KR-1020050026318-$ | WO-2009028922-$).DID.					
L138	26	(CN-101467461-$ | JP-2007503784-$ | WO-2004104930-$ | CN-101001382-$ | CN-101047860-$ | CN-101340581-$ | CN-1722836-$ | WO-2006004331-$ | CN-101150719-$ | CN-101273641-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM
L139	4	("20040028282"|"20060013299"|"20100086032"|"20120106629").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/03 02:46 PM
L140	9	(CN-1513268-$ | CN-1719905-$ | KR-102109244-$ | WO-2008004769-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 02:57 PM
L141	3	(WO-2010001862-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L142	2	("20090222177").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L143	2	(WO-2011002504-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:06 PM
L153	1	"100208827".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/01/27 09:42 PM
L154	1	("6084908").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:32 AM
L155	3	(WO-2007081908-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/05 06:33 AM
L156	1	("20100208827").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:36 AM
L161	754	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM

		OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L162	64	L161 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM
L163	21	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L164	26	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L165	44	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:25 PM
L166	3	L165 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:26 PM
L167	18	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:27 PM
L168	130	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:25 PM

		SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3)) SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L169	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L170	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L171	6	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:28 PM
L172	137	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:36 PM
L173	6	L172 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM
L174	171	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM

		OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME2 (MPM OR (most NEAR3 probab$3))					
L175	804	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) AND intra WITH (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:42 PM
L176	37	L175 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:43 PM
L177	53	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L178	8	L172 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L179	8	L174 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L180	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:45 PM
L181	0	"S158" AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 04:38 PM
L182	1	17/351649.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2022/07/16 08:03 AM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L183	4	(EP-2293567-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/16 08:24 AM
L184	29	("10455248" OR "20030202602" OR "20040028282" OR "20060013299" OR "20060109913" OR "20060251330" OR "20070237228" OR "20080123947" OR "20080123977" OR "20080267291" OR "20090022221" OR "20090046781" OR "20090060345" OR "20090154567" OR "20090232206" OR "20100086029" OR "20100086032" OR "20100208827" OR "20110286520" OR "20110292999" OR "20120106629" OR "5060285" OR "5640420" OR "6084908" OR "7302006" OR "8467450" OR "8483285" OR "9113168" OR "9288494").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/07/18 07:18 AM
L185	117	(CN-1513268-$ | CN-1719905-$ | CN-1722836-$ | CN-101001382-$ | CN-	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/07/18 09:38 PM

		101047860-$ | CN-101150719-$ | CN-101207810-$ | CN-101273641-$ | CN-101340581-$ | CN-101394565-$ | CN-101467461-$ | JP-H08280020-$ | JP-8280020$- \| JP-H8280020-$ | JP-2004128749-$ | JP-2005519543-$ | JP-2007503784-$ | JP-2007528675-$ | JP-2009118233-$ | KR-1020050026318-$ | KR-20050026318-$ | KR-1020090065401-$ | KR-20090065401-$ | KR-102109244-$ | KR-2109244-$ | WO-03075579-$ | WO-2003075579-$ | WO-2004104930-$ | WO-2005093661-$ | WO-2006004331-$ | WO-2007034918-$ | WO-2007081908-$ | WO-2008004769-$ | WO-2008027192-$ | WO-2008088140-$ | WO-2009028922-$ | WO-2009051719-$ | WO-2010039731-$).DID.	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L186	37	(Artificial Intelligence) More Like: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR KR-102194679-B1 OR US-20120106629-A1 OR KR-20200144159-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR JP-6193432-B2 OR WO-2011019383-A2 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/19 06:12 AM

		EP-2465265-A2).did.					
L188	2	15/806,893.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/08/12 02:51 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N144	44	(((large OR super OR jumbo OR mega OR giant) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N145	386	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N146	350	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:33 PM

		spatial$3) NEAR3 predict$3)).CLM.					
N147	45	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM. SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:34 PM
N148	5	N147 AND @ay<="2009"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N149	9	N147 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N150	47	N145 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N151	10	N150 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:41 PM
N152	14	N150 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:47 PM
N157	403	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N158	48	N157 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N159	14	N158 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N160	10	N158 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:39 AM
N182	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM

		OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N183	48	N182 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N184	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N185	48	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N186	1	N185 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N187	52	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:30 PM
N188	1	N187 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:31 PM

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	1	13/381538.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/19 04:11 PM
L3	56273	Thomson.AS. OR (Thomson WITH licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:34 AM
L4	412	(Zheng WITH Yunfei).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:37 AM
L5	20	(Abbas WITH Adeel).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L6	481	(Joel WITH Sole).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L7	1842	(Peng WITH Yin).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L8	344	(Xiaoan WITH Lu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L9	992	(Qian WITH Xu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L10	59046	L3 OR L4 OR L5 OR L6 OR L7 OR L8 OR L9	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L11	1	L10 AND ((intra WITH predict$4).CLM. SAME ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L12	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L13	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:48 AM
L14	526	((intra WITH predict$4)	(US-PGPUB; USPAT;	OR	ON	ON	2014/09/22

		SAME ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))))	USOCR; FPRS; EPO; JPO; DERWENT)				08:50 AM
L15	16	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((coding OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:55 AM
L16	17	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L17	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L18	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super OR "32" OR "64") NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:14 AM
L19	7540	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) OR ((coding OR predict$4) WITH unit)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:34 AM
L20	4104	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM

		OR macroblock OR (macro NEAR3 block))) OR ((coding AND predict$4) WITH unit)))					
L21	1154	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM
L22	5	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:36 AM
L23	56273	Thomson.AS. OR (Thomson WITH Licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:37 AM
L24	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:38 AM
L25	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND ((quad WITH tree) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:39 AM
L26	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:41 AM

		partit$6 OR struct$4))					
L27	45	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME tabl$4 OR (most WITH probab$4 WITH mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:44 AM
L28	22	("8467450"|"20080123977").PN. OR (WO-2010039731-$ | JP-2004128749-$ | JP-2009118233-$ | WO-2008027192-$ | WO-2007034918-$ | WO-2008088140-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 10:37 AM
L29	75	L26 AND ((H04N19/00278 OR H04N19/00042 OR H04N19/00781 OR H04N19/00884 OR H04N19/00763 OR H04N19/00545 OR H04N19/00969 OR H04N19/00024 OR H04N19/00018 OR H04N7/26244 OR H04N19/00733 OR H04N7/26026 OR G06T9/004).CPC.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:02 AM
L30	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L31	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM

		(divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))					
L32	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L33	18	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header OR syntax))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:32 AM
L34	5	("20090046781" | "20060251330").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:38 AM
L35	17	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:19 PM

		OR syntax) SAME (encod$4 AND decod$4))					
L36	30	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:31 PM
L37	1	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree)) SAME adapt$4) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:33 PM
L38	38	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:34 PM
L39	1	12/827195.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 08:07 AM
L40	6	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:07 PM

L41	34	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:09 PM
L42	4	WO-2009051719-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:04 AM
L44	3	(CN-101207810-$ | KR-20090065401-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:22 AM
L45	554219	(large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L46	244625	(large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L47	133	L46 AND ((MPM OR (most NEAR3 probab$4)) WITH intra)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L48	6	"5060285".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 09:37 AM
L49	5	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:02 AM
L50	8	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:16 AM
L51	870	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM
L52	307	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4))) SAME (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM

		OR mb OR macroblock)					
L53	327	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L54	255	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L55	162	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) WITH (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:07 AM
L56	250	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:11 AM
L57	21	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L58	10	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L59	10	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:21 AM
L60	25	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:22 AM

		split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"					
L61	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:23 AM
L62	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4 OR (top NEAR2 down) OR (bottom NEAR2 up))) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L63	30	intra SAME (((quad WITH tree) OR tree OR (top NEAR2 down) OR (bottom NEAR2 up)) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L64	2	12/310090.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L65	4	(WO-2008027192-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L66	7	("8885727" | "20110243244" | "20110096829").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:20 PM
L67	1160	merg$4 SAME ((intra OR spatial) WITH (mode OR type))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L68	124	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) AND (intra WITH DC)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L69	16	L68 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L70	674	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM

L71	92	L70 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM
L72	396	merg$4 SAME (intra WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L73	81	L72 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L74	4	WO-2009028922-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:46 AM
L75	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L76	2	L75 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L77	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR super OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:51 AM
L78	2	"20030099292".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/20 01:13 PM
L79	26448	(split$4 AND (merg$4 OR group$4)) WITH (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L80	8139	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L81	1826	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:42 AM

		decision) SAME2 (block OR macroblock OR MB)					
L82	976	L81 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:53 AM
L83	631	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:56 AM
L84	976	L82 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L85	222	L83 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L86	33	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) WITH (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:03 PM
L87	0	L86 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L88	0	L86 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L89	0	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) SAME (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L90	491	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:32 PM

		flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))					
L91	606	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:33 PM
L92	224	L91 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:37 PM
L93	434	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:46 PM
L94	152	L93 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L95	266	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L96	48	L95 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:49 PM

L97	378	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:13 PM
L98	72	L97 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:20 PM
L99	328	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:24 PM
L100	2	"20030202602".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 03:05 PM
L102	3	("20090110070").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 09:59 AM
L103	7	("20030099292" | "20030206594" | "20120106629").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 05:29 PM
L104	85	DC SAME (default OR intial$4) SAME (intra WITH predict$4) SAME mode	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:58 AM
L105	7	L104 AND (large OR super) WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:59 AM
L106	55	L104 AND (large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM

L107	8	L104 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM
L108	139	DC SAME (intra WITH predict$4) SAME mode SAME2 (default OR intial$4)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:01 AM
L109	23	L108 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:19 AM
L110	74	DC SAME (intra WITH predict$4) SAME mode SAME2 ((default OR intial$4) WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:20 AM
L112	5	(JP-2007503784-$ | WO-2004104930-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/12/14 09:08 AM
L113	4248	(spatial WITH intra WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:38 PM
L114	239	(spatial WITH intra WITH mode) SAME table	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L115	1	15/806893.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L116	222	(spatial WITH intra WITH mode) SAME table SAME (partition OR prediction)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L117	104	L116 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L118	98	L116 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L119	1008	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:42 PM
L120	162	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 (save OR store) AND((block OR unit)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:53 PM

		SAME (partition$4 OR merg$4 OR split$4))					
L121	14	L120 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:54 PM
L122	112	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 SAME2 (save OR store) AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:58 PM
L123	4	L122 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L124	4	L123 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L125	41	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4)) AND (DC WITH default)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L126	1	L125 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L127	2	L125 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:46 PM
L128	1171	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 (transmit$4 OR receiv$4 OR send OR sent) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:56 PM
L129	46	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:58 PM
L130	15	L129 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:59 PM

L131	65	(((spatial OR intra) NEAR3 mode) SAME table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:11 PM
L132	24	L131 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L133	21	L131 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L134	19	("100208827"|"20030202602"|"20060109913"|"20060251330"|"20070237228"|"20080123947"|"20080123977"|"20080267291"|"20090022221"|"20090046781"|"20090060345"|"20090154567"|"20100086029"|"20100208827"|"20110286520"|"20110292999"|"5060285"|"7302006"|"8467450"|"9288494").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2018/04/01 06:44 PM
L135	19	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO)	OR	ON	ON	2018/04/01 10:58 PM
L136	27	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 10:58 PM
L137	16	(JP-2009118233-$ | CN-101394565-$ | JP-2005519543-$ | JP-2007528675-$ | JP-08280020-$ | JP-H08280020-$ | WO-03075579-$ | KR-	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM

		20050026318-$ | KR-1020050026318-$ | WO-2009028922-$).DID.					
L138	26	(CN-101467461-$ | JP-2007503784-$ | WO-2004104930-$ | CN-101001382-$ | CN-101047860-$ | CN-101340581-$ | CN-1722836-$ | WO-2006004331-$ | CN-101150719-$ | CN-101273641-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM
L139	4	("20040028282"|"20060013299"|"20100086032"|"20120106629").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/03 02:46 PM
L140	9	(CN-1513268-$ | CN-1719905-$ | KR-102109244-$ | WO-2008004769-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 02:57 PM
L141	3	(WO-2010001862-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L142	2	("20090222177").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L143	2	(WO-2011002504-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:06 PM
L153	1	"100208827".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/01/27 09:42 PM
L154	1	("6084908").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:32 AM
L155	3	(WO-2007081908-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/05 06:33 AM
L156	1	("20100208827").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:36 AM
L161	754	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM

		OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L162	64	L161 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM
L163	21	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L164	26	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L165	44	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:25 PM
L166	3	L165 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:26 PM
L167	18	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:27 PM
L168	130	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:25 PM

		SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3)) SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L169	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L170	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L171	6	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:28 PM
L172	137	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:36 PM
L173	6	L172 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM
L174	171	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM

		OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME2 (MPM OR (most NEAR3 probab$3))					
L175	804	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) AND intra WITH (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:42 PM
L176	37	L175 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:43 PM
L177	53	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L178	8	L172 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L179	8	L174 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L180	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:45 PM
L181	0	"S158" AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 04:38 PM
L182	1	17/351649.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2022/07/16 08:03 AM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L183	4	(EP-2293567-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/16 08:24 AM
L184	29	("10455248" OR "20030202602" OR "20040028282" OR "20060013299" OR "20060109913" OR "20060251330" OR "20070237228" OR "20080123947" OR "20080123977" OR "20080267291" OR "20090022221" OR "20090046781" OR "20090060345" OR "20090154567" OR "20090232206" OR "20100086029" OR "20100086032" OR "20100208827" OR "20110286520" OR "20110292999" OR "20120106629" OR "5060285" OR "5640420" OR "6084908" OR "7302006" OR "8467450" OR "8483285" OR "9113168" OR "9288494").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/07/18 07:18 AM
L185	117	(CN-1513268-$ | CN-1719905-$ | CN-1722836-$ | CN-101001382-$ | CN-	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/07/18 09:38 PM

		101047860-$ | CN-101150719-$ | CN-101207810-$ | CN-101273641-$ | CN-101340581-$ | CN-101394565-$ | CN-101467461-$ | JP-H08280020-$ | JP-8280020$- \| JP-H8280020-$ | JP-2004128749-$ | JP-2005519543-$ | JP-2007503784-$ | JP-2007528675-$ | JP-2009118233-$ | KR-1020050026318-$ | KR-20050026318-$ | KR-1020090065401-$ | KR-20090065401-$ | KR-102109244-$ | KR-2109244-$ | WO-03075579-$ | WO-2003075579-$ | WO-2004104930-$ | WO-2005093661-$ | WO-2006004331-$ | WO-2007034918-$ | WO-2007081908-$ | WO-2008004769-$ | WO-2008027192-$ | WO-2008088140-$ | WO-2009028922-$ | WO-2009051719-$ | WO-2010039731-$).DID.	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L186	37	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR KR-102194679-B1 OR US-20120106629-A1 OR KR-20200144159-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR JP-6193432-B2 OR WO-2011019383-A2 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/19 06:12 AM

		EP-2465265-A2).did.					
L188	2	15/806,893.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/08/12 02:51 PM
L189	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR US-20210385460-A1 OR US-11082697-B2 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR JP-6688341-B2 OR JP-2020115669-A OR KR-20120052267-A OR DK-2449782-T3 OR KR-102109244-B1 OR KR-102194679-B1 OR US-20120106629-A1 OR JP-6415472-B2 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR ES-2744316-T3 OR KR-101939016-B1 OR KR-20170016015-A OR HU-E035547-T2 OR JP-5918128-B2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-104853203-B OR CN-102484709-B OR CN-104796717-B OR	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2023/02/06 07:39 AM

		CN-102484709-A OR CN-104853203-A OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104853204-A OR CN-104853202-A OR CN-104822066-A OR JP-20120532489-A5 OR KR-20160052471-A OR KR-101964382-B1 OR KR-101467175-B1 OR JP-2014534734-A).did.					
L190	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:47 AM
L191	78	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:56 AM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N144	44	(((large OR super OR jumbo OR mega OR giant) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N145	386	(((large OR super OR jumbo OR mega OR	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM

		giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N146	350	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:33 PM
N147	45	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM. SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:34 PM
N148	5	N147 AND @ay<="2009"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N149	9	N147 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N150	47	N145 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N151	10	N150 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:41 PM
N152	14	N150 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:47 PM
N157	403	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM

		macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N158	48	N157 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N159	14	N158 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N160	10	N158 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:39 AM
N182	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N183	48	N182 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N184	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N185	48	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N186	1	N185 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N187	52	N184 SAME (split$3	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12

		OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.					09:30 PM
N188	1	N187 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:31 PM

PE2E SEARCH - Search History (Similarity Search)

Ref #	Application Number	Text Selections	CPC Selections
L189	17/351,649	No selections	No selections

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	1	13/381538.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/19 04:11 PM
L3	56273	Thomson.AS. OR (Thomson WITH licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:34 AM
L4	412	(Zheng WITH Yunfei).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:37 AM
L5	20	(Abbas WITH Adeel).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L6	481	(Joel WITH Sole).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L7	1842	(Peng WITH Yin).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L8	344	(Xiaoan WITH Lu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L9	992	(Qian WITH Xu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L10	59046	L3 OR L4 OR L5 OR L6 OR L7 OR L8 OR L9	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L11	1	L10 AND ((intra WITH predict$4).CLM. SAME ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L12	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L13	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:48 AM
L14	526	((intra WITH predict$4)	(US-PGPUB; USPAT;	OR	ON	ON	2014/09/22

		SAME ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))))	USOCR; FPRS; EPO; JPO; DERWENT)				08:50 AM
L15	16	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((coding OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:55 AM
L16	17	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L17	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L18	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super OR "32" OR "64") NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:14 AM
L19	7540	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) OR ((coding OR predict$4) WITH unit)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:34 AM
L20	4104	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM

		OR macroblock OR (macro NEAR3 block))) OR ((coding AND predict$4) WITH unit)))					
L21	1154	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM
L22	5	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:36 AM
L23	56273	Thomson.AS. OR (Thomson WITH Licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:37 AM
L24	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:38 AM
L25	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND ((quad WITH tree) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:39 AM
L26	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:41 AM

		partit$6 OR struct$4))					
L27	45	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME tabl$4 OR (most WITH probab$4 WITH mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:44 AM
L28	22	("8467450"|"20080123977").PN. OR (WO-2010039731-$ | JP-2004128749-$ | JP-2009118233-$ | WO-2008027192-$ | WO-2007034918-$ | WO-2008088140-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 10:37 AM
L29	75	L26 AND ((H04N19/00278 OR H04N19/00042 OR H04N19/00781 OR H04N19/00884 OR H04N19/00763 OR H04N19/00545 OR H04N19/00969 OR H04N19/00024 OR H04N19/00018 OR H04N7/26244 OR H04N19/00733 OR H04N7/26026 OR G06T9/004).CPC.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:02 AM
L30	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L31	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM

		(divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))					
L32	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L33	18	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header OR syntax))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:32 AM
L34	5	("20090046781" | "20060251330").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:38 AM
L35	17	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:19 PM

		OR syntax) SAME (encod$4 AND decod$4))					
L36	30	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:31 PM
L37	1	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree)) SAME adapt$4) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:33 PM
L38	38	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:34 PM
L39	1	12/827195.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 08:07 AM
L40	6	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:07 PM

L41	34	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:09 PM
L42	4	WO-2009051719-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:04 AM
L44	3	(CN-101207810-$ | KR-20090065401-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:22 AM
L45	554219	(large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L46	244625	(large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L47	133	L46 AND ((MPM OR (most NEAR3 probab$4)) WITH intra)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L48	6	"5060285".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 09:37 AM
L49	5	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:02 AM
L50	8	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:16 AM
L51	870	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM
L52	307	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4))) SAME (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM

		OR mb OR macroblock)					
L53	327	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L54	255	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L55	162	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) WITH (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:07 AM
L56	250	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:11 AM
L57	21	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L58	10	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L59	10	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:21 AM
L60	25	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:22 AM

		split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"					
L61	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:23 AM
L62	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4 OR (top NEAR2 down) OR (bottom NEAR2 up))) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L63	30	intra SAME (((quad WITH tree) OR tree OR (top NEAR2 down) OR (bottom NEAR2 up)) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L64	2	12/310090.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L65	4	(WO-2008027192-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L66	7	("8885727" | "20110243244" | "20110096829").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:20 PM
L67	1160	merg$4 SAME ((intra OR spatial) WITH (mode OR type))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L68	124	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) AND (intra WITH DC)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L69	16	L68 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L70	674	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM

L71	92	L70 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM
L72	396	merg$4 SAME (intra WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L73	81	L72 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L74	4	WO-2009028922-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:46 AM
L75	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L76	2	L75 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L77	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR super OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:51 AM
L78	2	"20030099292".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/20 01:13 PM
L79	26448	(split$4 AND (merg$4 OR group$4)) WITH (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L80	8139	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L81	1826	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:42 AM

		decision) SAME2 (block OR macroblock OR MB)					
L82	976	L81 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:53 AM
L83	631	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:56 AM
L84	976	L82 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L85	222	L83 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L86	33	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) WITH (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:03 PM
L87	0	L86 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L88	0	L86 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L89	0	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) SAME (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L90	491	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:32 PM

		flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))					
L91	606	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:33 PM
L92	224	L91 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:37 PM
L93	434	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:46 PM
L94	152	L93 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L95	266	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L96	48	L95 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:49 PM

L97	378	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:13 PM
L98	72	L97 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:20 PM
L99	328	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:24 PM
L100	2	"20030202602".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 03:05 PM
L102	3	("20090110070").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 09:59 AM
L103	7	("20030099292" | "20030206594" | "20120106629").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 05:29 PM
L104	85	DC SAME (default OR intial$4) SAME (intra WITH predict$4) SAME mode	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:58 AM
L105	7	L104 AND (large OR super) WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:59 AM
L106	55	L104 AND (large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM

L107	8	L104 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM
L108	139	DC SAME (intra WITH predict$4) SAME mode SAME2 (default OR intial$4)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:01 AM
L109	23	L108 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:19 AM
L110	74	DC SAME (intra WITH predict$4) SAME mode SAME2 ((default OR intial$4) WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:20 AM
L112	5	(JP-2007503784-$ | WO-2004104930-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/12/14 09:08 AM
L113	4248	(spatial WITH intra WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:38 PM
L114	239	(spatial WITH intra WITH mode) SAME table	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L115	1	15/806893.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L116	222	(spatial WITH intra WITH mode) SAME table SAME (partition OR prediction)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L117	104	L116 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L118	98	L116 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L119	1008	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:42 PM
L120	162	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 (save OR store) AND((block OR unit)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:53 PM

		SAME (partition$4 OR merg$4 OR split$4))					
L121	14	L120 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:54 PM
L122	112	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 SAME2 (save OR store) AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:58 PM
L123	4	L122 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L124	4	L123 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L125	41	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4)) AND (DC WITH default)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L126	1	L125 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L127	2	L125 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:46 PM
L128	1171	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 (transmit$4 OR receiv$4 OR send OR sent) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:56 PM
L129	46	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:58 PM
L130	15	L129 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:59 PM

L131	65	(((spatial OR intra) NEAR3 mode) SAME table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:11 PM
L132	24	L131 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L133	21	L131 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L134	19	("100208827"|"20030202602"|"20060109913"|"20060251330"|"20070237228"|"20080123947"|"20080123977"|"20080267291"|"20090022221"|"20090046781"|"20090060345"|"20090154567"|"20100086029"|"20100208827"|"20110286520"|"20110292999"|"5060285"|"7302006"|"8467450"|"9288494").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2018/04/01 06:44 PM
L135	19	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO)	OR	ON	ON	2018/04/01 10:58 PM
L136	27	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 10:58 PM
L137	16	(JP-2009118233-$ | CN-101394565-$ | JP-2005519543-$ | JP-2007528675-$ | JP-08280020-$ | JP-H08280020-$ | WO-03075579-$ | KR-	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM

		20050026318-$ | KR-1020050026318-$ | WO-2009028922-$).DID.					
L138	26	(CN-101467461-$ | JP-2007503784-$ | WO-2004104930-$ | CN-101001382-$ | CN-101047860-$ | CN-101340581-$ | CN-1722836-$ | WO-2006004331-$ | CN-101150719-$ | CN-101273641-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM
L139	4	("20040028282"|"20060013299"|"20100086032"|"20120106629").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/03 02:46 PM
L140	9	(CN-1513268-$ | CN-1719905-$ | KR-102109244-$ | WO-2008004769-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 02:57 PM
L141	3	(WO-2010001862-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L142	2	("20090222177").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L143	2	(WO-2011002504-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:06 PM
L153	1	"100208827".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/01/27 09:42 PM
L154	1	("6084908").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:32 AM
L155	3	(WO-2007081908-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/05 06:33 AM
L156	1	("20100208827").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:36 AM
L161	754	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM

		OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L162	64	L161 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM
L163	21	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L164	26	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L165	44	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:25 PM
L166	3	L165 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:26 PM
L167	18	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:27 PM
L168	130	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:25 PM

		SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3)) SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L169	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L170	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L171	6	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:28 PM
L172	137	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:36 PM
L173	6	L172 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM
L174	171	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM

		OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME2 (MPM OR (most NEAR3 probab$3))					
L175	804	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) AND intra WITH (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:42 PM
L176	37	L175 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:43 PM
L177	53	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L178	8	L172 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L179	8	L174 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L180	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:45 PM
L181	0	"S158" AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 04:38 PM
L182	1	17/351649.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2022/07/16 08:03 AM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L183	4	(EP-2293567-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/16 08:24 AM
L184	29	("10455248" OR "20030202602" OR "20040028282" OR "20060013299" OR "20060109913" OR "20060251330" OR "20070237228" OR "20080123947" OR "20080123977" OR "20080267291" OR "20090022221" OR "20090046781" OR "20090060345" OR "20090154567" OR "20090232206" OR "20100086029" OR "20100086032" OR "20100208827" OR "20110286520" OR "20110292999" OR "20120106629" OR "5060285" OR "5640420" OR "6084908" OR "7302006" OR "8467450" OR "8483285" OR "9113168" OR "9288494").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/07/18 07:18 AM
L185	117	(CN-1513268-$ | CN-1719905-$ | CN-1722836-$ | CN-101001382-$ | CN-	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/07/18 09:38 PM

		101047860-$ | CN-101150719-$ | CN-101207810-$ | CN-101273641-$ | CN-101340581-$ | CN-101394565-$ | CN-101467461-$ | JP-H08280020-$ | JP-8280020$- \| JP-H8280020-$ | JP-2004128749-$ | JP-2005519543-$ | JP-2007503784-$ | JP-2007528675-$ | JP-2009118233-$ | KR-1020050026318-$ | KR-20050026318-$ | KR-1020090065401-$ | KR-20090065401-$ | KR-102109244-$ | KR-2109244-$ | WO-03075579-$ | WO-2003075579-$ | WO-2004104930-$ | WO-2005093661-$ | WO-2006004331-$ | WO-2007034918-$ | WO-2007081908-$ | WO-2008004769-$ | WO-2008027192-$ | WO-2008088140-$ | WO-2009028922-$ | WO-2009051719-$ | WO-2010039731-$).DID.	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L186	37	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR KR-102194679-B1 OR US-20120106629-A1 OR KR-20200144159-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR JP-6193432-B2 OR WO-2011019383-A2 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/19 06:12 AM

		EP-2465265-A2).did.					
L188	2	15/806,893.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/08/12 02:51 PM
L189	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR US-20210385460-A1 OR US-11082697-B2 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR JP-6688341-B2 OR JP-2020115669-A OR KR-20120052267-A OR DK-2449782-T3 OR KR-102109244-B1 OR KR-102194679-B1 OR US-20120106629-A1 OR JP-6415472-B2 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR ES-2744316-T3 OR KR-101939016-B1 OR KR-20170016015-A OR HU-E035547-T2 OR JP-5918128-B2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-104853203-B OR CN-102484709-B OR CN-104796717-B OR	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2023/02/06 07:39 AM

		CN-102484709-A OR CN-104853203-A OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104853204-A OR CN-104853202-A OR CN-104822066-A OR JP-20120532489-A5 OR KR-20160052471-A OR KR-101964382-B1 OR KR-101467175-B1 OR JP-2014534734-A).did.					
L190	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:47 AM
L191	78	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:56 AM
L192	35	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102505806-B1 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR US-20120106629-A1 OR CN-104853207-B OR KR-20230035152-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:28 PM

		OR CN-104796717-B OR CN-104853203-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A).did.					
L193	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR EP-3579558-A1 OR EP-2449782-B1 OR US-20210385460-A1 OR US-11082697-B2 OR KR-102505806-B1 OR EP-2449782-A1 OR WO-2011002504-A1 OR JP-2020115669-A OR KR-20120052267-A OR EP-3288268-A1 OR DK-2449782-T3 OR JP-6688341-B2 OR EP-3288268-B1 OR US-20120106629-A1 OR KR-102109244-B1 OR JP-2023029886-A OR KR-102194679-B1 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR JP-6415472-B2 OR ES-2744316-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR KR-101939016-B1 OR JP-5918128-B2 OR KR-20170016015-A OR HU-E035547-T2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A OR CN-104853203-B OR CN-104796717-B OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104822066-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR CN-104853204-A OR CN-104853202-A OR JP-2021083122-A OR JP-20120532489-A5 OR DK-2887670-T3).did.					
L194	46	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 1 Text Selection Text: (US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR US-20210385460-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR EP-3288268-A1 OR US-11082697-B2 OR ES-2657965-T3 OR ES-2744316-T3 OR JP-6688341-B2 OR HU-E035547-T2 OR KR-102505806-B1 OR KR-102109244-B1 OR US-20220021879-A1 OR KR-102194679-B1 OR CN-104853207-B OR KR-101939016-B1 OR KR-20180004329-A OR JP-20160178645-A5 OR KR-20200049909-A OR EP-3523968-B1 OR JP-2022191450-A OR JP-7180794-B2 OR EP-3468202-A1 OR KR-102487077-B1 OR KR-20230010063-A OR KR-102422711-B1 OR JP-2013048480-A OR JP-2013048481-A OR EP-4090023-A1 OR KR-102062894-B1 OR KR-20200005207-A OR KR-102258057-B1 OR KR-20220017380-A OR WO-2022031018-A1 OR US-9451264-B2 OR US-9491468-B2 OR US-9462281-B2 OR WO-2022108419-A1 OR KR-20220071130-A OR EP-3442231-A2 OR US-20190149828-A1 OR KR-20230070533-A OR KR-20110126567-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR JP-6672226-B2 OR US-20220070448-A1 OR JP-5513584-B2 OR EP-3197161-A1 OR EP-3197160-A1).did.					
L195	13809	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:32 PM
L196	2392	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L197	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L198	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA,	OR	ON	ON	2023/08/23 02:33 PM

			EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)				
L199	45	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:34 PM
L200	175	L196 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:51 PM
L201	123	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:52 PM
L202	46	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR (("32" OR "64") NEAR2 ("x" OR ".times." OR times))) NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA,	OR	ON	ON	2023/08/23 02:56 PM

			VN); EPO; DERWENT; IBM_TDB)				

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N144	44	(((large OR super OR jumbo OR mega OR giant) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N145	386	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N146	350	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:33 PM
N147	45	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM. SAME (split$3 OR partition$3	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:34 PM

		OR divid$3 OR divis$3).CLM.					
N148	5	N147 AND @ay<="2009"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N149	9	N147 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N150	47	N145 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N151	10	N150 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:41 PM
N152	14	N150 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:47 PM
N157	403	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N158	48	N157 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N159	14	N158 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N160	10	N158 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:39 AM
N182	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N183	48	N182 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N184	405	(((large OR super OR	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12

		jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					09:25 PM
N185	48	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N186	1	N185 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N187	52	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:30 PM
N188	1	N187 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:31 PM
N189	116	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:37 PM
N190	62	((((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3) NEAR3 (mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:38 PM
N191	4	((((large OR super OR jumbo OR mega OR giant) WITH ("32" OR "64" OR "128" OR "16"	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:40 PM

		OR "N" OR "2N")) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 (predict$3 OR mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.					

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	1	13/381538.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/19 04:11 PM
L3	56273	Thomson.AS. OR (Thomson WITH licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:34 AM
L4	412	(Zheng WITH Yunfei).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:37 AM
L5	20	(Abbas WITH Adeel).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L6	481	(Joel WITH Sole).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L7	1842	(Peng WITH Yin).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L8	344	(Xiaoan WITH Lu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L9	992	(Qian WITH Xu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L10	59046	L3 OR L4 OR L5 OR L6 OR L7 OR L8 OR L9	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L11	1	L10 AND ((intra WITH predict$4).CLM. SAME ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L12	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L13	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:48 AM
L14	526	((intra WITH predict$4)	(US-PGPUB; USPAT;	OR	ON	ON	2014/09/22

		SAME ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))))	USOCR; FPRS; EPO; JPO; DERWENT)				08:50 AM
L15	16	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((coding OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:55 AM
L16	17	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L17	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L18	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super OR "32" OR "64") NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:14 AM
L19	7540	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) OR ((coding OR predict$4) WITH unit)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:34 AM
L20	4104	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM

		OR macroblock OR (macro NEAR3 block))) OR ((coding AND predict$4) WITH unit)))					
L21	1154	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM
L22	5	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:36 AM
L23	56273	Thomson.AS. OR (Thomson WITH Licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:37 AM
L24	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:38 AM
L25	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND ((quad WITH tree) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:39 AM
L26	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:41 AM

		partit$6 OR struct$4))					
L27	45	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME tabl$4 OR (most WITH probab$4 WITH mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:44 AM
L28	22	("8467450"|"20080123977").PN. OR (WO-2010039731-$ | JP-2004128749-$ | JP-2009118233-$ | WO-2008027192-$ | WO-2007034918-$ | WO-2008088140-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 10:37 AM
L29	75	L26 AND ((H04N19/00278 OR H04N19/00042 OR H04N19/00781 OR H04N19/00884 OR H04N19/00763 OR H04N19/00545 OR H04N19/00969 OR H04N19/00024 OR H04N19/00018 OR H04N7/26244 OR H04N19/00733 OR H04N7/26026 OR G06T9/004).CPC.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:02 AM
L30	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L31	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM

		(divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))					
L32	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L33	18	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header OR syntax))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:32 AM
L34	5	("20090046781" | "20060251330").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:38 AM
L35	17	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:19 PM

		OR syntax) SAME (encod$4 AND decod$4))					
L36	30	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:31 PM
L37	1	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree)) SAME adapt$4) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:33 PM
L38	38	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:34 PM
L39	1	12/827195.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 08:07 AM
L40	6	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:07 PM

L41	34	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:09 PM
L42	4	WO-2009051719-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:04 AM
L44	3	(CN-101207810-$ | KR-20090065401-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:22 AM
L45	554219	(large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L46	244625	(large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L47	133	L46 AND ((MPM OR (most NEAR3 probab$4)) WITH intra)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L48	6	"5060285".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 09:37 AM
L49	5	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:02 AM
L50	8	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:16 AM
L51	870	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM
L52	307	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4))) SAME (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM

		OR mb OR macroblock)					
L53	327	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L54	255	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L55	162	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) WITH (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:07 AM
L56	250	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:11 AM
L57	21	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L58	10	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L59	10	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:21 AM
L60	25	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:22 AM

		split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"					
L61	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:23 AM
L62	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4 OR (top NEAR2 down) OR (bottom NEAR2 up))) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L63	30	intra SAME (((quad WITH tree) OR tree OR (top NEAR2 down) OR (bottom NEAR2 up)) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L64	2	12/310090.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L65	4	(WO-2008027192-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L66	7	("8885727" | "20110243244" | "20110096829").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:20 PM
L67	1160	merg$4 SAME ((intra OR spatial) WITH (mode OR type))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L68	124	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) AND (intra WITH DC)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L69	16	L68 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L70	674	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM

L71	92	L70 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM
L72	396	merg$4 SAME (intra WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L73	81	L72 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L74	4	WO-2009028922-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:46 AM
L75	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L76	2	L75 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L77	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR super OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:51 AM
L78	2	"20030099292".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/20 01:13 PM
L79	26448	(split$4 AND (merg$4 OR group$4)) WITH (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L80	8139	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L81	1826	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:42 AM

		decision) SAME2 (block OR macroblock OR MB)					
L82	976	L81 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:53 AM
L83	631	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:56 AM
L84	976	L82 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L85	222	L83 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L86	33	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) WITH (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:03 PM
L87	0	L86 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L88	0	L86 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L89	0	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) SAME (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L90	491	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:32 PM

		flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))					
L91	606	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:33 PM
L92	224	L91 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:37 PM
L93	434	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:46 PM
L94	152	L93 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L95	266	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L96	48	L95 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:49 PM

L97	378	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:13 PM
L98	72	L97 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:20 PM
L99	328	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:24 PM
L100	2	"20030202602".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 03:05 PM
L102	3	("20090110070").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 09:59 AM
L103	7	("20030099292" | "20030206594" | "20120106629").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 05:29 PM
L104	85	DC SAME (default OR intial$4) SAME (intra WITH predict$4) SAME mode	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:58 AM
L105	7	L104 AND (large OR super) WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:59 AM
L106	55	L104 AND (large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM

L107	8	L104 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM
L108	139	DC SAME (intra WITH predict$4) SAME mode SAME2 (default OR intial$4)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:01 AM
L109	23	L108 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:19 AM
L110	74	DC SAME (intra WITH predict$4) SAME mode SAME2 ((default OR intial$4) WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:20 AM
L112	5	(JP-2007503784-$ | WO-2004104930-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/12/14 09:08 AM
L113	4248	(spatial WITH intra WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:38 PM
L114	239	(spatial WITH intra WITH mode) SAME table	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L115	1	15/806893.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L116	222	(spatial WITH intra WITH mode) SAME table SAME (partition OR prediction)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L117	104	L116 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L118	98	L116 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L119	1008	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:42 PM
L120	162	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 (save OR store) AND((block OR unit)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:53 PM

		SAME (partition$4 OR merg$4 OR split$4))					
L121	14	L120 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:54 PM
L122	112	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 SAME2 (save OR store) AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:58 PM
L123	4	L122 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L124	4	L123 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L125	41	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4)) AND (DC WITH default)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L126	1	L125 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L127	2	L125 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:46 PM
L128	1171	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 (transmit$4 OR receiv$4 OR send OR sent) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:56 PM
L129	46	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:58 PM
L130	15	L129 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:59 PM

L131	65	(((spatial OR intra) NEAR3 mode) SAME table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:11 PM
L132	24	L131 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L133	21	L131 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L134	19	("100208827"|"20030202602"|"20060109913"|"20060251330"|"20070237228"|"20080123947"|"20080123977"|"20080267291"|"20090022221"|"20090046781"|"20090060345"|"20090154567"|"20100086029"|"20100208827"|"20110286520"|"20110292999"|"5060285"|"7302006"|"8467450"|"9288494").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2018/04/01 06:44 PM
L135	19	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO)	OR	ON	ON	2018/04/01 10:58 PM
L136	27	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 10:58 PM
L137	16	(JP-2009118233-$ | CN-101394565-$ | JP-2005519543-$ | JP-2007528675-$ | JP-08280020-$ | JP-H08280020-$ | WO-03075579-$ | KR-	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM

		20050026318-$ | KR-1020050026318-$ | WO-2009028922-$).DID.					
L138	26	(CN-101467461-$ | JP-2007503784-$ | WO-2004104930-$ | CN-101001382-$ | CN-101047860-$ | CN-101340581-$ | CN-1722836-$ | WO-2006004331-$ | CN-101150719-$ | CN-101273641-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM
L139	4	("20040028282"|"20060013299"|"20100086032"|"20120106629").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/03 02:46 PM
L140	9	(CN-1513268-$ | CN-1719905-$ | KR-102109244-$ | WO-2008004769-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 02:57 PM
L141	3	(WO-2010001862-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L142	2	("20090222177").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L143	2	(WO-2011002504-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:06 PM
L153	1	"100208827".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/01/27 09:42 PM
L154	1	("6084908").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:32 AM
L155	3	(WO-2007081908-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/05 06:33 AM
L156	1	("20100208827").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:36 AM
L161	754	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM

		OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L162	64	L161 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM
L163	21	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L164	26	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L165	44	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:25 PM
L166	3	L165 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:26 PM
L167	18	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:27 PM
L168	130	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:25 PM

		SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3)) SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L169	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L170	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L171	6	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:28 PM
L172	137	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:36 PM
L173	6	L172 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM
L174	171	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM

		OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME2 (MPM OR (most NEAR3 probab$3))					
L175	804	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) AND intra WITH (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:42 PM
L176	37	L175 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:43 PM
L177	53	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L178	8	L172 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L179	8	L174 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L180	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:45 PM
L181	0	"S158" AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 04:38 PM
L182	1	17/351649.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2022/07/16 08:03 AM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L183	4	(EP-2293567-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/16 08:24 AM
L184	29	("10455248" OR "20030202602" OR "20040028282" OR "20060013299" OR "20060109913" OR "20060251330" OR "20070237228" OR "20080123947" OR "20080123977" OR "20080267291" OR "20090022221" OR "20090046781" OR "20090060345" OR "20090154567" OR "20090232206" OR "20100086029" OR "20100086032" OR "20100208827" OR "20110286520" OR "20110292999" OR "20120106629" OR "5060285" OR "5640420" OR "6084908" OR "7302006" OR "8467450" OR "8483285" OR "9113168" OR "9288494").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/07/18 07:18 AM
L185	117	(CN-1513268-$ | CN-1719905-$ | CN-1722836-$ | CN-101001382-$ | CN-	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/07/18 09:38 PM

		101047860-$ | CN-101150719-$ | CN-101207810-$ | CN-101273641-$ | CN-101340581-$ | CN-101394565-$ | CN-101467461-$ | JP-H08280020-$ | JP-8280020$- \| JP-H8280020-$ | JP-2004128749-$ | JP-2005519543-$ | JP-2007503784-$ | JP-2007528675-$ | JP-2009118233-$ | KR-1020050026318-$ | KR-20050026318-$ | KR-1020090065401-$ | KR-20090065401-$ | KR-102109244-$ | KR-2109244-$ | WO-03075579-$ | WO-2003075579-$ | WO-2004104930-$ | WO-2005093661-$ | WO-2006004331-$ | WO-2007034918-$ | WO-2007081908-$ | WO-2008004769-$ | WO-2008027192-$ | WO-2008088140-$ | WO-2009028922-$ | WO-2009051719-$ | WO-2010039731-$).DID.	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L186	37	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR KR-102194679-B1 OR US-20120106629-A1 OR KR-20200144159-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR JP-6193432-B2 OR WO-2011019383-A2 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/19 06:12 AM

		EP-2465265-A2).did.					
L188	2	15/806,893.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/08/12 02:51 PM
L189	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR US-20210385460-A1 OR US-11082697-B2 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR JP-6688341-B2 OR JP-2020115669-A OR KR-20120052267-A OR DK-2449782-T3 OR KR-102109244-B1 OR KR-102194679-B1 OR US-20120106629-A1 OR JP-6415472-B2 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR ES-2744316-T3 OR KR-101939016-B1 OR KR-20170016015-A OR HU-E035547-T2 OR JP-5918128-B2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-104853203-B OR CN-102484709-B OR CN-104796717-B OR	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2023/02/06 07:39 AM

		CN-102484709-A OR CN-104853203-A OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104853204-A OR CN-104853202-A OR CN-104822066-A OR JP-20120532489-A5 OR KR-20160052471-A OR KR-101964382-B1 OR KR-101467175-B1 OR JP-2014534734-A).did.					
L190	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:47 AM
L191	78	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:56 AM
L192	35	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102505806-B1 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR US-20120106629-A1 OR CN-104853207-B OR KR-20230035152-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:28 PM

		OR CN-104796717-B OR CN-104853203-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A).did.					
L193	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR EP-3579558-A1 OR EP-2449782-B1 OR US-20210385460-A1 OR US-11082697-B2 OR KR-102505806-B1 OR EP-2449782-A1 OR WO-2011002504-A1 OR JP-2020115669-A OR KR-20120052267-A OR EP-3288268-A1 OR DK-2449782-T3 OR JP-6688341-B2 OR EP-3288268-B1 OR US-20120106629-A1 OR KR-102109244-B1 OR JP-2023029886-A OR KR-102194679-B1 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR JP-6415472-B2 OR ES-2744316-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR KR-101939016-B1 OR JP-5918128-B2 OR KR-20170016015-A OR HU-E035547-T2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A OR CN-104853203-B OR CN-104796717-B OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104822066-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR CN-104853204-A OR CN-104853202-A OR JP-2021083122-A OR JP-20120532489-A5 OR DK-2887670-T3).did.					
L194	46	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 1 Text Selection Text: (US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR US-20210385460-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR EP-3288268-A1 OR US-11082697-B2 OR ES-2657965-T3 OR ES-2744316-T3 OR JP-6688341-B2 OR HU-E035547-T2 OR KR-102505806-B1 OR KR-102109244-B1 OR US-20220021879-A1 OR KR-102194679-B1 OR CN-104853207-B OR KR-101939016-B1 OR KR-20180004329-A OR JP-20160178645-A5 OR KR-20200049909-A OR EP-3523968-B1 OR JP-2022191450-A OR JP-7180794-B2 OR EP-3468202-A1 OR KR-102487077-B1 OR KR-20230010063-A OR KR-102422711-B1 OR JP-2013048480-A OR JP-2013048481-A OR EP-4090023-A1 OR KR-102062894-B1 OR KR-20200005207-A OR KR-102258057-B1 OR KR-20220017380-A OR WO-2022031018-A1 OR US-9451264-B2 OR US-9491468-B2 OR US-9462281-B2 OR WO-2022108419-A1 OR KR-20220071130-A OR EP-3442231-A2 OR US-20190149828-A1 OR KR-20230070533-A OR KR-20110126567-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR JP-6672226-B2 OR US-20220070448-A1 OR JP-5513584-B2 OR EP-3197161-A1 OR EP-3197160-A1).did.					
L195	13809	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:32 PM
L196	2392	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L197	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L198	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA,	OR	ON	ON	2023/08/23 02:33 PM

			EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)				
L199	45	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:34 PM
L200	175	L196 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:51 PM
L201	123	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:52 PM
L202	46	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR (("32" OR "64") NEAR2 ("x" OR ".times." OR times))) NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA,	OR	ON	ON	2023/08/23 02:56 PM

			VN); EPO; DERWENT; IBM_TDB)				
L203	9	(EP-2081388-$ | KR-20090065401-$ | WO-2008083633-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/09/07 08:57 AM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N144	44	(((large OR super OR jumbo OR mega OR giant) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N145	386	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N146	350	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:33 PM

		unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N147	45	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM. SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:34 PM
N148	5	N147 AND @ay<="2009"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N149	9	N147 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N150	47	N145 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N151	10	N150 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:41 PM
N152	14	N150 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:47 PM
N157	403	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N158	48	N157 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N159	14	N158 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N160	10	N158 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:39 AM
N182	405	(((large OR super OR jumbo OR mega OR	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM

		giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N183	48	N182 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N184	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N185	48	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N186	1	N185 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N187	52	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:30 PM
N188	1	N187 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:31 PM
N189	116	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:37 PM
N190	62	((((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:38 PM

		(coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3) NEAR3 (mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.					
N191	4	((((large OR super OR jumbo OR mega OR giant) WITH ("32" OR "64" OR "128" OR "16" OR "N" OR "2N")) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 (predict$3 OR mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:40 PM

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	1	13/381538.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/19 04:11 PM
L3	56273	Thomson.AS. OR (Thomson WITH licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:34 AM
L4	412	(Zheng WITH Yunfei).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:37 AM
L5	20	(Abbas WITH Adeel).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L6	481	(Joel WITH Sole).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L7	1842	(Peng WITH Yin).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:38 AM
L8	344	(Xiaoan WITH Lu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L9	992	(Qian WITH Xu).IN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L10	59046	L3 OR L4 OR L5 OR L6 OR L7 OR L8 OR L9	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:40 AM
L11	1	L10 AND ((intra WITH predict$4).CLM. SAME ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L12	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (block OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:42 AM
L13	3	L10 AND ((intra WITH predict$4).CLM. AND ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))).CLM.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:48 AM
L14	526	((intra WITH predict$4)	(US-PGPUB; USPAT;	OR	ON	ON	2014/09/22

		SAME ((large OR super) WITH (((coding OR predict$4) WITH unit) OR block OR MB OR macroblock OR (macro NEAR3 block))))	USOCR; FPRS; EPO; JPO; DERWENT)				08:50 AM
L15	16	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((coding OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:55 AM
L16	17	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) WITH (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) WITH unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L17	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super) NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 08:56 AM
L18	8	L10 AND ((intra WITH predict$4).CLM. AND (((large OR super OR "32" OR "64") NEAR3 (block OR MB OR macroblock OR (macro NEAR3 block))).CLM. OR ((cod$4 OR predict$4) NEAR2 unit).CLM.))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:14 AM
L19	7540	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) OR ((coding OR predict$4) WITH unit)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:34 AM
L20	4104	((intra WITH predict$4) SAME (((large OR super OR "32" OR "64") WITH (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM

		OR macroblock OR (macro NEAR3 block))) OR ((coding AND predict$4) WITH unit)))					
L21	1154	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:35 AM
L22	5	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:36 AM
L23	56273	Thomson.AS. OR (Thomson WITH Licens$4).AS.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:37 AM
L24	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND (quad WITH tree WITH (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:38 AM
L25	14	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4)) AND ((quad WITH tree) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:39 AM
L26	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:41 AM

		partit$6 OR struct$4))					
L27	45	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME tabl$4 OR (most WITH probab$4 WITH mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 09:44 AM
L28	22	("8467450"|"20080123977").PN. OR (WO-2010039731-$ | JP-2004128749-$ | JP-2009118233-$ | WO-2008027192-$ | WO-2007034918-$ | WO-2008088140-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 10:37 AM
L29	75	L26 AND ((H04N19/00278 OR H04N19/00042 OR H04N19/00781 OR H04N19/00884 OR H04N19/00763 OR H04N19/00545 OR H04N19/00969 OR H04N19/00024 OR H04N19/00018 OR H04N7/26244 OR H04N19/00733 OR H04N7/26026 OR G06T9/004).CPC.)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:02 AM
L30	88	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L31	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM

		(divid$4 OR divis$4 OR partit$6 OR struct$4)) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))					
L32	42	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:22 AM
L33	18	((intra WITH predict$4) SAME ((large OR super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (encod$4 AND decod$4) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header OR syntax))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:32 AM
L34	5	("20090046781" | "20060251330").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 11:38 AM
L35	17	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR struct$4 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (tabl$4 OR (most WITH probab$4 WITH mode)) SAME (header	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:19 PM

		OR syntax) SAME (encod$4 AND decod$4))					
L36	30	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree))) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:31 PM
L37	1	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree)) SAME adapt$4) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:33 PM
L38	38	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4) SAME (header OR syntax) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/22 04:34 PM
L39	1	12/827195.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 08:07 AM
L40	6	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:07 PM

L41	34	((intra WITH predict$4) SAME ((super OR "32" OR "64") WITH (block OR MB OR macroblock OR (macro NEAR3 block))) SAME (divid$4 OR divis$4 OR partit$6 OR (quad WITH tree) OR adapt$4)) AND ((intra WITH predict$4 WITH table) SAME (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2014/09/23 02:09 PM
L42	4	WO-2009051719-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:04 AM
L44	3	(CN-101207810-$ | KR-20090065401-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/15 10:22 AM
L45	554219	(large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L46	244625	(large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (coding WITH unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L47	133	L46 AND ((MPM OR (most NEAR3 probab$4)) WITH intra)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/04/16 07:45 AM
L48	6	"5060285".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 09:37 AM
L49	5	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:02 AM
L50	8	(WO-2005093661-$ | JP-2007528675-$ | JP-H08280020-$ | WO-03075579-$ | JP-2005519543-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/09/30 10:16 AM
L51	870	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4)))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM
L52	307	intra SAME ((quad WITH tree) OR (tree WITH (merg$4 OR split$4 OR partition$4))) SAME (block OR MB	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:05 AM

		OR mb OR macroblock)					
L53	327	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L54	255	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:06 AM
L55	162	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) WITH (block OR MB OR mb OR macroblock)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:07 AM
L56	250	intra SAME (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:11 AM
L57	21	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L58	10	intra SAME2 (((quad WITH tree) OR tree) WITH (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:18 AM
L59	10	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) SAME (video OR frame) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:21 AM
L60	25	intra SAME2 (((quad WITH tree) OR tree) SAME (merg$4 OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:22 AM

		split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"					
L61	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:23 AM
L62	30	intra SAME (((quad WITH tree) OR tree) SAME (merg$4 OR split$4 OR partition$4 OR (top NEAR2 down) OR (bottom NEAR2 up))) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L63	30	intra SAME (((quad WITH tree) OR tree OR (top NEAR2 down) OR (bottom NEAR2 up)) SAME (merg$4 OR split$4 OR partition$4)) SAME (block OR MB OR mb OR macroblock) AND @ay <= "2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2015/10/01 09:32 AM
L64	2	12/310090.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L65	4	(WO-2008027192-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:09 PM
L66	7	("8885727" | "20110243244" | "20110096829").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/18 05:20 PM
L67	1160	merg$4 SAME ((intra OR spatial) WITH (mode OR type))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L68	124	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) AND (intra WITH DC)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L69	16	L68 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:00 AM
L70	674	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM

L71	92	L70 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:15 AM
L72	396	merg$4 SAME (intra WITH (mode OR type)) SAME (block OR macroblock OR MB OR mb)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L73	81	L72 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 08:20 AM
L74	4	WO-2009028922-$.DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:46 AM
L75	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L76	2	L75 AND @ay<"2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:50 AM
L77	38	merg$4 SAME ((intra OR spatial) WITH (mode OR type)) SAME2 ((large OR super OR "32" OR "64" OR "128" OR "2N") NEAR3 (block OR macroblock OR MB OR mb))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/19 09:51 AM
L78	2	"20030099292".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/07/20 01:13 PM
L79	26448	(split$4 AND (merg$4 OR group$4)) WITH (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L80	8139	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:41 AM
L81	1826	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:42 AM

		decision) SAME2 (block OR macroblock OR MB)					
L82	976	L81 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:53 AM
L83	631	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:56 AM
L84	976	L82 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L85	222	L83 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 11:58 AM
L86	33	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) WITH (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:03 PM
L87	0	L86 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L88	0	L86 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L89	0	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) SAME2 (block OR macroblock OR MB) SAME (large OR super) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:06 PM
L90	491	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:32 PM

		flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))					
L91	606	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR exten$4)) AND (video WITH (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:33 PM
L92	224	L91 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:37 PM
L93	434	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:46 PM
L94	152	L93 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L95	266	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR exten$4)) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:48 PM
L96	48	L95 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 12:49 PM

L97	378	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) SAME (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:13 PM
L98	72	L97 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:20 PM
L99	328	(split$4 AND (merg$4 OR group$4)) NEAR3 (signal$4 OR (syntax NEAR3 element) OR flag OR binary OR decision) AND ((block OR macroblock OR MB) WITH (large OR super OR "32" OR "64")) AND (video NEAR3 (encod$4 AND decod$4)) AND ((intra OR spatial) WITH (predict$4 OR mode))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 01:24 PM
L100	2	"20030202602".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2016/12/27 03:05 PM
L102	3	("20090110070").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 09:59 AM
L103	7	("20030099292" | "20030206594" | "20120106629").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/05/31 05:29 PM
L104	85	DC SAME (default OR intial$4) SAME (intra WITH predict$4) SAME mode	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:58 AM
L105	7	L104 AND (large OR super) WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 10:59 AM
L106	55	L104 AND (large OR super OR "32" OR "64") WITH (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM

L107	8	L104 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:00 AM
L108	139	DC SAME (intra WITH predict$4) SAME mode SAME2 (default OR intial$4)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:01 AM
L109	23	L108 AND (large OR super OR "32" OR "64") NEAR3 (block OR macroblock OR MB OR mb OR (macro NEAR2 block))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:19 AM
L110	74	DC SAME (intra WITH predict$4) SAME mode SAME2 ((default OR intial$4) WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/06/01 11:20 AM
L112	5	(JP-2007503784-$ | WO-2004104930-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2017/12/14 09:08 AM
L113	4248	(spatial WITH intra WITH mode)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:38 PM
L114	239	(spatial WITH intra WITH mode) SAME table	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L115	1	15/806893.APP.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:39 PM
L116	222	(spatial WITH intra WITH mode) SAME table SAME (partition OR prediction)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L117	104	L116 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L118	98	L116 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:40 PM
L119	1008	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:42 PM
L120	162	((spatial OR intra) NEAR3 mode) SAME table SAME predict$4 SAME2 (save OR store) AND((block OR unit)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:53 PM

		SAME (partition$4 OR merg$4 OR split$4))					
L121	14	L120 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:54 PM
L122	112	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 SAME2 (save OR store) AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 02:58 PM
L123	4	L122 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L124	4	L123 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:04 PM
L125	41	((spatial OR intra) NEAR3 mode) WITH table SAME predict$4 AND((block OR unit) SAME (partition$4 OR merg$4 OR split$4)) AND (DC WITH default)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L126	1	L125 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:29 PM
L127	2	L125 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:46 PM
L128	1171	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 (transmit$4 OR receiv$4 OR send OR sent) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:56 PM
L129	46	(((spatial OR intra) NEAR3 mode) WITH table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:58 PM
L130	15	L129 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 03:59 PM

L131	65	(((spatial OR intra) NEAR3 mode) SAME table) SAME2 ((transmit$4 OR receiv$4 OR send OR sent) SAME (store OR storing OR (pre ADJ stor$4))) AND ((block OR unit) SAME (partition$4 OR merg$4 OR split$4))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:11 PM
L132	24	L131 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L133	21	L131 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/03/28 04:14 PM
L134	19	("100208827"|"20030202602"|"20060109913"|"20060251330"|"20070237228"|"20080123947"|"20080123977"|"20080267291"|"20090022221"|"20090046781"|"20090060345"|"20090154567"|"20100086029"|"20100208827"|"20110286520"|"20110292999"|"5060285"|"7302006"|"8467450"|"9288494").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2018/04/01 06:44 PM
L135	19	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO)	OR	ON	ON	2018/04/01 10:58 PM
L136	27	(WO-2005093661-$ | WO-2009051719-$ | WO-2010039731-$ | WO-2008027192-$ | JP-2004128749-$ | KR-20090065401-$ | CN-101207810-$ | WO-2008088140-$ | WO-2007034918-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 10:58 PM
L137	16	(JP-2009118233-$ | CN-101394565-$ | JP-2005519543-$ | JP-2007528675-$ | JP-08280020-$ | JP-H08280020-$ | WO-03075579-$ | KR-	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM

		20050026318-$ | KR-1020050026318-$ | WO-2009028922-$).DID.					
L138	26	(CN-101467461-$ | JP-2007503784-$ | WO-2004104930-$ | CN-101001382-$ | CN-101047860-$ | CN-101340581-$ | CN-1722836-$ | WO-2006004331-$ | CN-101150719-$ | CN-101273641-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT)	OR	ON	ON	2018/04/01 11:04 PM
L139	4	("20040028282"|"20060013299"|"20100086032"|"20120106629").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/03 02:46 PM
L140	9	(CN-1513268-$ | CN-1719905-$ | KR-102109244-$ | WO-2008004769-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 02:57 PM
L141	3	(WO-2010001862-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L142	2	("20090222177").PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:05 PM
L143	2	(WO-2011002504-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2020/12/03 03:06 PM
L153	1	"100208827".PN.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/01/27 09:42 PM
L154	1	("6084908").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:32 AM
L155	3	(WO-2007081908-$).DID.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/05 06:33 AM
L156	1	("20100208827").PN.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:36 AM
L161	754	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM

		OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L162	64	L161 AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:20 PM
L163	21	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2010"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L164	26	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:21 PM
L165	44	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME DC SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:25 PM
L166	3	L165 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:26 PM
L167	18	L161 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/09 10:27 PM
L168	130	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:25 PM

		SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3)) SAME (block OR macroblock OR PU OR TU OR CU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))					
L169	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L170	2	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND (intra OR spatial) WITH (table OR plurality) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:27 PM
L171	6	L168 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:28 PM
L172	137	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:36 PM
L173	6	L172 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM
L174	171	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:41 PM

		OR super OR tree OR quadtree OR (quad NEAR2 tree)) SAME2 (MPM OR (most NEAR3 probab$3))					
L175	804	(((intra OR spatial) NEAR3 (predict$4 OR mode OR type)) OR "I16" OR "I_16x16" OR "I16x16") WITH (plural$3 OR table) SAME (merg$3 OR split$4 OR partition$3 OR super OR tree OR quadtree OR (quad NEAR2 tree)) AND intra WITH (MPM OR (most NEAR3 probab$3))	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:42 PM
L176	37	L175 AND ((size OR dimnesion OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:43 PM
L177	53	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L178	8	L172 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L179	8	L174 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2012"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:44 PM
L180	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 02:45 PM
L181	0	"S158" AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/03/12 04:38 PM
L182	1	17/351649.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2022/07/16 08:03 AM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L183	4	(EP-2293567-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/16 08:24 AM
L184	29	("10455248" OR "20030202602" OR "20040028282" OR "20060013299" OR "20060109913" OR "20060251330" OR "20070237228" OR "20080123947" OR "20080123977" OR "20080267291" OR "20090022221" OR "20090046781" OR "20090060345" OR "20090154567" OR "20090232206" OR "20100086029" OR "20100086032" OR "20100208827" OR "20110286520" OR "20110292999" OR "20120106629" OR "5060285" OR "5640420" OR "6084908" OR "7302006" OR "8467450" OR "8483285" OR "9113168" OR "9288494").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/07/18 07:18 AM
L185	117	(CN-1513268-$ | CN-1719905-$ | CN-1722836-$ | CN-101001382-$ | CN-	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU,	OR	ON	ON	2022/07/18 09:38 PM

		101047860-$ | CN-101150719-$ | CN-101207810-$ | CN-101273641-$ | CN-101340581-$ | CN-101394565-$ | CN-101467461-$ | JP-H08280020-$ | JP-8280020$- \| JP-H8280020-$ | JP-2004128749-$ | JP-2005519543-$ | JP-2007503784-$ | JP-2007528675-$ | JP-2009118233-$ | KR-1020050026318-$ | KR-20050026318-$ | KR-1020090065401-$ | KR-20090065401-$ | KR-102109244-$ | KR-2109244-$ | WO-03075579-$ | WO-2003075579-$ | WO-2004104930-$ | WO-2005093661-$ | WO-2006004331-$ | WO-2007034918-$ | WO-2007081908-$ | WO-2008004769-$ | WO-2008027192-$ | WO-2008088140-$ | WO-2009028922-$ | WO-2009051719-$ | WO-2010039731-$).DID.	CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L186	37	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR KR-102194679-B1 OR US-20120106629-A1 OR KR-20200144159-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR JP-6193432-B2 OR WO-2011019383-A2 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/07/19 06:12 AM

		EP-2465265-A2).did.					
L188	2	15/806,893.APP.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/08/12 02:51 PM
L189	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR US-20210385460-A1 OR US-11082697-B2 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR JP-6688341-B2 OR JP-2020115669-A OR KR-20120052267-A OR DK-2449782-T3 OR KR-102109244-B1 OR KR-102194679-B1 OR US-20120106629-A1 OR JP-6415472-B2 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR ES-2744316-T3 OR KR-101939016-B1 OR KR-20170016015-A OR HU-E035547-T2 OR JP-5918128-B2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-104853203-B OR CN-102484709-B OR CN-104796717-B OR	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2023/02/06 07:39 AM

		CN-102484709-A OR CN-104853203-A OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104853204-A OR CN-104853202-A OR CN-104822066-A OR JP-20120532489-A5 OR KR-20160052471-A OR KR-101964382-B1 OR KR-101467175-B1 OR JP-2014534734-A).did.					
L190	5	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:47 AM
L191	78	L175 AND ((size OR dimension OR height OR width) WITH ("16" OR "32")) AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/06 07:56 AM
L192	35	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102505806-B1 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR US-20120106629-A1 OR CN-104853207-B OR KR-20230035152-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:28 PM

		OR CN-104796717-B OR CN-104853203-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A).did.					
L193	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR EP-3579558-A1 OR EP-2449782-B1 OR US-20210385460-A1 OR US-11082697-B2 OR KR-102505806-B1 OR EP-2449782-A1 OR WO-2011002504-A1 OR JP-2020115669-A OR KR-20120052267-A OR EP-3288268-A1 OR DK-2449782-T3 OR JP-6688341-B2 OR EP-3288268-B1 OR US-20120106629-A1 OR KR-102109244-B1 OR JP-2023029886-A OR KR-102194679-B1 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR JP-6415472-B2 OR ES-2744316-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR KR-101939016-B1 OR JP-5918128-B2 OR KR-20170016015-A OR HU-E035547-T2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A OR CN-104853203-B OR CN-104796717-B OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104822066-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR CN-104853204-A OR CN-104853202-A OR JP-2021083122-A OR JP-20120532489-A5 OR DK-2887670-T3).did.					
L194	46	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 1 Text Selection Text: (US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR US-20210385460-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR EP-3288268-A1 OR US-11082697-B2 OR ES-2657965-T3 OR ES-2744316-T3 OR JP-6688341-B2 OR HU-E035547-T2 OR KR-102505806-B1 OR KR-102109244-B1 OR US-20220021879-A1 OR KR-102194679-B1 OR CN-104853207-B OR KR-101939016-B1 OR KR-20180004329-A OR JP-20160178645-A5 OR KR-20200049909-A OR EP-3523968-B1 OR JP-2022191450-A OR JP-7180794-B2 OR EP-3468202-A1 OR KR-102487077-B1 OR KR-20230010063-A OR KR-102422711-B1 OR JP-2013048480-A OR JP-2013048481-A OR EP-4090023-A1 OR KR-102062894-B1 OR KR-20200005207-A OR KR-102258057-B1 OR KR-20220017380-A OR WO-2022031018-A1 OR US-9451264-B2 OR US-9491468-B2 OR US-9462281-B2 OR WO-2022108419-A1 OR KR-20220071130-A OR EP-3442231-A2 OR US-20190149828-A1 OR KR-20230070533-A OR KR-20110126567-A	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:29 PM

		OR JP-6672226-B2 OR US-20220070448-A1 OR JP-5513584-B2 OR EP-3197161-A1 OR EP-3197160-A1).did.					
L195	13809	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/08/23 02:32 PM
L196	2392	(intra WITH (mode OR direction OR DC)) SAME (large OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L197	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:33 PM
L198	471	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA,	OR	ON	ON	2023/08/23 02:33 PM

			EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)				
L199	45	L196 AND @ay<="2011"	(US-PGPUB; USPAT; USOCR; EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:34 PM
L200	175	L196 AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:51 PM
L201	123	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR "32" OR "64") NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/08/23 02:52 PM
L202	46	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR (("32" OR "64") NEAR2 ("x" OR ".times." OR times))) NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA,	OR	ON	ON	2023/08/23 02:56 PM

			VN); EPO; DERWENT; IBM_TDB)				
L203	9	(EP-2081388-$ | KR-20090065401-$ | WO-2008083633-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/09/07 08:57 AM
L204	35	(Artificial Intelligence) More like doc: US-20210385460-A1 Text: (US-11082697-B2 OR US-20180070088-A1 OR EP-2449782-B1 OR EP-3579558-A1 OR EP-3288268-A1 OR EP-3288268-B1 OR DK-2449782-T3 OR KR-102505806-B1 OR KR-102109244-B1 OR WO-2011002504-A1 OR EP-2449782-A1 OR KR-20200049909-A OR US-20120106629-A1 OR KR-20230035152-A OR US-11044483-B2 OR JP-6672226-B2 OR EP-3474547-A1 OR EP-2465265-A2 OR WO-2011019383-A2 OR US-10225581-B2).did.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/12/06 04:14 PM
L205	47	(Artificial Intelligence) Similar to: 17/351,649 with 0 CPC Selections and 0 Text Selections Text: (US-20180070088-A1 OR EP-3579558-A1 OR EP-2449782-B1 OR US-20210385460-A1 OR US-11082697-B2 OR KR-102505806-B1 OR WO-2011002504-A1 OR EP-2449782-A1	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	OR	ON	ON	2023/12/06 04:19 PM

		OR JP-2020115669-A OR KR-20120052267-A OR EP-3288268-A1 OR DK-2449782-T3 OR JP-6688341-B2 OR EP-3288268-B1 OR US-20120106629-A1 OR KR-102109244-B1 OR JP-2023029886-A OR KR-102194679-B1 OR JP-2016178645-A OR JP-2012532489-A OR JP-2018174580-A OR ES-2657965-T3 OR JP-6415472-B2 OR ES-2744316-T3 OR KR-20200049909-A OR JP-7184841-B2 OR KR-20180004329-A OR KR-101939016-B1 OR JP-5918128-B2 OR KR-20170016015-A OR HU-E035547-T2 OR JP-20160178645-A5 OR CN-104853207-B OR CN-102484709-B OR CN-102484709-A OR CN-104853203-A OR CN-104853203-B OR CN-104796717-B OR CN-104796717-A OR CN-104853207-A OR CN-104811719-A OR CN-104869417-A OR CN-104853206-A OR CN-104853205-A OR CN-104853204-A OR CN-104822066-A OR CN-104853202-A OR JP-20120532489-A5 OR DK-2887670-T3 OR KR-101467175-B1).did.					
L206	1	("6978048").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/12/06 04:35 PM
L207	4	(JP-2000308060-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK,	OR	ON	ON	2023/12/06 04:36 PM

			TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L208	46	((intra WITH (mode OR direction OR DC)) SAME ((large WITH ("32" OR "64" OR "N")) OR super OR (("32" OR "64") NEAR2 ("x" OR ".times." OR times))) NEAR3 (macroblock OR block OR PU OR CU OR predict$4) SAME (partition$4 OR split$4)) AND @ay<="2009"	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); EPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/12/06 04:37 PM
L209	5	(WO-2009151232-$).DID.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/12/06 04:53 PM
L211	4	("20110090967").PN.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/12/06 04:54 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp

N144	44	(((large OR super OR jumbo OR mega OR giant) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N145	386	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:31 PM
N146	350	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:33 PM
N147	45	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit))) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM. SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:34 PM
N148	5	N147 AND @ay<="2009"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N149	9	N147 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM

N150	47	N145 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:40 PM
N151	10	N150 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:41 PM
N152	14	N150 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2020/12/14 12:47 PM
N157	403	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N158	48	N157 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N159	14	N158 AND @ay<="2012"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:37 AM
N160	10	N158 AND @ay<="2011"	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/05 06:39 AM
N182	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N183	48	N182 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 04:38 PM
N184	405	(((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit)	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM

		OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3)).CLM.					
N185	48	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N186	1	N185 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:25 PM
N187	52	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:30 PM
N188	1	N187 AND (MPM OR (most WITH probab$4)).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/03/12 09:31 PM
N189	116	N184 SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:37 PM
N190	62	((((large OR super OR jumbo OR mega OR giant OR "32" OR "64" OR "128" OR "16") WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit)) SAME ((intra OR spatial$3) NEAR3 predict$3) NEAR3 (mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:38 PM
N191	4	((((large OR super OR jumbo OR mega OR giant) WITH ("32" OR "64" OR "128" OR "16" OR "N" OR "2N")) WITH (block OR macroblock OR MB OR CU OR TU OR PU OR (coding NEAR2 unit) OR (predict$3 NEAR2 unit) OR data OR unit))	(US-PGPUB; USPAT)	OR	ON	ON	2023/08/23 03:40 PM

		SAME ((intra OR spatial$3) NEAR3 (predict$3 OR mode OR DC OR direct$4)).CLM.) SAME (split$3 OR partition$3 OR divid$3 OR divis$3 OR quad OR tree OR quadtree).CLM.					

US011936876B2

(12) **United States Patent**
Zheng et al.

(10) **Patent No.: US 11,936,876 B2**
(45) **Date of Patent: *Mar. 19, 2024**

(54) **METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS**

(71) Applicant: **InterDigital VC Holdings, Inc.**, Wilmington, DE (US)

(72) Inventors: **Yunfei Zheng**, San Jose, CA (US); **Qian Xu**, Folsom, CA (US); **Xiaoan Lu**, Yardley, PA (US); **Peng Yin**, Ithaca, NY (US); **Joel Sole Rojals**, San Diego, CA (US); **Adeel Abbas**, Passaic, NJ (US)

(73) Assignee: **InterDigital VC Holdings, Inc.**, Wilmington, DE (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **17/351,649**

(22) Filed: **Jun. 18, 2021**

(65) **Prior Publication Data**

US 2021/0385460 A1 Dec. 9, 2021

Related U.S. Application Data

(63) Continuation of application No. 15/806,893, filed on Nov. 8, 2017, now Pat. No. 11,082,697, which is a (Continued)

(51) **Int. Cl.**
H04N 19/136 (2014.01)
H04N 19/11 (2014.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04N 19/136*** (2014.11); ***H04N 19/11*** (2014.11); ***H04N 19/147*** (2014.11);
(Continued)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,060,285 A 10/1991 Dixit et al.
5,640,420 A 6/1997 Jung
(Continued)

FOREIGN PATENT DOCUMENTS

CN 1513268 A 7/2004
CN 1719905 A 1/2006
(Continued)

OTHER PUBLICATIONS

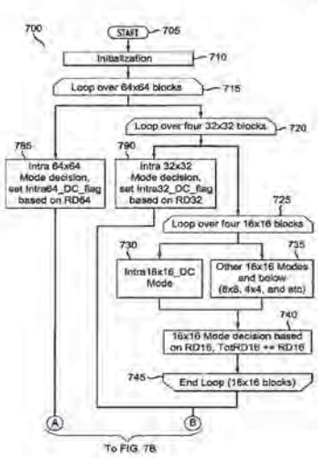
Chen et al., "Video Coding Using Extended Block Sizes", VCEG-AJ23, ITU—Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), 36th Meeting: San Diego, USA, Oct. 8-10, 2008, pp. 1-3.
(Continued)

Primary Examiner — Tyler W. Sullivan
(74) *Attorney, Agent, or Firm* — CONDO ROCCIA KOPTIW LLP

(57) **ABSTRACT**

Methods and apparatus are provided for signaling intra prediction for large blocks for video encoders and decoders. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

18 Claims, 9 Drawing Sheets



Related U.S. Application Data

continuation of application No. 13/381,538, filed as application No. PCT/US2010/001862 on Jun. 29, 2010, now abandoned.

(60) Provisional application No. 61/222,177, filed on Jul. 1, 2009.

(51) **Int. Cl.**

H04N 19/147	(2014.01)
H04N 19/157	(2014.01)
H04N 19/159	(2014.01)
H04N 19/176	(2014.01)
H04N 19/196	(2014.01)
H04N 19/46	(2014.01)
H04N 19/463	(2014.01)
H04N 19/593	(2014.01)
H04N 19/70	(2014.01)
H04N 19/96	(2014.01)

(52) **U.S. Cl.**

CPC ***H04N 19/157*** (2014.11); ***H04N 19/159*** (2014.11); ***H04N 19/176*** (2014.11); ***H04N 19/196*** (2014.11); ***H04N 19/46*** (2014.11); ***H04N 19/463*** (2014.11); ***H04N 19/593*** (2014.11); ***H04N 19/70*** (2014.11); ***H04N 19/96*** (2014.11)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,084,908	A	7/2000	Chiang et al.	
6,978,048	B1	12/2005	Higginbottom et al.	
7,302,006	B2	11/2007	Apostolopoulos et al.	
8,467,450	B2	6/2013	Moriya et al.	
8,483,285	B2	7/2013	Ye et al.	
9,113,168	B2	8/2015	Park et al.	
9,288,494	B2	3/2016	Xu et al.	
10,455,248	B2	10/2019	Park et al.	
2003/0202602	A1	10/2003	Apostolopoulos et al.	
2004/0028282	A1	2/2004	Kato et al.	
2006/0013299	A1	1/2006	Sato et al.	
2006/0109913	A1	5/2006	Wang et al.	
2006/0251330	A1	11/2006	Toth et al.	
2007/0237228	A1	10/2007	Han et al.	
2008/0123947	A1	5/2008	Moriya et al.	
2008/0123977	A1	5/2008	Moriya et al.	
2008/0267291	A1	10/2008	Vieron et al.	
2009/0022221	A1	1/2009	Xie et al.	
2009/0046781	A1	2/2009	Moriya et al.	
2009/0060345	A1	3/2009	Wheeler et al.	
2009/0154567	A1	6/2009	Lei et al.	
2009/0219991	A1*	9/2009	Po	H04N 19/176 375/E7.126
2009/0232206	A1	9/2009	Boon et al.	
2010/0086029	A1	4/2010	Chen et al.	
2010/0086030	A1*	4/2010	Chen	H04N 19/147 375/E7.243
2010/0086032	A1	4/2010	Chen et al.	
2010/0208827	A1	8/2010	Divorra et al.	
2011/0090967	A1*	4/2011	Chen	H04N 19/129 375/E7.125
2011/0286520	A1	11/2011	Xu et al.	
2011/0292999	A1	12/2011	Jeong et al.	
2012/0106629	A1	5/2012	Zheng et al.	

FOREIGN PATENT DOCUMENTS

CN	1722836	A	1/2006
CN	101001382	A	7/2007
CN	101047860	A	10/2007
CN	101150719	A	3/2008
CN	101207810	A	6/2008
CN	101273641	A	9/2008
CN	101340581	A	1/2009
CN	101394565	A	3/2009
CN	101467461	A	6/2009
EP	2081388	A1	7/2009
EP	2293567	A2	3/2011
JP	H08-280020	A	10/1996
JP	2000-308060	A	11/2000
JP	2004-128749	A	4/2004
JP	2005-519543	A	6/2005
JP	2007-503784	A	2/2007
JP	2007-528675	A	10/2007
JP	2009-118233	A	5/2009
KR	10-2005-0026318	A	3/2005
KR	10-2009-0065401	A	6/2009
KR	20090065401	A	6/2009
KR	10-2109244	B1	5/2020
WO	03/075579	A2	9/2003
WO	2004/104930	A2	12/2004
WO	2005/093661	A2	10/2005
WO	2006/004331	A1	1/2006
WO	2007/034918	A1	3/2007
WO	2007/081908	A1	7/2007
WO	2008/004769	A1	1/2008
WO	2008/027192	A2	3/2008
WO	2008-083633	A1	7/2008
WO	2008/088140	A1	7/2008
WO	2009/028922	A2	3/2009
WO	2009/051719	A2	4/2009
WO	2010/039731	A2	4/2010

OTHER PUBLICATIONS

ITU-T, "ITU-T Video Coding Experts Group (VCEG)", Jul. 1, 2009, pp. 1-3.

ITU, "Video Coding using Extended Block Sizes", ITU COM 16-C123-E Standard, International Telecommunication Union, Jan. 2009, 4 pages.

ITU-T, "International Telecommunication Union", H.264 Standard, Mar. 1, 2005, 343 pages.

Kim et al., "Enlarging MB Size for High Fidelity Video Coding Beyond Hd", VCEG-AJ21, ITU—Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), 36th Meeting: San Diego, CA, Oct. 8-10, 2008, pp. 1-6.

Lee et al., "Technical Considerations for Ad Hoc Group on New Challenges in Video Coding Standardization", ISO/IEC JTC1/SC29/WG11, Hannover, Germany, Jul. 2008, 4 pages.

Lim et al., "Intra Coding Using Extended Block Size", VCEG-AL28, ITU—Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), 38th Meeting: London, UK/Geneva, CH, Jul. 1-8, 2009, pp. 1-5.

Marpe et al., "H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas", IEEE International Conference on Image Processing, Genova, Italy, Sep. 14, 2005, 4 pages.

Naito et al., "Efficient Coding Scheme for Super High Definition Video Based on Extending H.264 High Profile", Visual Communications and Image Processing, 2006, 8 pages.

Vaisey et al., "Image Compression with Variable Block Size Segmentation", IEEE Transactions on Signal Processing, vol. 40, No. 8, Aug. 8, 1992, pp. 2040-2060.

Wien, Mathias, "Variable Block-Size Transforms for H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, No. 7, Jul. 2003, pp. 604-613.

Yoshino et al., "Preliminary Response for Draft Call for Evidence of High Performance Video Coding", ISO/IEC JTC1/SC29/WG11 m16082, MPEG2008, Lausanne, Switzerland, Feb. 2009.

CN 101001382 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.

CN 101150719 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.

CN 101207810 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.

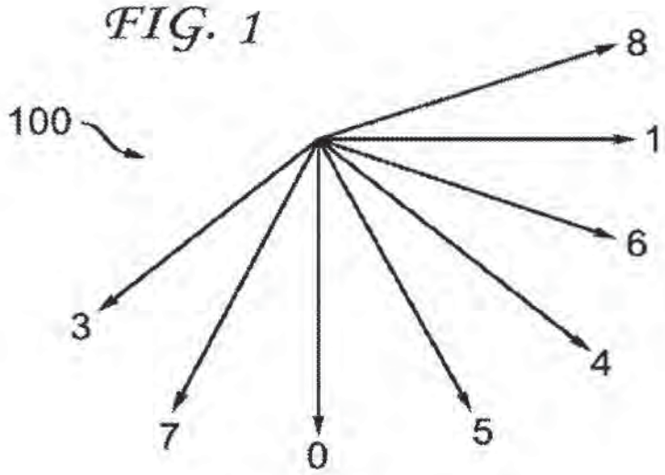
(56) **References Cited**

OTHER PUBLICATIONS

CN 101340581 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
CN 101394565 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
JP 2004-128749 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
JP 2009-118233 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
KR 10-2005-0026318 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
KR 10-2009-0065401 A, Documents cited w/ IDS dated Nov. 8, 2017, in related U.S. Appl. No. 15/806,893.
Vaisey, et al., "Image Compression with Variable Block Size Segmentation", IEEE, Transactions on Signal Processing, vol. 40, No. 8, Aug. 8, 1992, 3 pages.
Vaisey, et al., "Variable Block-Size Image Size Decoding", Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, IEEE, 1987, 4 pages.
Samet, et al., "The Quadtree and Related Hierarchical Data Structures", Computer Surverys, vol. 16, No. 2, Jun. 1984, 74 pages.
Chen, et al., "Video Coding Using Extended Block Sizes", ITU—Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), Document VCEG-AJ23, 36th Meeting: San Diego, USA, Oct. 8-10, 2008, 3 pages.
Chen, et al., "Video Coding Using Extended Block Sizes", International Telecommunication Union, Telecommunication Standardization Sector, COM 16- C123-E, Study Group 16 Contribution 123, Qualcomm Inc., ITU-T/COM-T/COM16/C/123E.DOC, 2009-2012, 4 pages.
Wien, et al., "Intra Coding using Variable Block Sizes", ITU-Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), VCEG-O31, 15th Meeting, Pattaya, Thailand, Dec. 4-6, 2001, 12 pages.
PatDocs English Translation of KR 20090065401 A.
KR 200960065401 A, PatDocs English Translation of KR200960065401 A.
WO 2008-0833633 A1, EP 2081388 A1.
Faundez-Zanuy et al., "Contributions to Interframe Coding", Workshop on image analysis & synthesis in image coding, Berlin, Oct. 1994, 6 pages.
McCann et al., "Samsung's Response to the Call for Proposals on Video Compression Technology", JCTVC A124, Samsung Electronics Co., Ltd., and British Broadcasting Corporation, Joint Collaborative Team on Video Coding (JCT-VC) of ITU T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 1st Meeting: Dresden, DE, Apr. 15-23, 2010, pp. 1-42.
Chen, et al., "U.S. Appl. No. 61/144,357", filed Jan. 13, 2009, 10 pages.
Chen, et al., "U.S. Appl. No. 61/166,631", filed Apr. 3, 2009, 97 pages.
JP 2000-308060 A, U.S. Pat. No. 6,978,048 B1.

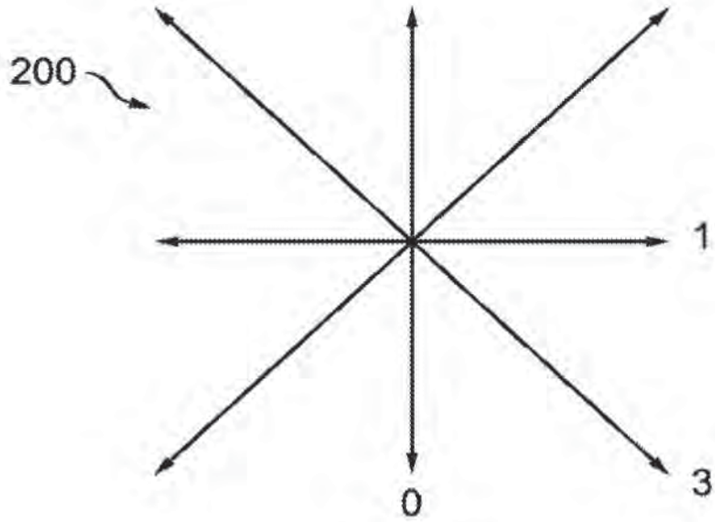
* cited by examiner

FIG. 1



PRIOR ART

FIG. 2



PRIOR ART

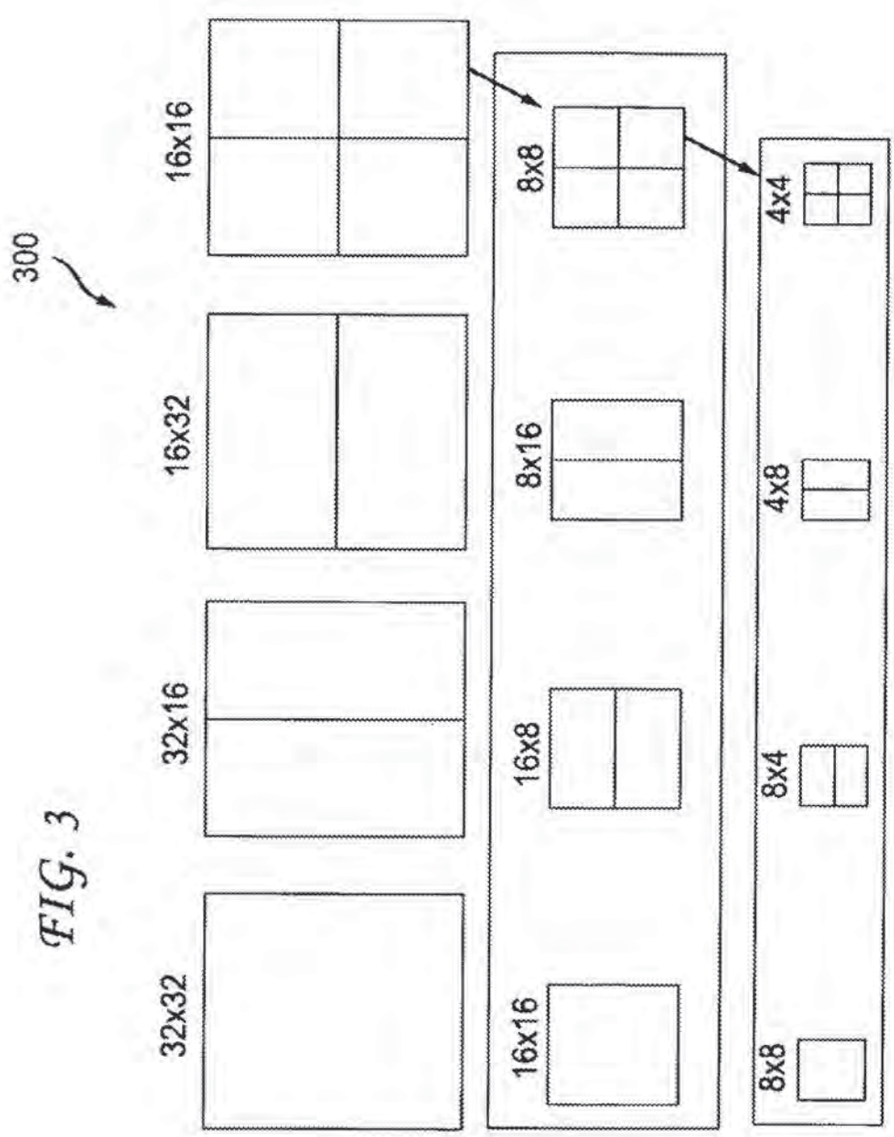

300
16x16
16x32
32x16
32x32
8x8
8x16
16x8
16x16
4x4
4x8
8x4
8x8

FIG. 3

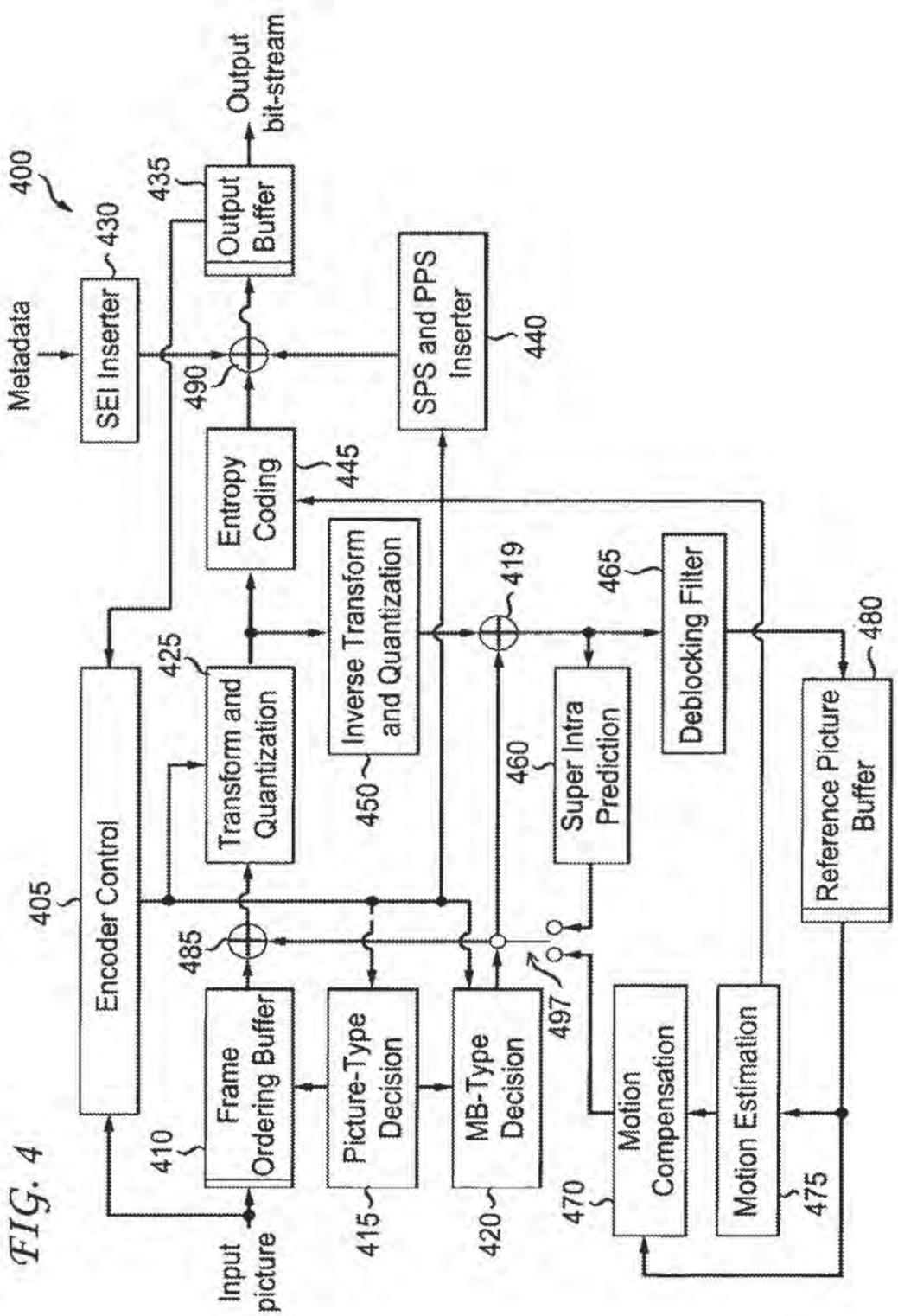

400
Metadata
SEI Inserter
430
Output Buffer
435
Output bit-stream
490
SPS and PPS Inserter
440
Entropy Coding
445
Inverse Transform and Quantization
450
419
465
Deblocking Filter
480
Reference Picture Buffer
425
Transform and Quantization
460
Super Intra Prediction
405
Encoder Control
485
497
Frame Ordering Buffer
410
Picture-Type Decision
415
MB-Type Decision
420
Motion Compensation
470
Motion Estimation
475
Input picture

FIG. 4

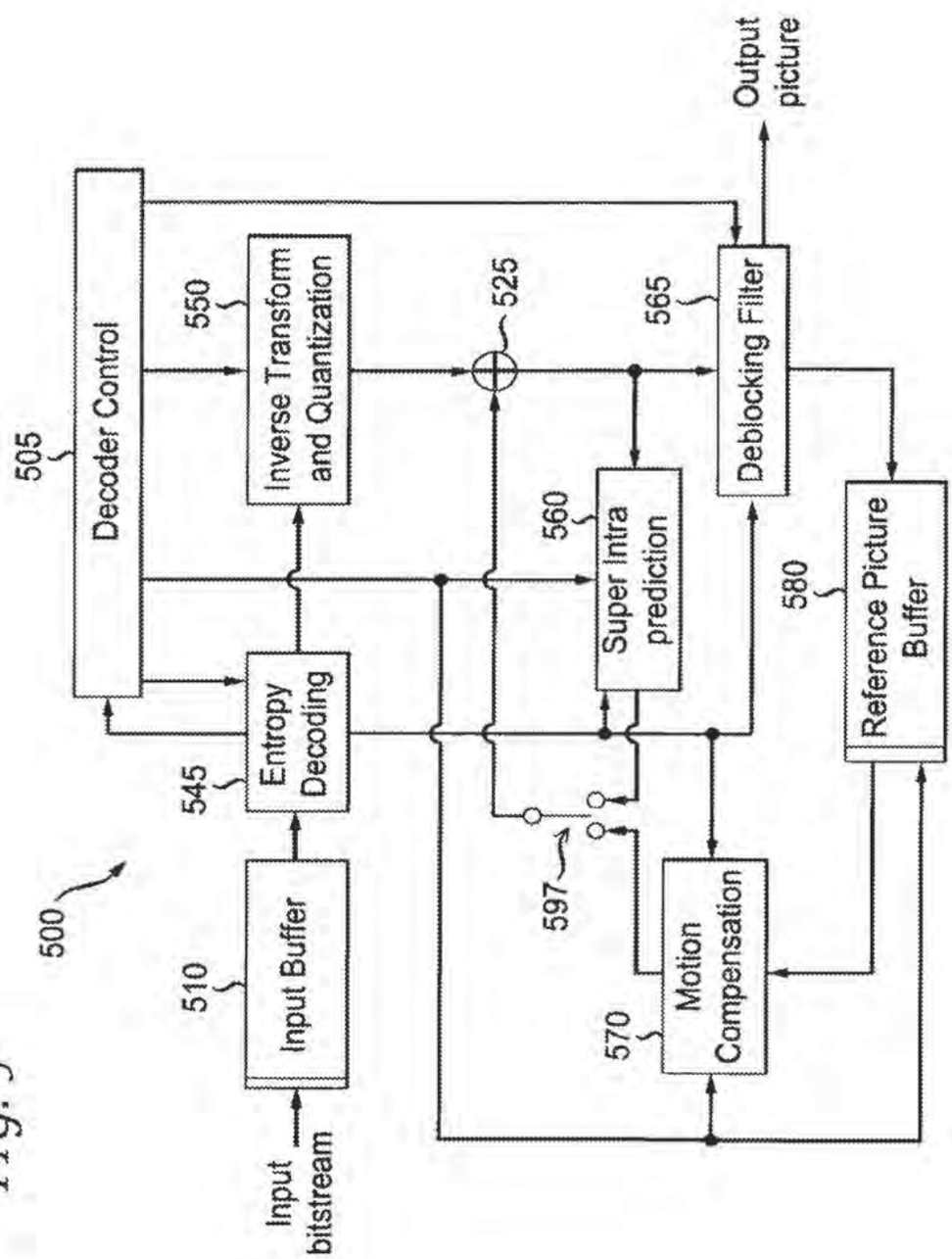

Input bitstream
500
Input Buffer
510
Entropy Decoding
545
Inverse Transform and Quantization
550
Decoder Control
505
Super intra prediction
560
Motion Compensation
570
597
525
Deblocking Filter
565
Reference Picture Buffer
580
Output picture

FIG. 5

FIG. 6

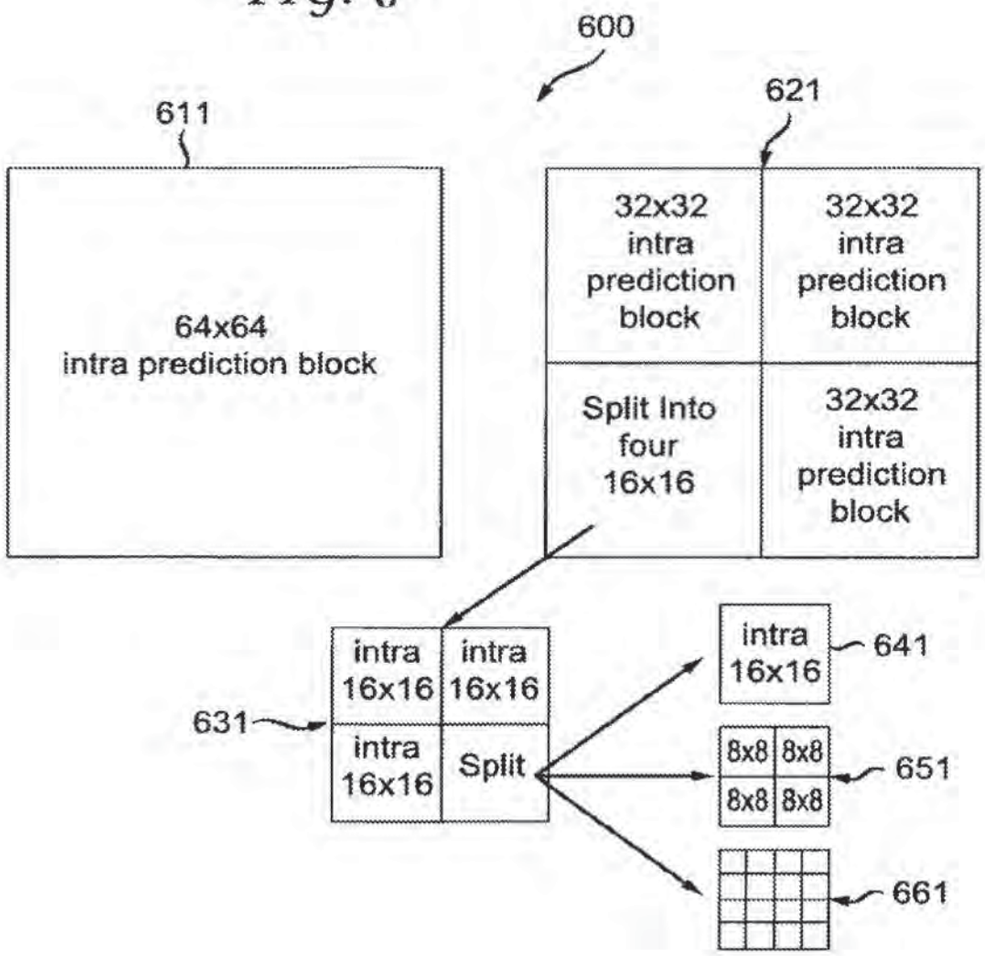

600
611
621
64x64
intra prediction block
32x32
intra
prediction
block
32x32
intra
prediction
block
Split Into
four
16x16
32x32
intra
prediction
block
631
intra
16x16
intra
16x16
intra
16x16
Split
intra
16x16
641
8x8
8x8
8x8
8x8
651
661

FIG. 7A

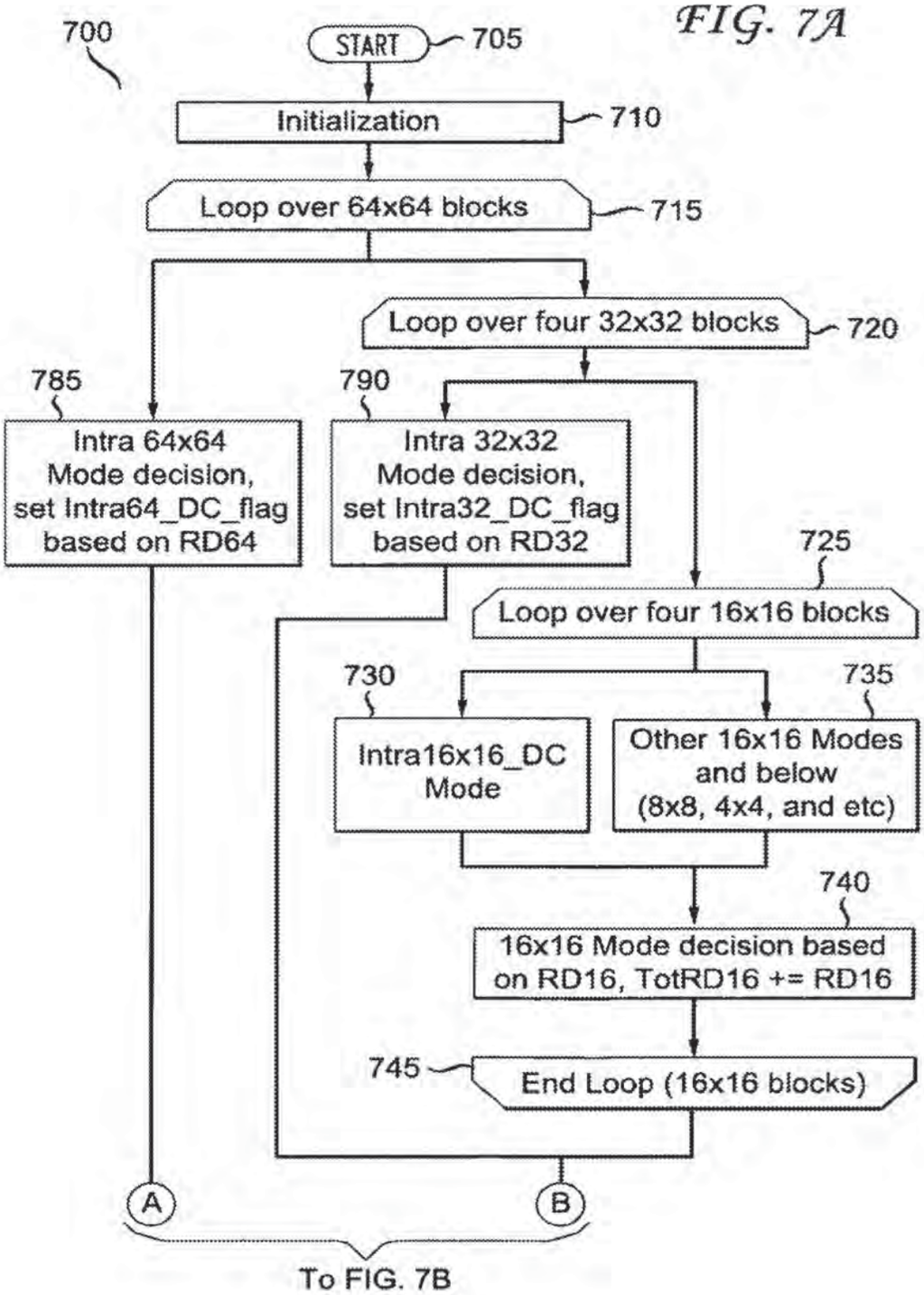

700
START
705
Initialization
710
Loop over 64x64 blocks
715
Loop over four 32x32 blocks
720
785
Intra 64x64
Mode decision,
set Intra64_DC_flag
based on RD64
790
Intra 32x32
Mode decision,
set Intra32_DC_flag
based on RD32
725
Loop over four 16x16 blocks
730
Intra16x16_DC
Mode
735
Other 16x16 Modes
and below
(8x8, 4x4, and etc)
740
16x16 Mode decision based
on RD16, TotRD16 += RD16
745
End Loop (16x16 blocks)
A
B
To FIG. 7B

FIG. 7B

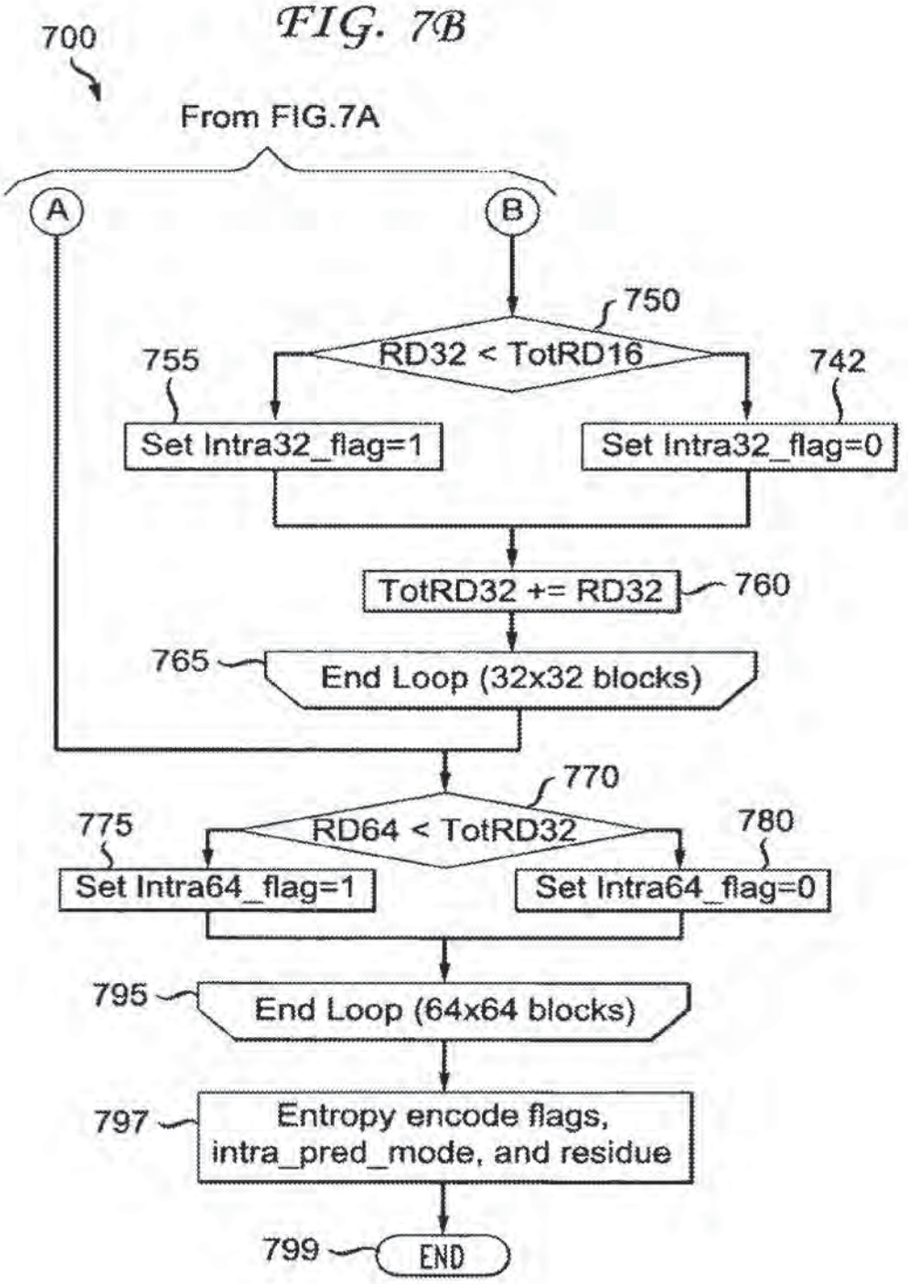

700
From FIG.7A
A
B
750
RD32 < TotRD16
755
742
Set Intra32_flag=1
Set Intra32_flag=0
TotRD32 += RD32
760
765
End Loop (32x32 blocks)
770
RD64 < TotRD32
775
780
Set Intra64_flag=1
Set Intra64_flag=0
795
End Loop (64x64 blocks)
797
Entropy encode flags, intra_pred_mode, and residue
799
END

FIG. 8A

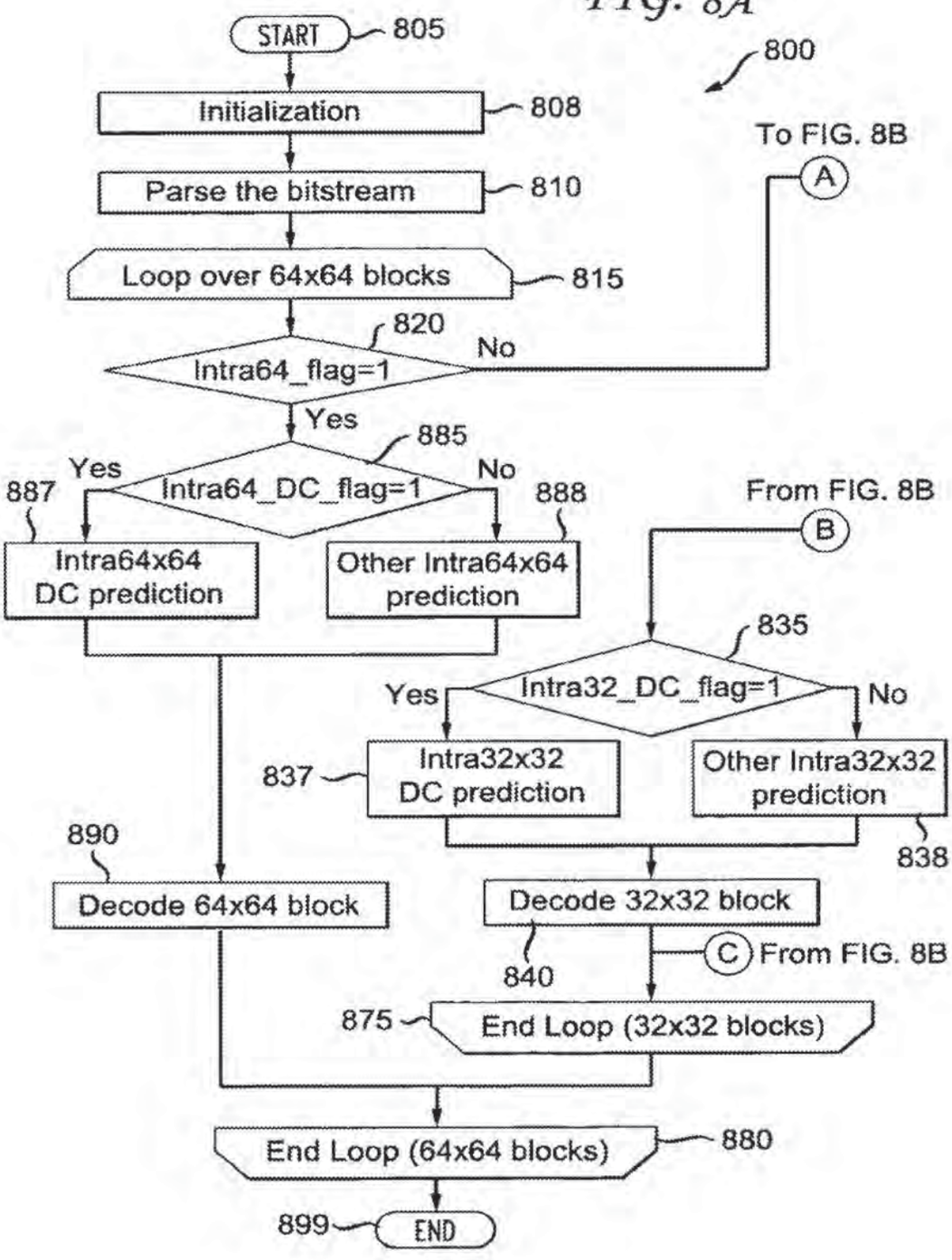
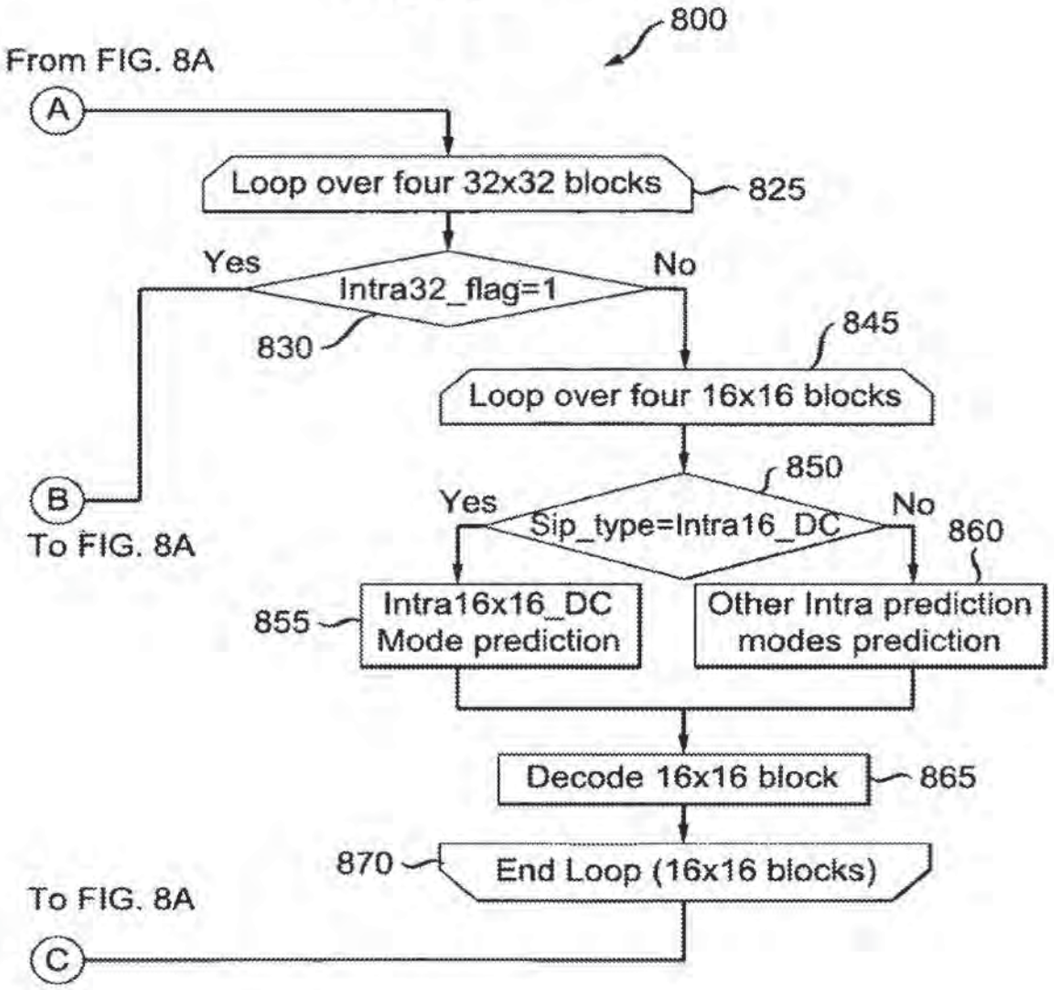

START
805
800
Initialization
808
To FIG. 8B
A
Parse the bitstream
810
Loop over 64x64 blocks
815
820
Intra64_flag=1
No
Yes
885
Yes
Intra64_DC_flag=1
No
887
888
From FIG. 8B
B
Intra64x64
DC prediction
Other Intra64x64
prediction
835
Yes
Intra32_DC_flag=1
No
837
Intra32x32
DC prediction
Other Intra32x32
prediction
838
890
Decode 64x64 block
Decode 32x32 block
840
C
From FIG. 8B
875
End Loop (32x32 blocks)
End Loop (64x64 blocks)
880
899
END

FIG. 8B


800
From FIG. 8A
A
Loop over four 32x32 blocks
825
Yes
Intra32_flag=1
No
830
845
Loop over four 16x16 blocks
B
To FIG. 8A
850
Yes
Sip_type=Intra16_DC
No
860
855
Intra16x16_DC Mode prediction
Other Intra prediction modes prediction
Decode 16x16 block
865
870
End Loop (16x16 blocks)
To FIG. 8A
C

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of co-pending U.S. application Ser. No. 15/806,893 filed Nov. 8, 2017, which is a continuation of U.S. application Ser. No. 13/381,538, filed Dec. 29, 2011, which is a National Phase entry under 35 U.S.C. § 371 of International Application No. PCT/US2010/001862, filed Jun. 29, 2010, which claims the benefit of U.S. Provisional Application Ser. No. 61/222,177, filed Jul. 1, 2009, each of which are incorporated by reference herein in their entireties.

TECHNICAL FIELD

The present principles relate generally to video encoding and decoding and, more particularly, to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

BACKGROUND

Most modern video coding standards employ various coding modes to efficiently reduce the correlations in the spatial and temporal domains. For example, in the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group-4 (MPEG-4) Part 10 Advanced Video Coding (AVC) standard/International Telecommunication Union, Telecommunication Sector (ITU-T) H.264 Recommendation (hereinafter the "MPEG-4 AVC Standard"), a picture can be intra or inter coded. In intra pictures, all macroblocks are coded in intra modes, thus exploiting spatial correlations within the picture. Intra modes can be classified into the following three types: INTRA4×4; INTRA8×8; and INTRA16×16. INTRA4×4 and INTRA8×8 support 9 intra prediction modes and INTRA16×16 supports 4 intra prediction modes.

INTRA4×4 and INTRA8×8 support the following 9 intra prediction modes: vertical; horizontal; DC; diagonal-down/left; diagonal-down/right; vertical-left; horizontal-down; vertical-right; and horizontal-up prediction. INTRA16×16 supports the following 4 intra prediction modes: vertical; horizontal; DC; and plane prediction. Turning to FIG. **1**, INTRA4×4 and INTRA8×8 prediction modes are indicated generally by the reference numeral **100**. In FIG. **1**, the reference numeral **0** indicates a vertical prediction mode, the reference numeral **1** indicates a horizontal prediction mode, the reference numeral **3** indicates a diagonal-down/left prediction mode, the reference numeral **4** indicates a diagonal-down/right prediction mode, the reference numeral **5** indicates a vertical-right prediction mode, the reference numeral **6** indicates a horizontal-down prediction mode, the reference numeral **7** indicates a vertical-left prediction mode, and the reference numeral **8** indicates a horizontal-up prediction mode. DC mode, which is part of the INTRA4×4 and INTRA8×8 prediction modes, is not shown. Turning to FIG. **2**, INTRA16×16 prediction modes are indicated generally by the reference numeral **200**. In FIG. **2**, the reference numeral **0** indicates a vertical prediction mode, the reference numeral **1** indicates a horizontal prediction mode, and the reference numeral **3** indicates a plane prediction mode. DC mode, which is part of the INTRA16×16 prediction modes, is not shown.

INTRA4×4 uses a 4×4 discrete cosine transform (DCT). INTRA8×8 uses 8×8 transforms. INTRA16×16 uses cascaded 4×4 transforms. For the signaling, INTRA4×4 and INTRA8×8 share the same macroblock type (mb_type) 0 and are differentiated by a transform size flag (transform_8×8_size_flag). Then the choice of intra prediction mode in INTRA4×4 or INTRA8×8 is signaled by the most probable mode possibly with a remaining mode if necessary. For INTRA16×16, all the intra prediction modes along with a coded block pattern (cbp) type are signaled in mb_type, which uses an mb_type value from 1 to 24. TABLE 1 shows the detailed signaling for macroblock types for Intra coded slices (I slices). If a larger block size than 16×16 is used for intra prediction, then several possible issues are faced, as follows.

(1) If INTRA32×32 or INTRA64×64 prediction are added, by simply extending the mb_type in the MPEG-4 AVC Standard, it will cause too much overhead for those two new modes and, in addition, will not allow a hierarchical type of intra prediction. An example of a hierarchical type of intra prediction is explained as follows. If a 32×32 block is used as a large block and we allow sub-partitions to be 16×16 then, for each 16×16 sub-partition, INTRA4×4, INTRA8×8, INTRA16×16 should be allowed.

(2) If a larger transform (such as a 16×16 transform) instead of a cascaded transform is used for INTRA16×16, then the current signaling cannot be applied.

(3) We should give different priority for intra prediction modes inside one intra partition type.

TABLE 1

mb_type	Name of mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	I_NxN	0	Intra_4x4	na	Equation 7-33	Equation 7-33
0	I_NxN	1	Intra_8x8	na	Equation 7-33	Equation 7-33
1	I_16x16_0_0_0	na	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0

TABLE 1-continued

mb_type	Name of mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15
25	I_PCM	na	na	na	na	na

Some prior art approaches exist relating to signaling large motion (inter) partitions in extensions of the MPEG-4 AVC Standard. One example of how large motion (Inter) partitions are signaled in extensions of the MPEG-4 AVC Standard is described with respect to a first prior art approach. The first prior art approach describes how signaling is done for 32×32 blocks or 64×64 blocks using a hierarchical coding structure.

Moreover, in addition to the existing motion partition sizes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4) in the MPEG-4 AVC Standard, inter coding has also been proposed for an extension of the MPEG-4 AVC Standard using 32×32, 32×16, and 16×32 partitions. Turning to FIG. **3**, motion partitions for use in 32×32 blocks are indicated generally by the reference numeral **300**. The partitions include 32×32, 32×16, 16×32, and 16×16. 16×16 partitions can be further partitioned into partitions of sizes 16×16, 16×8, 8×16, and 8×8. Moreover, 8×8 partitions can be further partitioned into partitions of sizes 8×8, 8×4, 4×8, and 4×4.

For each 32×32 block, SKIP mode or DIRECT mode is signaled using an mb32_skip_flag in a way similar to that performed for other modes of the MPEG-4 AVC Standard. In addition, the original mb_type for an M×N (M=8, or 16 and N=8, or 16) partition in the MPEG-4 AVC Standard is also used to signal a 2M×2N partition in a 32×32 block. If the mb32_type of a 32×32 indicates that a 16×16 partition is used, then the four 16×16 blocks are signaled in raster scan order by using the same syntax elements as macroblock_layer( ) in the MPEG-4 AVC Standard. Each 16×16 block may be partitioned further in the quadtree manner, from size 16×16 down to size 4×4.

For macroblock size 64×64, the following partitions are added above the partitions used in 32×32 blocks: 64×64; 64×32; and 32×64. Thus, one more hierarchical layer is added in the macroblock partition above block size 32×32. The original mb_type for an M×N (M=8, or 16 and N=8, or 16) macroblock partition in the MPEG-4 AVC Standard is used to signal a 4M×4N macroblock partition in 64×64 macroblocks. If a 32×32 macroblock partition is used for a 64×64 block, then each 32×32 block will be handled in the same manner as described above.

However, the existing literature does not address how large intra mode is signaled, where large intra mode is defined to mean intra prediction involving partition blocks having a size equal to or larger than 32×32.

SUMMARY

These and other drawbacks and disadvantages of the prior art are addressed by the present principles, which are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

According to an aspect of the present principles, an apparatus is provided. The apparatus includes a video encoder for encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to another aspect of the present principles, a method in a video encoder is provided. The method includes encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to yet another aspect of the present principles, an apparatus is provided. The apparatus includes a video decoder for decoding picture data for at least one large block in a picture by determining that intra prediction is to be performed for the at least one large block. The intra prediction is determined by determining a basic coding unit size and determining a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is determinable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to still another aspect of the present principles, a method is provided in a video decoder. The method

includes decoding picture data for at least one large block in a picture by determining that intra prediction is to be performed for the at least one large block. The intra prediction is determined by determining a basic coding unit size and determining a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is determinable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

These and other aspects, features and advantages of the present principles will become apparent from the following detailed description of exemplary embodiments, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present principles may be better understood in accordance with the following exemplary figures, in which:

FIG. **1** is a diagram showing INTRA4×4 and INTRA8×8 prediction modes to which the present principles may be applied;

FIG. **2** is a diagram showing INTRA16×16 prediction modes to which the present principles may be applied;

FIG. **3** is a diagram showing motion partitions for use in 32×32 blocks to which the present principles may be applied;

FIG. **4** is a block diagram for an exemplary video encoder to which the present principles may be applied in accordance with an embodiment of the present principles;

FIG. **5** is a block diagram for an exemplary video decoder to which the present principles may be applied in accordance with an embodiment of the present principles;

FIG. **6** is a block diagram for exemplary hierarchical partitions to which the present principles may be applied in accordance with an embodiment of the present principles;

FIGS. **7**A and **7**B represent a flow diagram for an exemplary method for encoding picture data for large blocks by signaling intra prediction for the large blocks in accordance with an embodiment of the present principles; and

FIGS. **8**A and **8**B represent a flow diagram showing an exemplary method for decoding picture data for large blocks by determining that intra prediction is to be applied to the large blocks in accordance with an embodiment of the present principles.

DETAILED DESCRIPTION

The present principles are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

The present description illustrates the present principles. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the present principles and are included within its spirit and scope.

All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the present principles and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions.

Moreover, all statements herein reciting principles, aspects, and embodiments of the present principles, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

Thus, for example, it will be appreciated by those skilled in the art that the block diagrams presented herein represent conceptual views of illustrative circuitry embodying the present principles. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudocode, and the like represent various processes which may be substantially represented in computer readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

The functions of the various elements shown in the figures may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" or "controller" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor ("DSP") hardware, read-only memory ("ROM") for storing software, random access memory ("RAM"), and non-volatile storage.

Other hardware, conventional and/or custom, may also be included. Similarly, any switches shown in the figures are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementer as more specifically understood from the context.

In the claims hereof, any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a) a combination of circuit elements that performs that function or b) software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The present principles as defined by such claims reside in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the claims call for. It is thus regarded that any means that can provide those functionalities are equivalent to those shown herein.

Reference in the specification to "one embodiment" or "an embodiment" of the present principles, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment of the present principles. Thus, the appearances of the phrase "in one embodiment" or "in an embodiment", as well any other variations, appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further

example, in the cases of "A, B, and/or C" and "at least one of A, B, and C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as readily apparent by one of ordinary skill in this and related arts, for as many items listed.

Moreover, it is to be appreciated that while one or more embodiments of the present principles are described herein with respect to an extension of the MPEG-4 AVC standard, the present principles are not limited to solely this extension and/or this standard and, thus, may be utilized with respect to other video coding standards, recommendations, and extensions thereof, while maintaining the spirit of the present principles.

As used herein, "high level syntax" refers to syntax present in the bitstream that resides hierarchically above the macroblock layer. For example, high level syntax, as used herein, may refer to, but is not limited to, syntax at the slice header level, Supplemental Enhancement Information (SEI) level, Picture Parameter Set (PPS) level, Sequence Parameter Set (SPS) level and Network Abstraction Layer (NAL) unit header level.

Also, as used herein, the words "picture" and "image" are used interchangeably and refer to a still image or a picture from a video sequence. As is known, a picture may be a frame or a field.

Additionally, as used herein, the word "signal" refers to indicating something to a corresponding decoder. For example, the encoder may signal that intra prediction is designated to be used for a particular large block (as defined herein) in order to make the decoder aware of which particular prediction type (e.g., intra or inter) was used on the encoder side. In this way, the same prediction type may be used at both the encoder side and the decoder side. Thus, for example, an encoder may transmit an indication (i.e., signal) for a particular large block that intra prediction is to be performed on that large block to simply allow the decoder to know and select the same prediction type for that large block. It is to be appreciated that signaling may be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth may be used to signal information to a corresponding decoder.

Turning to FIG. **4**, an exemplary video encoder to which the present principles may be applied in accordance with an embodiment of the present principles is indicated generally by the reference numeral **400**.

The video encoder **400** includes a frame ordering buffer **410** having an output in signal communication with a non-inverting input of a combiner **485**. An output of the combiner **485** is connected in signal communication with a first input of a transformer and quantizer **425**. An output of the transformer and quantizer **425** is connected in signal communication with a first input of an entropy coder **445** and a first input of an inverse transformer and inverse quantizer **450**. An output of the entropy coder **445** is connected in signal communication with a first non-inverting input of a combiner **490**. An output of the combiner **490** is connected in signal communication with a first input of an output buffer **435**.

A first output of an encoder controller **405** is connected in signal communication with a second input of the frame ordering buffer **410**, a second input of the inverse transformer and inverse quantizer **450**, an input of a picture-type decision module **415**, an input of a macroblock-type (MB-type) decision module **420**, a second input of a super intra prediction module **460**, a second input of a deblocking filter **465**, a first input of a motion compensator **470**, a first input of a motion estimator **475**, and a second input of a reference picture buffer **480**.

A second output of the encoder controller **405** is connected in signal communication with a first input of a Supplemental Enhancement Information (SEI) inserter **430**, a second input of the transformer and quantizer **425**, a second input of the entropy coder **445**, a second input of the output buffer **435**, and an input of the Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) inserter **440**.

A first output of the picture-type decision module **415** is connected in signal communication with a third input of a frame ordering buffer **410**. A second output of the picture-type decision module **415** is connected in signal communication with a second input of a macroblock-type decision module **420**.

An output of the Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) inserter **440** is connected in signal communication with a third non-inverting input of the combiner **490**.

An output of the inverse quantizer and inverse transformer **450** is connected in signal communication with a first non-inverting input of a combiner **419**. An output of the combiner **419** is connected in signal communication with a first input of the super intra prediction module **460** and a first input of the deblocking filter **465**. An output of the deblocking filter **465** is connected in signal communication with a first input of a reference picture buffer **480**. An output of the reference picture buffer **480** is connected in signal communication with a second input of the motion estimator **475**. A first output of the motion estimator **475** is connected in signal communication with a second input of the motion compensator **470**. A second output of the motion estimator **475** is connected in signal communication with a third input of the entropy coder **445**.

An output of the motion compensator **470** is connected in signal communication with a first input of a switch **497**. An output of the super intra prediction module **460** is connected in signal communication with a second input of the switch **497**. An output of the macroblock-type decision module **420** is connected in signal communication with a third input of the switch **497**. The third input of the switch **497** determines whether or not the "data" input of the switch (as compared to the control input, i.e., the third input) is to be provided by the motion compensator **470** or the super intra prediction module **460**. The output of the switch **497** is connected in signal communication with a second non-inverting input of the combiner **419** and with an inverting input of the combiner **485**.

Inputs of the frame ordering buffer **410** and the encoder controller **405** are available as input of the encoder **400**, for receiving an input picture **401**. Moreover, an input of the Supplemental Enhancement Information (SEI) inserter **430** is available as an input of the encoder **400**, for receiving metadata. An output of the output buffer **435** is available as an output of the encoder **400**, for outputting a bitstream.

Turning to FIG. **5**, an exemplary video decoder to which the present principles may be applied in accordance with an embodiment of the present principles is indicated generally by the reference numeral **500**.

The video decoder **500** includes an input buffer **510** having an output connected in signal communication with a

first input of the entropy decoder **545**. A first output of the entropy decoder **545** is connected in signal communication with a first input of an inverse transformer and inverse quantizer **550**. An output of the inverse transformer and inverse quantizer **550** is connected in signal communication with a second non-inverting input of a combiner **525**. An output of the combiner **525** is connected in signal communication with a second input of a deblocking filter **565** and a first input of a super intra prediction module **560**. A second output of the deblocking filter **565** is connected in signal communication with a first input of a reference picture buffer **580**. An output of the reference picture buffer **580** is connected in signal communication with a second input of a motion compensator **570**.

A second output of the entropy decoder **545** is connected in signal communication with a third input of the motion compensator **570** and a first input of the deblocking filter **565**. A third output of the entropy decoder **545** is connected in signal communication with an input of a decoder controller **505**. A first output of the decoder controller **505** is connected in signal communication with a second input of the entropy decoder **545**. A second output of the decoder controller **505** is connected in signal communication with a second input of the inverse transformer and inverse quantizer **550**. A third output of the decoder controller **505** is connected in signal communication with a third input of the deblocking filter **565**. A fourth output of the decoder controller **505** is connected in signal communication with a second input of the super intra prediction module **560**, with a first input of the motion compensator **570**, and with a second input of the reference picture buffer **580**.

An output of the motion compensator **570** is connected in signal communication with a first input of a switch **597**. An output of the super intra prediction module **560** is connected in signal communication with a second input of the switch **597**. An output of the switch **597** is connected in signal communication with a first non-inverting input of the combiner **525**.

An input of the input buffer **510** is available as an input of the decoder **500**, for receiving an input bitstream. A first output of the deblocking filter **565** is available as an output of the decoder **500**, for outputting an output picture.

As noted above, the present principles are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders. Moreover, as noted above, large blocks, to which the present principles may be applied, are defined to means blocks having a size equal to or larger than 32×32.

In an embodiment, for ease of notation, we split the signaling of intra prediction into the following two parts: sip_type (spatial intra partition type, which can be INTRA4×4, INTRA8×8, INTRA16×16, and so forth); and intra_pred_mode (such as, e.g., the 9 Intra Prediction Modes within INTRA4×4 and INTRA8×8) within each sip_type. In further detail with respect to a particular embodiment, we propose the following three rules for the present principles: (1) select a basic coding unit; (2) allow hierarchical layer intra prediction by either splitting from a largest intra prediction type or merging from the basic coding unit; and (3) for each sip_type, assign a higher priority to the intra_pred_mode which is most frequently used. With respect to Rule (1), we allow several sip_types for the basic coding unit.

An Embodiment

In an embodiment, we set the basic coding unit to be 16×16. In this coding unit, we allow sip_type to be INTRA4×4, INTRA8×8 and INTRA16×16. We also allow hierarchical layer intra prediction as shown in FIG. **6**.

Turning to FIG. **6**, exemplary hierarchical partitions to which the present principles may be applied are indicated generally by the reference numeral **600**. In this embodiment, if the largest block size is set to 64×64, then we use "split signaling" to allow hierarchical layer intra prediction. That is, in an embodiment, we add intra64_flag. If intra64_flag is equal to 1, then INTRA64×64 is used. Otherwise, if intra64_flag is equal to 0, then we split a 64×64 block **611** into four 32×32 blocks **621**. For each of the 32×32 blocks **621**, we add intra32_flag. If intra32_flag is equal to 1, then INTRA32×32 is used. Otherwise, if intra32_flag is equal to 0, then all sip_types allowed in the 16×16 basic coding unit are allowed here (i.e., for the 32×32 blocks **621**) as well. For intra_pred_mode in INTRA16×16, we have DC mode and the directional modes, the latter allowing different types of directional predictions by sending the mode information. Thus, a 32×32 intra prediction block **621** may be further split into 4 16×16 intra prediction blocks **631**. One or more of the 4 16×16 intra prediction blocks **631** may be further split to DC mode (not shown), a 16×16 mode **641**, an 8×8 mode **651**, and a 4×4 mode **661**. In this embodiment, we presume having the following four 16×16 intra prediction modes: DC; horizontal (HOR); vertical (VER); and multi-directional (Multi-DIR). The intra_pred_mode is signaled by considering the priority of each mode. In INTRA16×16, since the DC mode is used more often than other modes, we add INTRA16×16_DC in the sip_type table before INTRA16×16. Then we remove the most_probable_mode indication in intra_pred_mode for INTRA16×16. Instead, the other 3 modes (16×16, 8×8, and 4×4) are absolutely indicated.

Syntax

We illustrate an example of syntax for this embodiment in TABLE 2 and TABLE 3. In particular, TABLE 2 shows an exemplary specification of sip_type for a 16×16 coding unit in accordance with an embodiment of the present principles, and TABLE 3 shows exemplary INTRA16×16 prediction modes in accordance with an embodiment of the present principles. For INTRA32×32/INTRA64×64, the same modes as INTRA16×16 are used. For signaling, we replace the most_probable_mode indication with intra32_DC_flag and intra64_DC flag since DC is mostly used. Then we absolutely code other intra_pred_mode.

intra_pred_mode signaling for INTRA4×4 and INTRA8×8 may be performed exactly the same as in the MPEG-4 AVC Standard, so we will not list these modes in any of the tables.

TABLE 2

Sip_type	Index	Binarization bits
SIP8x8	0	0
SIP16x16DC	1	10
SIP16x16	2	110
SIP4x4	3	1110

TABLE 3

Intra Prediction Mode	Index	Binarization bits
VER	0	0
HOR	1	10
Multi-DIR	2	11

TABLE 4 shows exemplary macroblock layer syntax, in accordance with an embodiment of the present principles.

TABLE 4

macroblock_layer) {	C	Descriptor
. . .		
intra64_flag	2	u(1)
if (intra64_flag==1) {		
intra64_DC_flag	2	u(1)
if (intra64_DC_flag==0) {		
intra_pred_mode_64	2	ue(v)/se(v)
if (intra_pred_mode_64==Multi-DIR) {		
intra64_multidir_index	2	ue(v)/se(v)
}		
}		
}		
else {		
for (i32=0; i32<4; i32++) {		
intra32_flag[i32]	2	u(1)
if (intra32_flag[i32]==1) {		
intra32_DC_flag[i32]	2	u(1)
if (intra32_DC_flag[i32]==0) {		
intra_pred_mode_32[i32]	2	ue(v)/se(v)
if (intra_pred_mode_32[i32]==Multi-DIR) {		
intra32_multidir_index[i32]	2	ue(v)/se(v)
}		
}		
}		
else {		
for (i16=0; i16<4; i16++) {		
sip_type[i16]	2	ue(v)/se(v)
if (sip_type[i16]==SIP16x16) {		
intra_pred_mode_16[i16]	2	ue(v)/se(v)
if (intra_pred_mode_16[i16]==Multi-DIR) {		
intra16_multidir_index[i16]	2	ue(v)/se(v)
}		
}		
else if (sip_type[i16]!=SIP16x16_DC) {		
mb_intra_prediction_syntax( ); /*this is same as H.264*/		
}		
}		
}		
. . .		
}		

The semantics of some of the syntax elements of TABLE 4 are as follows:

Intra64_flag equal to 1 specifies that INTRA64×64 is used. Intra64_flag equal to 0 specifies that a 64×64 large block is further split into 32×32 partitions.

Intra64_DC_flag equal to 1 specifies that the intra_pred_mode is DC mode for INTRA64×64. Intra64_DC_flag equal to 0 specifies that the intra_pred_mode is not DC mode for INTRA64×64.

intra_pred_mode_64 specifies intra prediction mode (not including DC mode) for INTRA64×64.

intra64_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA64×64.

Intra32_flag [i] equal to 1 specifies that INTRA32×32 is used for the ith 32×32 large block. Intra32_flag [i] equal to 0 specifies that the ith 32×32 large block is further split into 16×16 partitions.

intra32_DC_flag [i] equal to 1 specifies that the intra_pred_mode is DC mode for INTRA32×32 for the ith 32×32 block. intra32_DC_flag[i] equal to 0 specifies that the intra_pred_mode is not DC mode for INTRA32×32 for the ith 32×32 block.

intra_pred_mode_32 [i] specifies the intra prediction mode (not including DC mode) for INTRA32×32 for the ith 32×32 large block.

intra32_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA32×32.

sip_type [i] specifies the spatial intra partition type for the basic block coding unit in the ith 16×16 block.

intra_pred_mode_16 [i] specifies the intra prediction mode (not including DC mode) for INTRA16×16 for the ith 16×16 block.

Intra16_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA16×16 for the ith 16×16 block.

Another Embodiment

In another embodiment, we adaptively select the large block unit to be 32×32 or 64×64. The selection may be signaled using one or more high level syntax elements. In an embodiment, if 32×32 is selected, then we just remove all the syntax related to 64×64.

In another embodiment, hierarchical intra layer prediction can involve merging from basic coding unit. For example, if the largest block unit is 64×64 and the basic coding unit is 16×16, then we use one flag (is_all_16×16_coding) to indicate if all of the 16×16 blocks inside one 64×64 block are of the 16×16 coding type. If is_all_16×16_coding is

equal to 1, then this indicates that 16×16 coding type is used and we stop signaling. Otherwise, we use one flag (is_all_32x32_coding) to indicate if all of the 32×32 blocks inside one 64×64 block are of the 32×32 coding type. If is_all_32x32_coding is equal to 1, then this indicates that all of the 32×32 blocks inside one 64×64 block are of the 32×32 coding type. Otherwise, if is_all_32x32_coding and is_all_16×16_coding are equal to 0, then this indicates that INTRA64×64 is used.

In another embodiment, we introduce a sip_type for a block unit (large_sip_type) having a size not less than 16×16. The three types are referred to as follows: large_intra_16×16, large_intra_32×32, and large_intra_64×64. large_intra_16×16 means that all 16×16 blocks inside one large block are of the 16×16 coding type. large_intra_32×32 means that all 32×32 blocks inside one large block are of the 32×32 coding type. In an embodiment, large_intra_32×32 can be combined with the embodiment described above with intra32_flag to allow hierarchical intra prediction. large_intra_64×64 means that all 64×64 blocks inside one large block are coded as INTRA64×64.

In another embodiment, we can introduce several sip/mode tables. The tables can be pre-stored at both an encoder and a decoder, or the tables can be user specified and transmitted using one or more high level syntax elements. TABLE 5 shows exemplary macroblock layer syntax, in accordance with an embodiment of the present principles.

TABLE 5

macroblock_layer( ) {	C	Descriptor
...		
is_all_16x16_coding	2	u(1)
if (is_all_16x16_coding == 0) {		
is_all_32x32_coding	2	u(1)
if (is_all_32x32_coding == 0) {		
decode_with_64x64_coding_type( )		
}		
else{		
decode_with_32x32_coding_type( )		
}		
}		
else {		
decode_with_16x16_coding_type( )		
}		
...		
}		

The semantics of the some of the syntax elements of TABLE 5 are as follows:

is_all_16×16_coding equal to 1 specifies that all 16×16 blocks inside a large block are coded by a 16×16 coding type. is_all_16×16_coding equal to 0 specifies that the large block is not coded by a 16×16 coding type.

is_all_32×32_coding equal to 1 specifies that all 32×32 blocks inside a large block are coded by a 32×32 coding type. is_all_32×32_coding equal to 0 specifies that the large block is not coded by a 32×32 coding type.

Turning to FIGS. 7A and 7B, which together represent an exemplary method for encoding picture data for large blocks by signaling intra prediction for the large blocks, indicated generally by the reference numeral **700**. The method **700** includes a start block **705** that passes control to a function block **710**. The function block **710** performs an initialization, and passes control to a loop limit block **715**. The loop limit block **715** performs a loop (hereinafter also loop **1**) over 64×64 blocks (i.e., blocks having a block size of 64×64), and passes control to a function block **785** and a loop limit block **720**.

The function block **785** performs an Intra 64×64 mode decision, sets Intra64_DC flag based on RD64 (i.e., the rate distortion resulting from the Intra 64×64 mode decision), and passes control to a decision block **770**.

The loop limit block **720** performs a loop (hereinafter also loop **2**) over four 32×32 blocks (i.e., the four blocks having a block size of 32×32 and obtained from the current 64×64 block being processed by loop **1**), and passes control to a function block **790** and a loop limit block **725**.

The function block **790** performs an Intra 32×32 mode decision, sets Intra32_DC flag based on RD32 (i.e., the rate distortion resulting from the Intra 32×32 mode decision), and passes control to a decision block **750**.

The loop limit block **725** performs a loop (hereinafter also loop **3**) over four 16×16 blocks (i.e., the four blocks having a block size of 16×16 and obtained from the current 32×32 block being processed by loop **2**), and passes control to a function block **730** and a function block **735**.

The function block **730** evaluates Intra16×16_DC mode, and passes control to a function block **740**. The function block **735** evaluates other 16×16 modes (i.e., other than Intra16×16_DC) and below (e.g., 8×8, 4×4, etc.), and passes control to the function block **740**.

The function block **740** performs a 16×16 mode decision based on RD16 (i.e., the rate distortion resulting from the intra 16×16 mode decision), then accumulates the RD16 of each 16×16 block to obtain TotRD16 (which indicates the total rate distortion of the whole 32×32 block when coded by four 16×16 blocks), and passes control to a loop limit block **745**. The loop limit block **745** ends the loop (i.e., loop **3**) over the 16×16 blocks, and passes control to the decision block **750**.

The decision block **750** determines whether or not RD32<TotRD16 (i.e., whether the rate distortion cost for a current 32×32 block is less than the total rate distortion cost for the four 16×16 blocks obtained from the current 32×32 block). If so, then control is passed to a function block **755**. Otherwise, control is passed to a function block **742**.

The function block **755** sets Intra32_flag equal to one, and passes control to a function block **760**. The function block **742** sets Intra32_flag equal to zero, and passes control to the function block **760**.

The function block **760** sets an accumulation of the RD32 of each 32×32 block to TotRD32 to indicate the total rate distortion of the whole 64×64 block when coded by four 32×32 blocks, and passes control to a loop limit block **765**. The loop limit block **765** ends the loop (i.e., loop **2**) over the 32×32 blocks, and passes control to a decision block **770**.

The decision block **770** determines whether or not RD64<TotRD32 (i.e., whether the rate distortion cost for a current 64×64 block is less than the total rate distortion cost for the four 32×32 blocks obtained from the current 64×64 block). If so, then control is passed to a function block **775**. Otherwise, control is passed to a function block **780**.

The function block **775** sets Intra64_flag equal to one, and passes control to a loop limit block **795**. The function block **780** sets Intra64_flag equal to zero, and passes control to the function block **795**.

The function block **795** ends the loop (i.e., loop **1**) over the 64×64 blocks, and passes control to a function block **797**. The function block **797** entropy codes flags, intra_pred_mode, and a residue, and passes control to an end block **799**.

Turning to FIGS. **8**A and **8**B, which together represent an exemplary method for decoding picture data for large blocks by determining that intra prediction is to be applied to the large blocks, indicated generally by the reference numeral

800. The method 800 includes a start block 805 that passes control to a function block 808. The function block 808 initializes the decoder and then passes the control to a function block 810. The function block 810 parses the bitstream, and passes control to a loop limit block 815. The loop limit block 815 performs a loop (hereinafter loop 1) over 64×64 blocks, and passes control to a decision block 820. The decision block 820 determines whether or not Intra64_flag is set equal to one. If so ("Yes"), then control is passed to a function block 885. Otherwise ("No"), control is passed to a loop limit block 825.

The function block 885 determines whether or not intra64_DC_flag is set equal to one. If so ("Yes"), then control is passed to a function block 887. Otherwise ("No"), control is passed to a function block 888. The function block 887 performs intra 64×64 DC prediction, and then passes control to a function block 890. The function block 888 performs intra 64×64 predictions other than intra 64×64 DC mode and then passes control to a function block 890. The function block 890 decodes a current 64×64 block, and passes control to a loop limit block 880. The loop limit block 880 ends the loop (i.e., loop 1) over the 64×64 blocks, and passes control to an end block 899.

The loop limit block 825 performs a loop (hereinafter loop 2) over four 32×32 blocks, and passes control to a decision block 830. The decision block 830 determines whether or not Intra32_flag is equal to one. If so ("Yes"), then control is passed to a function block 835. Otherwise ("No"), control is passed to a loop limit block 845.

The function block 835 determines whether or not Intra32_DC_flag is equal to one. If so ("Yes"), then control is passed to a function block 837. Otherwise ("No"), control is passed to a function block 838. The function block 837 performs an intra32×32 DC prediction, and passes control to a function block 840. The function block 838 performs an intra prediction other than intra 32×32 DC mode, and then passed the control to the function block 840. The function block 840 decodes a 32×32 block, and passes control to a loop limit block 875.

The loop limit block 875 ends the loop (i.e., loop 2) over the 32×32 blocks, and passes control to the loop limit block 880.

The loop limit block 845 performs a loop (hereinafter loop 3) over four 16×16 blocks, and passes control to a decision block 850. The decision block 850 determines whether or not sip_type=Intra16_DC. If so ("Yes"), then control is passed to a function block 855. Otherwise ("No"), control is passed to a function block 860.

The function block 855 performs Intra16×16_DC mode prediction, and passes control to a function block 865. The function block 860 performs mode prediction using other intra prediction modes (i.e., other than Intra16×16_DC mode), and passes control to the function block 865.

The function block 865 decodes a 16×16 block, and passes control to a loop limit block 870. The loop limit block 870 ends the loop (i.e., loop 3) over the 16×16 blocks, and passes control to the loop limit block 875.

A description will now be given of some of the many attendant advantages/features of the present invention, some of which have been mentioned above. For example, one advantage/feature is an apparatus having a video encoder for encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

Another advantage/feature is the apparatus having the video encoder as described above, wherein for each of the plurality of spatial intra partition types, a higher priority is assigned to a particular intra prediction mode that is most frequently used from among a plurality of available intra prediction modes.

Yet another advantage/feature is the apparatus having the video encoder as described above, wherein the large block size is adaptively selected.

Still another advantage/feature is the apparatus having the video encoder as described above, wherein the signaling is performed using one or more high level syntax elements.

Moreover, another advantage/feature is the apparatus having the video encoder as described above, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used by the video encoder to encode the at least one large block. The at least one of the spatial intra partition type table and the intra prediction mode table is arranged to be pre-stored and used by a corresponding video decoder to decode the at least one large block.

Further, another advantage/feature is the apparatus having the video encoder as described above, wherein at least one of a spatial intra partition type table and an intra prediction mode table, is used by the video encoder to encode the at least one large block, and is transmitted by the video encoder using one or more high level syntax elements.

These and other features and advantages of the present principles may be readily ascertained by one of ordinary skill in the pertinent art based on the teachings herein. It is to be understood that the teachings of the present principles may be implemented in various forms of hardware, software, firmware, special purpose processors, or combinations thereof.

Most preferably, the teachings of the present principles are implemented as a combination of hardware and software. Moreover, the software may be implemented as an application program tangibly embodied on a program storage unit. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units ("CPU"), a random access memory ("RAM"), and input/output ("I/O") interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU. In addition, various other peripheral units may be connected to the computer platform such as an additional data storage unit and a printing unit.

It is to be further understood that, because some of the constituent system components and methods depicted in the accompanying drawings are preferably implemented in software, the actual connections between the system components or the process function blocks may differ depending upon the manner in which the present principles are programmed. Given the teachings herein, one of ordinary skill

in the pertinent art will be able to contemplate these and similar implementations or configurations of the present principles.

Although the illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present principles is not limited to those precise embodiments, and that various changes and modifications may be effected therein by one of ordinary skill in the pertinent art without departing from the scope or spirit of the present principles. All such changes and modifications are intended to be included within the scope of the present principles as set forth in the appended claims.

The invention claimed is:

1. A method of decoding picture data for at least a large block in a picture, the method comprising:

determining that intra prediction is to be performed for the large block, wherein the large block has a block size equal to or larger than 32×32, and the large block size is greater than a size of a basic coding unit;

decoding a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks;

decoding an intra prediction mode for the large block that is not further split into four equally sized sub-blocks;

partitioning the large block into the four equally sized sub-blocks in response to the decoded binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks; and

decoding a second syntax element indicating a single spatial intra partition type for at least one basic coding unit sized sub-block, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

2. The method of claim **1**, further comprising decoding an intra prediction mode for the basic coding unit sized sub-block.

3. The method of claim **1**, further comprising decoding a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks greater than a basic coding unit size is further split into four smaller equally sized sub-blocks.

4. The method of claim **1**, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

5. An apparatus, comprising:

at least a memory and one or more processors, wherein the one or more processors are configured to:

decode picture data for at least a large block in a picture by determining that intra prediction is to be performed for the large block, wherein the large block has a large block size equal to or larger than 32×32, and the large block size is greater than a basic coding unit size, wherein the intra prediction is signaled for the large block and the one or more processors are configured to:

decode a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks;

decode an intra prediction mode for the large block that is not further split into four equally sized sub-blocks;

partition the large block into the four equally sized sub-blocks in response to the decoded binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks; and

decode a second syntax element indicating a single spatial intra partition type for at least one basic coding unit sized sub-block, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

6. The apparatus of claim **5**, wherein the one or more processors are configured to decode an intra prediction mode for the basic coding unit sized sub-block.

7. The apparatus of claim **5**, wherein the one or more processors are configured to decode a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks greater than a basic coding unit size is further split into four smaller equally sized sub-blocks.

8. The apparatus of claim **5**, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit.

9. The apparatus of claim **5**, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used by the one or more processors to decode picture data for the large block.

10. The apparatus of claim **5**, wherein at least one of a spatial intra partition type table and an intra prediction mode table is received by the one or more processors using one or more high level syntax elements and is used by the one or more processors to decode picture data for the large block.

11. A method of encoding picture data for at least a large block in a picture, the method comprising:

determining that intra prediction is to be performed for the large block, wherein the large block has a block size equal to or larger than 32×32, and the large block size is greater than a size of a basic coding unit;

encoding a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks;

encoding an intra prediction mode for the large block that is not further split into four equally sized sub-blocks;

partitioning the large block into the four equally sized sub-blocks in response to the decoded binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks; and

encoding a second syntax element indicating a single spatial intra partition type for at least one basic coding unit sized sub-block, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

12. The method of claim **11**, further comprising encoding an intra prediction mode for the basic coding unit sized sub-block.

13. The method of claim **11**, further comprising encoding a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks greater than a basic coding unit size is further split into four smaller equally sized sub-blocks.

14. The method of claim **11**, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

15. An apparatus comprising:

at least a memory and one or more processors, wherein the one or more processors are configured to:

encode picture data for at least a large block in a picture by determining that intra prediction is to be performed for the large block, wherein the large block has a large block size equal to or larger than 32×32, and the large block size is greater than a basic coding unit size, wherein the intra prediction is signaled for the large block and the one or more processors are configured to:

encode a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks;

encode an intra prediction mode for the large block that is not further split into four equally sized sub-blocks;

partition the large block into the four equally sized sub-blocks in response to the binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks; and

encode a second syntax element indicating a single spatial intra partition type for at least one basic coding unit sized sub-block, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

16. The apparatus of claim **15**, wherein, the one or more processors are configured to encode an intra prediction mode for the basic coding unit sized sub-block.

17. The apparatus of claim **15**, wherein, the one or more processors are configured to encode a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks greater than a basic coding unit size is further split into four smaller equally sized sub-blocks.

18. The method of claim **15**, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2654

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
17/351,649	06/18/2021 RULE	375	2487	ICVW_2009P00073 US02

APPLICANTS

InterDigital VC Holdings, Inc., Wilmington, DE;

INVENTORS

Yunfei Zheng, San Jose, CA;
Qian Xu, Folsom, CA;
Xiaoan Lu, Yardley, PA;
Peng Yin, Ithaca, NY;
Joel Sole Rojals, San Diego, CA;
Adeel Abbas, Passaic, NJ;

**** CONTINUING DATA ***************************

This application is a CON of 15/806,893 11/08/2017 PAT 11082697
which is a CON of 13/381,538 12/29/2011 ABN
which is a 371 of PCT/US2010/001862 06/29/2010
which claims benefit of 61/222,177 07/01/2009

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

06/30/2021

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/Tyler W Sullivan/ Examiner's Signature	CA	9	20	4
	☐ Met after Allowance	TWS Initials				

ADDRESS

Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103
UNITED STATES

TITLE

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

FILING FEE RECEIVED	FEES: Authority has been given in Paper
0.00	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:

☐ All Fees
☐ 1.16 Fees (Filing)
☐ 1.17 Fees (Processing Ext. of time)
☐ 1.18 Fees (Issue)
☐ Other _____

BIB (Rev. 05/07).

		❑ Credit

BIB (Rev. 05/07).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2654

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
17/351,649	06/18/2021 RULE	375	2487	ICVW_2009P00073 US02

APPLICANTS

InterDigital VC Holdings, Inc., Wilmington, DE;

INVENTORS

Yunfei Zheng, San Jose, CA;
Qian Xu, Folsom, CA;
Xiaoan Lu, Yardley, PA;
Peng Yin, Ithaca, NY;
Joel Sole Rojals, San Diego, CA;
Adeel Abbas, Passaic, NJ;

**** CONTINUING DATA ***************************

This application is a CON of 15/806,893 11/08/2017 PAT 11082697
which is a CON of 13/381,538 12/29/2011 ABN
which is a 371 of PCT/US2010/001862 06/29/2010
which claims benefit of 61/222,177 07/01/2009

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

06/30/2021

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/Tyler W Sullivan/ Examiner's Signature	CA	9	18 ~~20~~	4

☐ Met after Allowance
TWS Initials

ADDRESS

Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103
UNITED STATES

TITLE

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

FILING FEE RECEIVED	FEES: Authority has been given in Paper
0.00	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:

☐ All Fees
☐ 1.16 Fees (Filing)
☐ 1.17 Fees (Processing Ext. of time)
☐ 1.18 Fees (Issue)
☐ Other _____

BIB (Rev. 05/07).

		❑ Credit

BIB (Rev. 05/07).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2654

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
17/351,649	06/18/2021 RULE	375	2487	ICVW_2009P00073 US02

APPLICANTS

InterDigital VC Holdings, Inc., Wilmington, DE;

INVENTORS

Yunfei Zheng, San Jose, CA;
Qian Xu, Folsom, CA;
Xiaoan Lu, Yardley, PA;
Peng Yin, Ithaca, NY;
Joel Sole Rojals, San Diego, CA;
Adeel Abbas, Passaic, NJ;

**** CONTINUING DATA ***************************

This application is a CON of 15/806,893 11/08/2017 PAT 11082697
which is a CON of 13/381,538 12/29/2011 ABN
which is a 371 of PCT/US2010/001862 06/29/2010
which claims benefit of 61/222,177 07/01/2009

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

06/30/2021

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/Tyler W Sullivan/ Examiner's Signature	CA	9	18 ~~20~~	4

☐ Met after Allowance — TWS Initials

ADDRESS

Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103
UNITED STATES

TITLE

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
0.00	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____

BIB (Rev. 05/07).

		❑ Credit

BIB (Rev. 05/07).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2654

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
17/351,649	06/18/2021 RULE	375	2487	ICVW_2009P00073 US02

APPLICANTS

InterDigital VC Holdings, Inc., Wilmington, DE;

INVENTORS

Yunfei Zheng, San Jose, CA;
Qian Xu, Folsom, CA;
Xiaoan Lu, Yardley, PA;
Peng Yin, Ithaca, NY;
Joel Sole Rojals, San Diego, CA;
Adeel Abbas, Passaic, NJ;

**** CONTINUING DATA ***************************

This application is a CON of 15/806,893 11/08/2017 PAT 11082697
which is a CON of 13/381,538 12/29/2011 ABN
which is a 371 of PCT/US2010/001862 06/29/2010
which claims benefit of 61/222,177 07/01/2009

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

06/30/2021

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/Tyler W Sullivan/ Examiner's Signature	CA	9	18 ~~20~~	4

☐ Met after Allowance — TWS Initials

ADDRESS

Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103
UNITED STATES

TITLE

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

FILING FEE RECEIVED	FEES: Authority has been given in Paper
0.00	No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:

☐ All Fees
☐ 1.16 Fees (Filing)
☐ 1.17 Fees (Processing Ext. of time)
☐ 1.18 Fees (Issue)
☐ Other _____

BIB (Rev. 05/07).

		❑ Credit

Electronic Acknowledgement Receipt	
EFS ID:	43028756
Application Number:	17351649
International Application Number:	
Confirmation Number:	2654
Title of Invention:	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS
First Named Inventor/Applicant Name:	Yunfei Zheng
Customer Number:	15800
Filer:	Brian Hubbard/Cassandra Katz
Filer Authorized By:	Brian Hubbard
Attorney Docket Number:	ICVW_2009P00073 US02
Receipt Date:	18-JUN-2021
Filing Date:	
Time Stamp:	14:19:34
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$2300
RAM confirmation Number	E20216HE20020130
Deposit Account	505519
Authorized User	Cassandra Katz
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

37 CFR 1.19 (Document supply fees)

37 CFR 1.20 (Post Issuance fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	ICVW_2009P00073US02_ADS.pdf	1256775 0e8c88b5910b1b65046b42fb1ee261b2611cc7d8	no	10
Warnings:					
Information:					
2		ICVW_2009P00073US02_Application.pdf	8622926 13411187c232a2a992ca205b9c654d7514803be1	yes	31
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Specification		1	24	
	Claims		25	30	
	Abstract		31	31	
Warnings:					
Information:					
3	Drawings-only black and white line drawings	ICVW_2009P00073US02_Drawings.pdf	676417 6f53fd2c7a82decde3cbb0e916ea6de0c3d816c2	no	9
Warnings:					
Information:					
4	Power of Attorney	ICVW_2009P00073US02_POA.pdf	1038409 d74c192bf71922856ff9eb09de5180bddac21b94	no	2
Warnings:					
Information:					
5	Oath or Declaration filed	ICVW_2009P00073US02_Declaration.pdf	536168 b0a0e0bc7d41f144b55c2b1539d5b9a4ed408a80	no	6

Warnings:					
Information:					
6	Authorization for Extension of Time all replies	ICVW_2009P00073US02_EOTAuthorization.pdf	226707 e1b45cf74063b1a8d78b219a43c9d8afc6a4694c	no	1
Warnings:					
Information:					
7	Miscellaneous Incoming Letter	ICVW_2009P00073_US02_IDS_Trans.pdf	1023756 99c64501ca708495c01bb7a184546dc236f72ebe	no	2
Warnings:					
Information:					
8	Information Disclosure Statement (IDS) Form (SB08)	ICVW_2009P00073_US02_IDS_1449.pdf	1538163 bf6d8ba69fc45ef74ee5ef9de5f63a1b0876124c	no	8
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
9		ICVW_2009P00073US02_PreliminaryAmendment.pdf	4734006 62f5d3ff4721dda17efad519529351775a480e16	yes	11

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Preliminary Amendment	1	1
	Specification	2	5
	Claims	6	9
	Applicant Arguments/Remarks Made in an Amendment	10	10
	Abstract	11	11

Warnings:					
Information:					
10	Fee Worksheet (SB06)	fee-info.pdf	36816 55273815a11d22ae621fc1ef17bae515de0a8da4	no	2

Warnings:	
Information:	
Total Files Size (in bytes):	19690143

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Yunfei		Zheng	

Residence Information (Select One) ● US Residency Non US Residency Active US Military Service

City	San Jose	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2056 Sheffield Dr.		
Address 2			
City	San Jose	State/Province	CA
Postal Code	95131	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Qian		Xu	

Residence Information (Select One) ◉ US Residency Non US Residency Active US Military Service

City	Folsom	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2056 Sheffield Dr.		
Address 2			
City	Folsom	State/Province	CA
Postal Code	95630	Country	US

Inventor 3 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Xiaoan		Lu	

Residence Information (Select One) ◉ US Residency Non US Residency Active US Military Service

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

City	Yardley	State/Province	PA	Country of Residence	US

Mailing Address of Inventor:

Address 1	1023 Victory Drive
Address 2	
City	Yardley
State/Province	PA
Postal Code	19067
Country	US

Inventor 4 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Peng		Yin	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Ithaca	State/Province	NY	Country of Residence	US

Mailing Address of Inventor:

Address 1	6 John Street
Address 2	
City	Ithaca
State/Province	NY
Postal Code	14850
Country	US

Inventor 5 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Joel		Sole Rojals	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	San Diego	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	1011 F Street
Address 2	
City	San Diego
State/Province	CA
Postal Code	92101
Country	US

Inventor 6 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Adeel		Abbas	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Passaic	State/Province	NJ	Country of Residence	US

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

Mailing Address of Inventor:

Address 1	285 Aycrigg Ave		
Address 2	Apt 15G		
City	Passaic	**State/Province**	NJ
Postal Code	07055	**Country** i	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	15800		
Email Address		Add Email	Remove Email

Application Information:

Title of the Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS		
Attorney Docket Number	ICVW_2009P00073 US02	**Small Entity Status Claimed**	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	9	**Suggested Figure for Publication (if any)**	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	● Customer Number	US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	15800		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation of	15806893	2017-11-08
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15806893	Continuation of	13381538	2011-12-29
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
13381538	a 371 of international	PCT/US2010/001862	2010-06-29
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
PCT/US2010/001862	Claims benefit of provisional	61222177	2009-07-01

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	ICVW_2009P00073 US02
		Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS		

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 — Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

● Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.	Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	InterDigital VC Holdings, Inc.		
Mailing Address Information For Applicant:			
Address 1	200 Bellevue Parkway		
Address 2	Suite 300		
City	Wilmington	**State/Province**	DE
Country	US	Postal Code	19809
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Brian J. Hubbard/			Date (YYYY-MM-DD)	2021-06-18
First Name	Brian J.	Last Name	Hubbard	Registration Number	45873

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	ICVW_2009P00073 US02
	Application Number	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ICVW_2009P00073 US02

METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of co-pending U.S. Application Serial No. 13/381,538, filed December 29, 2011, which is a National Phase entry under 35 U.S.C. §371 of International Application No. PCT/US2010/001862, filed June 29, 2010, which claims the benefit of U.S. Provisional Application Serial No. 61/222,177, filed July 1, 2009, which is incorporated by reference herein in its entirety.

TECHNICAL FIELD

The present principles relate generally to video encoding and decoding and, more particularly, to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

BACKGROUND

Most modern video coding standards employ various coding modes to efficiently reduce the correlations in the spatial and temporal domains. For example, in the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Moving Picture Experts Group-4 (MPEG-4) Part 10 Advanced Video Coding (AVC) standard/International Telecommunication Union, Telecommunication Sector (ITU-T) H.264 Recommendation (hereinafter the "MPEG-4 AVC Standard"), a picture can be intra or inter coded. In intra pictures, all macroblocks are coded in intra modes, thus exploiting spatial correlations within the picture. Intra modes can be classified into the following three types: INTRA4x4; INTRA8x8; and INTRA16x16. INTRA4x4 and INTRA8x8 support 9 intra prediction modes and INTRA16x16 supports 4 intra prediction modes.

INTRA4x4 and INTRA8x8 support the following 9 intra prediction modes: vertical; horizontal; DC; diagonal-down/left; diagonal-down/right; vertical-left; horizontal-down; vertical-right; and horizontal-up prediction. INTRA16x16 supports the following 4 intra prediction modes: vertical; horizontal; DC; and plane prediction.

Turning to FIG. 1, INTRA4x4 and INTRA8x8 prediction modes are indicated generally by the reference numeral 100. In FIG. 1, the reference numeral 0 indicates a vertical prediction mode, the reference numeral 1 indicates a horizontal prediction mode, the reference numeral 3 indicates a diagonal-down/left prediction mode, the reference numeral 4 indicates a diagonal-down/right prediction mode, the reference numeral 5 indicates a vertical-right prediction mode, the reference numeral 6 indicates a horizontal-down prediction mode, the reference numeral 7 indicates a vertical-left prediction mode, and the reference numeral 8 indicates a horizontal-up prediction mode. DC mode, which is part of the INTRA4x4 and INTRA8x8 prediction modes, is not shown. Turning to FIG. 2, INTRA16x16 prediction modes are indicated generally by the reference numeral 200. In FIG. 2, the reference numeral 0 indicates a vertical prediction mode, the reference numeral 1 indicates a horizontal prediction mode, and the reference numeral 3 indicates a plane prediction mode. DC mode, which is part of the INTRA16x16 prediction modes, is not shown.

INTRA4x4 uses a 4x4 discrete cosine transform (DCT). INTRA8x8 uses 8x8 transforms. INTRA16x16 uses cascaded 4x4 transforms. For the signaling, INTRA4x4 and INTRA8x8 share the same macroblock type (mb_type) 0 and are differentiated by a transform size flag (transform_8x8_size_flag). Then the choice of intra prediction mode in INTRA4x4 or INTRA8x8 is signaled by the most probable mode possibly with a remaining mode if necessary. For INTRA16x16, all the intra prediction modes along with a coded block pattern (cbp) type are signaled in mb_type, which uses an mb_type value from 1 to 24. TABLE 1 shows the detailed signaling for macroblock types for Intra coded slices (I slices). If a larger block size than 16x16 is used for intra prediction, then several possible issues are faced, as follows.

(1) If INTRA32x32 or INTRA64x64 prediction are added, by simply extending the mb_type in the MPEG-4 AVC Standard, it will cause too much overhead for those two new modes and, in addition, will not allow a hierarchical type of intra prediction. An example of a hierarchical type of intra prediction is explained as follows. If a 32x32 block is used as a large block and we allow sub-partitions to be 16x16 then, for each 16x16 sub-partition, INTRA4x4, INTRA8x8, INTRA16x16 should be allowed.

(2) If a larger transform (such as a 16x16 transform) instead of a cascaded transform is used for INTRA16x16, then the current signaling cannot be applied.

(3) We should give different priority for intra prediction modes inside one intra partition type.

TABLE 1

mb_type	Name of mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16-PredMode	CodedBlock-PatternChroma	CodedBlock-PatternLuma
0	I_NxN	0	Intra_4x4	na	Equation 7-33	Equation 7-33
0	I_NxN	1	Intra_8x8	na	Equation 7-33	Equation 7-33
1	I_16x16_0_0_0	na	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15

25	I_PCM	na	na	na	na	na

Some prior art approaches exist relating to signaling large motion (inter) partitions in extensions of the MPEG-4 AVC Standard. One example of how large motion (Inter) partitions are signaled in extensions of the MPEG-4 AVC Standard is described with respect to a first prior art approach. The first prior art approach describes how signaling is done for 32x32 blocks or 64x64 blocks using a hierarchical coding structure.

Moreover, in addition to the existing motion partition sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4) in the MPEG-4 AVC Standard, inter coding has also been proposed for an extension of the MPEG-4 AVC Standard using 32x32, 32x16, and 16x32 partitions. Turning to FIG. 3, motion partitions for use in 32x32 blocks are indicated generally by the reference numeral 300. The partitions include 32x32, 32x16, 16x32, and 16x16. 16x16 partitions can be further partitioned into partitions of sizes 16x16, 16x8, 8x16, and 8x8. Moreover, 8x8 partitions can be further partitioned into partitions of sizes 8x8, 8x4, 4x8, and 4x4.

For each 32x32 block, SKIP mode or DIRECT mode is signaled using an mb32_skip_flag in a way similar to that performed for other modes of the MPEG-4 AVC Standard. In addition, the original mb_type for an MxN (M=8, or 16 and N =8, or 16) partition in the MPEG-4 AVC Standard is also used to signal a 2Mx2N partition in a 32x32 block. If the mb32_type of a 32x32 indicates that a 16x16 partition is used, then the four 16x16 blocks are signaled in raster scan order by using the same syntax elements as macroblock_layer() in the MPEG-4 AVC Standard. Each 16x16 block may be partitioned further in the quadtree manner, from size 16x16 down to size 4x4.

For macroblock size 64x64, the following partitions are added above the partitions used in 32x32 blocks: 64x64; 64x32; and 32x64. Thus, one more hierarchical layer is added in the macroblock partition above block size 32x32. The original mb_type for an MxN (M =8, or 16 and N =8, or 16) macroblock partition in the MPEG-4 AVC Standard is used to signal a 4Mx4N macroblock partition in 64x64 macroblocks. If a 32x32 macroblock partition is used for a 64x64 block, then each 32x32 block will be handled in the same manner as described above.

However, the existing literature does not address how large intra mode is signaled, where large intra mode is defined to mean intra prediction involving partition blocks having a size equal to or larger than 32x32.

SUMMARY

These and other drawbacks and disadvantages of the prior art are addressed by the present principles, which are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

According to an aspect of the present principles, an apparatus is provided. The apparatus includes a video encoder for encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to another aspect of the present principles, a method in a video encoder is provided. The method includes encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to yet another aspect of the present principles, an apparatus is provided. The apparatus includes a video decoder for decoding picture data for at least one large block in a picture by determining that intra prediction is to be

performed for the at least one large block. The intra prediction is determined by determining a basic coding unit size and determining a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is determinable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

According to still another aspect of the present principles, a method is provided in a video decoder. The method includes decoding picture data for at least one large block in a picture by determining that intra prediction is to be performed for the at least one large block. The intra prediction is determined by determining a basic coding unit size and determining a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is determinable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

These and other aspects, features and advantages of the present principles will become apparent from the following detailed description of exemplary embodiments, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present principles may be better understood in accordance with the following exemplary figures, in which:

FIG. 1 is a diagram showing INTRA4x4 and INTRA8x8 prediction modes to which the present principles may be applied;

FIG. 2 is a diagram showing INTRA16x16 prediction modes to which the present principles may be applied;

FIG. 3 is a diagram showing motion partitions for use in 32x32 blocks to which the present principles may be applied;

FIG. 4 is a block diagram for an exemplary video encoder to which the present principles may be applied in accordance with an embodiment of the present principles;

FIG. 5 is a block diagram for an exemplary video decoder to which the present principles may be applied in accordance with an embodiment of the present principles;

FIG. 6 is a block diagram for exemplary hierarchical partitions to which the present principles may be applied in accordance with an embodiment of the present principles;

FIGS. 7A and 7B represent a flow diagram for an exemplary method for encoding picture data for large blocks by signaling intra prediction for the large blocks in accordance with an embodiment of the present principles; and

FIGS. 8A and 8B represent a flow diagram showing an exemplary method for decoding picture data for large blocks by determining that intra prediction is to be applied to the large blocks in accordance with an embodiment of the present principles.

DETAILED DESCRIPTION

The present principles are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders.

The present description illustrates the present principles. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the present principles and are included within its spirit and scope.

All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the present principles and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions.

Moreover, all statements herein reciting principles, aspects, and embodiments of the present principles, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known

equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

Thus, for example, it will be appreciated by those skilled in the art that the block diagrams presented herein represent conceptual views of illustrative circuitry embodying the present principles. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudocode, and the like represent various processes which may be substantially represented in computer readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

The functions of the various elements shown in the figures may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" or "controller" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor ("DSP") hardware, read-only memory ("ROM") for storing software, random access memory ("RAM"), and non-volatile storage.

Other hardware, conventional and/or custom, may also be included. Similarly, any switches shown in the figures are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementer as more specifically understood from the context.

In the claims hereof, any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a) a combination of circuit elements that performs that function or b) software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The present principles as defined by such claims reside in the fact that the functionalities provided by the various recited means are combined and brought

together in the manner which the claims call for. It is thus regarded that any means that can provide those functionalities are equivalent to those shown herein.

Reference in the specification to "one embodiment" or "an embodiment" of the present principles, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment of the present principles. Thus, the appearances of the phrase "in one embodiment" or "in an embodiment", as well any other variations, appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of "A, B, and/or C" and "at least one of A, B, and C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as readily apparent by one of ordinary skill in this and related arts, for as many items listed.

Moreover, it is to be appreciated that while one or more embodiments of the present principles are described herein with respect to an extension of the MPEG-4 AVC standard, the present principles are not limited to solely this extension and/or this standard and, thus, may be utilized with respect to other video coding standards, recommendations, and extensions thereof, while maintaining the spirit of the present principles.

As used herein, "high level syntax" refers to syntax present in the bitstream that resides hierarchically above the macroblock layer. For example, high level syntax, as used herein, may refer to, but is not limited to, syntax at the slice header level, Supplemental Enhancement Information (SEI) level, Picture Parameter Set (PPS) level, Sequence Parameter Set (SPS) level and Network Abstraction Layer (NAL) unit header level.

Also, as used herein, the words "picture" and "image" are used interchangeably and refer to a still image or a picture from a video sequence. As is known, a picture may be a frame or a field.

Additionally, as used herein, the word "signal" refers to indicating something to a corresponding decoder. For example, the encoder may signal that intra prediction is designated to be used for a particular large block (as defined herein) in order to make the decoder aware of which particular prediction type (e.g., intra or inter) was used on the encoder side. In this way, the same prediction type may be used at both the encoder side and the decoder side. Thus, for example, an encoder may transmit an indication (i.e., signal) for a particular large block that intra prediction is to be performed on that large block to simply allow the decoder to know and select the same prediction type for that large block. It is to be appreciated that signaling may be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth may be used to signal information to a corresponding decoder.

Turning to FIG. 4, an exemplary video encoder to which the present principles may be applied in accordance with an embodiment of the present principles is indicated generally by the reference numeral 400.

The video encoder 400 includes a frame ordering buffer 410 having an output in signal communication with a non-inverting input of a combiner 485. An output of the combiner 485 is connected in signal communication with a first input of a transformer and quantizer 425. An output of the transformer and quantizer 425 is connected in signal communication with a first input of an entropy coder 445 and a first input of an inverse transformer and inverse quantizer 450. An output of the entropy coder 445 is connected in signal communication with a first non-inverting input of a combiner 490. An output of the combiner 490 is connected in signal communication with a first input of an output buffer 435.

A first output of an encoder controller 405 is connected in signal communication with a second input of the frame ordering buffer 410, a second input of the inverse transformer and inverse quantizer 450, an input of a picture-type decision module 415, an input of a macroblock-type (MB-type) decision module 420, a second input of a super intra prediction module 460, a second input of a

deblocking filter 465, a first input of a motion compensator 470, a first input of a motion estimator 475, and a second input of a reference picture buffer 480.

A second output of the encoder controller 405 is connected in signal communication with a first input of a Supplemental Enhancement Information (SEI) inserter 430, a second input of the transformer and quantizer 425, a second input of the entropy coder 445, a second input of the output buffer 435, and an input of the Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) inserter 440.

A first output of the picture-type decision module 415 is connected in signal communication with a third input of a frame ordering buffer 410. A second output of the picture-type decision module 415 is connected in signal communication with a second input of a macroblock-type decision module 420.

An output of the Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) inserter 440 is connected in signal communication with a third non-inverting input of the combiner 490.

An output of the inverse quantizer and inverse transformer 450 is connected in signal communication with a first non-inverting input of a combiner 419. An output of the combiner 419 is connected in signal communication with a first input of the super intra prediction module 460 and a first input of the deblocking filter 465. An output of the deblocking filter 465 is connected in signal communication with a first input of a reference picture buffer 480. An output of the reference picture buffer 480 is connected in signal communication with a second input of the motion estimator 475. A first output of the motion estimator 475 is connected in signal communication with a second input of the motion compensator 470. A second output of the motion estimator 475 is connected in signal communication with a third input of the entropy coder 445.

An output of the motion compensator 470 is connected in signal communication with a first input of a switch 497. An output of the super intra prediction module 460 is connected in signal communication with a second input of the switch 497. An output of the macroblock-type decision module 420 is connected in signal communication with a third input of the switch 497. The third input of the switch 497 determines whether or not the "data" input of the switch (as compared to the control input, i.e., the third input) is to be provided by the motion compensator 470 or the super intra prediction module 460. The output of the switch 497 is

connected in signal communication with a second non-inverting input of the combiner 419 and with an inverting input of the combiner 485.

Inputs of the frame ordering buffer 410 and the encoder controller 405 are available as input of the encoder 400, for receiving an input picture 401. Moreover, an input of the Supplemental Enhancement Information (SEI) inserter 430 is available as an input of the encoder 400, for receiving metadata. An output of the output buffer 435 is available as an output of the encoder 400, for outputting a bitstream.

Turning to FIG. 5, an exemplary video decoder to which the present principles may be applied in accordance with an embodiment of the present principles is indicated generally by the reference numeral 500.

The video decoder 500 includes an input buffer 510 having an output connected in signal communication with a first input of the entropy decoder 545. A first output of the entropy decoder 545 is connected in signal communication with a first input of an inverse transformer and inverse quantizer 550. An output of the inverse transformer and inverse quantizer 550 is connected in signal communication with a second non-inverting input of a combiner 525. An output of the combiner 525 is connected in signal communication with a second input of a deblocking filter 565 and a first input of a super intra prediction module 560. A second output of the deblocking filter 565 is connected in signal communication with a first input of a reference picture buffer 580. An output of the reference picture buffer 580 is connected in signal communication with a second input of a motion compensator 570.

A second output of the entropy decoder 545 is connected in signal communication with a third input of the motion compensator 570 and a first input of the deblocking filter 565. A third output of the entropy decoder 545 is connected in signal communication with an input of a decoder controller 505. A first output of the decoder controller 505 is connected in signal communication with a second input of the entropy decoder 545. A second output of the decoder controller 505 is connected in signal communication with a second input of the inverse transformer and inverse quantizer 550. A third output of the decoder controller 505 is connected in signal communication with a third input of the deblocking filter 565. A fourth output of the decoder controller 505 is connected in signal communication with a

second input of the super intra prediction module 560, with a first input of the motion compensator 570, and with a second input of the reference picture buffer 580.

An output of the motion compensator 570 is connected in signal communication with a first input of a switch 597. An output of the super intra prediction module 560 is connected in signal communication with a second input of the switch 597. An output of the switch 597 is connected in signal communication with a first non-inverting input of the combiner 525.

An input of the input buffer 510 is available as an input of the decoder 500, for receiving an input bitstream. A first output of the deblocking filter 565 is available as an output of the decoder 500, for outputting an output picture.

As noted above, the present principles are directed to methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders. Moreover, as noted above, large blocks, to which the present principles may be applied, are defined to means blocks having a size equal to or larger than 32x32.

In an embodiment, for ease of notation, we split the signaling of intra prediction into the following two parts: sip_type (spatial intra partition type, which can be INTRA4x4, INTRA8x8, INTRA16x16, and so forth); and intra_pred_mode (such as, e.g., the 9 Intra Prediction Modes within INTRA4x4 and INTRA8x8) within each sip_type. In further detail with respect to a particular embodiment, we propose the following three rules for the present principles: (1) select a basic coding unit; (2) allow hierarchical layer intra prediction by either splitting from a largest intra prediction type or merging from the basic coding unit; and (3) for each sip_type, assign a higher priority to the intra_pred_mode which is most frequently used. With respect to Rule (1), we allow several sip_types for the basic coding unit.

An Embodiment

In an embodiment, we set the basic coding unit to be 16x16. In this coding unit, we allow sip_type to be INTRA4x4, INTRA8x8 and INTRA16x16. We also allow hierarchical layer intra prediction as shown in FIG. 6.

Turning to FIG. 6, exemplary hierarchical partitions to which the present principles may be applied are indicated generally by the reference numeral 600. In this embodiment, if the largest block size is set to 64x64, then we use "split signaling" to allow hierarchical layer intra prediction. That is, in an embodiment, we

add intra64_flag. If intra64_flag is equal to 1, then INTRA64x64 is used. Otherwise, if intra64_flag is equal to 0, then we split a 64x64 block 611 into four 32x32 blocks 621. For each of the 32x32 blocks 621, we add intra32_flag. If intra32_flag is equal to 1, then INTRA32x32 is used. Otherwise, if intra32_flag is equal to 0, then all sip types allowed in the 16x16 basic coding unit are allowed here (i.e., for the 32x32 blocks 621) as well. For intra_pred_mode in INTRA16x16, we have DC mode and the directional modes, the latter allowing different types of directional predictions by sending the mode information. Thus, a 32x32 intra prediction block 621 may be further split into 4 16x16 intra prediction blocks 631. One or more of the 4 16x16 intra prediction blocks 631 may be further split to DC mode (not shown), a 16x16 mode 641, an 8x8 mode 651, and a 4x4 mode 661. In this embodiment, we presume having the following four 16x16 intra prediction modes: DC; horizontal (HOR); vertical (VER); and multi-directional (Multi-DIR). The intra_pred_mode is signaled by considering the priority of each mode. In INTRA16x16, since the DC mode is used more often than other modes, we add INTRA16x16_DC in the sip_type table before INTRA16x16. Then we remove the most_probable_mode indication in intra_pred_mode for INTRA16x16. Instead, the other 3 modes (16x16, 8x8, and 4x4) are absolutely indicated.

Syntax

We illustrate an example of syntax for this embodiment in TABLE 2 and TABLE 3. In particular, TABLE 2 shows an exemplary specification of sip types for a 16x16 coding unit in accordance with an embodiment of the present principles, and TABLE 3 shows exemplary INTRA16x16 prediction modes in accordance with an embodiment of the present principles. For INTRA32x32/INTRA64x64, the same modes as INTRA16x16 are used. For signaling, we replace the most_probable_mode indication with intra32_DC_flag and intra64_DC flag since DC is mostly used. Then we absolutely code other intra_pred_mode.

intra_pred_mode signaling for INTRA4x4 and INTRA8x8 may be performed exactly the same as in the MPEG-4 AVC Standard, so we will not list these modes in any of the tables.

TABLE 2

Sip_type	Index	Binarization bits
SIP8x8	0	0
SIP16x16DC	1	10
SIP16x16	2	110
SIP4x4	3	1110

TABLE 3

Intra Prediction Mode	Index	Binarization bits
VER	0	0
HOR	1	10
Multi-DIR	2	11

TABLE 4 shows exemplary macroblock layer syntax, in accordance with an embodiment of the present principles.

TABLE 4

macroblock_layer( ) {	C	Descriptor
...		
intra64_flag	2	u(1)
if (intra64_flag==1) {		
intra64_DC_flag	2	u(1)
if (intra64_DC_flag==0) {		
intra_pred_mode_64	2	ue(v)/se(v)
if (intra_pred_mode_64==Multi-DIR) {		
intra64_multidir_index	2	ue(v)/se(v)
}		
}		
}		
else {		
for (i32=0; i32<4; i32++) {		
intra32_flag[i32]	2	u(1)
if (intra32_flag[i32]==1) {		
intra32_DC_flag[i32]	2	u(1)
if (intra32_DC_flag[i32]==0) {		
intra_pred_mode_32[i32]	2	ue(v)/se(v)
if (intra_pred_mode_32[i32]==Multi-DIR) {		
intra32_multidir_index[i32]	2	ue(v)/se(v)
}		
}		
}		

else {		
for (i16=0; i16<4; i16++) {		
sip_type[i16]	2	ue(v)/se(v)
if (sip_type[i16]==SIP16x16) {		
intra_pred_mode_16[i16]	2	ue(v)/se(v)
if (intra_pred_mode_16[i16]==Multi-DIR) {		
intra16_multidir_index[i16]	2	ue(v)/se(v)
}		
}		
else if (sip_type[i16]!=SIP16x16_DC) {		
mb_intra_prediction_syntax(); /*this is same as H.264*/		
}		
}		
}		
...		
}		

The semantics of some of the syntax elements of TABLE 4 are as follows:

Intra64_flag equal to 1 specifies that INTRA64x64 is used. Intra64_flag equal to 0 specifies that a 64x64 large block is further split into 32x32 partitions.

Intra64_DC_ flag equal to 1 specifies that the intra_pred_mode is DC mode for INTRA64x64. Intra64_DC_flag equal to 0 specifies that the intra_pred_mode is not DC mode for INTRA64x64.

intra_pred_mode_64 specifies intra prediction mode (not including DC mode) for INTRA64x64.

intra64_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA64x64.

Intra32_flag [i] equal to 1 specifies that INTRA32x32 is used for the ith 32x32 large block. Intra32_flag [i] equal to 0 specifies that the ith 32x32 large block is further split into 16x16 partitions.

intra32_DC_flag [i] equal to 1 specifies that the intra_pred_mode is DC mode for INTRA32x32 for the ith 32x32 block. intra32_DC_flag[i] equal to 0 specifies that the intra_pred_mode is not DC mode for INTRA32x32 for the ith 32x32 block.

intra_pred_mode_32 [i] specifies the intra prediction mode (not including DC mode) for INTRA32x32 for the ith 32x32 large block.

intra32_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA32x32.

sip_type [i] specifies the spatial intra partition type for the basic block coding unit in the ith 16x16 block.

intra_pred_mode_16 [i] specifies the intra prediction mode (not including DC mode) for INTRA16x16 for the ith 16x16 block.

Intra16_multidir_index specifies the index of the angle for Multi_Dir mode in INTRA16x16 for the ith 16x16 block.

Another Embodiment

In another embodiment, we adaptively select the large block unit to be 32x32 or 64x64. The selection may be signaled using one or more high level syntax elements. In an embodiment, if 32x32 is selected, then we just remove all the syntax related to 64x64.

In another embodiment, hierarchical intra layer prediction can involve merging from basic coding unit. For example, if the largest block unit is 64x64 and the basic coding unit is 16x16, then we use one flag (is_all_16x16_coding) to indicate if all of the 16x16 blocks inside one 64x64 block are of the 16x16 coding type. If is_all_16x16_coding is equal to 1, then this indicates that 16x16 coding type is used and we stop signaling. Otherwise, we use one flag (is_all_32x32_coding) to indicate if all of the 32x32 blocks inside one 64x64 block are of the 32x32 coding type. If is_all_32x32_coding is equal to 1, then this indicates that all of the 32x32 blocks inside one 64x64 block are of the 32x32 coding type. Otherwise, if

is_all_32x32_coding and is_all_16x16_coding are equal to 0, then this indicates that INTRA64x64 is used.

In another embodiment, we introduce a SIP type for a block unit (large_sip_type) having a size not less than 16x16. The three types are referred to as follows: large_intra_16x16; large_intra_32x32; and large_intra_64x64. large_intra_16x16 means that all 16x16 blocks inside one large block are of the 16x16 coding type. large_intra_32x32 means that all 32x32 blocks inside one large block are of the 32x32 coding type. In an embodiment, large_intra_32x32 can be combined with the embodiment described above with intra32_flag to allow hierarchical intra prediction. large_intra_64x64 means that all 64x64 blocks inside one large block are coded as INTRA64x64.

In another embodiment, we can introduce several sip/mode tables. The tables can be pre-stored at both an encoder and a decoder, or the tables can be user specified and transmitted using one or more high level syntax elements. TABLE 5 shows exemplary macroblock layer syntax, in accordance with an embodiment of the present principles.

TABLE 5

macroblock_layer( ) {	C	Descriptor
...		
is_all_16x16_coding	2	u(1)
if (is_all_16x16_coding == 0) {		
is_all_32x32_coding	2	u(1)
if (is_all_32x32_coding == 0) {		
decode_with_64x64_coding_type()		
}		
else{		
decode_with_32x32_coding_type()		
}		
}		
else {		
decode_with_16x16_coding_type()		
}		
...		
}		

The semantics of the some of the syntax elements of TABLE 5 are as follows:

is_all_16x16_coding equal to 1 specifies that all 16x16 blocks inside a large block are coded by a 16x16 coding type. is_all_16x16_coding equal to 0 specifies that the large block is not coded by a 16x16 coding type.

is_all_32x32_coding equal to 1 specifies that all 32x32 blocks inside a large block are coded by a 32x32 coding type. is_all_32x32_coding equal to 0 specifies that the large block is not coded by a 32x32 coding type.

Turning to FIGS. 7A and 7B, which together represent an exemplary method for encoding picture data for large blocks by signaling intra prediction for the large blocks, indicated generally by the reference numeral 700. The method 700 includes a start block 705 that passes control to a function block 710. The function block 710 performs an initialization, and passes control to a loop limit block 715. The loop limit block 715 performs a loop (hereinafter also loop 1) over 64x64 blocks (i.e., blocks having a block size of 64x64), and passes control to a function block 785 and a loop limit block 720.

The function block 785 performs an Intra 64x64 mode decision, sets Intra64_DC flag based on RD64 (i.e., the rate distortion resulting from the Intra 64x64 mode decision), and passes control to a decision block 770.

The loop limit block 720 performs a loop (hereinafter also loop 2) over four 32x32 blocks (i.e., the four blocks having a block size of 32x32 and obtained from the current 64x64 block being processed by loop 1), and passes control to a function block 790 and a loop limit block 725.

The function block 790 performs an Intra 32x32 mode decision, sets Intra32_DC flag based on RD32 (i.e., the rate distortion resulting from the Intra 32x32 mode decision), and passes control to a decision block 750.

The loop limit block 725 performs a loop (hereinafter also loop 3) over four 16x16 blocks (i.e., the four blocks having a block size of 16x16 and obtained from the current 32x32 block being processed by loop 2), and passes control to a function block 730 and a function block 735.

The function block 730 evaluates Intra16x16_DC mode, and passes control to a function block 740. The function block 735 evaluates other 16x16 modes (i.e., other than Intra16x16_DC) and below (e.g., 8x8, 4x4, etc.), and passes control to the function block 740.

The function block 740 performs a 16x16 mode decision based on RD16 (i.e., the rate distortion resulting from the intra 16x16 mode decision), then accumulates the RD16 of each 16x16 block to obtain TotRD16 (which indicates the total rate distortion of the whole 32x32 block when coded by four 16x16 blocks), and passes control to a loop limit block 745. The loop limit block 745 ends the loop (i.e., loop 3) over the 16x16 blocks, and passes control to the decision block 750.

The decision block 750 determines whether or not RD32 < TotRD16 (i.e., whether the rate distortion cost for a current 32x32 block is less than the total rate distortion cost for the four 16x16 blocks obtained from the current 32x32 block). If so, then control is passed to a function block 755. Otherwise, control is passed to a function block 742.

The function block 755 sets Intra32_flag equal to one, and passes control to a function block 760. The function block 742 sets Intra32_flag equal to zero, and passes control to the function block 760.

The function block 760 sets an accumulation of the RD32 of each 32x32 block to TotRD32 to indicate the total rate distortion of the whole 64x64 block when coded by four 32x32 blocks, and passes control to a loop limit block 765. The loop limit block 765 ends the loop (i.e., loop 2) over the 32x32 blocks, and passes control to a decision block 770.

The decision block 770 determines whether or not RD64 < TotRD32 (i.e., whether the rate distortion cost for a current 64x64 block is less than the total rate distortion cost for the four 32x32 blocks obtained from the current 64x64 block). If so, then control is passed to a function block 775. Otherwise, control is passed to a function block 780.

The function block 775 sets Intra64_flag equal to one, and passes control to a loop limit block 795. The function block 780 sets Intra64_flag equal to zero, and passes control to the function block 795.

The function block 795 ends the loop (i.e., loop 1) over the 64x64 blocks, and passes control to a function block 797. The function block 797 entropy codes flags, intra_pred_mode, and a residue, and passes control to an end block 799.

Turning to FIGS. 8A and 8B, which together represent an exemplary method for decoding picture data for large blocks by determining that intra prediction is to be applied to the large blocks, indicated generally by the reference numeral 800. The method 800 includes a start block 805 that passes control to a function block 808. The function block 808 initializes the decoder and then passes the control to a function block 810. The function block 810 parses the bitstream, and passes control to a loop limit block 815. The loop limit block 815 performs a loop (hereinafter loop 1) over 64x64 blocks, and passes control to a decision block 820. The decision block 820 determines whether or not Intra64_flag is set equal to one. If so, then control is passed to a function block 885. Otherwise, control is passed to a loop limit block 825.

The function block 885 determines whether or not intra64_DC_flag is set equal to one. If so, then control is passed to a function block 887. Otherwise, control is passed to a function block 888. The function block 887 performs intra 64x64 DC prediction, and then passes control to a function block 890. The function block 888 performs intra 64x64 predictions other than intra 64x64 DC mode and then passes control to a function block 890. The function block 890 decodes a current 64x64 block, and passes control to a loop limit block 880. The loop limit block 880 ends the loop (i.e., loop 1) over the 64x64 blocks, and passes control to an end block 899.

The loop limit block 825 performs a loop (hereinafter loop 2) over four 32x32 blocks, and passes control to a decision block 830. The decision block 830 determines whether or not Intra32_flag is equal to one. If so, then control is passed to a function block 835. Otherwise, control is passed to a loop limit block 845.

The function block 835 determines whether or not Intra32_DC_flag is equal to one. If so, then control is passed to a function block 837. Otherwise, control is passed to a function block 838. The function block 837 performs an intra32x32 DC prediction, and passes control to a function block 840. The function block 838 performs an intra prediction other than intra 32x32 DC mode, and then passed the control to the function block 840. The function block 840 decodes a 32x32 block, and passes control to a loop limit block 875.

The loop limit block 875 ends the loop (i.e., loop 2) over the 32x32 blocks, and passes control to the loop limit block 880.

The loop limit block 845 performs a loop (hereinafter loop 3) over four 16x16 blocks, and passes control to a decision block 850. The decision block 850 determines whether or not sip_type = Intra16_DC. If so, then control is passed to a function block 855. Otherwise, control is passed to a function block 860.

The function block 855 performs Intra16x16_DC mode prediction, and passes control to a function block 865. The function block 860 performs mode prediction using other intra prediction modes (i.e., other than Intra16x16_DC mode), and passes control to the function block 865.

The function block 865 decodes a 16x16 block, and passes control to a loop limit block 870. The loop limit block 870 ends the loop (i.e., loop 3) over the 16x16 blocks, and passes control to the loop limit block 875.

A description will now be given of some of the many attendant advantages/features of the present invention, some of which have been mentioned above. For example, one advantage/feature is an apparatus having a video encoder for encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

Another advantage/feature is the apparatus having the video encoder as described above, wherein for each of the plurality of spatial intra partition types, a higher priority is assigned to a particular intra prediction mode that is most frequently used from among a plurality of available intra prediction modes.

Yet another advantage/feature is the apparatus having the video encoder as described above, wherein the large block size is adaptively selected.

Still another advantage/feature is the apparatus having the video encoder as described above, wherein the signaling is performed using one or more high level syntax elements.

Moreover, another advantage/feature is the apparatus having the video encoder as described above, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used by the video encoder to encode the at least one large block. The at least one of the spatial intra partition type table and the intra prediction mode table is arranged to be pre-stored and used by a corresponding video decoder to decode the at least one large block.

Further, another advantage/feature is the apparatus having the video encoder as described above, wherein at least one of a spatial intra partition type table and an intra prediction mode table, is used by the video encoder to encode the at least one large block, and is transmitted by the video encoder using one or more high level syntax elements.

These and other features and advantages of the present principles may be readily ascertained by one of ordinary skill in the pertinent art based on the teachings herein. It is to be understood that the teachings of the present principles may be implemented in various forms of hardware, software, firmware, special purpose processors, or combinations thereof.

Most preferably, the teachings of the present principles are implemented as a combination of hardware and software. Moreover, the software may be implemented as an application program tangibly embodied on a program storage unit. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units ("CPU"), a random access memory ("RAM"), and input/output ("I/O") interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU. In addition, various other peripheral units may be connected to the computer platform such as an additional data storage unit and a printing unit.

It is to be further understood that, because some of the constituent system components and methods depicted in the accompanying drawings are preferably implemented in software, the actual connections between the system components or the process function blocks may differ depending upon the manner in which the present principles are programmed. Given the teachings herein, one of ordinary skill in the pertinent art will be able to contemplate these and similar implementations or configurations of the present principles.

Although the illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present principles is not limited to those precise embodiments, and that various changes and modifications may be effected therein by one of ordinary skill in the pertinent art without departing from the scope or spirit of the present principles. All such changes and modifications are intended to be included within the scope of the present principles as set forth in the appended claims.

ICVW_2009P00073 US02

CLAIMS:

1. A decoding method comprising:

decoding picture data for at least a large block in a picture by determining that intra prediction is to be performed for the at least a large block,

wherein the at least a large block has a large block size greater than a basic coding unit size, the large block size being one of 32x32 and 64x64 and the basic coding unit size being 16x16, wherein the intra prediction is signaled for the at least a large block by:

- decoding a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;

- decoding an intra prediction mode for said large block in the case where said binary split signaling syntax element specifies the large block is not further split;

otherwise in the case where said binary split signaling syntax element specifies the large block is further split:

- decoding for each sub-block, in the case where said sub-block is 32x32, a binary split signaling syntax element specifying whether the 32x32 sub-block is further split into four equally basic coding unit sized blocks and decoding an intra prediction mode for said 32x32 sub-block in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and

- decoding for each sub-block, in the case where said sub-block is 16x16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

2. The method of claim 1, further comprising decoding at least one binary merge signaling syntax element specifying whether basic coding unit sized blocks are merged into a large sized block.

3. The method of claim 1, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used to decode the at least one large block.

4. The method of claim 1, wherein at least one of a spatial intra partition type table and an intra prediction mode table is received using one or more high level syntax elements and is used to decode the at least one large block.

5. An encoding method comprising:

encoding picture data for at least a large block in a picture by determining that intra prediction is to be performed for the at least a large block,

wherein the at least a large block has a large block size greater than a basic coding unit size, the large block size being one of 32x32 and 64x64 and the basic coding unit size being 16x16,

wherein the intra prediction is signaled for the at least a large block by:

- encoding a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;

- encoding an intra prediction mode for said large block in the case where said binary split signaling syntax element specifies the large block is not further split;

otherwise in the case where said binary split signaling syntax element specifies the large block is further split:

- encoding for each sub-block, in the case where said sub-block is 32x32, a binary split signaling syntax element specifying whether said 32x32 sub-block is further split into four equally basic coding unit sized blocks and encoding an intra prediction mode for said 32x32 sub-block in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and

- encoding for each sub-block, in the case where said sub-block is 16x16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

6. The method of claim 5, further comprising encoding at least one binary merge signaling syntax element specifying whether basic coding unit sized blocks are merged into a large sized block.

7. The method of claim 5, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used to encode the at least one large block.

8. The method of claim 5, wherein at least one of a spatial intra partition type table and an intra prediction mode table is encoded using one or more high level syntax elements and is used to encode the at least one large block.

9. A video decoder, comprising at least a memory and one or more processors, wherein the one or more processors are configured to:

decode picture data for at least a large block in a picture by determining that intra prediction is to be performed for the at least a large block,

wherein the at least a large block has a large block size greater than a basic coding unit size, the large block size being one of 32x32 and 64x64 and the basic coding unit size being 16x16,

wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:

- decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;

- decode an intra prediction mode for said large block in the case where said binary split signaling syntax element specifies the large block is not further split;

otherwise in the case where said binary split signaling syntax element specifies the large block is further split:

- decode for each sub-block, in the case where said sub-block is 32x32, a binary split signaling syntax element specifying whether said 32x32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32x32 sub-block in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and

- decode for each sub-block, in the case where said sub-block is 16x16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

10. The video decoder of claim 9, wherein the one or more processors are further configured to decode at least one binary merge signaling syntax element specifying whether basic coding unit sized blocks are merged into a large sized block.

11. The video decoder of claim 9, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used to decode the at least one large block.

12. The video decoder of claim 9, wherein at least one of a spatial intra partition type table and an intra prediction mode table is received using one or more high level syntax elements and is used to decode the at least one large block (810).

13. A video encoder, comprising at least a memory and one or more processors, wherein the one or more processors are configured to:

encode picture data for at least a large block in a picture by determining that intra prediction is to be performed for the at least a large block (820, 830),

wherein the at least a large block has a large block size greater than a basic coding unit size, the large block size being one of 32x32 and 64x64 and the basic coding unit size being 16x16,

wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:

- encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;

- encode an intra prediction mode for said large block in the case where said binary split signaling syntax element specifies the large block is not further split;

otherwise in the case where said binary split signaling syntax element specifies the large block is further split:

- encode for each sub-block, in the case where said sub-block is 32x32, a binary split signaling syntax element specifying whether said 32x32 sub-block is further split into four equally basic coding unit sized blocks and encode an intra

prediction mode for said 32x32 sub-block in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and

- encode for each sub-block, in the case where said sub-block is 16x16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

14. The video encoder of claim 13, wherein the one or more processors are further configured to encode at least one binary merge signaling syntax element specifying whether basic coding unit sized blocks are merged into a large sized block.

15. The video encoder of claim 13, wherein at least one of a spatial intra partition type table and an intra prediction mode table is pre-stored and used to encode the at least one large block.

16. The video encoder of claim 13, wherein at least one of a spatial intra partition type table and an intra prediction mode table is encoded using one or more high level syntax elements and is used to encode the at least one large block (810).

17. A non-transitory computer readable medium having a video signal structure comprising:

- coded picture data representative of at least a large block in a picture obtained by determining that intra prediction is to be performed for the at least a large block (820, 830), wherein the at least a large block has a large block size greater than a basic coding unit size, the large block size being one of 32x32 and 64x64 and the basic coding unit size being 16x16,

- a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;

- an intra prediction mode for said large block in the case where said binary split signaling syntax element specifies the large block is not further split;

otherwise in the case where said binary split signaling syntax element specifies the large block is further split:

- for each sub-block, in the case where said sub-block is 32x32, a binary split signaling syntax element specifying whether said 32x32 sub-block is further split into four equally basic coding unit sized blocks and an intra prediction mode for said 32x32 sub-block in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and

- for each sub-block, in the case where said sub-block is 16x16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types.

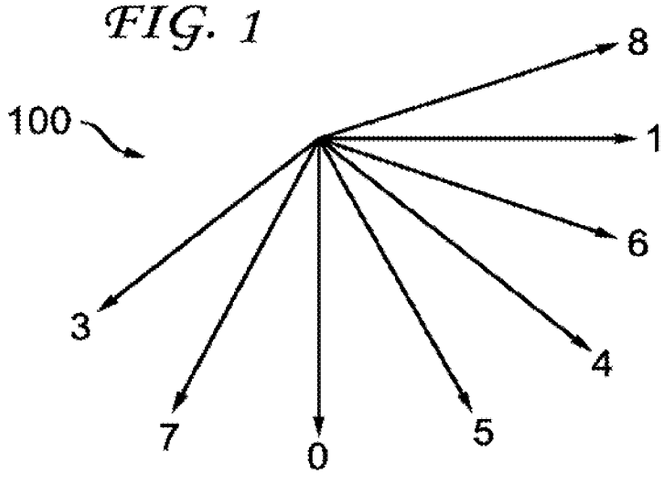
18. The medium of claim 17, wherein said video signal structure further comprises at least one binary merge signaling syntax element specifying whether basic coding unit sized blocks are merged into a large sized block.

19. The medium of claim 17, wherein said video signal structure further comprises data representative of at least one of a spatial intra partition type table and of an intra prediction mode table encoded in one or more high level syntax elements, said at least one of a spatial intra partition type table and an intra prediction mode table being used to encode said at least a large block.

ABSTRACT

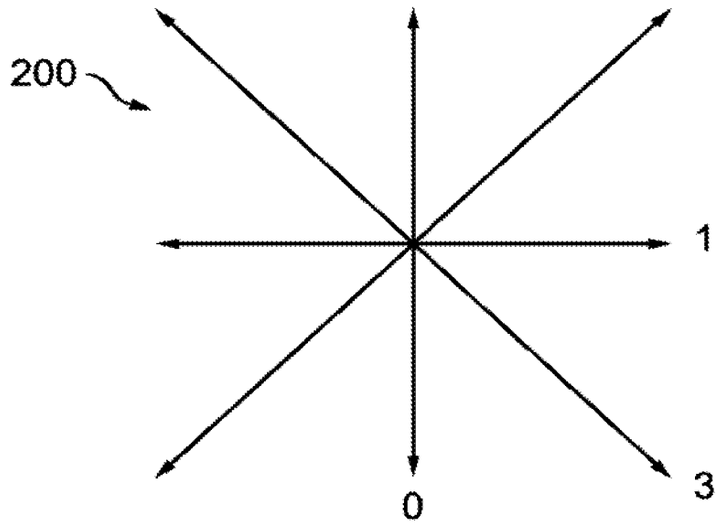
Methods and apparatus are provided for signaling intra prediction for large blocks for video encoders and decoders. An apparatus includes a video encoder (400) for encoding picture data for at least one large block in a picture by signaling intra prediction for the at least one large block. The intra prediction is signaled by selecting a basic coding unit size and assigning a single spatial intra partition type for the basic coding unit size. The single spatial intra partition type is selectable from among a plurality of spatial intra partition types. The at least one large block has a large block size greater than a block size of the basic coding unit. The intra prediction is hierarchical layer intra prediction and is performed for the at least one large block by at least one of splitting from the large block size to the basic coding unit size and merging from the basic coding unit size to the large block size.

FIG. 1



PRIOR ART

FIG. 2



PRIOR ART

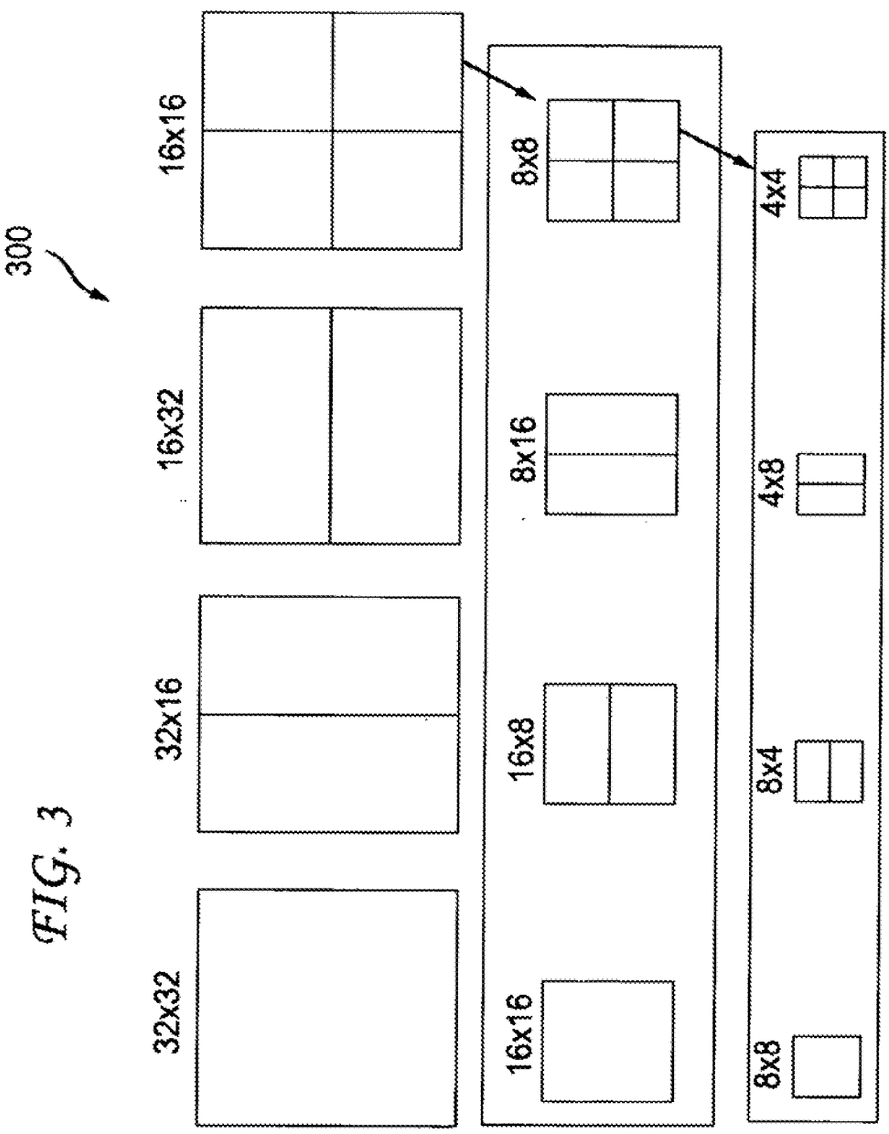

300
32x32
32x16
16x32
16x16
16x16
16x8
8x16
8x8
8x8
4x4
8x4
4x8
4x4

FIG. 3

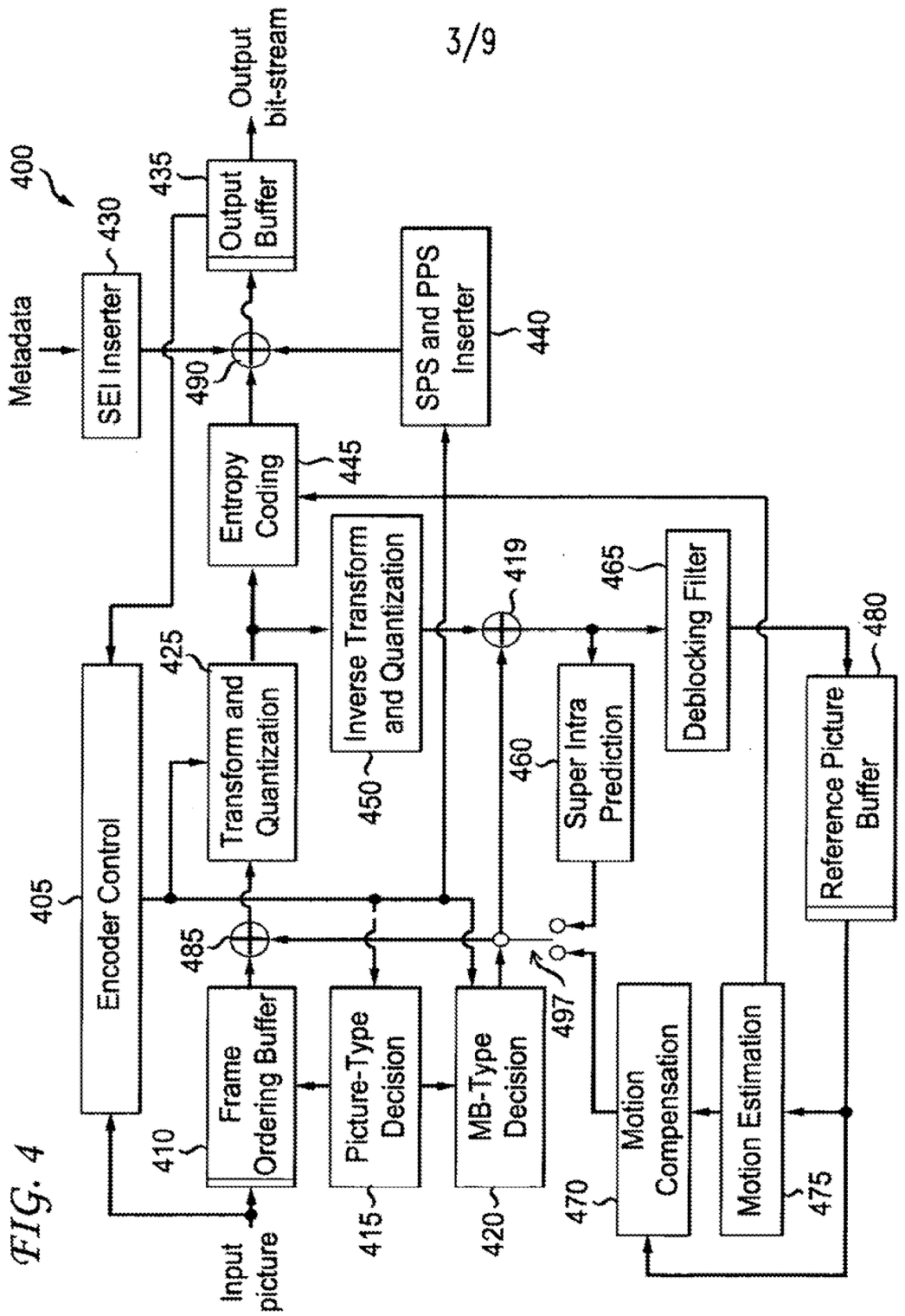

400
Metadata
SEI Inserter
430
435
Output Buffer
Output bit-stream
490
SPS and PPS Inserter
440
Entropy Coding
445
Inverse Transform and Quantization
450
419
465
Deblocking Filter
480
Reference Picture Buffer
425
Transform and Quantization
Super Intra Prediction
460
405
Encoder Control
485
497
Motion Compensation
470
Motion Estimation
475
Frame Ordering Buffer
410
Picture-Type Decision
415
MB-Type Decision
420
Input picture

FIG. 4

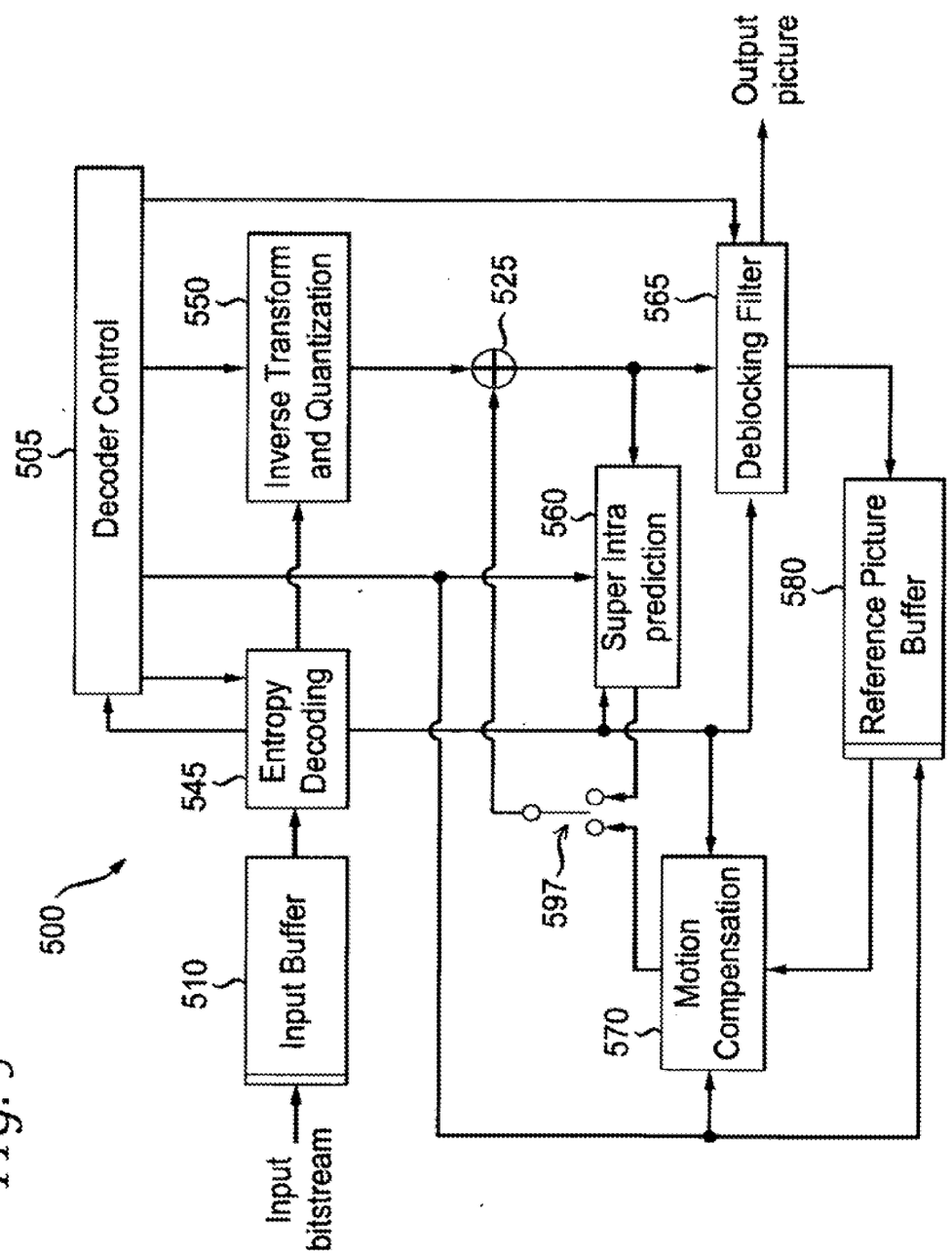

Output picture
Decoder Control
505
Inverse Transform and Quantization
550
525
565
Deblocking Filter
Super Intra prediction
560
580
Reference Picture Buffer
Entropy Decoding
545
597
Motion Compensation
570
Input Buffer
510
500
Input bitstream

FIG. 5

FIG. 6

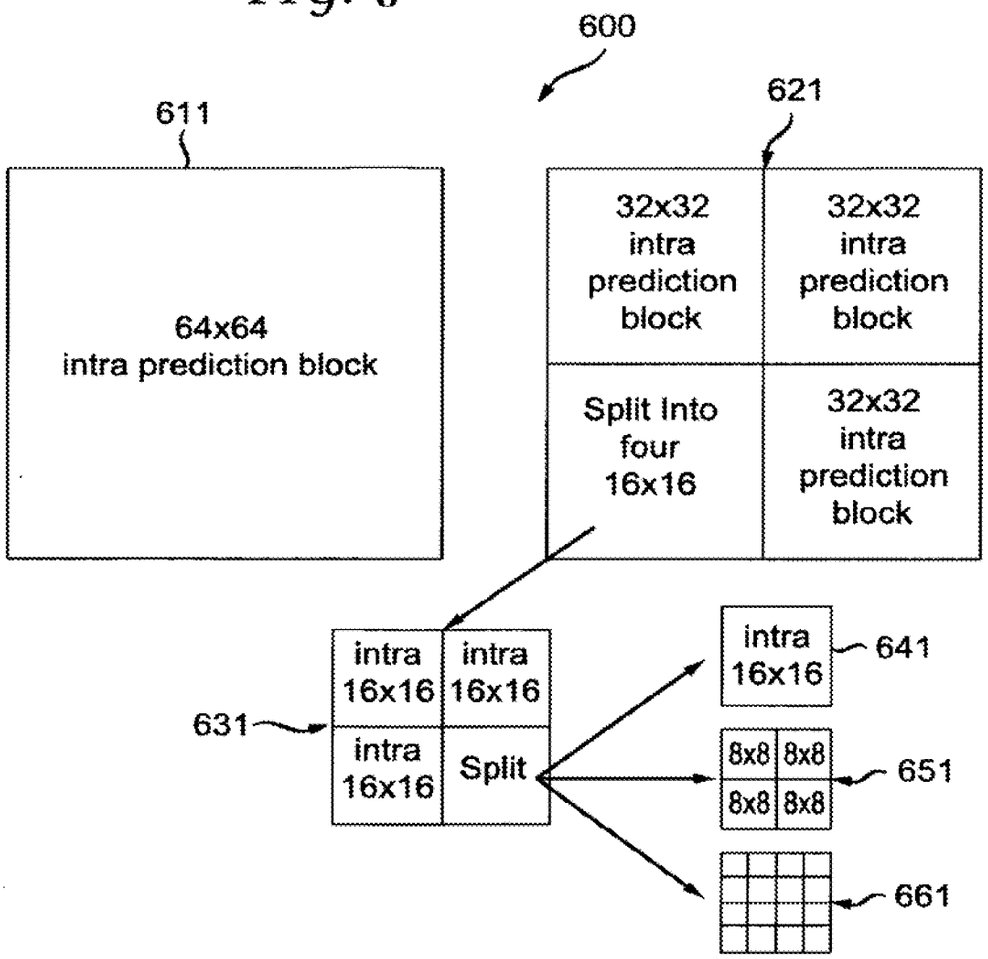

600
611
621
64x64
intra prediction block
32x32
intra
prediction
block
32x32
intra
prediction
block
Split Into
four
16x16
32x32
intra
prediction
block
631
intra
16x16
intra
16x16
intra
16x16
Split
intra
16x16
641
8x8
8x8
8x8
8x8
651
661

FIG. 7A

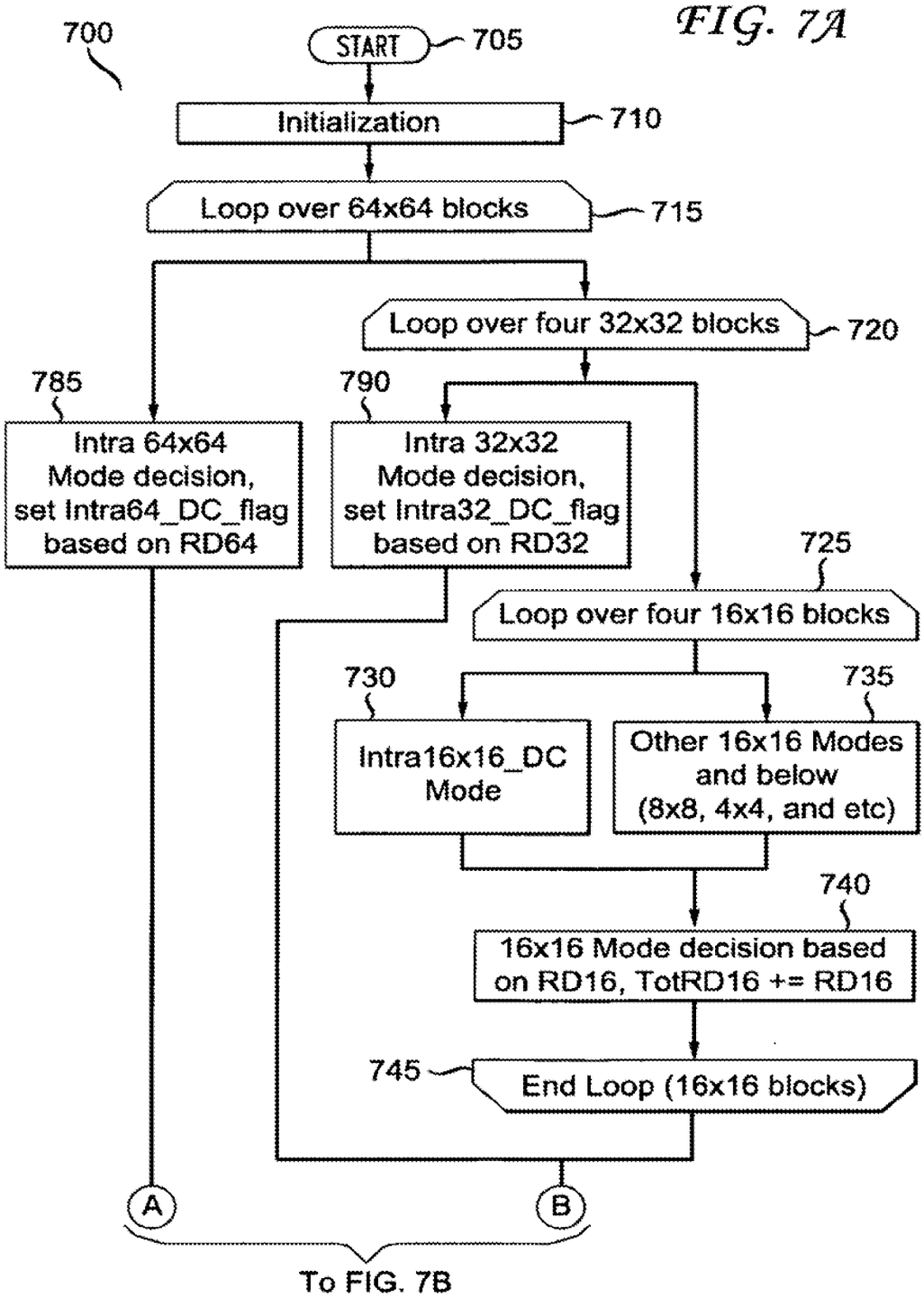

700
START
705
Initialization
710
Loop over 64x64 blocks
715
Loop over four 32x32 blocks
720
785
Intra 64x64
Mode decision,
set Intra64_DC_flag
based on RD64
790
Intra 32x32
Mode decision,
set Intra32_DC_flag
based on RD32
725
Loop over four 16x16 blocks
730
Intra16x16_DC
Mode
735
Other 16x16 Modes
and below
(8x8, 4x4, and etc)
740
16x16 Mode decision based
on RD16, TotRD16 += RD16
745
End Loop (16x16 blocks)
A
B
To FIG. 7B

FIG. 7B

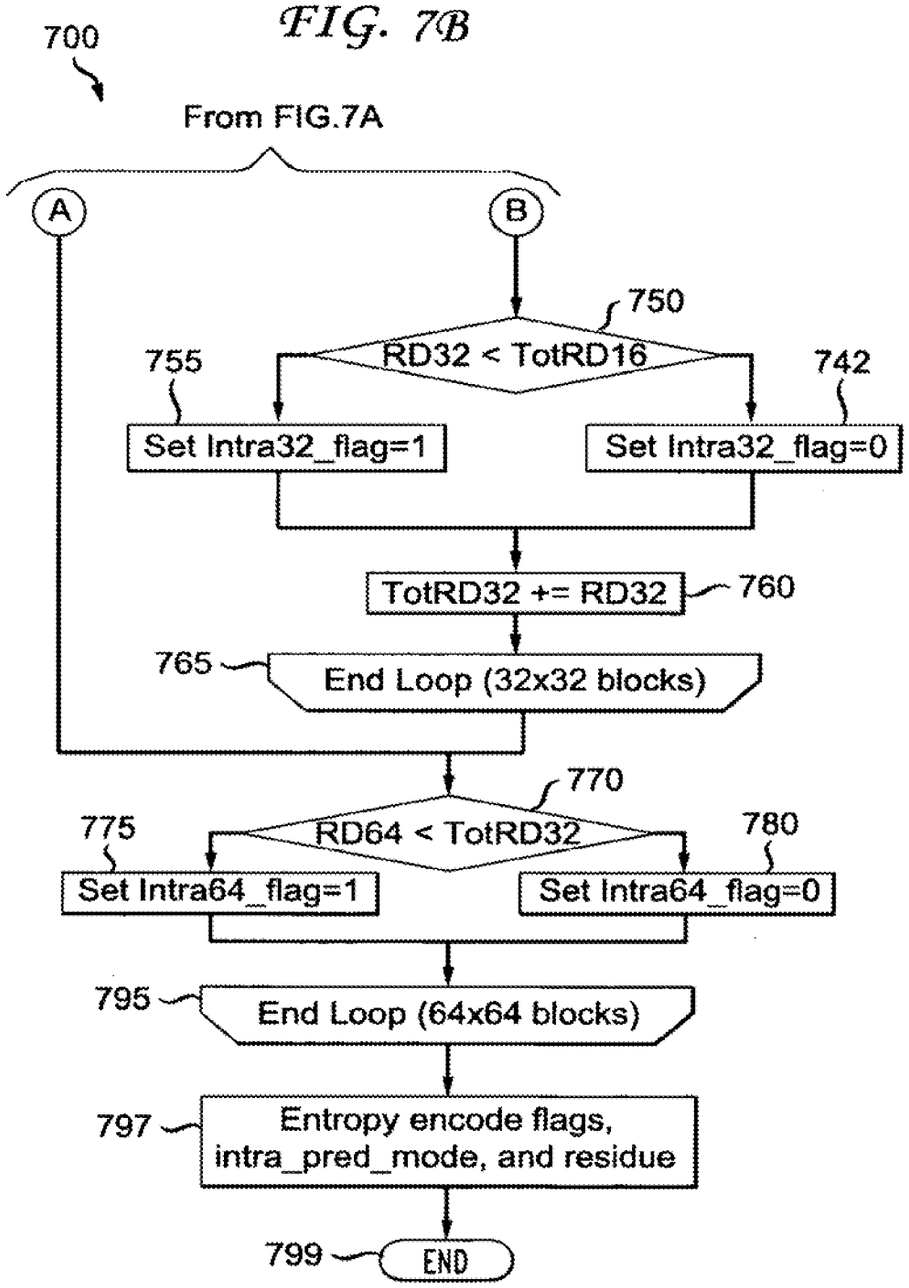

700
From FIG.7A
A
B
750
RD32 < TotRD16
755
742
Set Intra32_flag=1
Set Intra32_flag=0
TotRD32 += RD32
760
765
End Loop (32x32 blocks)
770
RD64 < TotRD32
775
780
Set Intra64_flag=1
Set Intra64_flag=0
795
End Loop (64x64 blocks)
797
Entropy encode flags, intra_pred_mode, and residue
799
END

FIG. 8A

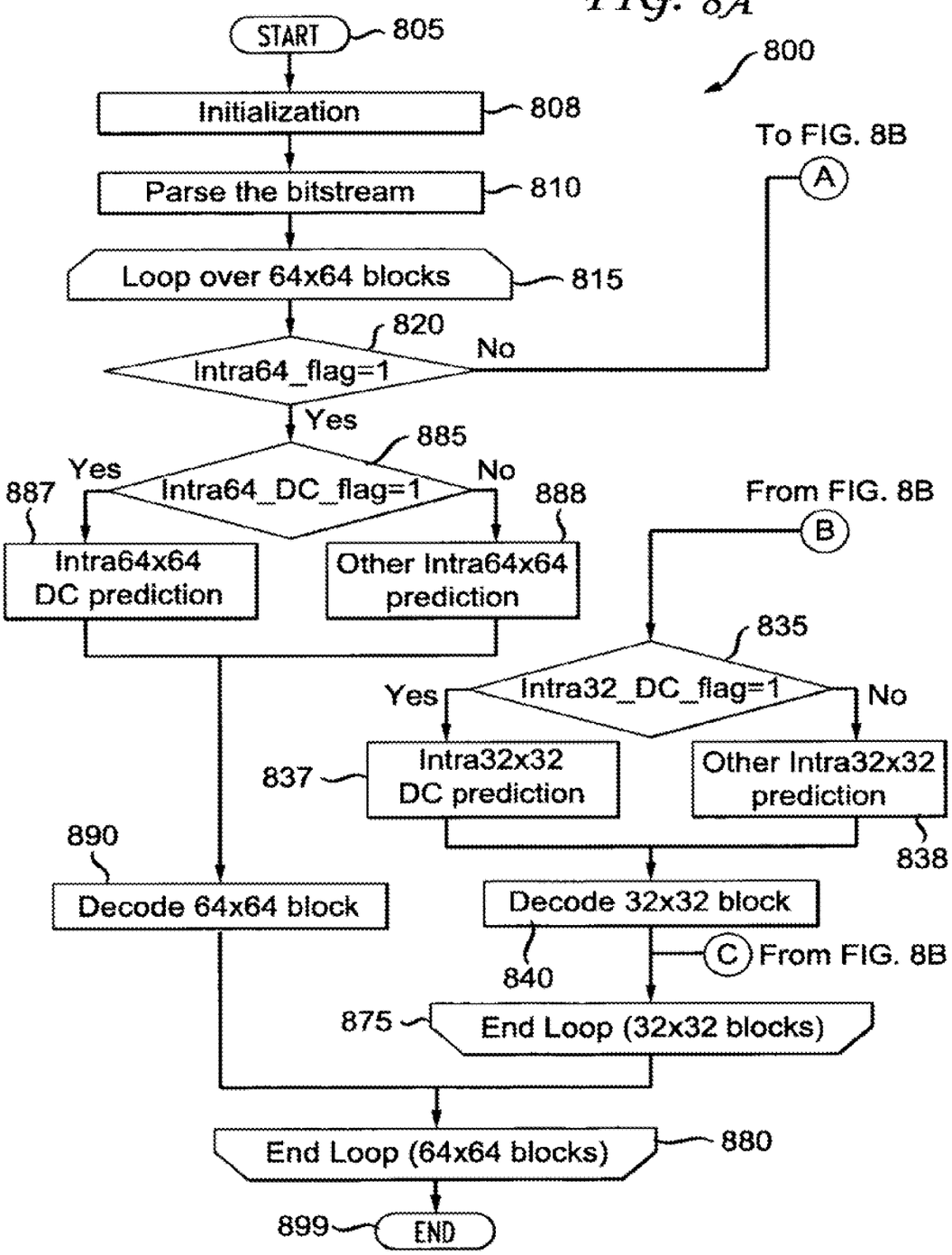
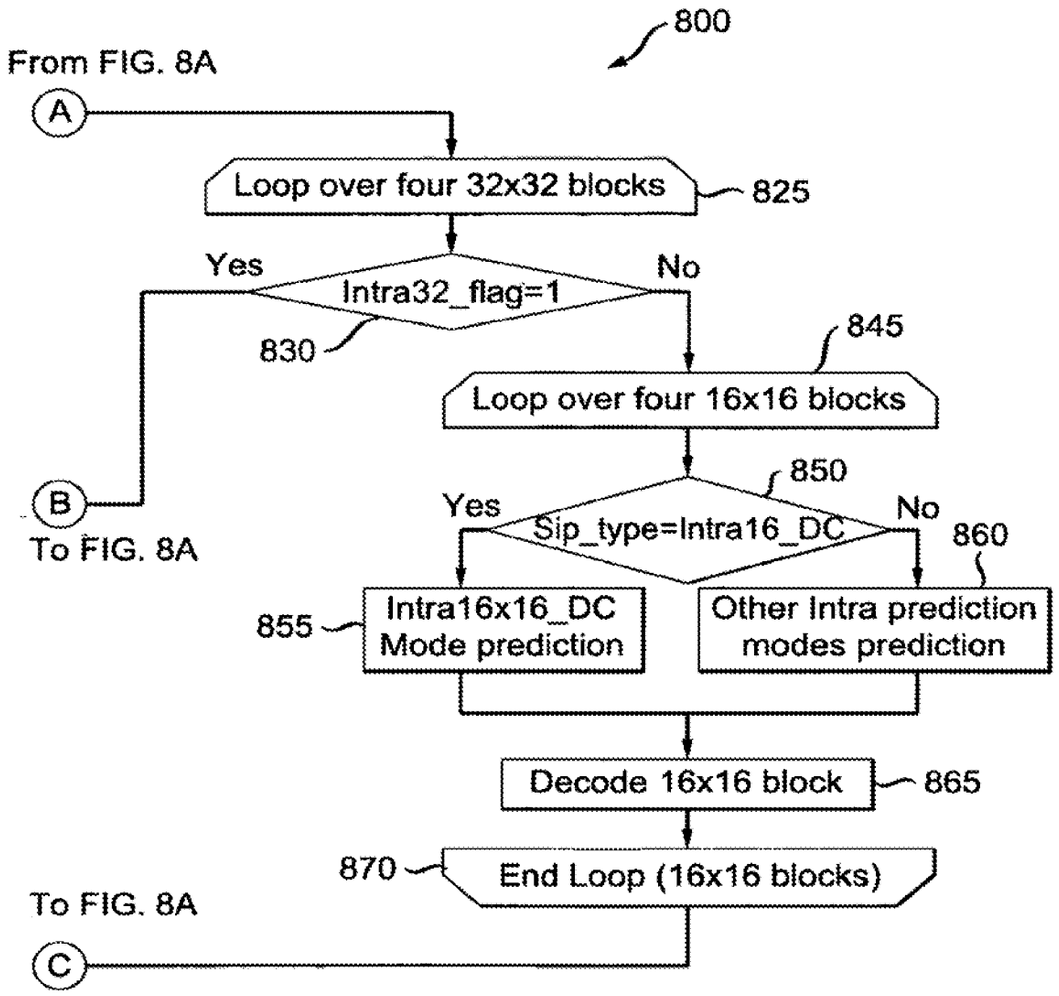

START
805
800
Initialization
808
To FIG. 8B
A
Parse the bitstream
810
Loop over 64x64 blocks
815
820
Intra64_flag=1
No
Yes
885
Intra64_DC_flag=1
Yes
No
887
888
Intra64x64
DC prediction
Other Intra64x64
prediction
From FIG. 8B
B
835
Intra32_DC_flag=1
Yes
No
837
Intra32x32
DC prediction
Other Intra32x32
prediction
838
890
Decode 64x64 block
Decode 32x32 block
840
C
From FIG. 8B
875
End Loop (32x32 blocks)
End Loop (64x64 blocks)
880
899
END

FIG. 8B


800
From FIG. 8A
A
Loop over four 32x32 blocks
825
Yes
Intra32_flag=1
No
830
845
Loop over four 16x16 blocks
850
Yes
Sip_type=Intra16_DC
No
860
B
To FIG. 8A
855
Intra16x16_DC Mode prediction
Other Intra prediction modes prediction
Decode 16x16 block
865
870
End Loop (16x16 blocks)
To FIG. 8A
C

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	To Be Assigned
Filing Date	June 18, 2021
First Named Inventor	Yunfei Zhen
Title	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS
Art Unit	To Be Assigned
Examiner Name	To Be Assigned
Attorney Docket Number	ICVW_2009P00073 US02

SIGNATURE of Applicant or Patent Practitioner

Signature	/Brian J. Hubbard/	Date (Optional)	June 18, 2021
Name	Brian J. Hubbard	Registration Number	45,873
Title (if Applicant is a juristic entity)	Patent Practitioner		
Applicant Name (if Applicant is a juristic entity)	InterDigital VC Holdings, Inc.		

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ *Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

DocuSign Envelope ID: 3487F87B-F448-450B-9D33-9020240B2ADF

Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 03/31/2021. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 15800

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

InterDigital VC Holdings, Inc.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	DocuSigned by: Christos A. Ioannou	Date (Optional)	10/9/2020
Name	Christos A. Ioannou		
Title	Senior Patent Executive		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Yunfei Zheng DocuSigned by: Date : 10/28/2020

Signature: 36C1A46D23064AF...

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Qian Xu

Signature: DocuSigned by: 01DED5E70D27480...

Date: 10/28/2020

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Xiaoan Lu

Signature: DocuSigned by: Xiaoan Lu E918052695DA490...

Date: 10/29/2020

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Peng Yin DocuSigned by: Date : 10/29/2020

Signature: Peng Yin 55FFCCC233584A0...

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Joel Sole Rojals DocuSigned by: Date : 11/6/2020

Signature: E882BECD46BB42D...

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DocuSign Envelope ID: B8A7B10D-865C-478D-8E7B-5A525E7CF362

DOCKET NO.: 2009P00073 US01

Substitute Form for PTO/AIA/01

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

As the below named inventor, I hereby declare that:

This declaration is directed to: ☐ The attached application, or

☒ United States application or PCT international application number 15/806,893

filed on November 8, 2017.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the above-identified application, including the claims.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR 1.56.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Adeel Abbas

DocuSigned by: Adeel Abbas 94F0FAA2FF21491...

Date: 11/6/2020

Signature:

Note: An application data sheet (PTO/AIA/14 or equivalent), including naming the entire inventive entity, must accompany this form. Use an additional form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14.

DOCKET NO.: ICVW_2009P00073 US02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

For: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AUTHORIZATION TO TREAT A REPLY AS INCORPORATING AN EXTENSION OF TIME UNDER 37 C.F.R. § 1.136(a)(3)

The Commissioner is hereby requested to grant an extension of time for the appropriate length of time, should one be necessary, in connection with this filing or any further filing submitted to the U.S. Patent and Trademark Office in the above-identified application during the pendency of this application. The Commissioner is further authorized to charge any fees related to any such extension of time to Deposit Account No. 50-5519.

Date: June 18, 2021

/Brian J. Hubbard/
Name: Brian J. Hubbard
Registration No.: 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
Telephone: (215) 558-5740
Facsimile: (215) 558-5676

DOCKET NO.: ICVW_2009P00073 US02 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: InterDigital VC Holdings, Inc. **Confirmation No.:** Not Yet Assigned

Application No.: Unfiled **Group Art Unit:** Not Yet Assigned

Filing Date: Unfiled **Examiner:** Not Yet Assigned

For: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

Filed Via EFS

INFORMATION DISCLOSURE STATEMENT

Pursuant to 37 CFR § 1.56 and in accordance with 37 CFR §§ 1.97-1.98, information relating to the above-identified application is hereby disclosed. Inclusion of information in this statement is not to be construed as an admission that this information is material as that term is defined in 37 CFR § 1.56(b).

☒ **IDS Filed Under 37 CFR 1.97(b)**

In accordance with § 1.97(b), since this Information Disclosure Statement is being filed either within three months of the filing date of the above-identified application, within three months of the date of entry into the national stage of the above identified application as set forth in § 1.491, before the mailing date of a first Office Action on the merits of the above-identified application, or before the mailing date of a first Office Action after the filing of request for continued examination under § 1.114, no additional fee is required.

CONTENT OF IDS PURSUANT TO 37 CFR 1.98

☒ Copies of reference numbers 01-29 listed on the attached Form PTO-1449 are not required to be submitted pursuant to 37 CFR § 1.98(a)(2)(ii).

☒ Copies of reference numbers 30-73 are not being submitted because they were previously cited by or submitted to the U.S. Patent and Trademark Office in patent application number 13/381,538, filed December 29, 2011 and 15/806,893, filed November 08, 2017 for which a claim for priority under 35 U.S.C. § 120 has been made in the instant application.

REFERENCES IN A LANGUAGE OTHER THAN ENGLISH

☒ The following documents are not in the English language. Accordingly, a concise explanation of the relevance of the document was incorporated in the specification passages identified below, the document was identified in a foreign communication as identified below or an English language counterpart application has been provided as indicated below.

Foreign Language Document	Cite No.	Pages of Reference in Specification or Relevance of Document
CN 101001382 A	33	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
CN 101150719 A	35	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
CN 101207810 A	36	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
CN 101340581 A	38	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
CN 101394565 A	39	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
JP 2004-128749 A	42	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
JP 2009-118233 A	46	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
KR 10-2005-0026318 A	47	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893
KR 10-2009-0065401 A	48	Documents cited w/ IDS dated November 08, 2017, in related US Patent Application No. 15/806,893

Foreign Language Document	Cite No.	English Language Counterpart	Cite No.
CN 1513268 A	30	US 2004/0028282 A1	11
CN 1719905 A	31	US 2006/0013299 A1	12
CN 1722836 A	32	WO 2006/004331 A1	53
CN 101047860 A	34	US 20090022221 A1	19
CN 101273641 A	37	US 8,467,450 B2	05
CN 101467461 A	40	US 2007/0237228 A1	15
JP H08-280020 A	41	US 5,640,420 A	02
JP 2005-519543 A	43	WO 03/075579 A2	50
JP 2007-503784 A	44	US 2006/0251330 A1	14
JP 2007-528675 A	45	WO 2005/093661 A2	52
KR 10-2109244 B1	49	US 2012/0106629 A1	29
WO 2007/034918 A1	54	US 8,467,450 B2	05

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: June 18, 2021

/Brian J. Hubbard/
Brian J. Hubbard
Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
Telephone: (215) 558-5740
Facsimile: (215) 558-5676

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	1	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

U.S. PATENTS						
Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1	5060285		1991-10-22	Dixit et al.	
	2	5640420	A	1997-06-17	Jung, Hae-Mook	
	3	6084908	A	2000-07-04	Chiang et al.	
	4	7302006	B2	2007-11-27	Apostolopoulos et al.	
	5	8467450	B2	2013-06-18	Moriya et al.	
	6	8483285	B2	2013-07-09	Ye et al.	
	7	9113168	B2	2015-08-18	Park et al.	
	8	9288494	B2	2016-03-15	Xu et al.	
	9	10455248	B2	2019-10-22	Park et al.	

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	2	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

U.S. PATENT APPLICATION PUBLICATIONS						
Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	10	20030202602	A1	2003-10-30	Apostolopoulos et al.	
	11	20040028282	A1	2004-02-12	Kato et al.	
	12	20060013299	A1	2006-01-19	Sato et al.	
	13	20060109913	A1	2006-05-25	Wang et al.	
	14	20060251330	A1	2006-11-09	Toth et al.	
	15	20070237228	A1	2007-10-11	Han et al.	
	16	20080123947	A1	2008-05-29	Moriya et al.	
	17	20080123977	A1	2008-05-29	Moriya et al.	
	18	20080267291	A1	2008-10-30	Vieron et al.	
	19	20090022221	A1	2009-01-22	Xie et al.	
	20	20090046781	A1	2009-02-19	Moriya et al.	

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	3	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

U.S. PATENT APPLICATION PUBLICATIONS						
Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	21	20090060345	A1	2009-03-05	Wheeler et al.	
	22	20090154567	A1	2009-06-18	Lei et al.	
	23	20090232206	A1	2009-09-17	Boon et al.	
	24	20100086029	A1	2010-04-08	Chen et al.	
	25	20100086032	A1	2010-04-08	Chen et al.	
	26	20100208827	A1	2010-08-19	Divorra et al.	
	27	20110286520	A1	2011-11-24	Xu et al.	
	28	20110292999	A1	2011-12-01	Jeong et al.	
	29	20120106629	A1	2012-05-03	Zheng et al.	

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	4	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T
	30	1513268	CN	A	2004-07-14	NTT DoCoMo Inc.		
	31	1719905	CN	A	2006-01-11	Sony Corp		
	32	1722836	CN	A	2006-01-18	Samsung Electronics Co., Ltd.		
	33	101001382	CN	A	2007-07-18	Hisense Group Co., Ltd.		
	34	101047860	CN	A	2007-10-03	Huawei Technology Co., Ltd.		
	35	101150719	CN	A	2008-03-26	Huawei Technologies Co., Ltd.		
	36	101207810	CN	A	2008-06-25	Lenovo (Beijing) Co., Ltd.		
	37	101273641	CN	A	2008-09-24	Mitsubishi Electric Corp		
	38	101340581	CN	A	2009-01-07	Shanghai Fullhan Microelectronics Co., Ltd.		
	39	101394565	CN	A	2009-03-25	Chengdu Jiuzhou Electronic Technology Inc.		
	40	101467461	CN	A	2009-06-24	Samsung Electronics Co., Ltd.		
	41	H08-280020	JP	A	1996-10-22	Daiu Denshi KK		

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	5	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T
	42	2004-128749	JP	A	2004-04-22	Toshiba Corp		
	43	2005-519543	JP	A	2005-06-30	Koninklijke Philips Electronics N.V.		
	44	2007-503784	JP	A	2007-02-22	AMT Advanced Multimedia Technology AB		
	45	2007-528675	JP	A	2007-10-11	Thomson Research Funding Corporation		
	46	2009-118233	JP	A	2009-05-28	KDDI Corp		
	47	10-2005-0026318	KR	A	2005-03-15	Samsung Electronics Co., Ltd.		
	48	10-2009-0065401	KR	A	2009-06-22	ICU Research and Industrial Cooperation Group		
	49	10-2109244	KR	B1	2020-05-11	Thomson Licensing		
	50	03/075579	WO	A2	2003-09-12	Koninklijke Philips Electronics N.V.		
	51	2004/104930	WO	A2	2004-12-02	AMT Advanced Multimedia Technology AB		
	52	2005/093661	WO	A2	2005-10-06	Thomson Research Funding Corporation		

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	6	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T
	53	2006/004331	WO	A1	2006-01-12	Samsung Electronics Co., Ltd.		
	54	2007/034918	WO	A1	2007-03-29	Mitsubishi Electric Corporation		
	55	2007/081908	WO	A1	2007-07-19	Thomson Licensing		
	56	2008/004769	WO	A1	2008-01-10	Samsung Electronics Co., Ltd.		
	57	2008/027192	WO	A2	2008-03-06	Thomson Licensing		
	58	2008/088140	WO	A1	2008-07-24	Samsung Electronics Co., Ltd.		
	59	2009/028922	WO	A2	2009-03-05	LG Electronics Inc.		
	60	2009/051719	WO	A2	2009-04-23	Thomson Licensing		
	61	2010/039731	WO	A2	2010-04-08	Qualcomm Incorporated		

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	7	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

NON-PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T
	62	CHEN et al., "Video Coding Using Extended Block Sizes", VCEG-AJ23, ITU - Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group {VCEG), 36th Meeting: San Diego, USA, October 08-10, 2008, pp. 1-3	
	63	ITU-T, "ITU-T Video Coding Experts Group (VCEG)", July 01, 2009, pp. 1-3	
	64	ITU, "Video Coding using Extended Block Sizes", ITU COM 16-C123-E Standard, International Telecommunication Union, January 2009, 4 pages	
	65	ITU-T, "International Telecommunication Union", H.264 Standard, March 01, 2005, 343 pages	
	66	KIM et al., "Enlarging MB Size for High Fidelity Video Coding Beyond HD", VCEG-AJ21, ITU - Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), 36th Meeting: San Diego, CA, October 08-10, 2008, pp. 1-6	
	67	LEE et al., "Technical Considerations for Ad Hoc Group on New Challenges in Video Coding Standardization", ISO/IEC JTC1/SC29/WG11, Hannover, Germany, July 2008, 4 pages	
	68	LIM et al., "Intra Coding Using Extended Block Size", VCEG-AL28, ITU - Telecommunications Standardization Sector, Study Group 16 Question 6, Video Coding Experts Group (VCEG), 38th Meeting: London, UK/Geneva, CH, July 01-08, 2009, pp. 1-5	

Substitute for 1449/PTO **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(use as many sheets as necessary)*				**Application Number**	Not Yet Assigned
				Filing Date	Not Yet Assigned
				First Named Inventor	Yunfei Zheng
				Art Unit	Not Yet Assigned
				Examiner Name	Not Yet Assigned
Sheet	8	of	8	**Attorney Docket Number**	ICVW_2009P00073 US02

NON-PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T
	69	MARPE et al., "H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas", IEEE International Conference on Image Processing, Genova, Italy, September 14, 2005, 4 pages	
	70	NAITO et al., "Efficient Coding Scheme for Super High Definition Video Based on Extending H.264 High Profile", Visual Communications and Image Processing, 2006, 8 pages	
	71	VAISEY et al., "Image Compression with Variable Block Size Segmentation", IEEE Transactions on Signal Processing, Vol. 40, No. 8, August 08, 1992, pp. 2040-2060	
	72	WIEN, MATHIAS, "Variable Block-Size Transforms for H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003, pp. 604-613	
	73	YOSHINO et al., "Preliminary Response for Draft Call for Evidence of High Performance Video Coding", ISO/IEC JTC1/SC29/WG11 m16082, MPEG2008, Lausanne, Switzerland, February 2009	

EXAMINER SIGNATURE			
Examiner Signature		Date Considered	
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.			

Publication number: CN101001382 A
Publication country: CHINA
Publication type: UNEXAMINED APPLIC. OPEN TO PUBLIC INSPECTION
Publication date: 20070718
Application number: CN200610170917
Application date: 20061225
Priority: CN200610170917200612 25CN20061170917;
Assignee: HISENSE GROUP CO ;
Assignee[std]: HISENSE GROUP CO LTD ;
Inventor: SHUMIN PEI TIAN ;
Inventor[std]: TIAN SHUMIN PEI ;
International class[8]: H04N19/176; H04N19/593;
Family members: CN101001382 A CN101001382 B
Title: FRAME PREDICTION AND RESET METHOD OF AVS VEDIO STANDARD
Title: 一种AVS视频标准的帧内预测及重建方法

Abstract:

This invention discloses an in-frame forecast and re-set method of an AVS video standard including an in-frame forecast process and a re-set-up process, in which, at least a part of the data in the macro block used in the current re-set-up and forecast comes from a chip storage of a process chip, which increases speed and efficiency of coding and decoding in the in-frame forecast and re-set-up.

Abstract:

本发明公开了一种AVS视频标准的帧内预测及重建的方法，该方法解决了现有的帧内预测及重建中效率低的问题，所述方法包括帧内预测过程及重建过程。其中，在所述帧内预测及重建过程中，当前宏块预测重建所利用的各个宏块中的数据至少有部分来自处理芯片的片内存储器。采用本发明，提高了帧内预测及重建中编解码的速度以及效率，打破了帧内预测及重建的一个瓶颈。

The present invention discloses a method of AVS video standard intra prediction and reconstruction, the method solves the problem of intra prediction and reconstruction problem of low efficiency, the method comprising intra prediction process and the reconstruction process. Wherein, in the intra prediction and reconstruction process, the current macroblock prediction data reconstruction utilized each macro block at least partially from the memory-chip processing chip.

According to the present invention, improved intra prediction and reconstruction codec speed and efficiency, breaking the intra prediction and reconstruction of a bottleneck.

Claims:

1、一种AVS视频标准的帧内预测及重建方法，包括帧内预测过程及重建过 程，其特征在于，所述帧内预测及重建过程中，当前宏块预测重建所利用的各 个法块中的数椐至少有部分来自处理芯片的片内存储器。

2、如权利要求1所述的方法，其特征在于，所述当前宏块预测重建所利用 的各个宏块中的数据包括当前宏块的上一宏块中的数据、当前宏块的左上宏块 中的数据、当前宏块的上宏块中的数椐以及当前宏块自身的数据。

3、如权利要求2所述的方法，其特征在于，所述当前宏块的上一宏块中的 数据、当前宏块的左上宏块中的数据、当前宏块的上宏块中的数椐以及当前宏 块自身的数据中至少一个存储于处理芯片的片内存储单元中。

4、如权利要求2或3所述的方法，其特征在于，所述当前宏块预测重建所 利用的上一宏块中的数据以及左上法块中的数椐是在上一法块预测重建结束后 存储于处理芯片的片内存储器中的；所述当前宏块预测重建所利用的上宏块中 的数据是在当前宏块所在行开始预测重建之前存储于处理芯片的片内存储器中 的；所述当前宏块预测重建所利用的当前宏块中的数据是在当前宏块部分预测 重建后存储于处理芯片的片内存储单元中的。

5、如权利要求1至3中任一项所述的方法，其特征在于，所述预测及重建 过程中所述当前宏块的4个8*8字节大小的子宏块全部预测重建完毕后一起输 出至处理芯片的片外数据存储器中。

6、如权利要求4所述的方法，其特征在于，所述领测及重建过程中所述当 前宏块的4个8*8字节大小的子宏块全部预测重建完毕后一起输出至处理芯片 的片外帧数椐存储器中。

1, one kind of AVS video standard intra prediction and reconstruction methods, including intra prediction process and the reconstruction process, wherein the intra prediction and reconstruction process, the current macroblock prediction reconstruction use each method in the block number noted at least partially from the memory-chip processing chip.

2, wherein the method of claim 1, characterized in that said current macroblock prediction data utilized by the reconstruction of the respective macroblocks include a macroblock of the current macroblock data, data of the current macroblock left macro block, the macro block in the current macroblock and the current macroblock number noted in its own data.

3, The method of claim 2, wherein the data of the current macroblock in a macroblock, macroblock left macroblock data Currently, on current macro block in the macro block and the current macro block number noted in the chip memory cell data itself is stored in at least one of the processing chip.

4, The method as claimed in claim 2 or 3, characterized in that said current macroblock prediction data reconstructed by the use of a macro-block and the left block method the number noted at the end of the previous block prediction method is stored in the reconstruction process of the chip-chip memory; the data of the current macroblock prediction reconstruction utilized on the macro block is before the current macroblock row start predicting reconstruction a memory chip in the

chip memory; said current reconstructed macroblock prediction data utilized in the current macroblock in the current macroblock is partially reconstructed prediction processing chip stored in the chip storage unit.

5, The method of claim 1 to 3 in any one of claims, wherein four 8 * 8-word prediction and the reconstruction process the current macroblock sub-section the size of the macro block all outputs together after reconstruction forecast to finish processing chip-chip data memory.

6, A method as claimed in claim 4, wherein said sub-macro-collar four 8 * 8 byte size measurement and reconstruction process the current macroblock all together the output prediction block reconstruction has been completed to the chip-chip memory frames noted.

Description:

一种AVS视频标准的帧内预测及重建方法 技术领域 本发明属于图像处理技术领域，尤其涉及一种AVS标准的视频编解码技术。

背景技术 AVS标准是中国自主制定的音视频编码技术标准，其中AVS视频当中具有 特征性的核心技术包括8*8帧数变换、量化、帧内预测、1/4精度像素插值、特 殊的帧间预测运动补偿、二维熵编码及去块效应环内滤波等。

其中，AVS帧内预测算法充分利用图像的空间相关性，用图像块周围的像 素来进行帧内预测，提高了编码效率，从而减少了编码输出码率。在解码端， 根据帧内预测模式的不同可能用其左上块、上块、右上块、左块、左下块的数 据对当前8*8块预测，得到的数据和该块相应的残差值重建得到解码块。

然而，帧内预测时所需要的数据要从片外帧数据存储器中读进数据，预测 重建后再输出到片外帧数据存储器中，频繁的与片外存储器交换数据严重影响 了编

One kind of AVS video standard intra prediction techniques in the field and reconstruction process of the present invention belongs to the field of image processing technology, particularly to a AVS standard video codec technology.

BACKGROUND AVS audio and video coding standard is China's independent development of technical standards, where AVS video core technology has a characteristic which comprises 8 * 8 frames transform, quantization, intra prediction, 1/4-pixel accuracy interpolation, special inter-frame motion compensation prediction, the two-dimensional entropy coding and deblocking in-loop filtering. wherein, AVS intra-prediction algorithm makes full use of spatial correlation of images, with an image like always around the block intra prediction to improve coding efficiency, thereby reducing the encoding output rate. The decoding side, noted in the intra prediction mode

解码速度及效率，成为帧内预测的一个瓶颈。

发明内容 本发明目的在于，提供一种速度更快、效率更高的帧内预测及重建方法。

为了解决上述问题，本发明提出了一种帧内预测及重建方法，包括帧内预 测过程及重建过程，其中，所述帧内预测及重建过程中，当前宏块预测重建所 利用的各个宏块中的数据至少有部分来自处理芯片的片内存储器。

其中，所述当前宏块预测重建所利用的各个宏块中的数据包括当前宏块的 上一宏块中的数据、当前宏块的左上法块中的数据、当前宏块的上宏块中的数 据以及当前宏块自身的数据。

其中，所述当前宏块的上一法块中的数据、当前宏块的左上宏块中的数据、 当前宏块的上宏块中的数据以及当前宏块自身的数据中至少一个存储于处理芯 片的片内存储单元。

另外，所述当前宏块预测重建所利用的上一宏块中的数椐以及左上宏块中 的数据是在上一宏块预测重建结束后存储于处理芯片的片内存储器中的；所述 当前宏块预测重建所利用的上宏块中的数据是在当前宏块所在行开始预测重建 之前存储于处理芯片的片内存储器中的；所述当前宏块预测重建所利用的当前 宏块中的数据是在当前宏块部分预测重建后存储于处理芯片片内存储单元中。

其中，所述预测及重建过程中所述当前宏块的4个8*8字节大小的子宏块 全部预测重建完毕后一起输出至处理芯片的片外数据存储器中。

本发明的AVS视频标准的帧内预测及重建方法具体包括以下步骤： A、指定第一存储器为当前宏块存储器，指定第二存储器为上一宏块存储器； B、按照AVS标准的方法进行帧内预测；对于所需的各个宏块中的数据， 从第一存储器、第二存储器、第三存储器或第四存储器中读取；其中，所述四 个存储器中至少一个是处理芯片的片内存储器； C、按照AVS标准的方法进行变换和重建；重建数据存储于当前宏块存储 器； D、判断当前宏块是否全部预测重建完

may differ by their upper left block, upper block, upper right block, left block, left block data of the current 8 * 8 prediction, and get the number noted in the block corresponding residuals rECONSTRUCTION decoded blocks.

However, the data required for intra prediction frame data from off-chip memory is read into the data, and then rebuild the prediction frame data is output to the memory chip, the frequent exchange of data with off-chip memory seriously affecting the provision decoding speed and efficiency, become a bottleneck intra prediction.

Objective

The present invention is to provide a faster, more efficient method of intra prediction and reconstruction.

In order to solve the above problems, the present invention provides a method of intra prediction and reconstruction, including intra prediction process and the reconstruction process, wherein the intra prediction and reconstruction process, the current macroblock prediction reconstruction utilized data of respective macro blocks at least partially from the on-chip memory processing chip. wherein the current macroblock prediction data reconstruction utilized each macroblock includes data on a macro block of the current macroblock, the data of the current macroblock law left block, the current macroblock data macroblock and the current macroblock own data.

Among them, a method of data blocks of the current macroblock, the data of the current macroblock left macroblock, macroblock data on the current macro block and the current macro block at least one of its own data stored in the memory unit within the chip of the chip.

In addition, the current macroblock prediction reconstruction utilized on a macro block number and the data noted

毕；若完毕，则继续执行，否则，转 向步骤B执行； E、将当前宏块的下一法块预测重建所需的上一宏块行的部分数据存储于所 述第四存储器中；将当前宏块行的下一宏块行预测重建所需的当前宏块行中的 当前宏块中的数据存储于所述第三存储器中相应位置； F、输出当前宏块存储器中的数据； G、指定第二存储器为当前法块存储器，第一存储器为上一宏块存储器； H、当前宏块帧内预测及重建结束。

其中，在所述步骤E中，当前宏块的下一宏块预测重建所需的上一宏块行 的部分数据是指所述下一宏块预测重建所需的位于上一宏块行的其左上宏块中 的数据；所述步骤E中所述相应位置是指当前宏块预测重建时所利用的上一宏 块中数据在第三存储器中的位置。

所述步骤D中全部预测重建完毕是指4个8*8大小的子宏块全部预测重建 完毕。

作为在本发明的一个实施例，所述步骤A及步骤G中对第一存储器、第二 存储器的指定通过互换指向两个存储器的数据指针来实现，存储器中的数据不 变。

本发明通过充分利用处理芯片的片内存储单元预先存储当前宏块进行预测 重建时所需要的其他宏块数据，在帧内预测及重建过程中不需要从片外帧数据 存储器中读取数据即可完成整个宏块的预测重建，从而极大地提高了整个编解 码系统的编解码速度和效率，减少了片内、片外频紫存取数据对帧内预测性能 的影响。

附图说明 图1是本发明基于的处理芯片的一个实施例的结构框图； 图2是基于图1所示处理芯片的工作流程图； 图3是基于图2所示流程的示意图。

具体实施方式 本发明通过有效利用处理芯片内存储器作为帧内预测及重建所用数据的存 储单元，从而避免了频繁的与片外数据存储器交换数据，进而提高了编解码速 度及效率，也打破了帧内预测及重建的一个瓶颈。

为了更有助于理解本发明，下面简要绘制出处理芯片的若干基本组成，借 助这些组成功能框图来阐述本发明。当然，真实地处理芯片十分复杂，本例仅 为有

in the upper left macro block is the memory chip at the end of the previous macroblock prediction stored in the reconstruction process in the chip; chip memory data of the current macroblock prediction reconstruction utilized on the macro block is before the current macroblock row start predicting reconstruction process is stored in the chip in; the current macroblock prediction reconstruction of current use macroblock data is after the current macroblock prediction part in the reconstruction process chip-chip memory storage unit. wherein the prediction and reconstruction process of the four 8 * 8-byte size of the current macro-block sub-macroblock prediction after all outputs together to rebuild the complete chip-chip data memory.

AVS video standard of the present invention, intra prediction and reconstruction method includes the following steps: A, designating the first memory to the current macro block memory, the second memory is designated on a macro block memory; B, in accordance with AVS Standard the method of intra prediction; the data required for each macroblock, from a first memory, a second memory, the third memory or the fourth memory read; wherein said at least one of the four memory processing chip within the on-chip memory; C, in accordance with the AVS standard methods transformation and reconstruction; to rebuild the data stored in the memory of the current macroblock; D, determine whether all the current macroblock prediction reconstruction is completed; if completed, continue, otherwise go to sTEP B execution; E, the current macroblock next block prediction method required to rebuild a part of the data stored on the macro-block line in the fourth memory; the desired current macroblock row to the next macroblock row reconstructed prediction storing data corresponding to the current position of the current macroblock in a macroblock row in the third memory; F,

助于理解本发明而简要的选择了其中的部分组成，本例中的处理芯片组成 框图不应限制本发明 参考图1,图示了本发明所基于的处理芯片的一个实施例的结构。如图所示， 处理芯片1包括： 输入单元14，用于接收输入数据； 存储单元12，用于存储中间或/和结果数据；其中，所述存储单元12又包 括四个部分：第一存储器，大小为16*16字节； 第二存储器，大小为16*16字节； 所述第一存储器与第二存储器用于轮流存放当前要预测和重建的宏块； 第三存储器，大小为一个帧的宽度，用于存储帧内预测和重建当前宏块的 部分子宏块所需的上宏块中的数据； 第四存储器，大小为一个字节，用于存储帧内预测和重建当前宏块的部分 子宏块所需的左上宏块中的数据； 处理单元11,作为芯片的数据计算处理部分； 输出单元14,用于输出数据。

参考图2,图示了基于图1所示处理芯片的一个帧内预测及重建流程。如图 所示，第一存储器、第二存储器、第三存储器及第四存储器均位于处理芯片内； 参考图2，同时参考图1,所述流程包括以下步骤： 步骤S100,指定当前宏块存储器为第一存储器，指定上一宏块处理器为第二存储器；这里的指定动作可以通过数据指针的指向来实现，也可以是其它适当的方式等，不限制本发明； 步骤S101,按照AVS标准的方法进行帧内预测，当前宏块预测所需的当前 宏块数据从当前宏块存储器读取，所需的左宏块中数据从上一宏块存储器读取， 部分子宏块所需的上宏块中和左上宏块中的数据分别从第三存储器和第四存储 器中读取；本步骤的更直观的表示可以参考图3; 步骤S102,按照AVS标准的方法进行IDCT（Inverse Digital Cosine Transform,离散余弦逆变换）变换和重建，重建数据存储在当前宏块存储器中； 在这里，可以采用DMA (DirectMemoryAccess,直接存储器存取）方式以降低 耗时； 步骤S103，判断当前宏块是否预测重建完毕；因为在这里以8*8字节的子 宏块为帧内预测的单位，因此所述大小为16*16字节的宏块，包括4个8*8字 节的子宏块，因此只有当这4个8*8子宏块全部重建完毕之后，所述宏块才称 为重建完毕；若当前宏块重建完毕，则继续步骤S104，否则执行步骤S101; 步骤S104，存储第三存储器中对应当前宏块位置的前一个字节

the output data of the current macroblock memory; G, specifies the second memory is a memory block of the current method, the first memory on a macro-block of memory; H, the current macroblock intra prediction and reconstruction ended.

Part of the data on a macro block row

wherein, in the step E, the current macroblock next macroblock prediction means needed for reconstruction of the next macroblock prediction needed for reconstruction is located on a macro its upper left macro block data of the block line; the step E in the corresponding locations on the macroblock in the current macroblock prediction data used by the reconstruction position in the third memory.

The step (D) refers to the complete reconstruction of all prediction sub-macro 4 8 * 8 size block all prediction reconstruction is completed.

As an embodiment of the present invention, the step A and step G in the first memory, a second memory specified by interchanging the two memory pointers pointing to data to achieve data memory unchanged.

The present invention chip memory chip unit by leveraging previously stored at the current macroblock prediction reconstruction needed another macro block data, do not need to be noted in the number of frames from the memory chip in the intra prediction and reconstruction process reads noted to complete the reconstruction of the entire macro block prediction, thereby greatly enhancing the speed and efficiency of the codec codec system, reducing the chip, affect access to data sheet FSB purple intra prediction performance.

FIG. 1 is a block diagram of one embodiment of the present invention is based on the chip; FIG. 2 is a flowchart

数据到第四 存储器中，其更直观的过程可以参考图3; 步骤S105,存储当前宏块的重建后的最后一行数椐到第三存储器中对应当 前宏块的位置，其更直观的过程可参考图3所示；同时，在这里可以启用DMA 方式以降低耗时； 步骤S106,输出当前宏块重建后的数据，也就是当前宏块存储器中的数据 到片外数据存储器中，例如帧数据存储器；同样，这里也可以采用DMA存取方 式以降彳氏存取耗时； 步骤S107,对当前宏块存储器和上一宏块存储器进行交换指定，也就是指定当前宏块存储器为第二存储器，上一宏块存储器为第一存储器，这里只是指定方向的变化，第一存储器和第二存储器中的数据不变；在这里指定方向的改变可以通过交换当前宏块存储器和上一宏块存储器的数据指针来实现； 步骤S108,结束一个宏块的帧内预测及重建过程。

以上只是一个宏块的帧内预测及重建过程，对于多个宏块可以重复执行本 流程即可。

参考图3，图示了图2所示流程的一个更为直观的示意图。如图所示，当前 宏块存储器（第一存储器）和上一宏块存储器（第二存储器）都为16*16字节 大小，包括4个8*8字节大小的子宏块，在第一存储器中可以直观的看出包括 b0、bl、b2> b3四个子宏块，第二存储器与第一存储器类似，故未示出， 在图2所示步骤SlOl中，所述的过程可以参考图3来理解：当前宏块中， 8*8子宏块b0帧内预测所需的左上宏块中的数据从第四存储器中读取；当前宏 块中，8*8子宏块b0和bl帧内预测所需的上法块中的数据从第三存储器中与当 前宏块对应的存储区a中读取；当前宏块中8*8子宏块b0和b2帧内预测所需 的左宏块中的数提从上一宏块存储器第二存储器中读取；当前宏块中子宏块bl 帧内预测所需要的左块数据从子宏块bO中读取；当前宏块中子宏块b2帧内预 测所需的上块数据从子宏块b0和bl中读取；当前宏块中子宏块b3所需的左块 和上块的数据分别从子宏块bO和b2以及子宏块bl中读取。

在图2所示的步骤S104中，所述的过程如图3中标记Pl所示，即保存第 三存储器中对应当前宏块存储器第一存储器的位置a的前一个字节数椐B至第 四存储器

of the operation based on the processing chip shown in FIG. 1; FIG. 3 is a schematic view of the process shown in Figure 2 based. particular embodiments of the present invention through the effective use of chip memory as intra prediction and reconstruction of the memory cell data used, thus avoiding the frequent exchange data with the memory chip data, thereby improving the speed and efficiency of the codec, also breaking the intra prediction and reconstruction of a bottleneck.

In order to contribute to understanding of the present invention, the following brief drawn a number of basic components processing chip, by means of these components a functional block diagram to illustrate the present invention. Of course, truly very complex processing chip, in this case only helpful in understanding the present invention and a brief selection of which parts, chips block diagram of the processing in this case should not be limited to the present invention with reference to FIG. 1, illustrates this a structure of an embodiment of the invention, the processing chip is based. As shown, the chip 1 comprises: an input unit 14 for receiving input data; storing unit 12 for storing intermediate and / or the results of data; wherein, the storage unit 12 also includes four parts: a first memory, 16 * 16 bytes in size; second memory, the size of 16 * 16 bytes; the first memory and the second memory for storing the current rotation and to the reconstructed macroblock prediction; the third memory, the size of a the width of the frame, for storing data and reconstruction of intra prediction macro block part of the sub macro blocks required on current macro block; fourth memory, the size of one byte for storing intra prediction and reconstruction of the current macro the data portion of the sub-blocks required for the upper left macroblock macroblock; processing unit 11, as the chip data calculation processing

中；在图2所示步骤S105中，所述的过程如图3中标记为P2的从A 到a的过程，即保存当前宏块存储器第一存储器的最后一行数据A至第三存储 器中对应当前宏块存储器第一存储器的位置a。

图3是用来更为直观的阐述本发明的流程，并不完全真实体现本发明，是 为了更有利于公众对技术的理解以充分的公开本发明。因此，不应以图3所示 的结构来限制本发明。

本领域普通技术人员应当理解，对于第一至第四存储器来说，也可以是部 分利用处理芯片的片外存储器、部分利用处理芯片片内存储器，本实施方式中 只是列举出全部为处理芯片片内存储器的情形，对于其他的情形在这里不一一 赘述。同样，由于帧内预测模式的不同，预测所需要的块也不同，在本实施方 式中只是列举了若干情况，其中利用了待预测重建块的左块、上块以及左上块 的数据，对于其他预测模式下的预测，则对应需要其他相应块的数据，例如还 可能需要右上块和左下块的数据等，对于这些所用到的数据同样可以通过处理 芯片的片内存储器存取。但由于对于不同的预测模式在原理上都M本相同的， 对于本领域普通技术人员来说，在本发明的精神的指导下，可以通过简单推理 得出其他模式的运作方法，因此不再对其他模式的方法进行——阐述。

以上所述是本发明的优选实施方式，应当指出，对于本技术领域的普通技 术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这 些改进和润饰也视为本发明的保护范围。

^ Pit HI

section; an output unit 14 for outputting data.

Referring to Figure 2, illustrates a prediction based on a frame processing chip shown in Figure 1 and the reconstruction process. As illustrated, the first memory, the second memory, the third memory and the fourth memory chips are located in the processing; refer to FIG. 2, while the reference to the flowchart in Figure 1 comprises the following steps: Step S100, the current macroblock designated memory a first memory, designated as a second macroblock processor memory; where the data designated by the operation of a pointer can be realized, and may be another suitable manner, which does not limit the present invention; step S101, the AVS standard in accordance with the method of intra prediction, the current macroblock is needed to predict the current macroblock from the current macroblock data read memory, the data required for the left macroblock are read from a memory macro block, the desired sub-macroblock portion on the upper left macroblock and macroblock number noted were read from the third memory and the fourth memory; a more visual representation of this step can refer to FIG. 3; step S102, in accordance with the AVS standard methods IDCT (Inverse digital cosine transform, inverse discrete cosine transform) transformation and reconstruction, reconstruction of the data stored in the current macro block memory; here, you can use DMA (DirectMemoryAccess, direct memory access) mode to reduce the time-consuming; step S103, the current macroblock sentenced Adams predict whether reconstruction is completed; because here 8 bytes * 8 sub-macroblock unit intra prediction, and therefore the size of 16 * 16-byte macroblock, comprising four 8 * 8-byte sub macro block, so only when these four 8x8 sub-macroblock rebuild all finished, it is called the macroblock

reconstruction is completed; if the current macroblock reconstruction is completed, continue to step S104, otherwise step S101; step S104, storage third memory corresponding to the current macroblock position before a byte of data to the fourth memory, which is more intuitive process can refer to Figure 3; step S105, stores the current number noted in the last line of the macro block reconstructed in the third memory the macro block corresponding to the current position, which is more intuitive process can be illustrated with reference to FIG. 3; the same time, where the DMA can be enabled to reduce time-consuming; step S106, the output data of the reconstructed current macro block, which is the current macroblock memory data to the chip data memory, for example, the frame data memory; similarly, where you can also preclude the use of DMA access mode's left foot to reduce access time-consuming; step S107, the current macroblock memory and on a macro block memory exchange specified, the current macroblock is designated as a second storage memory, on a macro block memory as the first memory, designated here just change direction, the data of the first and second memories of the same; here the specified direction changed by exchanging the current macro block memory and data pointer on a macro block of memory to achieve; step S108, the intra prediction and the end of the reconstruction process of a macro block.

These are just a macroblock intra prediction and reconstruction process for a plurality of macro blocks can repeat this process can be.

Referring to Figure 3, illustrates a more intuitive process shown in Fig. 2 Fig. As shown, the current macro block memory (first memory) and on a macro block memory (second memory) are 16 * 16 bytes in size, including four 8 * 8

sub-macro-byte blocks in the first a memory that can be seen include intuitive b0, bl, b2 and gt; b3 four sub-macro block, the second memory similar to the first memory, it is not shown in step SIOI shown in Figure 2, the process It can be understood with reference to FIG. 3: current MB, 8 * 8 sub-macroblock intra prediction data b0 desired upper left macro block is read from the fourth memory; current MB, 8 * 8 sub-macroblock data block b0 and bl law required in intra prediction is read from the third memory with the current macroblock in a corresponding storage area; the current macro block 8 * 8 sub-macroblock intra prediction b0 and b2 the required number of left macroblock reference macroblock read from the memory in a second memory; neutron current macroblock intra prediction macroblock bl needed left block data read in from the sub-macroblock bO; current macroblock the neutron data blocks required for the intra prediction macro block b2 is read from the sub macro block b0 and bl; the current block on the left block and macroblock data b3 required neutron macroblock respectively from the sub-macroblock bO and b2 bl and sub-macroblock read.

In the step shown in FIG. 2 S104, the process described in Figure 3 labeled PI, the third memory is saved in the memory corresponding to the current macroblock a first memory location of the number noted in the previous byte B to the fourth memory; in step S105, as shown in FIG. 2, 3 in the labeling process is shown in the process of P2 from a to a, that saves the current macro block memory of the first memory to the last line of data a third memory corresponding to the current position of a macroblock memory of the first memory.

3 is used to set forth a more intuitive process of the present invention is not entirely true embodiment of the present invention, is more conducive to the public understanding of technology with

full disclosure of the present invention. Therefore, not to the structure shown in Fig. 3 to limit the present invention.

This should be understood by those of ordinary skill, for the first through fourth memory, it may be a part of the use-chip memory chip, the inner part of the chip-chip memory utilization, but the present embodiment include all case processing chip chip memory for other cases where not enumerate. Also, since the different block intra prediction mode, prediction time is different, in the present embodiment, merely listed a number of cases in which the use of the block to be predicted reconstructed left block, upper block and the left block of data, other prediction prediction mode, the corresponding need for other corresponding data blocks, for example, may also need to block the upper right and lower left block of data, these data can also be used by the chip by the chip memory access. However, due to the different prediction mode on the same principles of the present M are, those of ordinary skill, under the guidance of the spirit of the present invention can be drawn from other modes of operation by a simple method of reasoning, and therefore no longer other methods were models - elaboration.

The above is a preferred embodiment of the present invention, it should be noted that those of ordinary skill in the art, in the present invention without departing from the principles of the premise, but also a number of improvements and modifications may be made, and these improvements retouching is also considered the scope of the present invention.

Pit HI

PatentOrder

Publication number: **CN101150719 A**

Publication country: CHINA

Publication type: UNEXAMINED APPLIC. OPEN TO PUBLIC INSPECTION

Publication date: 20080326

Application number: CN200610113256

Application date: 20060920

Priority: CN200610113256 20060920CN200611 1256;

Assignee: HUAWEI TECHNOLOGIES CO ;

Assignee[std]: HUAWEI TECH CO LTD ;

Inventor: MENG XINJIAN ;

Inventor[std]: XINJIAN MENG ;

International class[8]: H04N19/174; H04N19/436;

Family members: CN101150719 A CN101150719 B

Title: PARALLEL VIDEO CODING METHOD AND DEVICE

Title: 并行视频编码的方法及装置

Abstract:

The invention relates to a method of parallel video coding and a device. The invention comprises a host processor and a plurality of coder, the host processor is used for dividing present frame in the video sequence fixedly into macroblocks, distributing all macroblocks to one or a plurality of band groups according to the preset rule; according to the rule of processing load to balance between a plurality of coders, the band group of present frame is devided into one or a plurality of sub-band groups according to the raster scanning sequence; all sub-band groups of present frame which are determined by division and code configuration parameters are sent to a plurality of coders respectively, then parallel sub-band groups of present frame is coded by the coders, outputs codes and parameters respectively; finally, gathering codes and parameters output by each coder, further coding sub-band groups, frames and sequences, outputting whole sequence code. Thereby, the method and the device of the invention are suitable to realtime high definition video code.

Abstract:

本发明涉及一种并行视频编码的方法及装置。本发明主要包括主处理器及多个编码器，所述的主处理器用于将视频序列中的当前帧固定分割成宏块，按预定

The present invention relates to a method and apparatus for parallel video encoding. The present invention includes the main processor and a

规则将所有宏块分配给一个或多个条带组；以及将当前帧的条带组按使多个编码器之间处理负荷均衡原则，并依据光栅扫描的顺序划分为一个或多个子条带组；并将划分确定的当前帧所有子条带组和编码配置参数分别传送给多个编码器，之后，由所述的多个编码器对当前帧的子条带组并行进行编码处理，分别输出码流及参数；最后，将各编码器输出的码流及参数汇聚，进一步完成条带组、帧和序列的编码，输出整个序列码流。因此，本发明的方法和装置适用于实时高清视频编码。

plurality of encoders, the main processor for the current frame in a video sequence is divided into macro blocks fixed, according to a predetermined rule all the macro-blocks are allocated to one or more strips groups; and The bands set the current frame according to the principle of balancing the processing load among multiple encoders, and divided into one or more sub-band group based raster scan order; and dividing the determined current frame groups all sub-bands and coding Configuration parameters are transmitted to a plurality of encoders, followed by said plurality of sub-band encoder set the current frame coding processing is performed in parallel, respectively, and output stream parameters; finally, each encoder output stream and parameters aggregation, further complete strip group, coding frame and sequence the entire sequence of the output stream. Thus, the method and apparatus of the present invention is suitable for real-time HD video encoding.

Claims:

1、一种并行视频编码的方法，其特征在于，包括： 将视频序列中的当前帧分割成宏块，按预定规则将所有宏块分配给一个或多个条带 组； 依据光栅扫描的顺序将所述条带组分别划分为一个或多个子条带组，获得当前帧包含 的多个条带组，每个宏块被分配到同一个子条带组； 将子条带组根据其对应的处理负荷均衡地对应至各个编码器，每一个子条带组对应同 一个编码器，每个编码器对应一个或多个子条带组； 按子条带组和编码器对应关系，将当前帧所有子条带组和编码配置参数传送给多个编 码器； 当当前帧为I帧时，各编码器丢弃所有重建子条带组数据；当当前帧不为I帧时，每个 编码器从其它编码器重建子条带组获取所属子条带组运动估计所需的参考数据； 由所述的多个编码器对当前帧的子条带组并行进行编码处理，在编码过程中将每个子 条带组按光栅扫描的顺序划分为一个或多个条带，完成对所述子条带组中所有宏块的编 码，产生重建子条带组，输出编码码流和参数； 将各编

1, a parallel video coding method comprising: the video sequence of the current frame is divided into macro blocks, according to a predetermined rule all macro block is assigned to one or more strips Group; according to the order of raster scanning the strip group were divided into one or more sub-strip group, to get the current frame includes a plurality of strip group, each macro block is assigned to the same sub-band group; sub Article in accordance with the corresponding set of balanced processing load corresponding to the respective encoders, each sub-group corresponding to the same stripe encoder, each encoder corresponding to one or more sub-slice group; sub-bands by the corresponding group and encoder relations, the current frame and all sub-bands set encoding parameters configured to transmit multiple encoders; when the current

码器输出的码流及参数汇聚，完成条带组、帧和序列的编码，输出整个序列码流。

2、根据权利要求1所述的方法，其特征在于，当当前帧不为I帧时，所述的方法包括：根据各编码器当前帧子条带组和重建帧子条带组的图像区、存储区交叠关系，以及当 前帧子条带组中运动估计的最大搜索区、多参考帧属性，确定编码器之间交换的最少的参 考数据，各编码器之间通过交换操作获得所述参考数据，并利用所述参考数据更新各编码 器中缓存的每个子条带组运动估计搜索区参考数据。

3、根据权利要求1所述的方法，其特征在于，所述的预定规则包括：交错扫描模式、散乱扫描模式、前景和背景扫描模式、打开的盒子Box — out扫描模式、光栅扫描扫描模式、手绢扫描模式、以及需要采用编号指明每一个宏块所属的条带组 的显式扫描模式，或者，将当前帧划分为一个条带组。

4、根据权利要求1所述的方法，其特征在于，所述的子条带组为宽度等于帧宽度的一 个或多个连续的整宏块行。

5、根据权利要求1至4任一项所述的方法，其特征在于，所述的将子条带组根据其对应 的处理负荷均衡地对应至各个编码器的处理包括：根据各个编码器处理能力，以及参考数据传送代价，将当前帧包含的各子条带对应至 各个编码器，令多个编码器同时完成所分配子条带组的编码处理。

6、根据权利要求1所述的方法，其特征在于，在图像组编码过程中，所述的划分成多 个子条带组的处理包括：对于I帧，按照宏块数预测编码处理负荷，并根据编码器的处理能力及所述编码处理负 荷将子条带组对应至各个编码器，令多个编码器同时完成所分配子条带组的编码处理；对于第一个非I帧，当条带组划分不变时，采用I帧的子条带组划分方式进行子条带组 的划分；对于第2个非I帧，根据实际统计得到的第1个非I帧的各子条带组编码处理量预测所述 第2个非I帧各子条带组的编码处理负荷，并根据参考数据载入代价，调整所述第2个非I帧 子条带组的划分，令并行编码所述第

frame is an I frame, each encoder strip discards all sub-set of data reconstruction; when the current frame is not an I frame, Each encoder acquisition sub-strip group movement belongs to estimate the required reference data from other encoders reconstruction sub-strip group; by said plurality of encoders for sub-group of the current frame strip parallel coding, coding Each sub-strip process will set the order of raster scanning divided into one or more strips, the completion of the sub-strip group all macro block coding, generating reconstruction sub-strip group, the output code stream and parameters; the code stream and parameters for each encoder output aggregation, complete coding strip group, frame and sequence the entire sequence output stream.

2, according to the method of claim 1, wherein, when the current frame is not an I frame, said method comprising: a current frame based on each sub-encoder strip group and reconstruction of the sub-frame strip group image area, storage area overlapping relationship, as well as the current sub-frame strip groups in the largest motion estimation search area, multiple reference frames property, determine the least reference data exchange between the encoder, each swap operation between the encoder by obtaining the reference data, and the use of the reference data updating of the encoder buffer strip each sub-set of motion estimation search area reference data.

3, according to the method of claim 1, wherein said predetermined rules include: interlace scan mode, scattered scan mode, foreground and background scanning mode, playing ? box Box - out scanning mode, raster scan scan mode, scan mode handkerchief, and the need to adopt an explicit number specified scanning mode strip group each macro block belongs, or the current frame is divided into a stripe set.

2个非I帧时多个编码器处理时间一致；对于第2个非I帧以后的任一非I帧，根据该帧前一帧或前几帧各子条带组编码处理量预 测所述非I帧各子条带组的编码处理负荷，并根据考帧数据载入代价，调整所述非I帧子条 带组的划分，令并行编码所述非I帧时多个编码器处理时间一致。

7、根据权利要求1所述的方法，其特征在于，所述条带为宽度等于帧宽度的一个或多 个连续的整宏块行。

‘ 8、根据权利要求1所述的方法，其特征在于，所述条带是在对子条带组编码过程中按 照编码码流的长度截分。

9、根据权利要求1所述的方法，其特征在于，所述的每个编码器对子条带组中的每个 宏块进行的编码处理过程包括：. 运动估计，运动补偿； 帧内预测选择，帧内预测，帧内/帧间选择，求残差； 码率控制； 变换、量化； 重排序，熵编码； 反量化，反变换； 重建。

10、根据权利要求1所述的方法，其特征在于，所述的每个编码器对子条带组中的每个 宏块进行的编码处理过程包括：运动估计，运动补偿，使用多个后向或前向参考帧； 帧内预测选择，帧内预测，当前宏块帧内预测时使用同一条带已重建尚未环路滤波的 左边宏块和上边宏块数据；帧内/帧间选择，求残差； 码率控制； 整数变换、量化； 重排序，熵编码，熵编码是上下文基于自适应的变长编码或基于上下文的自适应二进 制算术编码； 反量化，反变换； 重建； 环路滤波。

11、根据权利要求8所述的方法，其特征在于，所述的每个编码器对子条带组中的每个 宏块进行的编码处理过程还包括： 设置选择当前帧所有条带的边界环路滤波模式为不滤波，各编码器各自独立完成每个 条带环路滤波处理，且各编码器、各条带之间不交换信息。

12、根据权利要求11所述的方法，其特征在于，所述的环路滤波包括： 条带的环路滤波在该条带第一个宏块重建完成后启动，或者，条带的环路滤波在该条带重建全部完成后开始。

4, according to the method of claim 1, wherein said group of sub-bands of width equal to the width of the frame one or more contiguous whole macroblock row.

5, the method according to one of claims 1 to 4, any one of the preceding claims, characterized in that the sub-set of equalizer bands correspond to the respective corresponding coded according to the processing load of the process comprising: According to various encoder processing capabilities, as well as reference data transfer cost of each sub-item of the current frame correspond to each band contains an encoder, so that a plurality of encoders encoding process while completing the assigned sub-band groups.

6, according to the method of claim 1, characterized in that, in the image encoding process in the group, the processing is divided into a plurality of sub-band group comprises: the I-frame, macro-blocks according to predictive coding processing load, and the processing capability and the encoding processing load encoder sub-bands correspond to the respective encoder group, make multiple encoders simultaneously assigned sub-band coding strip Group; For the first non-I-frame, when the strip group division unchanged, using sub-group dividing strip the way I frame is divided into sub-bands Group; for the first two non-I-frame, based on actual statistics to get a first coding processing load non-I-frame of each sub-band coding process group Forecasting the first two non-I-frame of each sub-band group and the cost of loading the reference data, the adjustment of the first two non-I-frame sub striped group, so that the parallel encoding the first two non-I-frame when processing a plurality of encoders consistent time; any of the non-I after the first two frames of a non-I-frame, based on the previous frame or previous frame each sub-band coding amount set frame predictive coding

13、一种并行视频编码的装置，其特征在于，包括主处理器和多个编码器，主处理器 用于将待编码的当前帧划分为子条带组，并分别传递给多个编码器，多个编码器并行编码 各子条带组，输出各自编码码流给主处理器，由主处理器生成序列码流并输出； 所述的主处理器包括条带组确定单元、子条带组确定单元及顶层编码单元，其中： 条带组确定单元，用于将视频帧序列中的当前帧固定分割成宏块，按预定规则将所有 宏块分配给一个或多个条带组； 子条带组确定单元，用于对当前帧的所有条带组，依据光栅扫描的顺序将所述条带组 分别划分为一个或多个子条带组，且每个宏块必须并且只能分配到一个子条带组，获得当 前帧包含的多个条带组，每个编码器对应一个或多个子条带组，每一个子条带组对应个编 码器； 子条带组数据传递单元，按子条带组和编码器对应关系将当前帧所有子条带组和编码 配置参数传送给多个编码器； 顶层编码单元：将各编码器输出的码流及参数汇聚，完成条带组、帧和序列的编码， 输出整个序列码流。

所述的编码器包括子条带组接收单元、参考数据输入输出单元及编码单元，其中： 子条带组接收单元，用于接收子条带组和编码配置参数； 参考数据输入输出单元，用于在各编码器之间交换参考数据，当当前帧不为I帧时，控 制编码器之间交换已重建的参考帧子条带组数据，更新各编码器中缓存的每个子条带组运 动估计搜索区参考数据； 编码单元，用于所分配子条带组进行编码处理，在编码过程中将每个子条带组按光栅 扫描的顺序划分为一个或多个条带，完成对所分配子条带组中所有宏块的编码，产生重建 子条带组，输出编码码流和参数。

14、根据权利要求13所述的装置，其特征在于，所述的主处理器的处理包括：将整个帧划分为一个条带组，并进一步划分为多个子条带组，且所有子条带组为宽度 等于帧宽度的一个或多个连续的整宏块行。

15、根据权利要求13所述的装置，其特征在于，所述的多个编码器中的每个编码器还 包括数据存储单元，用于缓存当前和重建的子条带组数据。

processing load of the non-I-frame of each sub-band group and the cost of loading data based on test frame, adjust the sub-division of the non-I-frame strip group, so that parallel encoding said plurality of non-I-frame encoder consistent processing time.

7, according to the method of claim 1, characterized in that the strip is of a width equal to the width of the frame or consecutive whole macroblock row.

'8, according to the method as claimed in claim 1, characterized in that the strip is a strip length in the sub-group of the encoding process in accordance with the code stream of cut-off points.

9, according to the method of claim 1, wherein each of said sub encoder strip groups in each macroblock coding process comprises: The motion estimation, motion compensation; select intra prediction, intra prediction, intra / inter selection, seeking residual; rate control; transform, quantization; reordering, entropy coding; inverse quantization, inverse transform; reconstruction.

10, according to the method of claim 1, wherein each of said sub encoder strip groups in each macroblock coding process comprises: motion estimation, motion compensation using a plurality of backward or forward reference frame; selecting intra prediction, intra prediction, using the same loop filter has not yet been rebuilt with the left and upper macro macroblock intra prediction of the current macroblock block data; intra / inter selection, seeking residual; rate control; integer transform, quantization; reordering, entropy coding, entropy coding is context adaptive binary arithmetic coding based on adaptive variable length coding or context-based; inverse quantization, inverse transform; reconstruction; loop filter.

16、根据权利要求13所述的装置，其特征在于，所述的条带为宽度等于帧宽度的一个 或多个连续的整宏块行，而且，所述的同一个宏块仅能被划分到同一条带中。

17、根据权利要求13所述的装置，其特征在于，所述的装置主处理器还包括控制单 元，用于初始化、配置及控制主处理器和多个编码器完成整个视频序列编码。

18、根据权利要求13至16任一项所述的装置，其特征在于，所述的多个编码器中的每 个编码器在进行编码过程中还包括； 将当前帧所有条带的边界环路滤波模式设置为不滤波，各编码器各自独立完成每个条 带环路滤波处理，各编码器、各条带之间不交换信息，且条带的环路滤波在该条带第一个 宏块重建完成后启动，或者，条带的环路滤波在该条带重建全部完成后开始。

19、根据权利要求13至16任一项所述的装置，其特征在于，所述的多个编码器仅完成 宏块级熵编码输出宏块码流和参数，主处理器顶层编码模块完成条带级熵编码。

11, according to the method of claim 8, wherein said encoding process for each sub-encoder strip groups in each macro block further comprises: Set the Select boundary loop filtering mode current frame all stripes not to filter, each coder independently complete each strip loop filter processing, and each encoder, do not exchange information between each strip.

12, according to the method of claim 11, wherein said loop filter comprising: a loop filter strips in the strip first macroblock reconstruction completed After starting, or loop filter strip after strip reconstruction completed ? beginning.

13, a parallel video encoding apparatus, characterized by comprising a main processor and a plurality of encoders, the main processor for a current frame to be encoded is divided into sub-strip group and passed separately to multiple encoders, multiple encoders parallel encoding each sub-strip group, each output code stream to the host processor to generate a sequence code stream by the main processor and output; said main processor include strip group determining unit, sub-slice group determination unit and the top coding unit, wherein: strip group determining unit for the sequence of video frames of the current frame is divided into macro blocks fixed, according to a predetermined rule all the macro-blocks are allocated to a or more strips groups; sub strip group determination unit for all of the current frame strip group, according to the order of raster scan the stripe group were divided into one or more sub-band group, and each macroblock must and can only be assigned to a sub-strip group, to get the current frame includes a plurality of strip group, each of the encoders corresponds to one or more sub-band groups, each sub-group corresponding to the strip encoder; child stripe set of data transfer unit, according to the sub-band group and

the correspondence between the encoder transmits the current frame and all sub-bands set encoding parameters to configure multiple encoders; top-level coding unit: stream and parameters of each encoder output aggregation, complete coding strip group, frame and sequence the entire sequence output stream.

Encoder

Comprises sub-band group receiving unit, reference data input unit and the encoding unit, wherein: the sub-strip group receiving unit for receiving the sub-band coding group and configuration parameters; reference data input and output units, for encoder switching between the reference data, when the current frame is not an I frame, the control switching between the encoder has reconstructed reference frames of data sub-strip group, updating of the encoder buffer Each sub-band set of motion estimation search area reference data; one or more bar coding unit for sub-band coding process group allocation, in the encoding process in each sub-strip group sequentially into raster scanning band, the completion of the assigned sub-band group all macro block coding, generating reconstruction sub-strip group, the output code stream and parameters.

14, according to the apparatus of claim 13, wherein the processing of the main processor comprises: the entire frame into a strip group, and further divided into a plurality of sub-strip group and all sub-bands set a width equal to the frame width of one or more contiguous whole macroblock row.

15, The apparatus of claim 13, wherein, wherein said plurality of encoders each encoder further comprises a data storage unit for buffering current and reconstruction The sub-band set of data.

16, apparatus according to claim 13, characterized in that said strip of width equal to the width of a frame or a plurality of contiguous whole macroblock row, and, the can only be described with one macro block is divided into the same band.

17, The apparatus of claim 13, wherein, wherein said main processor means further comprises a control unit, is used to initialize, configure and control of the main processor and a plurality of complete the entire video sequence encoder encoding.

18, apparatus according to any one of claims 13 to 16 claim wherein said plurality of encoders each encoder during the encoding process further comprises; not the boundary loop filter mode current frame all set to filter strips, each coder independently complete each strip loop filter processing, each encoder, do not exchange information between the strips, and strips loop filter in the strip after the first macro block reconstruction completed startup, or loop filter strip in the strip after the completion of the reconstruction to start all over.

19, apparatus according to claim any one of claims 13 to 16, characterized in that said plurality of encoders complete macroblock level only entropy coding macroblocks output stream and parameters, the main processor to complete stripe encoding module top level entropy coding.

Description:

并行视频编码的方法及装置 技术领域 本发明涉及视频编码技术领域，尤其涉及一种并行视频编码技术。

背景技术 视频编码技术是将数字视频信息压缩，以便实现更有效地被传送和存储。视频编码是 多媒体数据处理的核心技术。

Parallel video coding method and device TECHNICAL FIELD The present invention relates to a video coding technology, and more particularly relates to a parallel video coding techniques.

目前，视频压缩编码标准主要包括：国际电信联盟标准化部（ITU-T)视频编码专家 组（VCEG)制定的活动图像编码标准H.261、H. 263； IS0/IEC联合的活动图像专家组 (MPEG)制定的视频编码标准MPEGl、MPEG4-Part2； ITU-T视频编码专家组（VCEG)和 IS0/IEC MPEG联合专家组（JVT)共同制定的视频编码标准MPEG2/H. 262、H. 264/AVC (MPEG4-PartlO)；另外还有VC — 1 (前身是WMV — 9)和音视频标准组(AVS)制定的视频 编码标准AVS 1.0-P2等。

MPEG (活动图像专家组）、JVT (MPEG联合专家组）及VCEG (视频编码专家组）系列 视频编码标准基本框架如图1所示，采用基于块的运动补偿和变换编码的混合编码架构，包 括帧内预测、帧间预测、变换、量化和熵编码等。其中，帧间预测是使用基于块的运动矢 量来消除图像间的冗余；帧内预测是使用空间预测模式来消除图像内的冗余。再通过对预 测残差进行变换和量化消除图像内的视觉冗余。最后，运动矢量、预测模式、量化参数和 变换系数用熵编码进行压缩。视频解码过程的基本处理单元是宏块，且一个宏块通常包括 一个16x16的亮度样值块和对应的色度样值块。

不同的标准的所使用工具的有一定差别。对于新一代标准H.264/AVC（MPEG4-PartlO)、VC-K AVS1.0-P2而言，去块滤波器是必须的模块，被称为环路滤波而在 MPEG2、H.263、MPEG4-Part2标准中，去块滤波器仅是解码器中可选的一个后处理环节。

实时视频编码器的输入为高清视频信号，实时完成视频压缩编码，输出码流。实时视 频编码器是数字电视头端（Headend)系统的基本设备，也广泛用于视频会议、数字摄像、 可录的DVD播放机等设备中。

ll H. 264/AVC (MPEG4-PartlO)、VC—l、AVS1. 0-P2被称为新一代视频压缩编码标准， 同以MPEG2为代表的上一代标准相比，新一代标准的压缩率提供一倍以上，但复杂度也增加 2倍以上，实现的难度大大增加。

Background video coding technology is the digital video compression, in order to achieve more effectively transmitted and stored. Video coding is a core technology of the multimedia data processing.

Currently, video coding standards include: International Telecommunication Union Standardization Sector (ITU-T) standard video encoding Video Coding Experts Group (VCEG) developed H.261, H 263; IS0 / IEC Joint moving picture. video coding Experts Group standard MPEGl (MPEG) to develop, MPEG4-Part2;. video coding standard MPEG2 ITU-T video coding Experts Group (VCEG) and IS0 / IEC MPEG joint Expert Group (JVT) jointly developed / H 262, H. 264 / AVC (MPEG4-PartlO); in addition to VC - 1 (the predecessor of WMV - 9) and audio and video standards group (AVS) video coding standard developed by AVS 1.0-P2 and the like.

MPEG (Moving Picture Experts Group), JVT (MPEG Joint Experts Group) and VCEG (Video Coding Experts Group) series of video coding standard basic framework shown in Figure 1, the use of hybrid coding based on motion compensation and transform coding block architecture, including intra prediction, inter prediction, transformation, quantization and entropy coding. Among them, the inter prediction using block-based motion vectors to eliminate redundant between images; intra prediction is to use spatial prediction mode to eliminate redundancy within the image. Then through the prediction residual transform and quantization eliminate visual redundancy within the image. Finally, the motion vector, predictive mode, quantization parameter, and entropy coded transform coefficients is compressed. The basic process unit of the video decoding process is a macroblock, and a 16x16 macroblock typically includes a block of samples of

高清电视（HDTV)通常是指每帧扫描行数为逐行720线或隔行1080线及其以上的活动 图像。目前常见的高清格式有：720p (分辨率1280X720，帧频为24、30、60)，1080i (隔行扫描，每帧分辨率1920X1088，场频为60)，1080p (分辨率1920X 1088，帧频为 24、30)。未来更高分辨率的视频也会获得应用。高清视频能提供更高的视频质量，同 时，高清视频压缩编码的实现更为困难。

以新一代标准H.264/AVC为例，由于引入了多参考帧运动补偿、最小为4X4的可变块 尺寸预测、丰富的帧内预测模式、环路滤波、上下文自适应的变长编码（CAVLC)或上下文 自适应的算术编码（CABAC)等工具，使得编码器复杂度大大增加，据评估，在采用全搜索 运动估计的情况下，H.264/AVC HDTV720p编码器的计算复杂度约为3600 giga-instructions per second (GIPS)、Memory 访问流量约 5570 giga-bytes per second (GBytes/s)。1080i和1080p情况下更为巨大。

由于高清视频编码器巨大的计算复杂性，通常采用单个处理器难于实现实时编码。尤 其是在头端（Headend)等应用场合，需要支持多通道、多种视频格式和压缩编码标准以及 转码等应用环境中，采用单个编码器实现编码的难度更为突出。因此，高清编码器常常需 要采用多芯片（即多个编码器）并行进行编码处理实现。

然而，目前业界还没有一种并行视频编码处理方案能很好满足H.264/AVC (MPEG4-PartlO)、VC-K AVS1.0-P2等新一代视频压缩编码标准头端等应用的高性能实时高清编码要 求。

发明内容 本发明的目的是提供一种并行视频编码的方法及装置，从而提供一种高效的并行视频 编码实现，满足高性能实时高清视频编码。

本发明的目的是通过以下技术方案实现的： 本发明提供了一种并行视频编码的方法，该方法包括： 将视频序列中的当前帧分割成宏块，按预定规则将所有宏块分配给一个或多个条带 组； 依据光栅扫描的顺序将所述条带组分别划分为一个或多个子条带组，获得当前帧包含 的

the luminance and chrominance samples corresponding block.

There are some differences between the different standards used tools. For the next-generation standard H.264 / AVC (MPEG4-PartlO), VC-K AVS1.0-P2, deblocking filter module is required, and is known as a loop filter in MPEG2, H.263, MPEG4 -Part2 standard, the deblocking filter is only a decoder optional post-processing stage. real-time video encoder inputs for high-definition video signal in real time video coding, the output stream. Real-time video encoder is the basic equipment of digital TV head-end (Headend) system is also widely used for video conferencing, digital cameras, recordable DVD players and other devices.

Il H. 264 / AVC (MPEG4-PartlO), VC-l, AVS1. 0-P2 is called a new generation of video compression standard, the same generation as the representative of the MPEG2 standard compared to the standard of the new generation the compression ratio to provide more than doubled, but the complexity is increased by more than 2 times, difficult to achieve greatly increased. high-definition television (HDTV) usually refers to the number of scanning lines per frame progressive or interlaced 720 lines and 1080 lines of moving picture above. Current common HD formats: 720p (resolution of 1280X720, frame rate is 24,30,60), 1080i (interlaced, each frame resolution 1920X1088, field frequency is 60), 1080p (resolution of 1920X 1088, a frame rate of 24, 30). The future will get higher resolution video applications. HD video can provide higher video quality, at the same time, more difficult to achieve high-definition video compression.

In the new generation of standard H.264 / AVC, for example, the introduction of multiple reference frame motion compensation, a minimum of variable block size 4X4 forecasts, rich

多个条带组，每个宏块被分配到同一个子条带组；将子条带组根据其对应的处理负荷均衡地对应至各个编码器，每一个子条带组对应同 一个编码器，每个编码器对应一个或多个子条带组；按子条带组和编码器对应关系，将当前帧所有子条带组和编码配置参数传送给多个编码器；当当前帧为I帧时，各编码器丢弃所有重建子条带组数据；当当前帧不为I帧时，每个 编码器从其它编码器重建子条带组获取所属子条带组运动估计所需的参考数据；由所述的多个编码器对当前帧的子条带组并行进行编码处理，在编码过程中将每个子 条带组按光栅扫描的顺序划分为一个或多个条带，完成对所述子条带组中所有宏块的编 码，产生重建子条带组，输出编码码流和参数；将各编码器输出的码流及参数汇聚，完成条带组、帧和序列的编码，输出整个序列码 流。

当当前帧不为I帧时，所述的方法包括：根据各编码器当前帧子条带组和重建帧子条带组的图像区、存储区交叠关系，以及当 前帧子条带组中运动估计的最大搜索区、多参考帧属性，确定编码器之间交换的最少的参 考数据，各编码器之间通过交换操作获得所述参考数据，并利用所述参考数据更新各编码 器中缓存的每个子条带组运动估计搜索区参考数据。

所述的预定规则包括：交错扫描模式、散乱扫描模式、前景和背景扫描模式、打开的 盒子Box — out扫描模式、光栅扫描扫描模式、手绢扫描模式、以及需要采用编号指明每一 个宏块所属的条带组的显式扫描模式，或者，将当前帧划分为一个条带组。

所述的子条带组为宽度等于帧宽度的一个或多个连续的整宏块行。

所述的将子条带组根据其对应的处理负荷均衡地对应至各个编码器的处理包括：根据各个编码器处理能力，以及参考数据传送代价，将当前帧包含的各子条带对应至 各个编码器，令多个编码器同时完成所分配子条带组的编码处理。

在图像组编码过程中，所述的划分成多个子条带组的处理包括：对于I帧，按照宏块数预测编码处理负荷，并根据编码

intra-prediction mode, loop filter, context-adaptive variable length coding (CAVLC) or context adaptive arithmetic coding (CABAC) and other tools to make the encoder complexity greatly increased, according to the assessment, in the case of full search motion estimation situation, H.264 / AVC HDTV720p encoder computational complexity of about 3600 giga-instructions per second (GIPS), Memory traffic of about 5570 giga-bytes per second (GBytes / s). 1080i and 1080p under even greater situation.

The computational complexity due to the large high-definition video encoders typically use a single processor difficult to achieve real-time encoding. Especially at the head end (Headend) and other applications need to support multi-channel, multi-format video encoding and compression standard as well as transcoding application environments, the difficulty of using a single encoder implementation coding is more prominent. Thus, high-definition encoder often require multi-chip (i.e., a plurality of encoder) encodes the parallel processing implementation.

However, the industry there is not a parallel video coding scheme can satisfy the H.264 / AVC (MPEG4-PartlO), VC-K AVS1.0-P2 new generation of video compression standard applications such as headend high-performance real-time HD encoding requirements. object of the present invention to provide a method and apparatus for parallel video coding, thus providing an efficient parallel video Coding for high-performance real-time HD video encoding. object of the present invention is achieved by the following technical scheme: The present invention provides a parallel video coding, the method comprising: a current frame of a video sequence is divided into macro blocks, according to a predetermined rule all macro block allocation to one or more slice group; according to the order of the raster scan

器的处理能力及所述编码处理负 荷将子条带组对应至各个编码器，令多个编码器同时完成所分配子条带组的编码处理； 对于第一个非I帧，当条带组划分不变时，采用I帧的子条带组划分方式进行子条带组 的划分； 对于第2个非I帧，根据实际统计得到的第1个非I帧的各子条带组编码处理量预测所述 第2个非I帧各子条带组的编码处理负荷，并根据参考数据载入代价，调整所述第2个非I帧 子条带组的划分，令并行编码所述第2个非I帧时多个编码器处理时间一致； 对子第2个非I帧以后的任一非I帧，根据该帧前一帧或前几帧各子条带组编码处理量预测所述非I帧各子条带组的编码处理负荷，并根据考帧数据载入代价，调整所述非I帧子条 带组的划分，令并行编码所述非I帧时多个编码器处理时间一致。

所述条带为宽度等于帧宽度的一个或多个连续的整宏块行。

所述条带是在对子条带组编码过程中按照编码码流的长度截分。

所述的每个编码器对子条带组中的每个宏块进行的编码处理过程包括： 运动估计，运动补偿； 帧内预测选择，帧内预测，帧内/帧间选择，求残差； 码率控制； 变换、量化； 重排序，熵编码； 反量化，反变换； 重建。

所述的每个编码器对子条带组中的每个宏块进行的编码处理过程包括： 运动估计，运动补偿，使用多个后向或前向参考帧； 帧内预测选择，帧内预测，当前宏块帧内预测时使用同一条带已重建尚未环路滤波的 左边宏块和上边宏块数据；帧内/帧间选择，求残差； 码率控制； 整数变换、量化； 重排序，熵编码，熵编码是上下文基于自适应的变长编码或基于上下文的自适应二进 制算术编码； 反量化，反变换； 重建； 环路滤波。

所述的每个编码器对子条带组中的每个宏块进行的编码处理过程还包括：设置选择当 前帧所有条带的边界环路滤波模式为不滤波，各编码器各自独立完成每个条带环路滤波处 理，且各编码器、各条带之间不交换信息。

strip group were divided into one or more sub-strip group, to obtain the current frame includes a plurality of strip group, each macro block is assigned to the same sub-strip group; the group of sub-bands corresponding to a balanced manner according to the processing load corresponding to the respective encoders, each sub-group corresponding to the same band encoder, each encoder strip corresponding to one or more sub-groups; press sub-group and encoder strip correspondence between the current frame and all sub-strip coding group configuration parameters to the plurality of encoders; when the current frame as an I frame, the respective encoder strip discards all reconstruction sub-set of data; when the current frame is not an I frame, each encoder acquisition sub-band set of motion estimation belongs to the required reference data from other sub-band encoder reconstruction group; by said plurality of encoders for the current frame with sub-article the parallel encoding processing, the encoding process will be in each sub-strip group sequential raster scanning is divided into one or more strips, the completion of the sub-strip groups encoding all the macro blocks, resulting in sub-band reconstruction group, the output code stream and parameters; stream and parameters of each encoder output convergence, complete coding strip group, frames and sequences, the entire sequence output stream.

When the current frame is not an I frame, said method comprising: the current frame of each encoder strip groups and sub-sub-frame reconstruction slice group image area, storage area overlapping relationship, and the current sub-frame strip the group with the largest motion estimation search area, multiple reference frames attribute identifies the least reference data exchanged between the encoder, the encoder by switching between the operation of obtaining the reference

所述的环路滤波包括：条带的环路滤波在该条带第一个宏块重建完成后启动，或者， 条带的环路滤波在该条带重建全部完成后开始。

本发明还提供了一种并行视频编码的装置，包括主处理器和多个编码器，主处理器用 于将待编码的当前帧划分为子条带组，并分别传递给多个编码器，多个编码器并行编码各 子条带组，输出各自编码码流给主处理器，由主处理器生成序列码流； 所述的主处理器包括条带组确定单元、子条带组确定单元及顶层编码单元，其中： 条带组确定单元，用于将视频帧序列中的当前帧固定分割成宏块，按预定规则将所有 宏块分配给一个或多个条带组； 子条带组确定单元，用于对当前帧的所有条带组，依据光栅扫描的顺序将所述条带组 分别划分为一个或多个子条带组，且每个宏块必须并且只能分配到一个子条带组，获得当 前帧包含的多个条带组，每个编码器对应一个或多个子条带组，每一个子条带组对应个编 码器； 子条带组数据传递单元，按子条带组和编码器对应关系将当前帧所有子条带组和编码 配置参数传送给多个编码器； 顶层编码单元：将各编码器输出的码流及参数汇聚，完成条带组、帧和序列的编码， 输出整个序列码流。

所述的编码器包括子条带组接收单元、参考数据输入输出单元及编码单元，其中： 子条带组接收单元，用于接收子条带组和编码配置参数； 参考数据输入输出单元，用于在各编码器之间交换参考数据，当当前帧不为I帧时，控 制编码器之间交换已重建的参考帧子条带组数据，更新各编码器中缓存的每个子条带组运 动估计搜索区参考数据； 编码单元，用于所分配子条带组进行编码处理，在编码过程中将每个子条带组按光栅 扫描的顺序划分为一个或多个条带，完成对所分配子条带组中所有宏块的编码，产生重建 子条带组，输出编码码流和参数。

所述的主处理器的处理包括：将整个帧划分为一个条带组，并进一步划分为多个子条 带组，且所有子条带组为宽度等于帧宽度的一个或多个连续的整宏块行。

data, and using the reference data updating of coding each sub-band set of motion estimation search is in cache area reference data.

Predetermined rule

comprising: interlace scan mode, scan mode scattered, foreground and background scan mode, playing ? box Box - out scanning mode, scanning raster scan mode, scan mode handkerchief, and the need to use every specified number explicit macroblock scan mode strip group belongs, or the current frame is divided into a stripe set.

Sub Article

The band width set equal to the frame width of one or more contiguous whole macroblock row.

The sub-bands corresponding to a balanced group according to its processing load corresponding to each encoder process comprises: According to various encoder processing capabilities, as well as reference data transfer cost of each sub-frame contains the current tape strip correspond to the respective encoders, encoders simultaneously make a plurality of processing sub-encoded strip group assignment. group encoding process in the image, the processing is divided into a plurality of sub-band group comprises: the I frame, the number of macroblock prediction encoding processing load and the processing capability of the encoding processing load and encoder the corresponding sub-band set to respective encoders, make multiple encoders while completing the assigned sub-band coding group; for the first non-I-frame, when the tape group division unchanged, the use of sub-frame I strip group division mode is divided sub-band group; for the first two non-I-frame, according to a non-strip coding amount of each sub-group

所述的多个编码器中的每个编码器还包括数据存储单元，用于缓存当前和重建的子条 带组数据。

所述的条带为宽度等于帧宽度的一个或多个连续的整宏块行，而且，所述的同一个宏 块仅能被划分到同一条带中。

所述的装置主处理器还包括控制单元，用于初始化、配置及控制主处理器和多个编码 器完成整个视频序列编码。

所述的多个编码器中的每个编码器在进行编码过程中还包括： 将当前帧所有条带的边界环路滤波模式设置为不滤波，各编码器各自独立完成每个条 带环路滤波处理，各编码器、各条带之间不交换信息，且条带的环路滤波在该条带第一个 宏块重建完成后启动，或者，条带的环路滤波在该条带重建全部完成后开始。

所述的多个编码器仅完成宏块级熵编码输出宏块码流和参数，主处理器顶层编码模块 完成条带级熵编码。

由上述本发明提供的技术方案可以看出，本发明提供了一种并行视频编码处理方案，从而可以在 视频编码过程中选择并行的视频编码方式，使得本发明还可以很好地满足高性能实时高清视频编码所 需的处理能力。同时，本发明中，由于采用负载均衡的原则将粒度适中的待编码数据提供给多个编码 器，并由多个编码器并行对视频数据进行编码处理，从而使得各处理器之间尽可能地减少相互等待的 时间，并可以尽量减少编码器之间交互的开销尽量减少，因此，本发明提供的并行视频编码处理方案 还具有较高的视频编码效率。

附图说明 图1为现有技术中的视频编码框架示意图； 图2为本发明所述的方法的具体实现过程示意图•， 图3为本发明所述的装置的具体实现结构示意图。

具体实施方式 本发明主要是在视频编码过程，将条带组进一步划分为多个子条带组，每个子条带组 包括一个或多个条带，基于所述的子条带组利用多个编码器进行并行编码。

I-frame prediction of the actual statistics to get the first two non-I-frame coding processing load of each sub-band group and the cost of loading the reference data to adjust the division of the second non-I-frame sub-strip group, so that parallel encoding said second plurality of non-I-frame encoder processing time consistent; any of the first two pairs of non-I-frame after a non-I-frame, based on the previous frame or frames the first few frames of each sub-band coding amount prediction group of the non-I-frame of each sub-band group coding processing load and load test based on the cost of the frame data, adjust the division of non-I-frame sub-strip group, so that when I said non-parallel encoding multiple-frame coding processing time consistent. is the strip width equal to the width of the frame one or more contiguous whole macroblock row.

The strip is set in the sub-band coding process according to the length of the code stream of the cut-off.

Encoding process for each sub-encoder strip group

wherein each macroblock comprises: motion estimation, motion compensation; selected intra prediction, intra prediction, intra / inter selection, seeking residuals; rate control; transform, quantization; reordering, entropy coding; inverse quantization, inverse transform; reconstruction.

Encoding process for each sub-encoder strip group

wherein each macroblock comprises: motion estimation, motion compensation, using a plurality of forward or backward reference frame; intra prediction selection, intra prediction, using the current macro block is the same with the intra prediction macro block and the left upper macroblock data loop filter has

为便于对本发明中涉及的现有编码实现方案中技术概念的理解，首先对现有编码方案 中的部分概念进行说明。

H. 264按时间、空间由高到低分成序列、图像组、图像（即帧）、场、条带组、条带、 宏块、子宏块等不同层次。

(1)场、帧、图像 视频的一场或一帧可以用来产生一个编码图像。通常，视频帧可以分成两种类型：连 续或隔行视频。隔行视频情况下，两场（分别称为顶场和底场）组成一个帧； II 当前帧：正在编码的帧； 重建帧：编码器经过本地解码重建输出的帧； 参考图像（帧）：为了提高预测精度，H264编码可从一组前向或后向已编码重建图像 中选出一个或多个与当前最匹配的图像作为帧间编码的参考图像。H. 264中最多可从16个参考图像中进行选择，选出最佳的匹配图像。

(2)宏块(Macroblock, MB)和子宏块 一个编码图像通常划分成若干个宏块，一个宏块可以由一个16X16亮度象素和附加的 一个8X8Cb和一个8X8Cr彩色象素块组成。宏块是视频编码的基本尺寸单位。一个宏块可 进一步分割为块：即可分成16X8，8X16或8X8亮度象素块（以及附带的彩色象素）；对 8X8的子宏块，则可再分割成各种子宏块：8X4, 4X8或4X4亮度象素块（以及附带的彩色象素）。一个隔行扫描帧的顶场和底场的两个对应宏块组成宏块对。

根据所采用的预测模式是帧内还是帧间，将宏块分为帧内预测块（I宏块）和帧间预测 块（P宏块），I宏块利用从当前条带中已解码的象素作为参考进行帧内预测。P宏块利用前 面已编码的图像作为参考图像进行帧内预测。

(3)条带(Slice) 每个图像中，若干宏块被排列成条带的形式。帧内编码条带（I条带）只包含I宏块， 帧间编码条带（P条带）可包含P和I宏块。条带的编码是相互独立的，某条带的帧内预测不 能以其他条带中的宏块为参考图像。

全部由I条带组成的帧成为I帧；不全由I条带构成的帧称为非I帧。

(4)条带组 在H. 264标准中的图象宏块可以灵活的宏块组织顺序(FMO)划分为多

yet to rebuild; intra / inter selection, seeking residuals; rate control; integer transform, quantization; reordering, entropy coding, entropy coding is context-based adaptive variable length coding or adaptive binary arithmetic coding context; inverse quantization, inverse transform; reconstruction; loop filter.

Encoding process for each sub-encoder strip group

said each macro block further comprising: providing a loop filter mode selection border strip all the current frame is not filtering their respective encoders independently each strip loop filter processing, and each encoder, do not exchange information between each band.

The loop filter comprising: a loop filter strip in the strip first macro block start reconstruction after the completion of, or after the loop filter strip with a rebuilt at the beginning of the article completed ?.

The present invention also provides a parallel video coding apparatus includes a main processor and a plurality of encoders, the host processor for the current frame to be encoded is divided into sub-strip group, and passed to a plurality of coding, respectively, a plurality of parallel encoder encodes each sub-strip group, each output code stream to the host processor to generate a sequence stream by the host processor; said strip group includes a host processor determining unit, sub-band group coding determination unit and the top unit, wherein: strip group determining unit for the sequence of video frames of the current frame is divided into macro blocks fixed, predetermined rule all macro blocks are allocated to one or more strips groups; sub-bands group determination unit for all of the current frame strip group, according to the order of the raster scan strip group

个条带组（slice group）；条带组是一个编码图像中若干MB的一个子集，它可包含一个或若干个条带。通过 条带组(slice group)的使用，FMO改变了图像划分为条带和宏块的方式，可以进一步提高 条带的差错恢复能力。

宏块到条带组的映射定义了宏块属于哪一个条带组。利用FMO技术，H. 264定义了7种宏 块扫描模式，所述的七种扫描模式包括：交错、散乱、前景和背景、打开的盒子（Box — out)、光栅扫描、手绢及显式（用编号指明每一个宏块所属的条带组）。

不同的标准及档次（Profile)对FMO的支持不同。H. 264基本档次（Baseline Profile)和扩展档次支持FMO 7种扫描模式。H. 264主档次（Main Profile)、VC-K AVS1.0-P2不支持FMO模式，仅有"光栅扫描"一种扫描模式；只有一个条带组，它的尺寸 II 等于帧。

(5)图像组(GOP) 多个连续的图像（帧），起始帧就是I帧。

(6)序列 视频序列（sequence)，编码比特流的最高层语法结构，包括一个或多个连续的编码 图像。

在编码处理过程中，档次是规定的语法、语义及算法的子集。符合某个档次规定的解 码器必须完全支持该档次定义的子集。H.264/AVC标准分为3个档次（Profile)和4种高保 真扩展（High Extended)。

1、基本档次（Baseline Profile)： 利用I条带和P条带支持帧内和帧间编码，支持基于上下文的自适应的变长编码进行的熵编码（CAVLC)。主要应用于可视电话、会议电视、无线通信等实时视频通信。

2、主档次(Main Profile)： 支持隔行视频，采用B条带的帧间编码和采用加权预测的帧内编码；支持利用基于上下 文的自适应的算术编码（CABAC)。主要用于数字广播电视与数字视频存储。

3、扩展档次(Extended Profile)： 支持码流之间有效的切换（SP和SI条带）、改进误码性能（数据分割），但不支持

were divided into one or more sub-strip group, and each macro block and must only be assigned to a child strip group, to obtain the current frame includes a plurality of strip group, each of the encoders corresponds to one or more sub-strip group, each sub-group corresponding to the stripe encoder; sub-band set of data transfer unit, according to sub-article band group and encoder transmits the correspondence between the current frame and all sub-band coding group configuration parameters to multiple encoder; encoding unit top: the code stream and parameters for each encoder output convergence, complete set of strips, frames and sequences the coding sequence of the entire output stream.

Encoder

comprises sub-band group receiving unit, reference data input and output units and coding unit, wherein: the sub-strip group receiving unit for receiving a sub-band coding and group configuration parameters; the reference data input and output each sub-strip unit, for switching between the reference data encoder, when the current frame is not an I frame, controlling a switching between the encoder sub reconstructed reference frame strip group data, updating of the encoder buffer with a set of motion estimation search area reference data; encoding unit for the allocated sub-band group encoding process in the encoding process in each sub-band group sequential raster scanning is divided into one or more strips, complete sub strip groups encoding all the macro blocks allocated, generated reconstruction sub-strip group, the output code stream and parameters.

Main processor

said process comprising: the entire frame is divided into a strip group, and further divided into a plurality of sub-strip group, and all sub-strip group

隔 行视频和CABAC,主要应用于流媒体。

在具体的编码处理过程中，为方便适配各种网络标准协议，H.264/AVC的功能分为两 层，即视频编码层（VCL)和网络提取层（NAL，Network Abstraction Layer)。VCL数据 即编码处理后的输出，它表示被压缩编码后的视频数据序列。在VCL数据传输或存储之前， 这些编码的VCL数据，先被映射或封装进NAL单元中。

H. 264/AVC (MPEG4-PartlO)视频编码标准的视频编码层采用变换和预测混合编码方 法，相应的框图仍如图1所示。如果采用帧内预测编码，其预测值PRED (图中用P表示）是 由当前条带中已编码的参考图像经运动补偿（MC)后得出的，其中，参考图像用F' n — 1 表示。为了提高预测精度，从而提高压缩比，实际的参考图像可在过去或未来（指显示次 序上）已编码解码重建和滤波的帧中进行选择。预测值PRED和当前块相减后，产生一个残 差块Dn,经块变换、量化后产生一组量化后的变换系数X，再经熵编码，与解码所需的一些边信息（如预测模式量化参数、运动矢量等）一起组成一个压缩后的码流，经NAL (网络自 适应层）供传输和存储用。

II 正如上述，为了提供进一步预测用的参考图像，编码器必须有重建图像的功能。因此 必须使残差图像经反量化、反变换后得到的D' n与预测值P相加，得到uF' n (未经滤波的 帧）。为了去除编码解码环路中产生的噪声，提高参考帧的图像质量，从而提高压缩图像 性能，设置了一个环路滤波器，滤波后的输出即为重建图像，可用作参考图像。

运动估计占编码器运算量的50%以上，是编码器实现的瓶颈。所谓运动估计是对于当 前帧中的每一块（亮度宏块及其子宏块）到前一帧或后一帧某给定搜索范围内根据一定的 匹配准则找出与当前块最相似的块，即匹配块，由匹配块与当前块的相对位移计算出运动 矢量（Motion Vector)。这里常用的准则为绝对误差和（SAD)最小。运动估计的越准确， 补偿的残差就越小，编码效率就越高，编码出来的图像质量也就越好。为

width equal to the width of one frame or more consecutive the entire macro block line.

A plurality of encoders

said each encoder also includes a data storage unit for the sub-bands of the current set of data cache and reconstruction. is the strip width equal to the width of the frame one or more contiguous whole macroblock row, and the same macro block is divided into only a band of the same.

Apparatus main processor

further includes a control unit for initialization, configuration and control of the main processor and a plurality of encoders complete the entire video sequence encoding.

A plurality of encoders

said each encoder during the encoding process further comprising: boundary current mode loop filter frame strips all set to not filtering their respective encoders complete independence of each strip belt loop filter processing, each encoder, do not exchange information between each strip and loop filter strip in the strip first macro block start after the completion of the reconstruction, or loop filter strip in the strip all ? start after the completion of the reconstruction.

A plurality of encoders

Only the complete macroblock level entropy coding macroblock output stream and parameters, the main processor to complete the top stripe encoding module level entropy coding. technical solution provided by the present invention can be seen, the present invention provides a parallel video coding scheme, so that you can select the parallel video encoding in the video encoding process, so that the present invention also can satisfy High-performance real-time HD video

了块运动估计，需要读入该块对应搜索窗的参考帧数据（也称参考数据）。对于一个16X16宏块，如果运 动估计搜索位置范围水平[-64，+63]/竖直[-32，+31],那么，需要读入参考数据为参考帧中对应于本宏块及其周围的位置图像区域，大小为（64 + 16 + 64) X (32+16 + 32) =144 X 80。在多参考帧情况下，可能需要读入多个参考帧搜索窗数据。由于每帧宏块数很大， 上述存储访问量巨大，成为视频编码实现的瓶颈。为此，需要通过相邻宏块间重用参考帧 搜索窗数据，这样可使得读入的参考帧数据量大大下降，当很多相邻宏块在一个单元中进 行运动估计时，所需读入的参考数据仅比这些宏块区域略大。

H.264/MPEG-4 AVC标准定义了一个对16X16宏块和4X4块边界的解块过滤过程。在宏块 这种情况下，过滤的目的是消除由于相邻宏块有不同的帧内、帧间预测或者不同的量化参 数导致的人工方块效应。在块边界这种情况下，过滤的目的是消除可能由于变换/量化和来 自于相邻块运动矢量的差别引起的人工痕迹。环路滤波通过一个内容自适应的非线性算法 修改在宏块/块边界的同一边的两个像素。

其它新一代视频编码标准，即VC— 1、AVS1.0-P2,与H. 264/AVC具有相同的编码框 架，仅部分模块细节不同；如环路滤波：H. 264/AVC可以选择对条带的边界滤波或者不滤 波，VC — 1、AVS1.0-P2的条带边界总是不滤波，具体的滤波算法也有差别。同样，上一代视频编码标准，包括MPEG4、H .263、MPEG4 — Part2,与H. 264/AVC、VC—1、AVS1.0-P2具 有类似的编码框架，仅部分模块不同，如MPEG4、H .263、MPEG4—Part2编码器没有环路滤 波，运动估计参考帧也只有一个。

为便于对本发明的理解，下面将对本发明的具体实现过程进行说明。

本发明的具体实现过程如图2所示，具体包括： 步骤21:将当前帧划分至少一个条带组； II 具体为将视频帧序列中的当前帧分割成宏块，按预定规则将所有宏块分配给一个或多 个条带组； 在该步骤中，可以将视频帧划分为条带组具体可以按灵活的宏块组织顺序（FMO)的各

encoding processing power required. Meanwhile, in the present invention, since the principle of load balancing granularity will be moderate encoded data to the plurality of encoders, by a plurality of encoders for encoding the video data in parallel processing, so that as much as possible among the processors reduce the mutual waiting time, and you can try to reduce as much as possible to reduce the interaction between the encoder ? pin, therefore, the present invention provides a parallel video coding scheme also Nuisance video encoding with higher efficiency.

BRIEF DESCRIPTION OF THE DRAWINGS Figure 1 is a schematic diagram of a video coding framework prior art; specific implementation process according to the present invention. FIG. 2 is a schematic view of the method □, the specific implementation structure diagram of the apparatus of the present invention, FIG. specific embodiment of the present invention mainly in the video encoding process, the strip group is further divided into a plurality of sub-strip group, each sub-band group includes one or more strips, with the group based on the use of sub-article a plurality of parallel encoding encoder.

In order to facilitate the technical concept of the present invention relates to conventional coding implementations of understanding, first part of the existing coding scheme will be described concept.

H. 264 by time, space descending into a sequence of image group, the image (ie, frame), field strip group, striped, macro block, sub macroblock at different levels.

(1) field, a frame, an image of a video or an image can be used to generate an encoded. Typically, video frames can be divided into two types: continuous or interlaced video. Under circumstances

种扫描方式划分，所述的各种扫描模式包括：交错扫描模式、散乱扫描模式、前景和背景 扫描模式、打开的盒子（Box — out)扫描模式、光栅扫描扫描模式、手绢扫描模式和显式 扫描模式，其中，在显式扫描模式中，需要采用编号指明每一个宏块所属的条带组；若编码器不支持FMO时，则可以将一帧所有宏块划分到一个条带组。

而且，在所述的光栅扫描扫描模式中，所述的条带组可以选择为在图像中大小和位置 固定、宽度等于帧宽度的一个或多个连续的整宏块行；在所述的显式扫描模式中，所述的 条带组的形状可以均选择为矩形。

步骤22:基于所述的条带组将当前帧划分为多个子条带组； 在H.264编码方案中，分成序列、图像组（GOP)、图像（帧）、条带组、条带、宏 块、子宏块等不同层次；本发明中，为了用多个编码器并行编码，需要选择一种恰当的粒度将整个一个序列的编码任务划分为多个子任务（模块），即将所述的条带组进一步划分 为一个或多个子条带组，或当前帧只有一个条带组，则需要将该条带组划分为多个子条带 组，以便于后续可以基于当前帧的多个子条带组进行负荷均衡的并行编码处理。

在该步骤中，对当前帧的所有条带组分别进一步划分为多个子条带组，具体是按归光 栅扫描的顺序划分为一个或多个子条带组； 其中，所有子条带组可以为宽度等于帧宽度的一个或多个连续的整宏块行。

需要说明的是，在划分子条带组过程中，每个宏块必须并且只能分配到一个子条带 组，将每个编码器对应一个或多个子条带组，每一个子条带组仅能对应一个编码器，但每 个编码器可以对应一个或多个子条带组，即可以用于对一个或多个子条带组进行编码； 另外，在该步骤中，一个编码器还可以对应多个子条带组，例如，某种条带组和子条 带组划分方案导致部分子条带组尺寸偏小时，可将若干个小的子条带组分给一个编码器。

多个编码器之间处理负荷均衡的原则具体为：按与多个编码器处理能力匹配的

interlaced video, two (referred to the top and bottom fields) to form a frame; II current frame: the frame being encoded; reconstructed frame: After the encoder outputs locally decoded reconstructed frame; reference image (frame): To improve the prediction accuracy, H264 encoding one or more selected image that best matches the current inter-coded as a reference picture or backward coded reconstructed image from a set of front. H. 264 can select from up to 16 reference images, select the best matching images.

(2) macroblock (Macroblock, MB) and a sub-macroblock coded images are often divided into several macro block, a macro block may consist of a 16X16 pixel brightness and additional 8X8Cr a 8X8Cb and a color pixel blocks. Macroblock is a video encoding basic size units. A macroblock can be further divided into blocks: 16X8,8X16 or 8X8 brightness can be divided into blocks of pixels (and the included color pixels); on the 8X8 sub-macroblock can be further divided into various sub-macroblock: 8X4, 4X8 or 4X4 luminance pixel block (and the included color pixels). One of the two interlaced frames corresponding to the top and bottom fields of macro blocks macroblock right.

According to the prediction mode is used intra or inter, intra prediction macro block is divided into blocks (I macro block) and inter prediction block (P macro block), I take advantage of the current macro-block strip decoded pixel intra prediction as a reference. P macroblock is coded using the front face image as the reference image intra prediction.

(3) strips (Slice) each image, several macro blocks are arranged in the form of strips. Intra-coded strips (I strip) contains only I macroblock, inter-coded strips (P strip) may contain P and I macro block. Stripe encoding are

原则来 划分子条带组，在划分过程中结合参考数据传送代价，令多个编码器同时完成所分配子条 带组的编码处理。

在该步骤中，在对一个图像组（GOP)进行编码的过程中，具体的子条带组划分处理方 案包括如下处理方式： 对于I帧，按照宏块数预测编码处理负荷，按与多编码器处理能力匹配的原则来划分子 II 条带组，令多个编码器同时完成所分配子条带组的编码处理； 对于第一个非I帧，当条带组划分不变时，采用I帧的子条带组划分方式进行子条带组 的划分； 对于第2个非I帧，根据实际统计得到的第1个非I帧的各子条带组编码处理量预测所述 第2个非I帧各子条带组的编码处理负荷，并根据参考数据载入代价，调整所述第2个非I帧 子条带组的划分，令并行编码所述第2个非I帧时多个编码器处理时间一致； 对于第2个非I帧以后的任一非I帧，根据该帧前一帧或前几帧各子条带组编码处理量预 测所述非I帧各子条带组的编码处理负荷，并根据考帧数据载入代价，调整所述非I帧子条带组的划分，令并行编码所述非I帧时多个编码器处理时间一致。

步骤23:将划分为子条带组后的待编码数据分发给各个编码器； 具体的分发方式按步骤22中描述的子条带组和编码器对应关系进行，具体的分发内容 不仅包括当前帧所有子条带组，还包括相关的编码配置参数信息，这些信息包括条带的描 述信息，如位置、宏块数等；所述的编码配置参数信息具体包括但不限于：编码标准信息 (标准、档次），FMO及扫描模式，参考帧数，运动估计搜索范围，码率控制要求，以及环 路滤波模式等。

步骤24:各编码器之间执行交换参考数据的操作； 具体的交换参考数据的处理过程可以为：当当前帧不为I帧时，编码器之间交换已重建 的参考帧子条带组数据，更新各编码器中缓存的每个子条带组运动估计搜索区参考数据； 进一步讲，所述的交换参考数据的处理为：当当前帧不为I帧时，编码器进行帧间预测，其中最核心的处理是运动估计；在所述运动估计前，编码器需要获得搜索区参考数 据；参考数据来源于前面已重建的参考帧，这个参考帧由各编码器输出的各重建子条带组 合成；参考数据的

independent, intra prediction of a strip can not be in other bands in the macroblock reference image.

All frame strips composed by I become an I frame; frame failure by the I band is referred to as non-I-frame.

(4) strip group in H. 264 standard image macroblock flexibility macroblock organization order (FMO) is divided into a plurality of strip group (slice group); stripe set is a coded image Several MB of a subset, which may contain one or several slices. By strip group (slice group) use, FMO changed image into strips and the way macroblock can be further enhanced error recovery capability strip.

Mapping macroblock to slice group defines the macro-block strip group to which they belong. FMO use technology, H 264 defines seven kinds macroblock scan mode, the seven scanning modes include: staggered, scattered, foreground and background, playing ? box (Box - out), raster scanning, handkerchiefs and explicit (with the number specified in the strip group each macro block belongs). different standards and grades (Profile) support for FMO different. H. 264 basic grade (Baseline Profile) and extended support FMO 7 grades scan-mode. H. 264 primary grades (Main Profile), VC-K AVS1.0-P2 does not support FMO mode, only "raster scan" A scanning mode; only one stripe group, its size is equal to the frame II.

A plurality of successive images (frames)

(5) Group of Pictures (GOP), the start frame is an I frame.

(6) a sequence of video sequences (sequence), the highest level of the grammatical structure of the encoded bit stream, comprising one or more consecutive coded picture.

读入是视频编码存储访问量中最大的部分，对编码性能影响很大；为此 需要最大限度地减少参考数据的读入量，这需要采用各种参考数据重用策略。

即在该步骤中，若当前帧不为I帧，则根据各编码器当前帧子条带组和重建帧子条带组 的图像区、存储区交叠关系，以及当前帧子条带组中运动估计最大搜索区、参考帧，确定 编码器之间交换的最少的参考数据，并利用所述交换的参考数据更新各编码器中缓存的每 个子条带组运动估计搜索区参考数据。

也就是说，本发明中，一个编码器编码过程中产生的重建子条带组存储在本编码器本 地存储器中，当下一帧同一个子条带组位置大小不变时，这个重建子条带组的数据能够全 部作为下帧运动估计的参考数据，但超出本编码器所分配子条带组区域以外的搜索区部分 II 参考数据是由其它编码器产生的，必须从其它编码器获得。

例如，在运动估计过程中采用大小固定的搜索窗，一个参考帧，根据搜索窗的尺寸确 定编码器之间交换的最小参考数据，并在编码器之间交换获得相应的参考数据，以便于用 来更新各编码器中缓存的每个子条带组的运动估计搜索区参考数据。

步骤25:由各个编码器并行将待编码的当前帧所有子条带组包含的宏块进行编码处 理； 对当前帧所有子条带组用多个编码器并行编码的处理具体可以为：每个编码器完成所 分配子条带组的编码，在编码过程中，将每个子条带组按光栅扫描的顺序划分为一个或多 个条带，每个宏块必须并且只能分配到一个条带；每个编码器完成所分配子条带组中所有 宏块的编码，产生重建子条带组，输出编码码流和参数。

在该步骤中，每个编码器对子条带组中所有宏块的编码过程具体可以包括：运动估 计，运动补偿；帧内预测选择，帧内预测，帧内/帧间选择，求残差；码率控制；变换、量 化；重排序，熵编码；反量化，反变换；以及重建。

若根据H. 264、VC-K AVS1.0 — P2标准，则每个编码器对子条带组中所有宏

In the encoding process, the grade is specified subset of syntax, semantics and algorithms. In line with the provisions of a grade decoder must fully support the grades defined subset. H.264 / AVC standard is divided into three grades (Profile) and four kinds of high-fidelity extended (High Extended).

1, the basic grade (Baseline Profile): I use strips and P bands support intra and inter coding, entropy coding support context-based adaptive variable-length coding performed (CAVLC). Mainly used in real-time video communications videophone, video conferencing, wireless communications.

2, the main grade (Main Profile): support for interlaced video, the use of inter-coded B bands and weighted intra prediction coding; support the use of context-based adaptive arithmetic coding (CABAC). Primarily for digital broadcast television and digital video storage.

3, extended profile (Extended Profile): supporting effective switching (SP and SI strip) between the stream and improved error performance (data segmentation), but does not support interlaced video and CABAC, mainly used in streaming.

In the specific encoding process, in order to facilitate the adaptation of various network standard protocols, H.264 / AVC function is divided into two layers, namely video coding layer (VCL) and network abstraction layer (NAL, Network Abstraction Layer). VCL data is output after the coding process, which represents a sequence of video data are compressed encoded. VCL data before transmission or storage, these codes VCL data, the first to be mapped or encapsulated NAL unit.

H. 264 / AVC (MPEG4-PartlO) video coding standard video coding layer using transform and predictive coding

块的的编 码过程具体可以包括以下步骤： (1)使用多个后向或前向参考帧进行运动估计，及运动补偿； (2)帧内预测选择，即当前宏块帧内预测时，使用同一条带已重建尚未环路滤波的左 边宏块和上边宏块数据；帧内/帧间选择，求残差； (3)码率控制处理； (4)整数变换、量化处理； (5)重排序，熵编码，熵编码是上下文基于自适应的变长编码或基于上下文的自适应 二进制算术编码； (6)反量化，及反变换处理； (7)重建处理； (8)环路滤波处理。

基于H. 264标准，环路滤波是编码过程产生重建帧必须的环节。H. 264每条带的边界是 否进行环路滤波可以通过“条带边界环路滤波模式”设置选择，本发明优选将当前帧所有 条带“边界环路滤波模式”设置为“不滤波”。当前帧所有条带“边界环路滤波模式”设 置为“不滤波”时，各编码器各自独立完成每个条带环路滤波处理，各编码器、各条带之 间不交换数据。

ll 由于各条带的环路滤波是独立的，为节省环路滤波的处理时间，本发明中针对一个条 带包含的宏块的环路滤波在该条带第一个宏块重建完成后启动，宏块的重建和环路滤波构 成宏块级流水线，为了降低环路滤波处理对主存储器的访问频度，可以在片上缓存上存储 上边一整行宏块中的最下边4行象素值及块信息和左边宏块最右4列的象素值及块信息，环 路滤波处理时仅需要访问这块片上缓存器。当然，一个条带的环路滤波也可以在该条带重 建全部完成后开始，同样，各编码器对各条带的环路滤波各自独立进行，各编码器、各条 带之间不需要交换数据。

步骤26:对编码器编码后的输出码流及参数进行汇聚处理和顶层编码，产生输出序列 码流； 具体为：主处理器汇集各编码器输出的码流及相关参数，生成条带组、帧码流，合并 处理生成序列码流后输出；例如，对于H. 264,需要完成包括NAL在内的所有编码处理。

经过上述步骤21至步骤26的处理，便可以实现针对视频数据的并行编码处理，而且， 在针对视频数据进行编码处理过程中需要重复执行所述的步骤21至步骤

method mix, the corresponding block diagram is still shown in Figure 1. If intra prediction encoding, the prediction value PRED (Fig represented by P) is a current stripe coded reference picture motion-compensated (MC) after the draw, in which, with the reference image F 'n - 1 FIG. In order to improve the prediction accuracy, thereby increasing the compression ratio, the actual reference image can be in the past or the future (of the order of display) has been codec reconstruction and filtered frames to choose. PRED and the predicted value after subtracting the current block to generate a residual block Dn, by block transform, to produce a set of quantized transform coefficients quantized after X, and then by entropy coding, a number of side information required for decoding (such as prediction mode quantization parameters, motion vectors, etc.) to form a compressed stream together by NAL (Network Adaptation Layer) for transmission and storage.

ll as, in order to provide further reference picture for prediction, the encoder must have the function of said reconstructed image. It is necessary to make the residual image by inverse quantization, inverse transform obtained after D 'n with the predicted value P added to obtain uF' n (unfiltered frames). In order to remove noise generated in the loop codec, improve the image quality of the reference frame, thereby improving the compression performance of an image, it sets up a loop filter, the output filtered reconstructed picture is used as a reference image.

More than 50 percent of motion estimation computation encoder is encoder implementation bottlenecks. The so-called motion estimation for each block in the current frame (luminance macroblock and sub-macroblock) to the previous frame or subsequent frame within a given search

26，以便于重复进 行下一帧编码处理，直到编码过程结束。

本发明在具体实现过程中，划分子带分组的处理过程为本发明实现的关键，下面将结 合具体的应用实例对划分子带分组的处理过程进行说明。

本发明中，为了通过并行处理获得更大编码处理能力，满足实时高清视频编码的要 求，需要采用多个编码器并行进行编码处理，所述的多个编码器的架构和编码处理能力可 以不相同，但其编码器总的处理能力应大于所要求的编码处理，并有一定余量应付并行编 码处理器的附加开销。

以高清视频720p (1280X720, 30 fps) H.264实时编码为例。编码器架构设计考虑支 持1或2个参考帧，运动估计搜索范围水平[-64，+63]/竖直[-32，+31]。单个编码器采用 VLIW DSP或FPGA实现，根据预先的评估，大约需要5个所述的编码器累计起来能够达到720p 总的编码处理能力并有一定余量，所述的编码器编号为编码器1～5。

本发明中，为了获得最佳的并行处理性能，主要考虑以下两方面：(1)各编码器之间负荷均衡，相互等待尽量少；(2)各编码器之间通信等开销尽量小。

本发明的方法正是通过灵活的子条带组划分，使并行视频编码系统中多个处理器（即 编码器）满足上述两方面处理，以达到整个并行视频编码系统的性能最佳。

为达到上述（1)所述的需求，需要使预计的子条带组的编码处理负荷与所分配的编码 II 器的处理能力匹配。子条带组的编码处理负荷与该子条带组大小（即包含的宏块数数）、 图像内容特征、编码配置参数（帧内预测、帧间预测、参考帧数量、最小块、量化等级、 熵编码方式等）密切相关，通常，宏块数越多，编码处理负荷越大，因此可以通过调整子 条带组的宏块数和对应编码器处理能力相匹配。

对于上述（2)所述的需求，多编码器并行处理的通信开销主要是参考数据的交换，若 运动估计参数（参考帧数、搜索范围等）固定，当子条带组划分不变

range to find the most similar to the current block is a block according to certain matching criteria, That matching block by block match the current block is calculated relative displacement motion vector (Motion Vector). Here common guidelines and absolute error (SAD) is minimized. The more accurate motion estimation, compensation residuals smaller, higher coding efficiency, coding out of the picture quality is better. To block motion estimation, we need to read the block search window corresponding to the reference frame data (also called reference data). For a 16X16 macro block, if the motion estimation search range of horizontal position [-64, 63] / vertical [-32, 31], then, need to read the reference data as a reference frame corresponding to the present macro-block and its surroundings The position of the image area, the size of (64 16 64) X (32 16 32) = 144 X 80. In the case of multiple reference frames, you may need to read multiple reference frame search window data. Since the macro blocks per frame and a huge amount of the memory access, video encoding realization become a bottleneck. To this end, through neighboring macroblocks reuse reference frame search window data, so that reference can be made to read the frame data is greatly decreased, when many neighboring macroblocks for motion estimation in a single unit, the desired read Reference data is only slightly larger than the macro block area.

H.264 / MPEG-4 AVC standard defines one pair of 16X16 and 4X4 block boundary macroblock deblocking filter process. In this case the macro block, the purpose of filtering is to eliminate the adjacent macro block has a different frame, inter prediction or artificial block effect different quantization parameters lead. In this case the block boundary filter is designed to eliminate possible because the transform / quantization and artifacts from the difference in the adjacent block motion vectors caused.

时，通信开销不变， 如果当前帧子条带组划分发生改变，则参考数据交换的通信开销明显增大。故子条带组划 分的调整需要计算参考数据通信开销增大的代价影响。

具体以高清视频720p (1280X720)为例，对本发明提供的条带组、子条带组的划分方 式进行说明。即如表1所示，表3中是720p—帧的宏块划分，共有3600个16X16宏块，分为 45个宏块行（表中最右边一列是宏块行的编号，MBRI～45)，每个宏块行有80个宏块，每 个小格中的数字是宏块编号，按光栅扫描（从左向右，从上到下）递增； 表1 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3、7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII ... MBR12~16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20 ... ... MBR2T25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 ... MBR3(f'34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39~43 3440 3519 MBR44 3520 3599 MBR45 针对表1所示的一帧，以按前景和后景划分条带组为例。假设，仅有一个前景条带组， 共216个宏块（8宏块列、27个宏块行）。剩余宏块其余属于背景条带组。在实际应用中， 如会议电视中，常把视场中的人（或人脸）视为最关心的部分，将这部分划分为前景条带 组单独进行编码。对两个条带组又进一步划分为总共6个子条带组，其中背景条带组分为5 个子条带组，前景条带组分为一个子条带组。

如表2所示，相应的划分后的子条带组具体可以为如下划分结果： 子条带组1:为最下方5个整宏块行，共400个宏块（80宏块列、5个宏块行），宏块编 号0～719; 子条带组2:宏块编号720～1439,前景部分（带双下划线的黑体字区）除外； 子条带组3:宏块编号1400～2159，

Loop filter through a content adaptive nonlinear algorithm modify two pixels in the same side of the macroblock / block boundary of. other next-generation video coding standard, VC- 1, AVS1.0-P2, and H. 264 / AVC encoding the same frame, only the details of the different parts of the module; such as a loop filter: H 264 / AVC can. Select boundary filtering bands or no filtering, VC - 1, AVS1.0-P2 border strip is not always filter specific filtering algorithm also makes a difference. Similarly, the previous generation of video coding standards, including MPEG4, H.263, MPEG4 - Part2, and H. 264 / AVC, VC-1, AVS1.0-P2 has a similar coding framework, different only part of the module, such as MPEG4, H.263, MPEG4-Part2 encoder no loop filter, motion estimation is only a reference frame.

Specific implementation process

To facilitate understanding of the invention, the present invention will be described.

Specific implementation process

of the present invention shown in Figure 2, specifically includes: Step 21: The current frame is divided into at least one strip group; Il in particular video frame in the sequence of the current frame is divided into macro blocks, according to a predetermined rule All macro blocks assigned to one or more strips groups; in this step can be specifically divided by a flexible organization macroblock order (FMO) various scanning video frame is divided into strip group, said each Species scan modes include: interlace scan mode, scattered scan mode, foreground and background scanning mode, playing Jian box (Box - out) scanning mode, raster scan scan mode, scan mode and explicit handkerchief scan mode, in which explicit scanning mode, you need to use numbered strips specified group each macro block belongs; if the

前景部分（带双下划线的黑体字区）除外； 子条带组4:宏块编号2160～2879，前景部分（带双下划线的黑体字区）除外； 子条带组5:为最下方5个整宏块行，共400个宏块（80宏块列、5个宏块行），宏块编 号2880～3599; 子条带组6:前景条带组（带双下划线的黑体字区）所有216个宏块。

这种子条带组的划分方式适用于H. 264基本档次和扩展档次支持FMO的情形。

表2 II 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3"7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII = •二 = = — = — 二 MBR12"16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20 = 二 = = 二 = 二 — MBR2T25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 = = = = = = 二 = MBR3 (T34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39~43 3440 3519 MBR44 3520 3599 MBR45 仍以对表1所示的帧为例，且按显式划分条带组的实现方式如表3所示，一帧划分成4个 条带组，又进一步划分为总共5个子条带组。

条带组1为左上角的斜体字区，共1600个宏块（40宏块列、40个宏块行）；该条带组进 一步上下等划分成两个子条带组（子条带组1、2)，每个子条带组共800个宏块（40宏块 列、20个宏块行)。

条带组2为右上角的带下划线的黑体字区，共1600个宏块（40宏块列、40个宏块行）； 该条带组进一步上下等划分成两个子条带组（子条带组3、4)，每个子条带组共800个宏块 (40宏块列、20个宏块行）。

条带组3为下方5个整宏块行，共400个宏块（80宏块列、5个宏块行）；该条带

encoder does not support the FMO, you can divide all macro blocks of one frame to a stripe set.

Also, in the raster scan scan mode, the strip group can choose the size and location of the fixed image, frame width is equal to a width of one or more contiguous whole macroblock row; in the Explicit scan mode above, the shape of the strip group may have chosen to be rectangular.

Step 22: Based on the stripe set the current frame is divided into a plurality of sub-strip group; in the H.264 coding scheme, is divided into a sequence, group of pictures (GOP), images (frames), strip group, stripe, macroblock, sub-macroblock different levels; the present invention, in order to use multiple encoders in parallel coding, you need to select an appropriate size of a sequence encoding the entire task is divided into multiple sub-tasks (modules), will said strip group is further divided into one or more sub bands group, or the current frame is only one stripe set, you will need to strip the group divided into a plurality of sub-strip group, in order to follow-up can be more based on the current frame sub-band group load balancing parallel encoding process.

In this step, all of the current frame strip groups were further divided into a plurality of sub-strip group, in particular a normalization of raster scanning sequence is divided into one or more sub bands group; wherein all sub-bands group may be a width equal to the width of the frame one or more contiguous whole macroblock row.

It should be noted that, in the division of the sub-strip group process, each macro block must and can only be assigned to a sub-strip group, each of the encoders corresponds to one or more sub-strip group, each child strip group only correspond to an encoder, but each encoder may correspond to

组整个 II 划分成一个子条带组（子条带组5)。

这种划分适用于H. 264基本档次和扩展档次支持FMO的情形。

表3 O 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 — ... ••• ... ... ••• ... ... ••• ■■• ... ... MBR3~7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII ••• ... ••• ••• ... ... ••• ... ••• ... ... MBR12、16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20 ... ... — ... ... ••• ... ... ••• ... ... MBR2广25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 ••• ••• ... ••• ••• — ••• ••• ••• ••• ... ••• ... MBR30"34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39、3 3440 3519 MBR44 3520 3599 MBR45 再以按光栅扫描划分条带组为例，相应的划分结果如表4所示，一帧划分成1个条带组，又进一步等分为总共5个子条带组，每个子条带组9个整宏块行，共720个宏块。

子条带组1:宏块编号0∽719; 子条带组2:宏块编号720∽1439; 子条带组3:宏块编号1400∽2159; 子条带组4:宏块编号2160∽ 子条带组5:宏块编号2880∽3599; 这种划分方案适用于H. 264、VC—1、AVS1.0-P2、H. 263、MPEG2、MPEG4 — Part2标准 及其各档次。

表4 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3"7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836

one or more sub bands group, which can be used for one or more sub bands set for encoding; Also, in this step, an encoder You can also have multiple sub-strip group, for example, some kind of strip groups and sub-strip groups led to the partial sub-division scheme strip group size is too small, it can be a number of small sub-band component to an encoder. processing load among multiple encoders in particular the principle of balance: according to the principle and the ability to match multiple encoders process to divide sub-strip group, referring to the cost of the data transfer process incorporated in the division, so that a plurality of encoders while completing the processing sub-band coding group assignment.

In this step, in the process of a group of pictures (GOP) encoding the specific sub-group division strip processing scheme includes the following approach: for I-frame, in accordance with the number of slices predictive coding processing load, press and principles of multi-coder processing capabilities matched to divide sub II strip group, to make a plurality of encoders while completing the assigned sub-band coding group; for the first non-I-frame, when the tape group division unchanged, the use of sub-band group division way I frame is divided into sub-bands group; for the first two non-I-frame, according to a non-strip coding amount of each sub-group I-frame prediction of the actual statistics obtained the first two non-coding processing load on I-frame of the sub-band group and the cost of loading the reference data, adjusts the second division of non-I-frame sub-strip group, so that parallel encoding the first two non-I time frame consistent with the processing time of a plurality of encoders; after the first two for any non-I-frame, the non-I-frame, said non-predictive I-frame from the previous frame or previous frame of each sub-band group encoding processing

837 838 839 840 841 842 843 877 878 879 MBRll ••• ••• … … — — … ••• … ••• — … ••• … — MBR12、.16 1280 1359 MBR17 1360 1439 MBR18 1440 = 1519 MBR19 1520 = = 1599 MBR20 = 二 = = 二 二 = = 二 = = = = = = MBR2T25 2000 — = 2039 MBR26 2080 二 = 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 … … … … … … ••• ••• — … ••• — ••• ••• MBR30∽34 2720 2799 MBR35 2800 2879 MBR36 2880 二 2959 MBR37 2960 二 3039 MBR38 = 二 = = = — 二 二 = = = — = = MBR39~43 3440 — 3519 MBR44 3520 = — 3599 MBR45 这里对本发明中步骤24所述的交换参考数据举例说明如下： 以表4所示的子带分组划分方式为例，以720p、运动估计搜索范围水平[-64，+63]/竖 直[-32，+31]为例，假设编码器3对应子条带组3，需要的参考数据区为对应宏块编号为 1280∽2319的前一个或多个参考帧图像区域。

对于一个参考帧的运动估计情况下，若在编码器3的当前帧与上一帧所编码子条带组划 II 分不变，即同为宏块编号为1440∽2159区域，由于所需参考帧区域中宏块编号为1440∽ 2159区域是本编码器上一帧重建的子条带组，故需要从编码器3外部读入的参考数据仅为以 下两块： 宏块编号为1280∽1439,来自于（编码器2输出的）重建子条带组2;和，宏块编号为2160∽2319，来自于（编码器4输出的）重建子条带组4。

同样，对于一个参考帧的运动估计情况下，若在编码器2的当前帧与上一帧所编码子条 带组划分不变，即同为宏块编号为720∽1439区域，由于所需参考帧区域中宏块编号为 720∽1439区域是本编码器上一帧重建的子条带组，故需要从编码器3外部读入的参考数据 仅为以下两块： 宏块编号为1440∽1559，来自于（编码器3输出的）重建子条带组3;和，宏块编号为560∽719，，来自于（编码器1输出的）重建子条带组1。

在此，当前编码子条带组相邻的两个编码器2和3相互将重建子条带组数据传给对方作 为参考数据，用于进行编码操作。

amount of each of the frame coding sub-band group load and load test based on the cost of the frame data, adjust the division of non-I-frame sub-strip group, so that when I said non-parallel encoding multiple-frame coding processing time consistent.

Step 23: The article is divided into sub-band data to be encoded after the group distributed to the encoder; sub-band group and encoder specific distribution manner described in Step 22 corresponding relationship, not only the specific content distribution including the current frame all sub-strip group, including related configuration parameters encoded information, which includes descriptions of the strip, such as location, number of slices, etc; encoding the specific configuration parameter information include, but are not limited to: coding standards information (standard grade), FMO and scanning mode, the reference frame, motion estimation search range, the rate control requirements, and loop filter mode.

Step 24: The encoder exchange performed between the operation of the reference data; specific reference data exchange process may be as follows: when the current frame is not an I frame, the encoder is exchanged between the reconstructed reference frame Subclause with a set of data, updating of the encoder buffer strip each sub-set of motion estimation search area reference data; further, the reference data exchange process is as follows: when the current frame is not an I frame, the inter-frame encoder forecast, the core of which is the motion estimation process; before the motion estimation, the encoder needs to obtain the search area reference data; data from previously reconstructed reference frame, each reference frame by the reconstruction of the sub-encoder output combined into strips; reference data is read into memory access amount in video encoding largest part,

若使用多参考帧运动估计，则需要从其它编码器载入所需的多个重建子条带组数据部 分。

本发明提供的方法可以适用于各种视频格式，包括高清视频或标清视频，而且，可以 是逐行也可以为隔行扫描；其中，对于隔行扫描视频，一个隔行扫描帧的顶场和底场的两 个对应宏块组成宏块对，需要将一个宏块对分配给同一个条带组、子条带组、条带编码。

基于前面的条带组及子条带组划分的实例，下面对针对子条带组编码过程中的条带的 划分的应用实例进行描述： 应用实例一 将子条带组划分为条带分割的一个例子如表5所示。在表5中，子条带组1在编码过程中 分成7个条带，按光栅扫描顺序编号为条带1～7，条带1、2各为两个宏块行，条带3～7各为 一个宏块行； 表5 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 160 161 162 ... 196 197 198 199 200 201 202 203 ... 237 238 239 MBR3 II 240 241 242 — 276 277 278 279 280 281 282 283 317 318 319 MBR4 320 321 322 356 357 358 359 360 361 362 363 397 398 399 MBR5 400 401 402 416 417 418 419 420 421 422 423 477 478 479 MBR6 480 481 482 516 517 518 519 520 521 522 523 557 558 559 MBR7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 应用实例二 第二个条带分割的例子如表6所示，在表2中，子条带组1在编码过程中分成5个条带, 按光栅扫描顺序编号为条带1～5，各条带包含的宏块如下： 条带1:宏块编码0～123; 条带2:宏块编码124～237; 条带5:宏块编码238～319; 条带3:宏块编码320～521; 条带4:宏块编码522～719; 表6 0 1 2 36 37 38 39 40 41 42 43 44 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 124 157 158 159 MBR2 160 161 162 196 197 198 199 200 201 202 203 204 237 238 239 MBR3 240 241 242 276 277 278 279 280 281 282 283 284 317 318 319 MBR4 320 321 322 356 357 358 359 360 361 362 363 364 397 398 399 MBR5 400 401 402 416 417 418 419

a great impact on encoding performance; for this need to minimize the amount of reference data is read, which requires the use of a variety of reference data reuse strategy.

In this step, that is, if the current frame is not an I frame, the current frame based on each encoder strip groups and sub-sub-frame reconstruction slice group image area, storage area overlapping relationship, and the current sub-frame strip with the largest group of motion estimation search area, a reference frame, a reference to determine the minimal exchange of data between the encoder and the reference data using the exchange of the encoder buffer updating of each sub-band set of motion estimation search area reference data.

That is, the present invention, the local memory of the encoder, the next frame of the same sub-group of the stripe position of the same size, the reconstruction of the sub-sub-stripe group storage rebuilding an encoder encoding process generated data strip group to be fully used as the reference frame motion estimation data, but the search area beyond the data part II reference encoder strip sub-regional group is assigned by other than produced by the encoder, the encoder must be obtained from other.

For example, using the motion estimation process in fixed-size search window, a reference frame, according to the search window size determines the minimum reference data exchanged between the encoder and the exchange between the encoder to obtain the corresponding reference data, to facilitate movement of each sub-band group is used to update each encoder buffer estimation search area reference data.

Step 25: The macro block of the current frame by each parallel encoder to be encoded for all sub-bands group

420 421 422 423 424 477 478 479
MBR6 480 481 482 516 517 518 519
520 521 522 523 524 557 558 559
MBR7 560 561 562 596 597 598 599
600 601 602 603 604 637 638 639
MBR8 640 641 642 ... 676 677 678 679
680 691 692 693 694 717 718 719
MBR9 本发明还包括一种并行视频编码的装置，该装置的具体实现结构如图3所示，包括主处 理器和多个编码器，主处理器用于将待编码的当前帧划分为子条带组，并分别传递给多个 编码器，多个编码器并行编码各子条带组，输出各自编码码流给主处理器，由主处理器生 成序列码流并输出。

(一）主处理器 所述的主处理器进一步包括条带组确定单元、子条带组确定单元、子条带组数据传递 单元及顶层编码单元，各单元具体为： II 条带组确定单元，用于从数据传送单元接收数字视频序列数据，将视频帧序列中的当 前帧固定分割成宏块，按预定规则将所有宏块分配给一个或多个条带组； 子条带组确定单元，用于对当前帧的所有条带组，按照光栅扫描的顺序划分为一个或 多个子条带组，每个宏块必须并且只能分配到一个子条带组； 子条带组数据传递单元，用于根据子条带组和编码器的对应关系配置控制多个编码器 读入所述的子条带组数据，且每个编码器读入一个或多个子条带组，每一个子条带组包含 的待编码数据均传递到同一个编码器；在子条带组数据传递过程中可以使用的总线信号格 式包括但不限于：并行接口、高速串行接口、BT656或CCIR601数字视频信号格式、HD-SDI 信号、AMBA总线规范中的AHB或AXI总线或以太网接口对应的信号格式； 顶层编码单元：用于汇集各编码器输出的码流及相关参数，生成条带组、帧码流，合 并处理生成序列码流后输出； 在主处理器中具体可以将整个帧为一个条带组，划分为多个子条带组，且所有子条带 组为在图像中大小和位置固定、宽度等于帧宽度的一个或多个连续的整宏块行。

另外，在主处理器中还完成预先对接收到当前帧进行图像预处理，包括：二维去噪处 理，和或，缩放处理，和/或，数字视频4: 4： 4到4: 2： 2格式变换处理，和/或，数字视 频4: 2： 2到4: 2： 0的格式变换处理等。

contains coded; all sub-processing of the current frame with a plurality of sets of parallel strips encoded by the encoder may be specifically: each encoder complete coding sub-band group assigned in the encoding process, each sub-strip group sequential raster scanning is divided into one or more strips, each macro block must and can only be assigned to a strip; each encoder to complete the assigned sub-strip groups encoding all the macro blocks, resulting in the reconstruction sub-strip group, the output code stream and parameters.

In this step, each encoder strip groups in all sub-macroblock encoding process may include: motion estimation, motion compensation; selected intra prediction, intra prediction, intra / inter selection, seeking residuals; rate control; transform, quantization; reordering, entropy coding; inverse quantization, inverse transform; and reconstruction.

If according to H. 264, VC-K AVS1.0 - P2 standard encoding process, each sub-encoder strip groups in all macro blocks specifically includes the following steps: (1) using multiple post or to carry out before the reference frame motion estimation and motion compensation; (2) to select intra prediction that the current macroblock intra prediction, using the same band has not yet left the reconstruction loop filter and upper macroblock macroblock data; intra / inter selection, residual demand; (3) rate control process; (4) the integer transform, quantization process; (5) reordering, entropy coding, entropy coding is context-based adaptive variable length coding or context-based adaptive binary arithmetic coding; and (6) inverse quantization, and inverse transform processing; (7) for the reconstruction process; (8) the loop filter processing.

Based on H. 264 standard, loop filter generates reconstructed frame encoding process is necessary to link.

而且，本发明所述主处理器具体中可以是一个计算机系统，包括：CPU、存储器、与数 据传送单元的接口（与数据传送单元的接口和外部通信网络输入接口），以及输出接口， 用于输出已编码码流，和，配置控制信息接口，用于接受外部配置控制信息。

(二）编码器，所述的编码器的结构具体可以包括：(1)子条带组接收单元，用于接收子条带组和编码配置参数；(2)参考数据输入输出单元，用于在各编码器之间交换参考数据，具体为：当当前帧 不为I帧时，每个编码器通过数据传送单元中设置的交换参考数据单元，交换从其它编码器 获取所需的参考数据，即控制编码器之间交换已重建的参考帧子条带组数据，更新各编码 器中缓存的每个子条带组运动估计搜索区参考数据。

(3)编码单元，用于并行对当前帧所有子条带组编码：每个编码器完成所分配子条带 组的编码，在编码过程中将每个子条带组按光栅扫描的顺序划分为一个或多个条带，每个 宏块必须并且只能分配到一个条带；每个编码器完成所分配子条带组中所有宏块的编码， 产生重建子条带组，产生并输出码流和相关参数；而且，所述条带优选为宽度等于帧宽度的一个或多个连续的整宏块行； II 本发明所述多个编码器中每个编码器可以为： 通用处理器，包括：超标量结构的处理器、超长指令字信号处理器（VLIW DSP)；或 现场可编程大规模阵列芯片（FPGA);或 定制的超大规模集成电路芯片（VLSI)；或 指令集可配置处理器。

每个编码器还配置数据随机访问存储器（RAM),即存储单元，用于缓存当前和重建的 子条带组数据等信息。数据RAM容量不大时可以使用芯片内大容量RAM，所需容量较大时可 以配置使用片外的同步静态随机访问存储器（SSRAM)或同步动态随机访问存储器(SDRAM)。

进一步地，为获得更好的并行处理，用于H. 264、VC— 1、AVS1.0 — P2标准编码时，所 述编码器编码中的环路滤波方式为： 设置选择当前帧所有条带的边界环路滤波模式为不滤波，各编码器各自独立完成环路 滤波处理，各编码器、各条带之间不交换数据； 具体为：一个

H. 264 is the boundary of each band whether to perform loop filter can be set by selecting "border strip loop filtering mode", the present invention preferably all the current frame strip "boundary-loop filtering mode" is set to "no filter." All slices of the current frame "boundary-loop filtering mode" is set to "no filter", each coder independently complete each strip loop filter processing, each encoder between each strip as not to exchange data.

II Since the loop filter each band is independent, in order to save processing time loop filter, the loop filter macro block of the present invention for a strip contained in subsection a reconstruction with macroblock after the start, the reconstruction and loop filter macro blocks constituting the macro block level pipeline, in order to reduce the loop filter processing for main memory access frequency, can be stored on top of a whole line of the macro block in the lowermost row 4 as an on-chip cache pixel value and the pixel value information block and the block information and macro block on the left of the rightmost four, only need to access on-chip buffer when this loop filtering process. Of course, a loop filter strips can also be completed after the strip began to rebuild the same, each encoder loop filter each band performed independently of each encoder, no exchange between the strip data.

Step 26: the output stream encoder and parameter-coded top-level coding and aggregation process, producing an output sequence code stream; specifically: the main processor and associated collection stream parameters for each encoder output, generate bar with a group frame stream, the combined process of generating sequence code stream output; for example, for H. 264, including the need to complete all the coding process, including the NAL.

条带的环路滤波在该条带第一个宏块重建完成后启动，宏块的重建和环 路滤波构成宏块级流水线；或者，一个条带的环路滤波在该条带重建全部完成后开始，各 编码器对各条带的环路滤波各自独立进行，各编码器、各条带之间不需要交换数据。

所述的编码器在编码过程中，相应的参考数据交换过程为： 当当前帧不为I帧时，编码器之间交换参考帧子条带组数据，根据各编码器当前帧子条 带组和重建帧子条带组的图像区、存储区交叠关系，并考虑当前帧子条带组中运动估计最 大搜索区、参考帧，确定编码器之间交换的最少的参考数据，更新各编码器中缓存的每个 子条带组运动估计搜索区参考数据；其中，运动估计可以采用大小固定的搜索窗，一个参 考帧，根据搜索窗尺寸确定编码器之间交换的最少的参考数据，更新各编码器中缓存的每 个子条带组运动估计搜索区参考数据。

需要说明的是，在编码器执行编码操作过程中，通常还需要对当前帧进行视频编码前 还需要预处理，包括：二维去噪，缩放和格式变换，格式变换指将输入数字视频格式转为 编码器要求的格式等处理；常见的编码数字视频格式为4: 2： 0，常见的输入的数字视频格 式为BT656或CCIR601规定的4: 2： 2。H.264高保真扩展编码支持4: 2： 2和4: 4： 4，此时 输入数字视频为4: 2： 2和4: 4： 4。高清视频物理接口常为HD—SDI。

而且，多个编码器所需的当前帧数据预处理可采用以下两种形式之一： 1、主处理器接收当前帧，集中预处理并缓存后发再转发给多个编码器；这种方案需要 II 主处理器处理能力较强，数据帧存储容量较大； 2、多个编码器各自预处理处理所分配的子条带组。

本发明所述的装置主处理器中还可以设置有控制单元，用于初始化、配置及控制主处 理器中其它处理单元和多个编码器完成整个视频序列编码。

本发明所述的装置中，为了简化每个编码器的启动、配置电路，降低成本，所述的编 码器的启动、配置程序或数据可以来自主处理器；对于编码器为FPGA的

After the above-described processing of steps 21 to 26, we can achieve the parallel process for the encoded video data, and, in the process for the encoded video data need to repeat the process of steps 21 to 26, in order to repeat the next frame coding process, until the end of the encoding process.

In the present invention, the specific implementation process, the process of dividing the key sub-band packet of the present invention to achieve, the process will be in conjunction with specific application examples of the division of the sub-band packet will be described.

The present invention, in order to obtain greater parallel processing by the encoding processing capacity to meet the requirements of real-time HD video encoding, requires the use of a plurality of encoders for encoding processing in parallel, a plurality of the architecture of the encoder and coding processing capacity It may not be the same, but the total processing capacity of the encoder should be greater than the required coding process, and a certain margin to cope with parallel encoding processor overhead.

HD video 720p (1280X720, 30 fps) H.264 real-time encoding for example. Consider the encoder architecture supports 1 or 2 reference frames, motion estimation search range level [-64, 63] / vertical [-32, 31]. Encoder uses single VLIW DSP or FPGA, according to a pre-evaluation, the encoder takes about 5 up to achieve the cumulative total number of coded 720p handling capacity and a certain margin, the encoder is the encoder number 1 ~ 5.

The present invention, in order to get the best parallel processing performance, consider the following two aspects: (1) between the encoder load balancing, wait for each other as little as possible; (2) the communication

情况，可以由主处 理器通过并行或串行配置接口配置FPGA;对于编码器为通用处理器的情形，可选择并口或 串口启动（Boot),启动过程中应用的BootROM程序来自主处理器。

本发明所述的装置典型地是由一个或多个电路板构成的一个嵌入式计算机系统；或 者，也可以是个人计算机或服务器通过通信网络连接在一起的系统，其中编码器可以是PC 或服务器。

本发明所述的方法及装置适用多种编码标准及其各档次。

本发明提供的装置适用于各种视频格式，包括高清视频或标清视频，可以是逐行也可 以为隔行扫描，对于隔行扫描视频，一个隔行扫描帧的顶场和底场的两个对应宏块组成宏 块对，需要将一个宏块对分配给同一个条带组、子条带组、条带编码。

以上所述，仅为本发明较佳的具体实施方式，但本发明的保护范围并不局限于此，任 何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到的变化或替换，都 应涵盖在本发明的保护范围之内。因此，本发明的保护范围应该以权利要求的保护范围为 准。

between the encoder and other expenses as much as possible small.

The method of the present invention is through flexible sub-band group division, so that a plurality of parallel video coding system processor (ie, encoder) satisfies the above two aspects, so as to achieve the performance of the parallel video coding system best.

In order to achieve the above-mentioned requirement (1) above, the processing load is expected to require coding sub-band coding group II's processing power allocated match. The processing load of the sub-band coding group and the sub-band group size (ie, the number of macro blocks included), image content features, configuration parameters coding (intra prediction, inter prediction, the number of reference frames, most small, quantization level closely related to entropy coding, etc.), in general, macro-blocks, the greater the coding processing load, and therefore can be matched by adjusting the sub-band group and the corresponding number of macro blocks encoder processing capabilities.

The above-mentioned requirement (2) for the communication overhead parallel processing of multiple encoder reference mainly to exchange data, if motion estimation parameters (reference frames, search, etc.) is fixed, when the sub-group dividing strip not change, the communication ? pin unchanged if the current sub-frame strip group division changed, the reference data communications ? pin exchange increased significantly. Therefore, the sub-group adjustment strip the divided reference data necessary to calculate the cost impact of increased communication overhead. particular HD video 720p (1280X720) as an example, the strip of the present invention provides a group of sub-band group division manner described. That is, as

shown in Table 1, Table 3 is divided 720p- frame macroblock, a total of 3600 16X16 macro block is divided into 45 macro block row (right-most column of the table is a macro block row of numbers, MBRI~45) each macro block row 80 macro blocks, each small grid of numbers are the macro block number, raster scanning (from left to right, top to bottom) is incremented; table 10123637383940 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3,7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII... MBR12 ~ 16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20... MBR2T25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29... MBR3 (f'34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39 ~ 43 3440 3519 MBR44 3520 3599 MBR45 for a frame shown in table 1, according to foreground and background striped set an example. Suppose that only one outlook strip group, a total of 216 macro blocks (8 macroblock row, 27 macro block line). the remaining part of the background of the rest of the macro block strip group. in practical applications, such as video conferencing, often to the field of view of the people (or face) considered as part of most concern, this part of the article is divided into foreground with a separate set of coding. two strip group is further divided into a total of six sub-strip group, in which the background is divided into five sub-groups strip strip group, prospects strip into a sub-group of slice group.

As shown in Table 2, the respective sub-bands of the divided specific group can be divided into the following results: sub-band group 1: 5 for the bottom line

of the entire macro-blocks, a total of 400 macro blocks (80 macroblock column, five macro block line), the macro block number 0~719; sub-strip group 2: macro block number 720~1439, foreground portion (zone bold double underlined) except; sub-band group 3: macroblock except for the macro block number 2160~2879, foreground portion (zone bold double underlined); sub-bands group 5:; number 1400~2159, foreground portion (zone bold double underlined) except sub-strip group 4: the bottom five whole macro block row, a total of 400 macro blocks (80 macroblock row, five macro block line), the macro block number 2880~3599; sub-band group 6: Prospects strip group (with double underlined boldface region) All 216 macro blocks. this division sub-band group apply to H. 264 baseline and extended grade support FMO situation.

Table 2 Il 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3 "7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII = □ two = = - = - two MBR12 "16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20 = = = two = two two - MBR2T25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 = = = = = = two = MBR3 (T34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39 ~ 43 3440 3519 MBR44 3520 3599 MBR45 still frames shown in table 1, for example, and by explicit implementation striped groups such as shown in table 3, a strip is divided into four groups, further divided into a total of five sub-band group.

1 is the top left corner strip group italics area, a total of 1600 macro block

(macro block row 40, 40 macroblock row); and so on up and down the strip group further divided into two sub-bands group (sub-article band groups 1, 2), each sub-band group of 800 macro blocks (40 macroblock row, 20 macroblock row).

2 strip group underlined bold upper right corner of area, a total of 1600 macro block (macro block row 40, 40 macroblock row); and so on up and down the strip group further divided into two sub-strip group (sub-band group 3, 4), each sub-band group of 800 macro blocks (40 macroblock row, 20 macroblock row). strip group 3 below five the entire macro block row, a total of 400 macro blocks (80 macroblock row, five macro block line); strip the entire group is divided into a sub-II strip group (sub-article with a group of 5).

This division applies to H. 264 baseline and extended grade support FMO situation.

Table 3 O 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 -... □□□... □□ □... □□□ □□ □... MBR3 ~ 7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRII □□□... □□□ □□□... □□□... □□□... MBR12,16 1280 1359 MBR17 1360 1439 MBR18 1440 1519 MBR19 1520 1599 MBR20... -... □□ □... □□□... MBR2 wide 25 2000 2039 MBR26 2080 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29 □□□ □□□... □□□ □□□ - □□□ □□□ □□□ □□□... □□□... MBR30 "34 2720 2799 MBR35 2800 2879 MBR36 2880 2959 MBR37 2960 3039 MBR38 MBR39,3 3440 3519 MBR44 3520 3599 MBR45 then to raster scan striped group, for example, the corresponding results of divisions shown in table 4, a strip is

divided into a group, and further divided into a total of five sub-strip group, each sub-band group nine whole macroblock row, a total of 720 macro blocks. sub-band group 1: macro block number 0~719; sub-strip group 2: macro block number 720~1439; sub-band group 3: macro block number 1400~2159; sub-bands Group 4: macroblock No sub 2160~ strip group 5: macro block number 2880~3599; this division scheme applies to H. 264, VC-1, AVS1.0-P2, H 263, MPEG2, MPEG4 - Part2 standard and its quality..

Table 4 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRl 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 MBR3 "7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 720 721 722 756 757 758 759 760 761 762 763 797 798 799 MBRIO 800 801 802 836 837 838 839 840 841 842 843 877 878 879 MBRll □□ □ □□□... - -... □□□... □□□ -... □□□... - MBR12,.16 1280 1359 MBR17 1360 1439 MBR18 1440 = 1519 MBR19 1520 = = 1599 MBR20 two = = = = = twenty-two two = = = = = = MBR2T25 2000 - = 2039 MBR26 2080 two = 2159 MBR27 2160 2239 MBR28 2240 2319 MBR29... □□□ □□□ -... □□□ - □□□ □□□ MBR30~34 2720 2799 MBR35 2800 2879 MBR36 2880 two 2959 MBR37 2960 two two 3039 MBR38 = = = = - twenty-two = = = = - = = MBR39 ~ 43 3440 - 3519 MBR44 3520 = - 3599 MBR45 here to exchange the present invention, the step 24 of the reference data described as follows: as shown in table 4 in the sub-division with packet mode, for example, to 720p, the horizontal motion estimation search range [-64, 63] / vertical [-32, 31] as an example, assume that the encoder strip 3 corresponding sub-group 3, you need to reference the data area corresponding to the macro block number in front of one or more reference frame image 1280~2319 area.

For the next reference frame motion estimation, the encoder 3, if the current frame and the previous sub-frame encoded slice group II zoned constant fraction, i.e., the same macroblock number 1440~2159 region, since desired reference frame region macroblock number 1440~ 2159 is a sub-regional group on this strip encoder frame reconstruction, they need only two from the following external encoder 3 reads the reference data: macro block number 1280~1439, from (2 encoder output) reconstruction sub-band group 2; and, the macroblock number 2160~2319, from (the output of the encoder 4) rebuilding the sub-band group 4.

Similarly, a reference to the next frame of motion estimation, the encoder if the current frame and the previous sub-frame encoded slice group 2 of the same division, i.e., the same macroblock number 720~1439 region, since desired reference frame region macroblock number 720~1439 strip group is a sub-region of the encoder on a frame reconstruction, we need only two from the following external encoder 3 reads the reference data: macro block number 1440~1559, from (the output of the encoder 3) reconstruction of the sub-band group 3; and macroblock number 560~719,, from (1 encoder output) reconstruction sub-band group 1.

In this case, the current sub-band coding group of adjacent two encoders 2 and 3 will rebuild mutual sub-strip group data to each other as the reference data for performing the encoding operation.

If you use multiple reference frame motion estimation, it needs to be loaded from other coding required for a plurality of sub-band reconstruction unit set of data points.

The present invention provides a method may be applied to a variety of

video formats, including high-definition video or standard definition video, but can also be a progressive interlaced; wherein for interlaced video, a top field and interlaced frames two corresponding bottom field of macro blocks macro block needs to be allocated to a macro-block with a strip group, sub-group of the strip, the strip encoding.

Based on a sample strip in front of the group and sub-group division strip, the following application examples allotting strip groups encoding process for the strip will be described: Application example of a sub-group is divided into strips an example of the strip division as shown in table 5. In Table 5, Group 1 is divided into sub-bands in the encoding process seven bands, raster scan sequence number of bands 1~7, two strips 1 and 2 are each macro block row, the strip 3 to 7 each row of a macroblock; table 5 0 1 2 36 37 38 39 40 41 42 43 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 157 158 159 MBR2 160 161 162... 196 197 198 199 200 201 202 203... 237 238 239 MBR3 II 240 241 242 - 276 277 278 279 280 281 282 283 317 318 319 MBR4 320 321 322 356 357 358 359 360 361 362 363 397 398 399 MBR5 400 401 402 416 417 418 419 420 421 422 423 477 478 479 MBR6 480 481 482 516 517 518 519 520 521 522 523 557 558 559 MBR7 560 561 562 596 597 598 599 600 601 602 603 637 638 639 MBR8 640 641 642 676 677 678 679 680 691 692 693 717 718 719 MBR9 application example two examples of the second division with a bar as shown in table 6. in table 2, the sub-band group 1 is divided into five bands in the encoding process, the raster scan sequence number of bands ~ 5 macro blocks each strip contains the following: Lane 1: macro block coding 0~123; Lane 2: macro block coding 124~237; strip 5: macro block coding 238~319; strip 3: macro block coding 320~521; strip 4: 522~719 macroblock coding; table 6 0 1 2 36 37

38 39 40 41 42 43 44 77 78 79 MBRI 80 81 82 116 117 118 119 120 121 122 123 124 157 158 159 MBR2 160 161 162 196 197 198 199 200 201 202 203 204 237 238 239 MBR3 240 241 242 276 277 278 279 280 281 282 283 284 317 318 319 MBR4 320 321 322 356 357 358 359 360 361 362 363 364 397 398 399 MBR5 400 401 402 416 417 418 419 420 421 422 423 424 477 478 479 MBR6 480 481 482 516 517 518 519 520 521 522 523 524 557 558 559 MBR7 560 561 562 596 597 598 599 600 601 602 603 604 637 638 639 MBR8 640 641 642... 676 677 678 679 680 691 692 693 694 717 718 719 MBR9 the present invention further comprises a parallel video coding apparatus, the specific implementation structure of the apparatus shown in FIG. 3, it comprises a main processor and a plurality of encoders, the main processing the current frame is divided used to be encoded as a sub-strip group, and passed to a plurality of encoders respectively, a plurality of parallel encoder encodes each sub-strip group, each output code stream to the host processor, the host processor generate a sequence stream and outputs. main processor (a) the main processor further includes a stripe set determining unit, sub-strip group determining unit, sub-band set of data transfer unit and the top coding unit, each unit in particular: Il strip group determining unit for sequence data from the data transfer unit receiving digital video, video frame sequence of the current frame is divided into macro blocks fixed, predetermined rule all macro blocks are allocated to one or more strips groups; sub-strip group determination unit for all groups of the current frame strip, in the order of raster scanning into one or more sub-strip group, each macro block and can only be assigned to a group of sub-bands; sub-set of data stripes a transfer unit for sub-bands according to the corresponding set of data between sub-band group and encoder configuration control multiple encoders read into the,

and each encoder reads one or more sub-strip group, each data to be encoded sub-bands included in the group are delivered to the same encoder; bus signal format in the sub-strip group data transfer process can be used include, but are not limited to: a parallel interface, high-speed serial interfaces, BT656 or digital video CCIR601 signal format, HD-SDI signal, AMBA AHB bus specification or AXI bus or Ethernet interface corresponding to the signal format; encoding unit top: a collection of stream and associated parameters for each encoder output, the stripe set, a frame stream, combined processing sequence generated stream output; specifically the entire frame of a strip group is divided into a plurality of sub-strip group, and all sub-strip group fixed to the main processor for image size and location, a width equal to the width of the frame one or more contiguous whole macroblock row.

In addition, the host processor also completed the previously received current frame image pre-processing, comprising: a two-dimensional de-noising, and or, scaling processing, and / or digital video 4: 4: 4-4: 2: 2 format conversion process, and / or digital video 4: 2: 2 to 4: 2: 0 format conversion processing.

In addition, the main processor of the present invention in particular can be a computer system comprising: CPU, memory, and data transfer unit interfaces (interfaces and external communications network and data transmission unit input interface), and the output interface for outputting an encoded bit stream, and is configured to control the interface for receiving external configuration control information.

(b) encoder, said encoder structure may include: (1) sub-strip group receiving unit for receiving a sub-band coding and group configuration parameters; (2)

the reference data input and output units for switching between the reference data encoder, in particular: when the current frame is not an I frame, each reference data encoder switching means via the data transfer unit set, switching from the second encoder to obtain the desired reference data, which controls the switching between the encoder has been reconstructed reference frame sub-band set of data, updating of the encoder buffer strip of each sub-set of motion estimation search area reference data.

(3) coding unit for all sub-parallel to the current frame strip groups encoding: coding sub-band coding each group to complete the assignments, the encoding process for each sub-strip group raster scan sequence into one or more strips, each macro block and must only be assigned to a strip; each of the encoders complete the assigned sub-strip groups encoding all the macro blocks, resulting in the reconstruction sub-strip group, to produce and outputting a stream and parameters; moreover, the strip is preferably of a width equal to the width of the frame or the entire plurality of successive macroblock row; II of the present invention, the plurality of encoders each encoder may be: General processor, comprising: a processor's superscalar, VLIW signal processor (VLIW DSP); or large-scale field programmable array chip (FPGA); or custom VLSI chips (VLSI); or instructions set configurable processor.

Each encoder further configuration data random access memory (RAM), that is a storage unit for caching and reconstruction of the current sub-band set of data and other information. You can use the data in RAM capacity is not large-capacity on-chip RAM, required larger capacity chip can be configured to use synchronous static random access memory (SSRAM) or

synchronous dynamic random access memory (SDRAM).

Furthermore, in order to obtain a better parallel processing for H. 264, VC- 1, AVS1.0 - P2 standard encoding, the encoder encodes the loop filter way: Set to select the current frame boundary-loop filtering mode for all bands is not filtered, each coder independently complete loop filtering process, each encoder, no exchange of data between each strip; specifically: the loop filter in which a strip after the band completed the first macro block start rebuilding, reconstruction and loop filter macro blocks constituting the macro block level pipeline; or, a loop filter strip after the strip began to rebuild completed, each encoder Articles loop filter band performed independently of each encoder does not need to exchange data between each band.

Encoder

in the encoding process, the corresponding reference data exchange process: when the current frame is not an I frame, the reference frame exchange of data between the sub-strip group encoder, each encoder according to the current sub-frame strip groups and sub-frame reconstruction slice group image area, storage area overlapping relationship, and considering the current sub-frame strip groups estimate the maximum motion search area, the reference frame of reference to determine the minimum data exchange between the encoder, updating of the encoder buffer strip of each sub-set of motion estimation search area reference data; wherein the motion estimation can be used fixed-size search window, a reference frame, according to the search window size determines the least reference data exchanged between the encoder, updating of the encoder buffer strip each sub-set of motion estimation search area reference data.

Note that, in the process of the encoder performs the encoding operation, usually required before the current frame video coding also requires preprocessing, comprising: a two-dimensional de-noising, scaling and format conversion, format conversion means by the input digital video format into the format required by the encoder processing such as; common encoding digital video format is 4: 2: 0, common input digital video format BT656 or CCIR601 specified 4: 2: 2. H. 264 hi-spreading code supports 4: 2: 2 and 4: 4: 4, where the input digital video with 4: 2: 2 and 4: 4: 4. HD video is often physical interface HD-SDI.

In addition, a plurality of encoded current frame data required for pre-treatment can take one of two forms of the following: 1, the main processor receives the current frame, and after pretreatment centralized cache hair forwarded to multiple encoders; such programs need strong II host processor capability, large data storage capacity frame; 2, a plurality of encoders each pretreatment processing sub-band group allocation. means the main processor of the present invention may also be provided with a control unit for initialization, configuration and control of the main processor for other processing units and a plurality of encoders complete the entire video sequence encoding. means according to the present invention, in order to simplify start of each encoder, a circuit configuration, reduce costs, the encoder of the start, the configuration program or data from a host processor may be; for the FPGA encoder case, you can configure the interface by the host processor configures the FPGA via parallel or serial; encoder for a general purpose processor case, choose parallel or serial boot (boot), the BootROM startup process application program from the host processor.

The device of the present invention is typically an embedded computer

system consists of one or more circuit boards constituted; or may be a personal computer or server connected through the communication network system in which the encoder can be PC or server.

The method and apparatus of the present invention applies a variety of coding standards and various grades.

The present invention provides for a variety of video formats, including HD video or SD video, which can be both progressive and to be interlaced, for interlaced video, a top and bottom fields of the interlaced frame macro blocks corresponding to macro block needs to be allocated to a macro-block with a strip group, sub-group of the strip, the strip encoding.

The above are only preferred embodiments of the present invention, but the scope of the present invention is not limited thereto, and any skilled in the art in the art within the technical scope disclosed by the present invention can easily think of changes or replacement shall fall within the scope of the present invention. Accordingly, the scope of the present invention should be in the scope of the claims prevail.

Cited by: CN101969560 A ; CN102231631 A ; CN102281441 B ; CN102340659 A ; CN102461168 A ; CN103124345 A ; CN103141098 A ; CN103141098 B ; CN103268263 A ; CN103268263 B ; CN103299641 A ; CN103299641 B ; CN103414902 A ; CN103442196 A ; CN103442196 B ; CN103458244 A ; CN103716625 A ; CN103716626 A ; CN103780909 A ; CN103916675 A ; CN103916675 B ; CN103918258 A ; CN104038766 A ; CN104205834 A ; CN104243992 A ; CN104780377 A ; US8218642 BB ; WO09149671 A1 ; WO12171400 A1 ; WO13071460 A1 ;

PatentOrder

Publication number: CN101207810 A
Publication country: CHINA
Publication type: UNEXAMINED APPLIC. OPEN TO PUBLIC INSPECTION
Publication date: 20080625
Application number: CN200610165565
Application date: 20061221
Priority: CN200610165565200612 21CN20061165565;
Assignee: LENOVO BEIJING CO ;
Assignee[std]: LENOVO BEIJING CO LTD ;
Inventor: LI WENJIE ;
Inventor[std]: WENJIE LI ;
International class[8]: H04N7/26; H04N7/26; H04N7/32; H04N7/32;
Family members: CN100571390 C CN101207810 A

Title: Method and apparatus for fast mode selecting of H264 video coding

Title: 一种H264视频编码快速模式选择方法和装置

Abstract:
The invention discloses a quick mode selection method and a device of an H264 video coding. The method comprises the selection of a mode in frames and the selection of the mode between the frames, wherein, for the selection of the mode in the frames, a prediction method with different fineness is chosen according to the dramatic degree of the image information changes in the prior macro block, thereby decreasing the number of the model to be chosen and confirming the final selection mode through the method of code rate-distortion factor optimizing; for the selection of the mode between the frames, the sum of the absolute difference of the corresponding pixel point of the corresponding macro block of the prior frame and the reference frame is used as a dividing basis to divide the macro block into different fineness and to correspondingly choose a prediction method with different fineness for decreasing the number of the model to be chosen and confirming the final selection mode through the method of code rate-distortion factor optimizing. The invention can decrease the number of the model to be chosen and solve the application limited problems caused by over-complicated code, over-loaded operation and over time consumption in the prior art.

Abstract:

本发明公开了一种H264视频编码快速模式选择方法和装置，方法包括：对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程度来选择不同精细度的预测方式，以减少候选模式的数量，并通过"码率—失真度优化"的方式确定最后的选定模式；对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式以减少候选模式的数量，并通过"码率—失真度优化"的方式确定最后的选定模式。本发明

The present invention discloses a H264 video coding fast mode selection method and device, the method including: intra mode, change the intensity of the internal image information according to the current macroblock to select the prediction of different fineness in order to reduce the candidate mode by the number of bit rate - distortion optimization "to determine the final selected model; for the inter mode, the current frame and corresponding to the reference frame macroblock corresponding pixel absolute

可以减少候选模式的数量，解决现有技术中因编码复杂度过高、运算量过大、耗时过多而造成的应用限制的问题。

difference and as a division in accordance with, macroblock is divided into different fineness corresponds to select the prediction of different fineness in order to reduce the number of candidate patterns, and through the bit rate - distortion optimization "to determine the final selected model. The present invention can reduce the number of candidate patterns, coding complexity in the existing technology is too high, excessive computing, consuming too much of the application constraints.

Claims:

1. 一种H264视频编码快速模式选择方法，其特征在于，包括如下步骤： 步骤一、对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程 度来选择不同精细度的预测方式，以减少候选模式的数量，并通过“码率一 失真度优化”的方式确定最后的选定模式； 步骤二、对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素 点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同 精细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的 方式确定最后的选定模式。

1. a H264 video coding fast mode selection method, characterized in that includes the following steps: Step 1, the intra mode, change the intensity of the current the macroblock internal image information to select the prediction of different fineness way to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; Step 2 for the inter mode, the current frame and reference frame corresponding to the macro block corresponding to as absolute difference and as a division basis, the macroblock is divided into different fineness and corresponds to choose a different fineness prediction to reduce the number of candidate patterns, and through the bit rate a distortion optimization "way of the prime point of determining the final the selected mode.

2.根据权利要求1所述的方法，其特征在于，在所述步骤二中，在选择 不同精细度的预测方式后，还通过宏块的帧间运动的方向信息来减少候选模式的数量。

2 The method according to claim 1, characterized in that, in as described in Step 2, choose a different forecast of finesse, but also by the direction information of the macroblocks of the inter-frame movement to reduce the candidate patterns. number.

3.根据权利要求1或2所述的方法，其特征在于，所述步骤二包括： 步骤a，根据图像信息或用户输入，确定全局预测门限值TV和局部预测 门限值PV; 步骤b,将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤c，对目标宏块进行16X16像素大小的全搜索，找到最佳的匹配块， 将目标宏块和最佳匹配块进行比较，得到目标宏块的全局对应象素点的绝对 差之和T-SAD; 步骤d，判断T-SAD是否小于TV,是则执行步骤e,否则执行步骤g; 步骤e,说明16X16像素大小的全搜索已经可以满足图像质量的要求， 对该目标宏块进行粗精度的预测方式，得到对应的候选模式，然后在16X16 像素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选 定模式，结束帧间模式的选择流程； 步骤g，说明16X16像素大小的全搜索并不能满足图像质量的要求，对 该目标宏块进行细精度的预测方式，得到对应的候选模式，然后结合所述候 选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间模式的选择流程。

3. method according to claim 1 or 2 above, characterized in that, as described in Step 2: Step a, according to the image data or user input to determine the global prediction threshold TV and local predict threshold PV; steps b, the image of each frame is divided into 16X16 pixel size as the basic coding unit macroblock; step c, the target macroblock 16X16 pixel size of the search, to find the best matching block, the target macroblock good matching block were compared to obtain the corresponding pixel in the target macroblock global absolute difference and T-the SAD; step d, to determine T-SAD is less than TV, is to step e, otherwise to step g; step e 16X16 pixel size of the search has to meet the image quality requirements, the rough accuracy of the target macroblock prediction mode, the corresponding candidate patterns described in the candidate model, and then in the 16X16 pixel precision,

4.根据权利要求3所述的方法，其特征在于，在所述步骤e中，所述粗 精度的预测方式包括：将目标宏块划分为能体现方向信息的两个16X8子块 和两个8X16子块，计算四个子块的对应象素点的绝对差之和SAD，并选出 两个16X8子块中的SAD最大

值MAX16X8和两个8X 16子块中的SAD最 大值MAX8X16,选取MAX16X8和MAX8X16中数值较小的所对应的候 选模式。

5.根据权利要求3所述的方法，其特征在于，所述步骤g包括： 步骤gl，将当前宏块进一步划分为4个8X8像素大小的子块，并在全 搜索的基础上进行局部范围的搜索找到各子块的最佳匹配块，将目标子块和 最佳匹配块进行比较，得到该目标子块的局部对应象素点的绝对差之和 P-SAD； 步骤g2,判断P-SAD是否小于PV，是则执行步骤g3，否则执行步骤g4; 步骤g3，说明8X8像素大小的搜索已经可以满足图像质量的要求，将8 X 8像素大小的目标子块划分为能体现方向信息的两个8 X 4子块和两个4X 8 子块，计算四个子块的对应象素点的绝对差之和SAD，并选出两个8X4子 块中的SAD最大值MAX8 X4和两个4X 8子块中的SAD最大值MAX4X 8， 选取MAX8X4和MAX4X8中数值小的所对应的候选模式，然后在8X8像 素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选定 模式，结束帧间的模式选择流程； 步骤g4,说明8X8像素大小的搜索并不能满足图像质量的要求，所以 只能对该目标子块采用精细的4X4预测方式，进行“码率一失真度优化”， 得到最后的选定模式，结束帧间的模式选择流程。

6.根据权利要求1所述的方法，其特征在于，在所述步骤一中，在选择 不同精细度的预测方式后，还通过当前宏块的方向矢量信息来减少候选模式的数量。

7.根据权利要求1或6所述的方法，其特征在于，所述步骤一包括： 步骤A,根据图像信息或用户输入，确定预测门限值VH; 步骤B，将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C,计算目标宏块内部图像信息的变化激烈程度VT; 步骤D，判断VT的值是否大于VH，是则执行步骤E，否则执行步骤F 步骤E,说明宏块内部图像信息的变化剧烈，采用细精度的4X4预测方式，计算当前4X4像素大小的子块的局部方向矢量P-DV，根据方向矢量的 信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码率—失真度优化”，得到最后的选定模式，结束帧内的模式选择流程； 步骤F，说明宏块内部图像信息的变化平缓，采用粗精度的16X16预测 方式，计算当前16X16像素大小的宏块的全局方向矢量T-DV，根据方向矢 量的信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码 率一失真度优化”，得到最后的选定模式，结束帧内的模式选择流程。

8.一种H264视频编码快速模式选择装置，其特征在于，包括： 帧内模式选择模块，用于：根据当前宏块内部图像信息的变化剧烈程度 来选择不同精细度的预测方式，以减少候选模式的数量，并通

combined with code a rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode; step g, 16X16 pixel size of the search and can not meet the requirements of the image quality of the target macroblock fine accuracy of the forecast way, the corresponding candidate patterns, and then combined with the candidate patterns described in a bit rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode.

4 The method according to claim 3, characterized in that the rough accuracy of the forecast include: the target macroblock is divided into to reflect the direction information of the two 16X8 sub-block in said step e and two 8X16 sub-block, calculate the four sub-blocks of the corresponding pixel absolute difference and the SAD, and elected two 16X8 sub-block, the SAD maximum MAX16X8 and two 8X 16 sub-blocks in the SAD maximum MAX8X16 smaller the select MAX16X8 and MAX8X16 value corresponding candidate patterns.

5. method according to claim 3, characterized in that, as described in Step g: Step gl, the current macroblock is further divided into four sub-block size 8X8 pixels and on the basis of the whole search The local scope of a search to find the best match for each sub-block block, the target sub-block and the best matching block, get the target sub-block local corresponding pixel absolute difference and P-the SAD; step g2, to determine P-SAD is less than the PV, then perform step g3, otherwise steps g4 and; step g3, 8X8 pixel size of the search has to meet the requirements of the image quality will target sub-block size of 8 x 8 pixels is divided into to reflects the directional information of two 8 x 4 sub-blocks and two 4X 8 sub-block to calculate the four sub-blocks of the corresponding pixel absolute difference and the SAD, and elected two 8X4 sub-block SAD maximum value MAX8 X4 select the MAX8X4 and MAX4X8 values in the corresponding candidate model, then the combination of 8X8 pixel precision and two 4X 8 sub-blocks in the SAD maximum value MAX4X candidate patterns described in the "bit rate a distortion optimization "to get the last selected mode, the end of the interframe the mode selection process; step g4 and 8X8 pixel size of the search does not meet the requirements of image quality, so only the target sub-block fine 4X4 prediction methods, "rate distortion degree of optimization, the last selected mode, the end of the interframe the mode selection process.

6. The method according to claim 1,

过“码率一失 真度优化”的方式确定最后的选定模式； 帧间模式选择模块，用于：将当前帧和参考帧的对应宏块的对应象素点 的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同精 细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的方 式确定最后的选定模式。

characterized in that, to reduce the number of candidate patterns described in step one, choose a different forecast of finesse, but also by the current macroblock direction vector information.

7. The method according to claim 1 or 6 as described, characterized in that the steps include: the step A, in accordance with image information or user input to determine the predictive threshold the VH; Step B, the image of each frame is divided into 16X16 pixel size as the basic coding unit macroblock; step C, calculate the internal target macroblock image information changes the intensity of the VT; Step D, to determine the value of the VT is greater than VH, is to step E, otherwise implementation of Step F Step E, the rapid change of the macroblock internal image information, the fine precision 4X4 forecast to calculate the current 4X4 pixel size of the local direction of the sub-block vector P-DV, under the direction vector of the information obtained is most likely to two candidate model, then the candidate with the DC mode to "rate a distortion optimization, the last selected mode, the end of the intra mode selection process; Step F, indicating moderate changes in the macro block of image information within 16X16 prediction mode using coarse accuracy, calculate the current size of 16X16 pixels the macroblock global direction vector T-DV, according to the information of the direction vector draw the two most likely candidate mode, then the candidate with the DC mode "bit rate a distortion optimization ", the last selected mode, the end of the frame of the model selection process.

8. - kinds H264 video encoder fast mode to select a device, characterized in that, including: intra mode decision module, used for: changes in the intensity of the internal image information according to the current macroblock to select the prediction of different fineness to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; inter mode decision module for: the current frame and corresponding to the reference frame macroblock corresponding pixel the absolute difference and as a division basis, the macroblock is divided into different fineness and corresponds to choose a different fineness prediction to reduce the number of candidate patterns, and through the bit rate a distortion optimization "way to determine the final election given pattern.

Description:

一种H264视频编码快速模式选择方法和装置 技术领域 本发明涉及视频编码模式的选择方法，特别是涉及一种H264视频编码 快速模式选择方法和装置。

背景技术 同人们熟悉的H.263相比，H.264标准下的视频图像质量更好，同时编码 效率可以提高50%以上，并且具有更好的容错能力，这使得它在未来的视频 通信领域具有广泛的应用前景。然而现有技术中H.264/AVC编码技术是有缺 陷的。这是因为，依据H.264/AVC中码率失真度优化（RDO)的定义，对视 频序列进行编码时，对帧图像必须使用所有的预测模式各测试一遍，然后找到一个从RDO角度来讲最佳的预测模式，使得编码计算的复杂度大幅增加， 增大编码耗时，使得H.264/AVC的实时性变差。

因此，尽管率失真优化可以带来非常好的编码性能，但是对于编码时所 有模式的搜索消耗大量时间，这使得H.264编码器很难用于实时业务。很多快速的运动估计和模式决定的方法被提出用于改善这个情况，比如2004年多 媒体国际会议（ICME2004)中发表的《帧内模式预测》，以及2004年图像处 理国际会议中发表的《H.264中低复杂度的宏块模式预测》和《H.264中自适 应门限值的帧间预测模式选择》中，提出了利用门限值提前终止的算法，用 来减少对于候选预测模式的搜索。

然而这些快速实现的模式选择方法依然有缺陷，主要体现在对于优选的 预测模式的选择仍然过于复杂，需要进行相对较多的计算。

发明内容 本发明的目的在于提供一种H264视频编码快速模式选择方法和装置， 解决现有技术中因编码复杂度过高、运算量过大、耗时过多而造成的应用限 制的问题。

为了实现上述目的，本发明提供了一种H264视频编码快速模式选择方 法，其中，包括如下步骤：步骤一、对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程 度来选择不同精细度的预测方式，以减少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模式； 步骤二、对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素 点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同 精细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的 方式确定最后的选定模式。

上述的方法，其中，在所述步骤二中，在选择不同精细度的预测方式后， 还通过宏块的帧间运动的方向信息来减少候选模式的数量。

上述的方法，其中，所述步骤二包括： 步骤a,根据图像信息或用户输入，确定全局预测门限值TV

The fast mode of the

a H264 video encoding method and device technology of the present invention relates to video coding mode selection method, especially the fast mode selection method and apparatus involving an H264 video encoder. background with people familiar with the H.263, the H.264 standard video image quality better, while the coding efficiency can be improved by more than 50 percent, and has better fault tolerance which makes it in the future the field of video communications has wide application prospect. However, the existing technology in H.264/AVC encoding technology is flawed. This is because, when the video sequence is encoded in accordance with the definition of H.264/AVC rate distortion optimization (RDO), the frame must use all of the forecasting model test again, then find one in terms of the angle from the RDO the best forecasting model, making the coding complexity of computing a substantial increase, increasing the encoding time-consuming, making H.264/AVC real-time variation.

Therefore, although the rate-distortion optimization can be a very good coding performance, but time-consuming search for the encoding mode, which allows the H.264 encoder is difficult for real-time business. A lot of fast motion estimation and mode decision methods have been proposed for the improvement of the situation, such as Multimedia International Conference 2004 (ICME2004), intra mode prediction, as well as published in the International Conference on Image Processing 2004 H. 264 low complexity macroblock mode prediction and inter prediction modes of H.264 adaptive threshold select "proposed threshold of early termination algorithm used to reduce the search for candidate forecasting models.

However, the mode selection method quickly realize there are still defects, mainly reflected in the choice of the optimal prediction model is still too complex, the need for relatively large computing. contents of the invention of the present invention aims to a H264 video coding fast mode selection methods and devices, existing technologies due to coding complexity is too high, operator overload, time-consuming too much caused by the application of restrictions problem.

To achieve these goals, the present invention provides a H264 video coding fast mode selection method, which includes the following steps: Step 1, the intra mode, to select on the basis of the current macroblock internal changes in the

和局部预测 门限值PV; 步骤b,将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C，对目标宏块进行16X16像素大小的全搜索，找到最佳的匹配块， 将目标宏块和最佳匹配块进行比较，得到目标宏块的全局对应象素点的绝对 差之和T-SAD; 步骤d,判断T-SAD是否小于TV，是则执行步骤e，否则执行步骤g; 步骤e，说明16X16像素大小的全搜索已经可以满足图像质量的要求， 对该目标宏块进行粗精度的预测方式，得到对应的候选模式，然后在16X16 像素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选 定模式，结束帧间模式的选择流程； 步骤g,说明16X16像素大小的全搜索并不能满足图像质量的要求，对 该目标宏块进行细精度的预测方式，得到对应的候选模式，然后结合所述候 选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间模式的选 择流程。

上述的方法，其中，在所述步骤e中，所述粗精度的预测方式包括：将 目标宏块划分为能体现方向信息的两个16X8子块和两个8X16子块，计算 四个子块的对应象素点的绝对差之和SAD，并选出两个16X8子块中的SAD 最大值MAX16X8和两个8X16子块中的SAD最大值MAX8X16，选取 MAX16X8和MAX8X16中数值较小的所对应的候选模式。

上述的方法，其中，所述步骤g包括： 步骤g1,将当前宏块进一步划分为4个8X8像素大小的子块，并在全 搜索的基础上进行局部范围的搜索找到各子块的最佳匹配块，将目标子块和 最佳匹配块进行比较，得到该目标子块的局部对应象素点的绝对差之和 P-SAD； 步骤g2，判断P-SAD是否小于PV，是则执行步骤g3，否则执行步骤g4; 步骤g3，说明8X8像素大小的搜索已经可以满足图像质量的要求，将8 X 8像素大小的目标子块划分为能体现方向信息的两个8 X 4子块和两个4 X 8 子块，计算四个子块的对应象素点的绝对差之和SAD，并选出两个8X4子 块中的SAD最大值MAX8X4和两个4X8子块中的SAD最大值MAX4X 8， 选取MAX8X4和MAX4X8中数值小的所对应的候选模式，然后在8X8像 素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间的模式选择流程； 步骤g4，说明8X8像素大小的搜索并不能满足图像质量的要求，所以 只能对该目标子块采用精细的4X4预测方式，进行“码率一失真度优化”， 得到最后的选定模式，结束帧间的模式选择流程。

上述的方法，其中，在所述步骤一中，在选择不同精细度的预测方式后， 还通过当前宏块的方向矢量信息来减少候选模式的数量。

上述的方法，其中，所述步骤一包括： 步骤A,根据图像信息或用户输入，确定预测门限值VH; 步骤B，将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C,计算目标宏块内

intensity of the image information different fineness prediction mode, in order to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; Step 2 for the inter mode, the current frame and reference frame corresponding to macroblock corresponding pixel absolute difference and division basis, the macroblock is divided into different granularity and corresponding select different fineness prediction to reduce the number of candidate patterns, and through "rate distortion degree of optimization "way to determine the selected mode. the above method, in which the steps described in two, choose a different forecast of finesse, but also by the direction information of the macroblocks of the inter-frame movement to reduce the number of candidate patterns. the above method, wherein Step 2 include: steps a, according to the image data or user input to determine the global prediction threshold TV and local predict threshold PV; step b, the image of each frame is divided into as the basic coding unit of 16X16 pixel size of the macro block; step C, the target macroblock 16X16 pixel size of the search, to find the best matching block, the target macroblock and the best matching block, get the target macroblock corresponds to the global pixel absolute difference and T-the SAD; step d, to determine T-SAD is less than TV, is to step e, otherwise to step g; to step e, 16X16 pixel size of the search have to meet image quality requirements of the target macroblock coarse accuracy of prediction, the corresponding candidate patterns described candidate model "rate distortion optimization", and then combined 16X16 pixel precision, and the last election given mode, the end of the selection process of the inter-frame mode; step g 16X16 pixel size of the search and can not meet the requirements of the image quality, fine accuracy of the prediction in the target macroblock corresponding candidate patterns, and then combined with the candidate patterns described a bit rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode. rough accuracy of prediction methods include:

the method described above, which, in said step e, as described in the target macroblock is divided into two 16X8 sub reflect the direction of information blocks and two 8X16 sub-block to calculate the four The absolute difference of the corresponding sub-block pixel SAD, and elected two 16X8 sub-block, the SAD maximum MAX16X8 and two 8X16 sub-block in the SAD maximum MAX8X16, the select MAX16X8 and MAX8X16 value smaller corresponding to the candidate mode. the above method, wherein step g including: steps to gl, the current macroblock is

部图像信息的变化激烈程度VT; 步骤D,判断VT的值是否大于VH,是则执行步骤E，否则执行步骤F 步骤E,说明宏块内部图像信息的变化剧烈，采用细精度的4X4预测方 式，计算当前4X4像素大小的子块的局部方向矢量P-DV，根据方向矢量的 信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码率一 失真度优化”，得到最后的选定模式，结束帧内的模式选择流程； 步骤F,说明宏块内部图像信息的变化平缓，采用粗精度的16X16预测 方式，计算当前16X16像素大小的宏块的全局方向矢量T-DV，根据方向矢 量的信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码 率一失真度优化”，得到最后的选定模式，结束帧内的模式选择流程。

为了实现本发明的目的，本发明还提供了一种H264视频编码快速模式 选择装置，其中，包括：帧内模式选择模块，用于：根据当前宏块内部图像信息的变化剧烈程度来选择不同精细度的预测方式，以减少候选模式的数量， 并通过“码率一失真度优化”的方式确定最后的选定模式；帧间模式选择模 块，用于：将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为划 分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式以减 少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模 式。

本发明的优点在于： 1)本发明根据宏块内部图像信息的变化剧烈程度来选择不同精细度的预 测方式，可以减少帧内模式选择中的候选模式的数量，进而减少了运算量。

2)本发明将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为 划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式，从而减少了帧间模式选择中的候选模式的数量，进而减少了运算量。

3)本发明还通过宏块的帧间运动的方向信息来减少帧间模式选择中的候 选模式的数量。

4)本发明还通过当前宏块的方向矢量信息来减少帧内模式选择中的候选 模式的数量。

5)本发明不但有帧内的模式选择还有帧间的模式选择，本发明应用于 H.264/AVC标准，可以有效的减少4x4大小的候选模式的数目，是一种能有 效的减少运算量和运算时间的视频编码方法。

附图说明 图1为本发明方法的步骤流程图； 图2为本发明提供的自适应帧间预测模式选择算法流程图； 图3为本发明中宏块的四个参数的含义示意图； 图4为本发明提供的自适应帧内预测模式选择算法流程图。

具体实施方式 本发明提出了一种H264视频编码快

further divided into four sub-block size 8X8 pixels, and a full search based on local-wide search to find all the sub-block the best matching block, the target sub-block and the best matching blocks were compared to obtain the target sub-block of the local corresponding pixel absolute difference and P-the SAD; step g2, to determine P-the SAD is less than the PV perform the steps g3, otherwise to step g4 and; step g3, 8X8 pixel size of the search has to meet the requirements of the image quality will be 8 X 8 pixel size of the target sub-block partition in order to reflect the direction information of the two X-4 sub-block and two 4 X 8 sub-blocks, and calculate the four sub-blocks corresponding pixel absolute difference, SAD, and elected in the SAD maximum of two 8X4 sub-block the MAX8X4 and two 4X8 sub-block SAD the maximum MAX4X 8, corresponding to the candidate to select the MAX8X4 and MAX4X8 numerical mode, and then combined in the 8X8 pixel precision candidate patterns described in the "rate distortion optimization", the last selected mode, the end of inter-frame mode selection process; step g4 and 8X8 pixel size of the search and can not meet the requirements of the image quality, so only the target sub-block fine 4X4 prediction methods, the bit rate is a distortion optimization ", to get the last selected mode, the end of the inter-frame mode selection process. the method described above, which, in the steps described in a, in the choice of different fineness forecast, but also by the current macroblock direction vector information to reduce the number of candidate patterns. the method described above, which, as described in step one, including: Step A predictive threshold based on image information or user input to determine the VH; Step B, the image of each frame is divided into 16X16 pixel size as the basic coding unit the macro block; step C, the calculation of the target macroblock internal image information change the intensity of the VT; Step D, to determine the value of the VT is greater than VH, is to step E, otherwise step E step F, indicating the macroblock internal image rapid change of information, fine precision 4x4 prediction mode, calculate the current 4X4 pixel size of the local orientation of the sub-block vector P-DV, the two most likely candidate mode information obtained under the direction vector, and then with the DC candidate mode "with a bit rate distortion optimization", the last selected mode, the end of the frame of the model selection process; step F, indicating gentle the macroblock internal image information changes, 16X16 prediction mode using coarse accuracy to calculate the current 16X16 pixel size of the macroblock of the global direction vector T-DV to draw the two most likely candidate mode, according to the information of the direction vector, then the candidate with the

速模式选择方法，图1为本发明方 法的步骤流程图，如图，本发明分为帧内算法流程101和帧间算法流程102 两个部分。在快速帧间模式选择算法中，本发明使用当前帧和参考帧对应宏 块的对应象素点的绝对差之和（SAD，Sum of Absolute Difference)作为相关 性的判断，把宏块划分为不同精细度的模式•，然后根据宏块的帧间运动的方向信息，进一步获得优选的预测模式，最后在优选的预测模式中使用码率一 失真度优化（RDO)，从而确定最佳模式。而对于快速帧内模式选择算法，本 发明可以首先根据当前宏块内部图像信息的变化剧烈来选择不同精细度的预 测模式，并且可以根据当前宏块的方向矢量信息，从中选择出最有可能的几 种预测模式，然后对这些优选出来的预测模式进行“码率一失真度优化” (RDO)算法选择出最佳模式。

图2为本发明提供的自适应帧间预测模式选择算法流程图，如图，帧间 模式选择算法主要包括以下步骤.• 步骤201.根据图像信息或用户输入，确定全局预测门限值TV和局部预 测门限值PV; 步骤202.将图像的每一帧划分成为基本的编码单元，即16X16像素大 小的宏块； 步骤203.对其中的目标宏块进行16X16像素大小的全搜索，找到最佳 的匹配块，将目标宏块和最佳匹配块进行比较，得到该目标宏块的全局SAD 值（T-SAD); 步骤204.判断T-SAD的值是否小于TY，是则执行步骤205，否则执行 步骤207; 步骤205,如果T-SAD的值小于TV，说明16X 16像素大小的全搜索已 经可以满足图像质量的要求，所以只需要接着对该目标宏块进行粗精度的模 式选择，即从16X16, 16X8，8X16模式中选择一个最合适的； 步骤206，对于目标宏块粗精度的模式选择，首先按照图3所示分别计 算目标宏块的上半部分、下半部分、左半部分和右半部分所对应的四个SAD 值，并分别得出在上/下、左/右这两种模式下的SAD的最大值，表示为MAX16 X8和MAX8X16，然后比较并选出MAX16X8和MAX8X 16中较小的所对 应的模式，说明在该模式下如果进行运动搜索和运动补偿会取得更好的效果。

然后就选择16X16和该模式进行码率一失真度优化（RDO)算法，选择出最 佳模式，转步骤213。

步骤207,如果T-SAD的值大于TV，说明16X 16像素大小的全搜索并 不能满足图像质量的要求，所以还需要接着对该目标宏块进行更细精度的模 式选择，即从8X8，4X8，8X4, 4X4模式中选择一个最合适的； 步骤208，对于目标宏块细精度的模式选择，将当前宏块进一步划分为4 个8X8像素大小的子块； 步骤209,在全搜索的基础上进行小范围的搜索找到各自最佳的匹配块， 将目标子块和最佳匹配块进行比较，得到该目标子块的局部SAD值(P-SAD); 步骤210,判断P-SAD的值是否小于PV，是则执行步骤211，否则执行 步骤212; 步骤211，如果P-SAD的值小于PV，说明8X8像素大小的搜索已经可 以满足图像质量的要求，所以只需要接着

DC mode to "rate a distortion optimization, the final selected mode, the end of the frame of the model selection process.

In order to achieve the purpose of the present invention, the present invention also provides a H264 video coding fast mode selection device, which include: intra mode decision module, used for: changes in intensity to the internal image information according to the current macroblock select the prediction of different fineness, in order to reduce the number of candidate patterns, and through "rate is a distortion optimization" to determine the final selected model; inter mode decision module for: the current frame and reference frame corresponds to the absolute difference and as a division of the corresponding pixel in the macroblock basis, the macroblock is divided into different fineness corresponds to select different fineness prediction to reduce the number of candidate patterns, and through the bit rate distortion The Optimization determine the last selected mode. the advantages of the present invention in

is: 1) The present invention is to select the changes in intensity according to the macro block of image information within the forecast of different fineness, can reduce the number of candidate patterns in the intra mode, thereby reducing the amount of computation.

2) of the present invention to the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference and as a division basis, the macroblock is divided into different fineness corresponds to choose a different fineness prediction, thereby reducing the number of candidate patterns in the inter mode, thereby reducing the amount of computation.

3) The present invention also interframe macroblock motion direction information to reduce the number of candidate patterns in the inter mode.

4) of the present invention through the current macroblock direction vector information to reduce the number of candidate patterns in the intra mode.

5) of the present invention not only intra mode selection in the inter-frame mode selection, the present invention is applied to the H.264/AVC standard, can effectively reduce the 4x4 size of the number of candidate patterns is an effective video coding method to reduce the computational

对该目标子块进行粗精度的模式选 择，即从8X8, 8X4, 4X8模式中选择一个最合适的； 对于目标子块粗精度的模式选择，按照图3所示分别计算目标子块（8 X8像素大小）的上半部分、下半部分、左半部分和右半部分所对应的四个 SAD值，并分别得出在上/下，左/右这两种模式下的SAD的最大值，表示为 MAX8X4和MAX4X8，然后比较并选出MAX8X4和MAX4X8中较小的 所对应的模式，说明在该模式下如果进行运动搜索和运动补偿会取得更好的 效果。然后就选择8X8和该模式进行码率一失真度优化（RDO)算法，选择 出最佳模式，转步骤213。

步骤212,如果P-SAD的值大于PV,说明8X8像素大小的搜索并不能 满足图像质量的要求，所以只能对该目标子块采用最精细的模式，即4X4模 式，转步骤213。

步骤213,完成目标宏块的帧间模式选择。

图4为本发明提供的自适应帧内预测模式选择算法流程图。如图，快速 帧内模式选择算法主要包括以下步骤： 步骤401.根据图像信息或用户输入，确定预测门限值VH; 步骤402.将图像的每一帧划分成为基本的编码单元，即16X16像素大 小的宏块； 步骤403.计算目标宏块内部图像信息的变化激烈程度，用VT表示； 步骤404.判断VT的值是否大于VH,是则执行步骤405，否则执行步 骤 408; 步骤405，如果VT的值大于VH,说明宏块内部图像信息的变化剧烈， 就需要采用细精度的预测模式，即4X4预测模式； 步骤406.如果采用细精度的预测模式，即4X4预测模式，就计算当前 子块（4X4像素大小）的局部方向矢量P-DV; 步骤407.根据方向矢量的信息得出最有可能的2种候选预测模式，转 步骤411; 步骤408.如果VT的值小于VH，说明宏块内部图像信息的变化不大， 就可以采用粗精度的预测模式，即16X16预测模式； 步骤409.如果采用粗精度的预测模式，即16X16预测模式，就计算当 前宏块（16X16像素大小）的全局方向矢量T-DV; 步骤410，根据方向矢量的信息得出最有可能的2种候选预测模式； 步骤411.将候选预测模式同DC预测模式（DC模式是帧内预测4x4大 小的9种预测模式中的一种，DC模式下4x4块的所有像素点的预测值是相同 的）一起进行码率一失真度优化（RDO)算法，选择出最佳模式。

步骤412.完成目标宏块的帧内预测模式选择。

由以上论述可知，本发明可以带来以下技术效果：1)本发明根据宏块内部图像信息的变化剧烈程度来选择不同精细度的预 测方式，可以减少帧内模式选择中的候选模式的数量，进而减少了运算量。

2)本发明将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为 划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式，

complexity and computation time.

Description of the Drawings Figure 1 step flowchart of the present invention; the present invention to provide adaptive inter-frame prediction mode selection algorithm flowchart; meaning schematic of the present invention the macroblock four parameters; Figure 4 the present invention to provide adaptive intra prediction mode selection algorithm flowchart. specific implementation modalities of the present invention proposed a H264 video coding fast mode selection method, a flowchart of the steps of the present invention shown, the present invention is divided into 102 two frame algorithm processes 101 and interframe algorithm processes parts. Fast inter mode selection algorithm, the present invention uses the corresponding pixel of the current frame and reference frame corresponding to the macroblock of absolute differences (SAD, Sum, of the Absolute Difference,) as a correlation between the judgment of the macroblock divided into different finesse mode, and then under the direction of the macroblocks of the inter-frame movement of information, and further obtain the optimal prediction mode, the bit rate distortion optimization (RDO), the last in the optimal forecasting model, to determine the best mode. For fast intra mode selection algorithm, the present invention can be first based on the rapid change of the current macroblock internal image information to select a different fineness prediction models, and according to the current macroblock direction vector information, choose the most likely to several forecasting models, these preferred forecasting model, the bit rate is a distortion degree of optimization (RDO) algorithm to select the best mode.

adaptive interframe the

Figure 2 the present invention to provide prediction mode selection algorithm flow chart shown, inter mode selection algorithm includes the following steps. ◆ Step 201 based on image information or user input to determine the global prediction threshold TV and local forecast threshold PV; to step 202. image of each frame is divided into the basic coding unit, the size of 16X16 pixel macro block; to step 203. one of the target macroblock 16X16 pixel size of the full search, to find the best matching block, the target macroblock and the best matching block compared to get the target macroblock global SAD value (T-SAD); step 204 to determine the value of the T-SAD is less than the TY is executed Step 205, otherwise perform step 207; step 205, if the T-the SAD value is less than TV,

从而减少了帧间模式选择中的候选模式的数量，进而减少了运算量。

3)本发明还通过宏块的帧间运动的方向信息来减少帧间模式选择中的候 选模式的数量。

4)本发明还通过当前宏块的方向矢量信息来减少帧内模式选择中的候选 模式的数量。

5)本发明不但有帧内的模式选择还有帧间的模式选择，本发明应用于 H.264/AVC标准，可以有效的减少4x4大小的候选模式的数目，是一种能有 效的减少运算量和运算时间的视频编码方法。

以上所述仅是本发明的优选实施方式，应当指出，对于本技术领域的普 通技术人员来说，在不脱离本发明原理的前提下，还可以作出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

^ Pit HI 对于帧内模式选择，根据当前宏块内部图像信 息的变化剧烈程度来选择不同精细度的预测方 式，以减少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模式； 对于帧间模式选择，将当前帧和参考帧的对应 宏块的对应象素点的绝对差之和作为划分依 据，把宏块划分为不同的精细度并对应选择不 同精细度的预测方式以减少候选模式的数量， 并通过“码率一失真度优化”的方式确定最后 的选定模式。

indicating that the search has to meet the requirements of the image quality of 16X 16 pixels in size, so only need then to the target macroblock coarse accuracy mode selection, from the 16X16, 16X8, 8X16 mode, select one of the most appropriate; step 206, select the mode of target macroblock rough accuracy, first shown in Figure 3 were calculated for the upper part of the target macroblock, the lower half of the left part and right part of the SAD value corresponding to the four, and were drawn up / down, left / right mode of Both the maximum value of the SAD, said that for the MAX16 X8 and MAX8X16-and then compare and choose smaller MAX16X8 and MAX8X 16 corresponding mode In this mode, the motion search and motion compensation to achieve better results. and then choose the 16X16 and the mode bit rate to a distortion optimization (RDO) algorithm to select the best mode, turn to step 213. step 207, if the value of the T-SAD than TV, description of the search and can not meet the requirements of the image quality of 16X 16 pixels in size, so you need then the target macroblock finer precision mode selection, ie 8X8, 4X8, 8X4, 4X4 mode, select one of the most appropriate; step 208, the mode selection of the target macroblock fine accuracy, the current macroblock further divided into four sub-block size 8X8 pixels; step 209, in the whole search on the basis of small-scale search to find their best matching block, the target sub-block and the best matching block to get the target sub-block partial SAD value (P-SAD); step 210 to determine the P- SAD value is smaller than the PV, then perform step 211, otherwise perform step 212; step 211, if the value is less than the P-SAD PV 8X8 pixel size of the search has to meet the image quality requirements, we only need then the mode selection of the coarse accuracy of the target sub-block, from the 8X8, 8X4, 4X8 mode, select one of the most appropriate; mode select for the target sub-block coarse accuracy were calculated in accordance with Figure 3 shows the target sub-block (8 X8 pixel size) in the upper half of the bottom half, left part and right part corresponding to the four SAD values were obtained in both modes of the up / down, left / right maximum SAD, the MAX8X4 and MAX4X8 then compare and select the mode the MAX8X4 and MAX4X8 smaller the corresponding instructions in this mode if the motion search and motion compensation to achieve better results. 8X8 and the mode and then choose to carry out the bitrate a distortion optimization (RDO) algorithm to select the best mode, turn to step 213. step 212, if the SAD of the P-value is greater than the PV 8X8 pixel size of the search does not meet the requirements of image quality, so only the target sub-block using the most sophisticated model, the 4X4 mode, go to step 213. steps 213,

the target macroblock inter mode.

the

Figure 4 the present invention to provide adaptive intra prediction mode selection algorithm flowchart. Figure fast intra mode selection algorithm includes the following steps: Step 401 based on image information or user input to determine the forecast threshold the VH; step 402 the image of each frame is divided into the basic coding unit, that is 16X16 pixel size macro block; to step 403. calculate the change in the intensity of the target macroblock internal image information, said in VT; step 404 to determine the value of VT is greater than VH, then perform step 405, otherwise perform step 408; step 405, if VT value is greater than VH and description of intense the macroblock internal image information changes, you need to fine the accuracy of forecasting models, forecasting models namely 4X4; step 406 if the fine accuracy of prediction models, namely 4X4 forecasting model, the purpose of calculating the current sub-block (4x4 pixels in size), the local direction vector P-DV; to step 407. the most likely candidate forecasting models drawn under the direction vector information, go to step 411; to step 408. If the value of VT is less than the VH that little change in the macroblock internal image information, you can use the rough accuracy of the forecasting model, ie, 16X16 prediction mode; to step 409. rough accuracy of the prediction model, that is 16X16 prediction mode of calculating the current macroblock (16X16 pixel size) global direction vector T-DV; step 410, the most likely candidate prediction mode information obtained under the direction vector; to step 411. the candidate forecasting model with the DC prediction mode (DC mode is the intra prediction 4x4 size a nine kinds of forecasting models, the predictive value of all pixels in the DC mode, the 4x4 block is the same) with one of bit-rate distortion optimization (RDO) algorithm to select the best mode. step 412. target macroblock intra prediction mode selection. discourse we can see from the above, the present invention can have the following effects: 1) The present invention is to select the intensity changes according to the macroblock internal image information of different fineness prediction can reduce the candidate patterns in the intra mode the number, thereby reducing the amount of computation.

2) of the present invention to the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference and as a division basis, the macroblock is divided into different fineness corresponds to

choose a different fineness prediction, thereby reducing the number of candidate patterns in the inter mode, thereby reducing the amount of computation.

3) The present invention also interframe macroblock motion direction information to reduce the number of candidate patterns in the inter mode.

4) of the present invention through the current macroblock direction vector information to reduce the number of candidate patterns in the intra mode.

5) of the present invention not only intra mode selection in the inter-frame mode selection, the present invention is applied to the H.264/AVC standard, can effectively reduce the 4x4 size of the number of candidate patterns is an effective video coding method to reduce the computational complexity and computation time. above is only preferred embodiments of the present invention, it should be pointed out that the person of ordinary skill for this technology, not out of principle of this invention, can also make a number of improvements and polish, these improvements and The polish should also be regarded as the scope of protection of the present invention. of Pit HI for intra mode selection, changes in the intensity of the internal image information according to the current macroblock to choose a different forecast of finesse in order to reduce the number of candidate patterns, and by "bit-rate distortion optimization" determine the selected mode; for the inter mode, the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference, and as divided in accordance with the macroblock is divided into different fineness and the corresponding select different fineness prediction to reduce the number of candidate patterns, and by the "bit rate is a distortion optimization" to determine the final selected model.

Cited by: CN101783951 A ; WO12068817 A1 ; WO12088806 A1 ;

Publication number: CN101207810 A
Publication country: CHINA
Publication type: UNEXAMINED APPLIC. OPEN TO PUBLIC INSPECTION
Publication date: 20080625
Application number: CN200610165565
Application date: 20061221
Priority: CN200610165565200612 21CN200611 65565;
Assignee: LENOVO BEIJING CO ;
Assignee[std]: LENOVO BEIJING CO LTD ;
Inventor: LI WENJIE ;
Inventor[std]: WENJIE LI ;
International class[8]: H04N7/26; H04N7/26; H04N7/32; H04N7/32;
Family members: CN100571390 C CN101207810 A
Title: Method and apparatus for fast mode selecting of H264 video coding

Title: 一种H264视频编码快速模式选择方法和装置

Abstract:
The invention discloses a quick mode selection method and a device of an H264 video coding. The method comprises the selection of a mode in frames and the selection of the mode between the frames, wherein, for the selection of the mode in the frames, a prediction method with different fineness is chosen according to the dramatic degree of the image information changes in the prior macro block, thereby decreasing the number of the model to be chosen and confirming the final selection mode through the method of code rate-distortion factor optimizing; for the selection of the mode between the frames, the sum of the absolute difference of the corresponding pixel point of the corresponding macro block of the prior frame and the reference frame is used as a dividing basis to divide the macro block into different fineness and to correspondingly choose a prediction method with different fineness for decreasing the number of the model to be chosen and confirming the final selection mode through the method of code rate-distortion factor optimizing. The invention can decrease the number of the model to be chosen and solve the application limited problems caused by over-complicated code, over-loaded operation and over time consumption in the prior art.

Abstract:

本发明公开了一种H264视频编码快速模式选择方法和装置，方法包括：对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程度来选择不同精细度的预测方式，以减少候选模式的数量，并通过"码率—失真度优化"的方式确定最后的选定模式；对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式以减少候选模式的数量，并通过"码率—失真度优化"的方式确定最后的选定模式。本发明

The present invention discloses a H264 video coding fast mode selection method and device, the method including: intra mode, change the intensity of the internal image information according to the current macroblock to select the prediction of different fineness in order to reduce the candidate mode by the number of bit rate - distortion optimization "to determine the final selected model; for the inter mode, the current frame and corresponding to the reference frame macroblock corresponding pixel absolute

可以减少候选模式的数量，解决现有技术中因编码复杂度过高、运算量过大、耗时过多而造成的应用限制的问题。

difference and as a division in accordance with, macroblock is divided into different fineness corresponds to select the prediction of different fineness in order to reduce the number of candidate patterns, and through the bit rate - distortion optimization "to determine the final selected model. The present invention can reduce the number of candidate patterns, coding complexity in the existing technology is too high, excessive computing, consuming too much of the application constraints.

Claims:

1. 一种H264视频编码快速模式选择方法，其特征在于，包括如下步骤： 步骤一、对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程 度来选择不同精细度的预测方式，以减少候选模式的数量，并通过“码率一 失真度优化”的方式确定最后的选定模式； 步骤二、对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素 点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同 精细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的 方式确定最后的选定模式。

1. a H264 video coding fast mode selection method, characterized in that includes the following steps: Step 1, the intra mode, change the intensity of the current the macroblock internal image information to select the prediction of different fineness way to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; Step 2 for the inter mode, the current frame and reference frame corresponding to the macro block corresponding to as absolute difference and as a division basis, the macroblock is divided into different fineness and corresponds to choose a different fineness prediction to reduce the number of candidate patterns, and through the bit rate a distortion optimization "way of the prime point of determining the final the selected mode.

2.根据权利要求1所述的方法，其特征在于，在所述步骤二中，在选择 不同精细度的预测方式后，还通过宏块的帧间运动的方向信息来减少候选模式的数量。

2 The method according to claim 1, characterized in that, in as described in Step 2, choose a different forecast of finesse, but also by the direction information of the macroblocks of the inter-frame movement to reduce the candidate patterns. number.

3.根据权利要求1或2所述的方法，其特征在于，所述步骤二包括： 步骤a，根据图像信息或用户输入，确定全局预测门限值TV和局部预测 门限值PV; 步骤b,将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤c，对目标宏块进行16X16像素大小的全搜索，找到最佳的匹配块， 将目标宏块和最佳匹配块进行比较，得到目标宏块的全局对应象素点的绝对 差之和T-SAD; 步骤d，判断T-SAD是否小于TV,是则执行步骤e,否则执行步骤g; 步骤e,说明16X16像素大小的全搜索已经可以满足图像质量的要求， 对该目标宏块进行粗精度的预测方式，得到对应的候选模式，然后在16X16 像素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选 定模式，结束帧间模式的选择流程； 步骤g，说明16X16像素大小的全搜索并不能满足图像质量的要求，对 该目标宏块进行细精度的预测方式，得到对应的候选模式，然后结合所述候 选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间模式的选择流程。

3. method according to claim 1 or 2 above, characterized in that, as described in Step 2: Step a, according to the image data or user input to determine the global prediction threshold TV and local predict threshold PV; steps b, the image of each frame is divided into 16X16 pixel size as the basic coding unit macroblock; step c, the target macroblock 16X16 pixel size of the search, to find the best matching block, the target macroblock good matching block were compared to obtain the corresponding pixel in the target macroblock global absolute difference and T-the SAD; step d, to determine T-SAD is less than TV, is to step e, otherwise to step g; step e 16X16 pixel size of the search has to meet the image quality requirements, the rough accuracy of the target macroblock prediction mode, the corresponding candidate patterns described in the candidate model, and then in the 16X16 pixel precision,

4.根据权利要求3所述的方法，其特征在于，在所述步骤e中，所述粗 精度的预测方式包括：将目标宏块划分为能体现方向信息的两个16X8子块 和两个8X16子块，计算四个子块的对应象素点的绝对差之和SAD，并选出 两个16X8子块中的SAD最大

值MAX16X8和两个8X 16子块中的SAD最 大值MAX8X16,选取MAX16X8和MAX8X16中数值较小的所对应的候 选模式。

5.根据权利要求3所述的方法，其特征在于，所述步骤g包括： 步骤gl，将当前宏块进一步划分为4个8X8像素大小的子块，并在全 搜索的基础上进行局部范围的搜索找到各子块的最佳匹配块，将目标子块和 最佳匹配块进行比较，得到该目标子块的局部对应象素点的绝对差之和 P-SAD； 步骤g2,判断P-SAD是否小于PV，是则执行步骤g3，否则执行步骤g4; 步骤g3，说明8X8像素大小的搜索已经可以满足图像质量的要求，将8 X 8像素大小的目标子块划分为能体现方向信息的两个8 X 4子块和两个4X 8 子块，计算四个子块的对应象素点的绝对差之和SAD，并选出两个8X4子 块中的SAD最大值MAX8 X4和两个4X 8子块中的SAD最大值MAX4X 8， 选取MAX8X4和MAX4X8中数值小的所对应的候选模式，然后在8X8像 素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选定 模式，结束帧间的模式选择流程； 步骤g4,说明8X8像素大小的搜索并不能满足图像质量的要求，所以 只能对该目标子块采用精细的4X4预测方式，进行“码率一失真度优化”， 得到最后的选定模式，结束帧间的模式选择流程。

6.根据权利要求1所述的方法，其特征在于，在所述步骤一中，在选择 不同精细度的预测方式后，还通过当前宏块的方向矢量信息来减少候选模式的数量。

7.根据权利要求1或6所述的方法，其特征在于，所述步骤一包括： 步骤A,根据图像信息或用户输入，确定预测门限值VH; 步骤B，将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C,计算目标宏块内部图像信息的变化激烈程度VT; 步骤D，判断VT的值是否大于VH，是则执行步骤E，否则执行步骤F 步骤E,说明宏块内部图像信息的变化剧烈，采用细精度的4X4预测方式，计算当前4X4像素大小的子块的局部方向矢量P-DV，根据方向矢量的 信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码率—失真度优化”，得到最后的选定模式，结束帧内的模式选择流程； 步骤F，说明宏块内部图像信息的变化平缓，采用粗精度的16X16预测 方式，计算当前16X16像素大小的宏块的全局方向矢量T-DV，根据方向矢 量的信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码 率一失真度优化”，得到最后的选定模式，结束帧内的模式选择流程。

8.一种H264视频编码快速模式选择装置，其特征在于，包括： 帧内模式选择模块，用于：根据当前宏块内部图像信息的变化剧烈程度 来选择不同精细度的预测方式，以减少候选模式的数量，并通

combined with code a rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode; step g, 16X16 pixel size of the search and can not meet the requirements of the image quality of the target macroblock fine accuracy of the forecast way, the corresponding candidate patterns, and then combined with the candidate patterns described in a bit rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode.

4 The method according to claim 3, characterized in that the rough accuracy of the forecast include: the target macroblock is divided into to reflect the direction information of the two 16X8 sub-block in said step e and two 8X16 sub-block, calculate the four sub-blocks of the corresponding pixel absolute difference and the SAD, and elected two 16X8 sub-block, the SAD maximum MAX16X8 and two 8X 16 sub-blocks in the SAD maximum MAX8X16 smaller the select MAX16X8 and MAX8X16 value corresponding candidate patterns.

5. method according to claim 3, characterized in that, as described in Step g: Step gl, the current macroblock is further divided into four sub-block size 8X8 pixels and on the basis of the whole search The local scope of a search to find the best match for each sub-block block, the target sub-block and the best matching block, get the target sub-block local corresponding pixel absolute difference and P-the SAD; step g2, to determine P-SAD is less than the PV, then perform step g3, otherwise steps g4 and; step g3, 8X8 pixel size of the search has to meet the requirements of the image quality will target sub-block size of 8 x 8 pixels is divided into to reflects the directional information of two 8 x 4 sub-blocks and two 4X 8 sub-block to calculate the four sub-blocks of the corresponding pixel absolute difference and the SAD, and elected two 8X4 sub-block SAD maximum value MAX8 X4 select the MAX8X4 and MAX4X8 values in the corresponding candidate model, then the combination of 8X8 pixel precision and two 4X 8 sub-blocks in the SAD maximum value MAX4X candidate patterns described in the "bit rate a distortion optimization "to get the last selected mode, the end of the interframe the mode selection process; step g4 and 8X8 pixel size of the search does not meet the requirements of image quality, so only the target sub-block fine 4X4 prediction methods, "rate distortion degree of optimization, the last selected mode, the end of the interframe the mode selection process.

6. The method according to claim 1,

过“码率一失 真度优化”的方式确定最后的选定模式； 帧间模式选择模块，用于：将当前帧和参考帧的对应宏块的对应象素点 的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同精 细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的方 式确定最后的选定模式。

characterized in that, to reduce the number of candidate patterns described in step one, choose a different forecast of finesse, but also by the current macroblock direction vector information.

7. The method according to claim 1 or 6 as described, characterized in that the steps include: the step A, in accordance with image information or user input to determine the predictive threshold the VH; Step B, the image of each frame is divided into 16X16 pixel size as the basic coding unit macroblock; step C, calculate the internal target macroblock image information changes the intensity of the VT; Step D, to determine the value of the VT is greater than VH, is to step E, otherwise implementation of Step F Step E, the rapid change of the macroblock internal image information, the fine precision 4X4 forecast to calculate the current 4X4 pixel size of the local direction of the sub-block vector P-DV, under the direction vector of the information obtained is most likely to two candidate model, then the candidate with the DC mode to "rate a distortion optimization, the last selected mode, the end of the intra mode selection process; Step F, indicating moderate changes in the macro block of image information within 16X16 prediction mode using coarse accuracy, calculate the current size of 16X16 pixels the macroblock global direction vector T-DV, according to the information of the direction vector draw the two most likely candidate mode, then the candidate with the DC mode "bit rate a distortion optimization ", the last selected mode, the end of the frame of the model selection process.

8. - kinds H264 video encoder fast mode to select a device, characterized in that, including: intra mode decision module, used for: changes in the intensity of the internal image information according to the current macroblock to select the prediction of different fineness to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; inter mode decision module for: the current frame and corresponding to the reference frame macroblock corresponding pixel the absolute difference and as a division basis, the macroblock is divided into different fineness and corresponds to choose a different fineness prediction to reduce the number of candidate patterns, and through the bit rate a distortion optimization "way to determine the final election given pattern.

Description:

一种H264视频编码快速模式选择方法和装置 技术领域 本发明涉及视频编码模式的选择方法，特别是涉及一种H264视频编码 快速模式选择方法和装置。

背景技术 同人们熟悉的H.263相比，H.264标准下的视频图像质量更好，同时编码 效率可以提高50%以上，并且具有更好的容错能力，这使得它在未来的视频 通信领域具有广泛的应用前景。然而现有技术中H.264/AVC编码技术是有缺 陷的。这是因为，依据H.264/AVC中码率失真度优化（RDO)的定义，对视 频序列进行编码时，对帧图像必须使用所有的预测模式各测试一遍，然后找到一个从RDO角度来讲最佳的预测模式，使得编码计算的复杂度大幅增加， 增大编码耗时，使得H.264/AVC的实时性变差。

因此，尽管率失真优化可以带来非常好的编码性能，但是对于编码时所 有模式的搜索消耗大量时间，这使得H.264编码器很难用于实时业务。很多快速的运动估计和模式决定的方法被提出用于改善这个情况，比如2004年多 媒体国际会议（ICME2004)中发表的《帧内模式预测》，以及2004年图像处 理国际会议中发表的《H.264中低复杂度的宏块模式预测》和《H.264中自适 应门限值的帧间预测模式选择》中，提出了利用门限值提前终止的算法，用 来减少对于候选预测模式的搜索。

然而这些快速实现的模式选择方法依然有缺陷，主要体现在对于优选的 预测模式的选择仍然过于复杂，需要进行相对较多的计算。

发明内容 本发明的目的在于提供一种H264视频编码快速模式选择方法和装置， 解决现有技术中因编码复杂度过高、运算量过大、耗时过多而造成的应用限 制的问题。

为了实现上述目的，本发明提供了一种H264视频编码快速模式选择方 法，其中，包括如下步骤：步骤一、对于帧内模式选择，根据当前宏块内部图像信息的变化剧烈程 度来选择不同精细度的预测方式，以减少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模式； 步骤二、对于帧间模式选择，将当前帧和参考帧的对应宏块的对应象素 点的绝对差之和作为划分依据，把宏块划分为不同的精细度并对应选择不同 精细度的预测方式以减少候选模式的数量，并通过“码率一失真度优化”的 方式确定最后的选定模式。

上述的方法，其中，在所述步骤二中，在选择不同精细度的预测方式后， 还通过宏块的帧间运动的方向信息来减少候选模式的数量。

上述的方法，其中，所述步骤二包括： 步骤a,根据图像信息或用户输入，确定全局预测门限值TV

The fast mode of the

a H264 video encoding method and device technology of the present invention relates to video coding mode selection method, especially the fast mode selection method and apparatus involving an H264 video encoder. background with people familiar with the H.263, the H.264 standard video image quality better, while the coding efficiency can be improved by more than 50 percent, and has better fault tolerance which makes it in the future the field of video communications has wide application prospect. However, the existing technology in H.264/AVC encoding technology is flawed. This is because, when the video sequence is encoded in accordance with the definition of H.264/AVC rate distortion optimization (RDO), the frame must use all of the forecasting model test again, then find one in terms of the angle from the RDO the best forecasting model, making the coding complexity of computing a substantial increase, increasing the encoding time-consuming, making H.264/AVC real-time variation.

Therefore, although the rate-distortion optimization can be a very good coding performance, but time-consuming search for the encoding mode, which allows the H.264 encoder is difficult for real-time business. A lot of fast motion estimation and mode decision methods have been proposed for the improvement of the situation, such as Multimedia International Conference 2004 (ICME2004), intra mode prediction, as well as published in the International Conference on Image Processing 2004 H. 264 low complexity macroblock mode prediction and inter prediction modes of H.264 adaptive threshold select "proposed threshold of early termination algorithm used to reduce the search for candidate forecasting models.

However, the mode selection method quickly realize there are still defects, mainly reflected in the choice of the optimal prediction model is still too complex, the need for relatively large computing. contents of the invention of the present invention aims to a H264 video coding fast mode selection methods and devices, existing technologies due to coding complexity is too high, operator overload, time-consuming too much caused by the application of restrictions problem.

To achieve these goals, the present invention provides a H264 video coding fast mode selection method, which includes the following steps: Step 1, the intra mode, to select on the basis of the current macroblock internal changes in the

和局部预测 门限值PV; 步骤b,将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C，对目标宏块进行16X16像素大小的全搜索，找到最佳的匹配块， 将目标宏块和最佳匹配块进行比较，得到目标宏块的全局对应象素点的绝对 差之和T-SAD; 步骤d,判断T-SAD是否小于TV，是则执行步骤e，否则执行步骤g; 步骤e，说明16X16像素大小的全搜索已经可以满足图像质量的要求， 对该目标宏块进行粗精度的预测方式，得到对应的候选模式，然后在16X16 像素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选 定模式，结束帧间模式的选择流程； 步骤g,说明16X16像素大小的全搜索并不能满足图像质量的要求，对 该目标宏块进行细精度的预测方式，得到对应的候选模式，然后结合所述候 选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间模式的选 择流程。

上述的方法，其中，在所述步骤e中，所述粗精度的预测方式包括：将 目标宏块划分为能体现方向信息的两个16X8子块和两个8X16子块，计算 四个子块的对应象素点的绝对差之和SAD，并选出两个16X8子块中的SAD 最大值MAX16X8和两个8X16子块中的SAD最大值MAX8X16，选取 MAX16X8和MAX8X16中数值较小的所对应的候选模式。

上述的方法，其中，所述步骤g包括： 步骤gl,将当前宏块进一步划分为4个8X8像素大小的子块，并在全 搜索的基础上进行局部范围的搜索找到各子块的最佳匹配块，将目标子块和 最佳匹配块进行比较，得到该目标子块的局部对应象素点的绝对差之和 P-SAD； 步骤g2，判断P-SAD是否小于PV，是则执行步骤g3，否则执行步骤g4; 步骤g3，说明8X8像素大小的搜索已经可以满足图像质量的要求，将8 X 8像素大小的目标子块划分为能体现方向信息的两个8 X 4子块和两个4 X 8 子块，计算四个子块的对应象素点的绝对差之和SAD，并选出两个8X4子 块中的SAD最大值MAX8X4和两个4X8子块中的SAD最大值MAX4X 8， 选取MAX8X4和MAX4X8中数值小的所对应的候选模式，然后在8X8像 素的精细度下结合所述候选模式进行“码率一失真度优化”，得到最后的选定模式，结束帧间的模式选择流程； 步骤g4，说明8X8像素大小的搜索并不能满足图像质量的要求，所以 只能对该目标子块采用精细的4X4预测方式，进行“码率一失真度优化”， 得到最后的选定模式，结束帧间的模式选择流程。

上述的方法，其中，在所述步骤一中，在选择不同精细度的预测方式后， 还通过当前宏块的方向矢量信息来减少候选模式的数量。

上述的方法，其中，所述步骤一包括： 步骤A,根据图像信息或用户输入，确定预测门限值VH; 步骤B，将图像的每一帧划分成作为基本编码单元的16X16像素大小的 宏块； 步骤C,计算目标宏块内

intensity of the image information different fineness prediction mode, in order to reduce the number of candidate patterns, and through "rate distortion optimization" to determine the final selected model; Step 2 for the inter mode, the current frame and reference frame corresponding to macroblock corresponding pixel absolute difference and division basis, the macroblock is divided into different granularity and corresponding select different fineness prediction to reduce the number of candidate patterns, and through "rate distortion degree of optimization "way to determine the selected mode. the above method, in which the steps described in two, choose a different forecast of finesse, but also by the direction information of the macroblocks of the inter-frame movement to reduce the number of candidate patterns. the above method, wherein Step 2 include: steps a, according to the image data or user input to determine the global prediction threshold TV and local predict threshold PV; step b, the image of each frame is divided into as the basic coding unit of 16X16 pixel size of the macro block; step C, the target macroblock 16X16 pixel size of the search, to find the best matching block, the target macroblock and the best matching block, get the target macroblock corresponds to the global pixel absolute difference and T-the SAD; step d, to determine T-SAD is less than TV, is to step e, otherwise to step g; to step e, 16X16 pixel size of the search have to meet image quality requirements of the target macroblock coarse accuracy of prediction, the corresponding candidate patterns described candidate model "rate distortion optimization", and then combined 16X16 pixel precision, and the last election given mode, the end of the selection process of the inter-frame mode; step g 16X16 pixel size of the search and can not meet the requirements of the image quality, fine accuracy of the prediction in the target macroblock corresponding candidate patterns, and then combined with the candidate patterns described a bit rate distortion optimization ", the last selected mode, the end of the selection process of the inter-frame mode. rough accuracy of prediction methods include:

the method described above, which, in said step e, as described in the target macroblock is divided into two 16X8 sub reflect the direction of information blocks and two 8X16 sub-block to calculate the four The absolute difference of the corresponding sub-block pixel SAD, and elected two 16X8 sub-block, the SAD maximum MAX16X8 and two 8X16 sub-block in the SAD maximum MAX8X16, the select MAX16X8 and MAX8X16 value smaller corresponding to the candidate mode. the above method, wherein step g including: steps to gl, the current macroblock is

部图像信息的变化激烈程度VT；步骤D,判断VT的值是否大于VH,是则执行步骤E，否则执行步骤F 步骤E,说明宏块内部图像信息的变化剧烈，采用细精度的4X4预测方 式，计算当前4X4像素大小的子块的局部方向矢量P-DV，根据方向矢量的 信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码率一 失真度优化”，得到最后的选定模式，结束帧内的模式选择流程； 步骤F,说明宏块内部图像信息的变化平缓，采用粗精度的16X16预测 方式，计算当前16X16像素大小的宏块的全局方向矢量T-DV，根据方向矢 量的信息得出最有可能的两种候选模式，然后同DC候选模式一起进行“码 率一失真度优化”，得到最后的选定模式，结束帧内的模式选择流程。

为了实现本发明的目的，本发明还提供了一种H264视频编码快速模式 选择装置，其中，包括：帧内模式选择模块，用于：根据当前宏块内部图像信息的变化剧烈程度来选择不同精细度的预测方式，以减少候选模式的数量， 并通过“码率一失真度优化”的方式确定最后的选定模式；帧间模式选择模 块，用于：将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为划 分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式以减 少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模 式。

本发明的优点在于： 1)本发明根据宏块内部图像信息的变化剧烈程度来选择不同精细度的预 测方式，可以减少帧内模式选择中的候选模式的数量，进而减少了运算量。

2)本发明将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为 划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式，从而减少了帧间模式选择中的候选模式的数量，进而减少了运算量。

3)本发明还通过宏块的帧间运动的方向信息来减少帧间模式选择中的候 选模式的数量。

4)本发明还通过当前宏块的方向矢量信息来减少帧内模式选择中的候选 模式的数量。

5)本发明不但有帧内的模式选择还有帧间的模式选择，本发明应用于 H.264/AVC标准，可以有效的减少4x4大小的候选模式的数目，是一种能有 效的减少运算量和运算时间的视频编码方法。

附图说明 图1为本发明方法的步骤流程图； 图2为本发明提供的自适应帧间预测模式选择算法流程图； 图3为本发明中宏块的四个参数的含义示意图； 图4为本发明提供的自适应帧内预测模式选择算法流程图。

具体实施方式 本发明提出了一种H264视频编码快

further divided into four sub-block size 8X8 pixels, and a full search based on local-wide search to find all the sub-block the best matching block, the target sub-block and the best matching blocks were compared to obtain the target sub-block of the local corresponding pixel absolute difference and P-the SAD; step g2, to determine P-the SAD is less than the PV perform the steps g3, otherwise to step g4 and; step g3, 8X8 pixel size of the search has to meet the requirements of the image quality will be 8 X 8 pixel size of the target sub-block partition in order to reflect the direction information of the two X-4 sub-block and two 4 X 8 sub-blocks, and calculate the four sub-blocks corresponding pixel absolute difference, SAD, and elected in the SAD maximum of two 8X4 sub-block the MAX8X4 and two 4X8 sub-block SAD the maximum MAX4X 8, corresponding to the candidate to select the MAX8X4 and MAX4X8 numerical mode, and then combined in the 8X8 pixel precision candidate patterns described in the "rate distortion optimization", the last selected mode, the end of inter-frame mode selection process; step g4 and 8X8 pixel size of the search and can not meet the requirements of the image quality, so only the target sub-block fine 4X4 prediction methods, the bit rate is a distortion optimization ", to get the last selected mode, the end of the inter-frame mode selection process. the method described above, which, in the steps described in a, in the choice of different fineness forecast, but also by the current macroblock direction vector information to reduce the number of candidate patterns. the method described above, which, as described in step one, including: Step A predictive threshold based on image information or user input to determine the VH; Step B, the image of each frame is divided into 16X16 pixel size as the basic coding unit the macro block; step C, the calculation of the target macroblock internal image information change the intensity of the VT; Step D, to determine the value of the VT is greater than VH, is to step E, otherwise step E step F, indicating the macroblock internal image rapid change of information, fine precision 4x4 prediction mode, calculate the current 4X4 pixel size of the local orientation of the sub-block vector P-DV, the two most likely candidate mode information obtained under the direction vector, and then with the DC candidate mode "with a bit rate distortion optimization", the last selected mode, the end of the frame of the model selection process; step F, indicating gentle the macroblock internal image information changes, 16X16 prediction mode using coarse accuracy to calculate the current 16X16 pixel size of the macroblock of the global direction vector T-DV to draw the two most likely candidate mode, according to the information of the direction vector, then the candidate with the

速模式选择方法，图1为本发明方 法的步骤流程图，如图，本发明分为帧内算法流程101和帧间算法流程102 两个部分。在快速帧间模式选择算法中，本发明使用当前帧和参考帧对应宏 块的对应象素点的绝对差之和（SAD，Sum of Absolute Difference)作为相关 性的判断，把宏块划分为不同精细度的模式，然后根据宏块的帧间运动的方向信息，进一步获得优选的预测模式，最后在优选的预测模式中使用码率一 失真度优化（RDO），从而确定最佳模式。而对于快速帧内模式选择算法，本 发明可以首先根据当前宏块内部图像信息的变化剧烈来选择不同精细度的预 测模式，并且可以根据当前宏块的方向矢量信息，从中选择出最有可能的几 种预测模式，然后对这些优选出来的预测模式进行“码率一失真度优化” (RDO)算法选择出最佳模式。

图2为本发明提供的自适应帧间预测模式选择算法流程图，如图，帧间 模式选择算法主要包括以下步骤。步骤201.根据图像信息或用户输入，确定全局预测门限值TV和局部预 测门限值PV; 步骤202.将图像的每一帧划分成为基本的编码单元，即 16X16像素大 小的宏块； 步骤203.对其中的目标宏块进行16X16像素大小的全搜索，找到最佳 的匹配块，将目标宏块和最佳匹配块进行比较，得到该目标宏块的全局SAD 值（T-SAD); 步骤204.判断T-SAD的值是否小于TY，是则执行步骤205，否则执行 步骤207; 步骤205,如果T-SAD的值小于TV，说明16X 16像素大小的全搜索已 经可以满足图像质量的要求，所以只需要接着对该目标宏块进行粗精度的模 式选择，即从16X16, 16X8，8X16模式中选择一个最合适的； 步骤206，对于目标宏块粗精度的模式选择，首先按照图3所示分别计 算目标宏块的上半部分、下半部分、左半部分和右半部分所对应的四个SAD 值，并分别得出在上/下、左/右这两种模式下的SAD的最大值，表示为MAX16 X8和MAX8X16，然后比较并选出MAX16X8和MAX8X 16中较小的所对 应的模式，说明在该模式下如果进行运动搜索和运动补偿会取得更好的效果。

然后就选择16X16和该模式进行码率一失真度优化（RDO)算法，选择出最 佳模式，转步骤213。

步骤207,如果T-SAD的值大于TV，说明16X 16像素大小的全搜索并 不能满足图像质量的要求，所以还需要接着对该目标宏块进行更细精度的模 式选择，即从8X8，4X8，8X4, 4X4模式中选择一个最合适的； 步骤208，对于目标宏块细精度的模式选择，将当前宏块进一步划分为4 个8X8像素大小的子块； 步骤209,在全搜索的基础上进行小范围的搜索找到各自最佳的匹配块， 将目标子块和最佳匹配块进行比较，得到该目标子块的局部SAD值(P-SAD); 步骤210,判断P-SAD的值是否小于PV，是则执行步骤211，否则执行 步骤212; 步骤211，如果P-SAD的值小于PV，说明8X8像素大小的搜索已经可 以满足图像质量的要求，所以只需要接着

DC mode to "rate a distortion optimization, the final selected mode, the end of the frame of the model selection process.

In order to achieve the purpose of the present invention, the present invention also provides a H264 video coding fast mode selection device, which include: intra mode decision module, used for: changes in intensity to the internal image information according to the current macroblock select the prediction of different fineness, in order to reduce the number of candidate patterns, and through "rate is a distortion optimization" to determine the final selected model; inter mode decision module for: the current frame and reference frame corresponds to the absolute difference and as a division of the corresponding pixel in the macroblock basis, the macroblock is divided into different fineness corresponds to select different fineness prediction to reduce the number of candidate patterns, and through the bit rate distortion The Optimization determine the last selected mode. the advantages of the present invention in

is: 1) The present invention is to select the changes in intensity according to the macro block of image information within the forecast of different fineness, can reduce the number of candidate patterns in the intra mode, thereby reducing the amount of computation.

2) of the present invention to the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference and as a division basis, the macroblock is divided into different fineness corresponds to choose a different fineness prediction, thereby reducing the number of candidate patterns in the inter mode, thereby reducing the amount of computation.

3) The present invention also interframe macroblock motion direction information to reduce the number of candidate patterns in the inter mode.

4) of the present invention through the current macroblock direction vector information to reduce the number of candidate patterns in the intra mode.

5) of the present invention not only intra mode selection in the inter-frame mode selection, the present invention is applied to the H.264/AVC standard, can effectively reduce the 4x4 size of the number of candidate patterns is an effective video coding method to reduce the computational

对该目标子块进行粗精度的模式选 择，即从8X8, 8X4, 4X8模式中选择一个最合适的； 对于目标子块粗精度的模式选择，按照图3所示分别计算目标子块（8 X8像素大小）的上半部分、下半部分、左半部分和右半部分所对应的四个 SAD值，并分别得出在上/下、左/右这两种模式下的SAD的最大值，表示为 MAX8X4和MAX4X8，然后比较并选出MAX8X4和MAX4X8中较小的 所对应的模式，说明在该模式下如果进行运动搜索和运动补偿会取得更好的 效果。然后就选择8X8和该模式进行码率一失真度优化（RDO)算法，选择 出最佳模式，转步骤213。

步骤212.如果P-SAD的值大于PV,说明8X8像素大小的搜索并不能 满足图像质量的要求。所以只能对该目标子块采用最精细的模式，即4X4模 式，转步骤213。

步骤213,完成目标宏块的帧间模式选择。

图4为本发明提供的自适应帧内预测模式选择算法流程图。如图，快速 帧内模式选择算法主要包括以下步骤： 步骤401.根据图像信息或用户输入，确定预测门限值VH; 步骤402.将图像的每一帧划分成为基本的编码单元，即16X16像素大 小的宏块； 步骤403.计算目标宏块内部图像信息的变化激烈程度，用VT表示； 步骤404.判断VT的值是否大于VH,是则执行步骤405，否则执行步 骤 408; 步骤405，如果VT的值大于VH,说明宏块内部图像信息的变化剧烈， 就需要采用细精度的预测模式，即4X4预测模式； 步骤406.如果采用细精度的预测模式，即4X4预测模式，就计算当前 子块（4X4像素大小）的局部方向矢量P-DV; 步骤407.根据方向矢量的信息得出最有可能的2种候选预测模式，转 步骤411; 步骤408.如果VT的值小于VH，说明宏块内部图像信息的变化不大， 就可以采用粗精度的预测模式，即16X16预测模式； 步骤409.如果采用粗精度的预测模式，即16X16预测模式，就计算当 前宏块（16X16像素大小）的全局方向矢量T-DV; 步骤410，根据方向矢量的信息得出最有可能的2种候选预测模式； 步骤411.将候选预测模式同DC预测模式（DC模式是帧内预测4x4大 小的9种预测模式中的一种，DC模式下4x4块的所有像素点的预测值是相同 的）一起进行码率一失真度优化（RDO)算法，选择出最佳模式。

步骤412.完成目标宏块的帧内预测模式选择。

由以上论述可知，本发明可以带来以下技术效果：1)本发明根据宏块内部图像信息的变化剧烈程度来选择不同精细度的预 测方式，可以减少帧内模式选择中的候选模式的数量，进而减少了运算量。

2)本发明将当前帧和参考帧的对应宏块的对应象素点的绝对差之和作为 划分依据，把宏块划分为不同的精细度并对应选择不同精细度的预测方式，

complexity and computation time.

Description of the Drawings Figure 1 step flowchart of the present invention; the present invention to provide adaptive inter-frame prediction mode selection algorithm flowchart; meaning schematic of the present invention the macroblock four parameters; Figure 4 the present invention to provide adaptive intra prediction mode selection algorithm flowchart. specific implementation modalities of the present invention proposed a H264 video coding fast mode selection method, a flowchart of the steps of the present invention shown, the present invention is divided into 102 two frame algorithm processes 101 and interframe algorithm processes parts. Fast inter mode selection algorithm, the present invention uses the corresponding pixel of the current frame and reference frame corresponding to the macroblock of absolute differences (SAD, Sum, of the Absolute Difference,) as a correlation between the judgment of the macroblock divided into different finesse mode, and then under the direction of the macroblocks of the inter-frame movement of information, and further obtain the optimal prediction mode, the bit rate distortion optimization (RDO), the last in the optimal forecasting model, to determine the best mode. For fast intra mode selection algorithm, the present invention can be first based on the rapid change of the current macroblock internal image information to select a different fineness prediction models, and according to the current macroblock direction vector information, choose the most likely to several forecasting models, these preferred forecasting model, the bit rate is a distortion degree of optimization (RDO) algorithm to select the best mode.

adaptive interframe the

Figure 2 the present invention to provide prediction mode selection algorithm flow chart shown, inter mode selection algorithm includes the following steps. ◆ Step 201 based on image information or user input to determine the global prediction threshold TV and local forecast threshold PV; to step 202. image of each frame is divided into the basic coding unit, the size of 16X16 pixel macro block; to step 203. one of the target macroblock 16X16 pixel size of the full search, to find the best matching block, the target macroblock and the best matching block compared to get the target macroblock global SAD value (T-SAD); step 204 to determine the value of the T-SAD is less than the TY is executed Step 205, otherwise perform step 207; step 205, if the T-the SAD value is less than TV,

从而减少了帧间模式选择中的候选模式的数量，进而减少了运算量。

3)本发明还通过宏块的帧间运动的方向信息来减少帧间模式选择中的候 选模式的数量。

4)本发明还通过当前宏块的方向矢量信息来减少帧内模式选择中的候选 模式的数量。

5)本发明不但有帧内的模式选择还有帧间的模式选择，本发明应用于 H.264/AVC标准，可以有效的减少4x4大小的候选模式的数目，是一种能有 效的减少运算量和运算时间的视频编码方法。

以上所述仅是本发明的优选实施方式，应当指出，对于本技术领域的普 通技术人员来说，在不脱离本发明原理的前提下，还可以作出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

^ Pit HI 对于帧内模式选择，根据当前宏块内部图像信 息的变化剧烈程度来选择不同精细度的预测方 式，以减少候选模式的数量，并通过“码率一失真度优化”的方式确定最后的选定模式； 对于帧间模式选择，将当前帧和参考帧的对应 宏块的对应象素点的绝对差之和作为划分依 据，把宏块划分为不同的精细度并对应选择不 同精细度的预测方式以减少候选模式的数量， 并通过“码率一失真度优化”的方式确定最后 的选定模式。

indicating that the search has to meet the requirements of the image quality of 16X 16 pixels in size, so only need then to the target macroblock coarse accuracy mode selection, from the 16X16, 16X8, 8X16 mode, select one of the most appropriate; step 206, select the mode of target macroblock rough accuracy, first shown in Figure 3 were calculated for the upper part of the target macroblock, the lower half of the left part and right part of the SAD value corresponding to the four, and were drawn up / down, left / right mode of Both the maximum value of the SAD, said that for the MAX16 X8 and MAX8X16-and then compare and choose smaller MAX16X8 and MAX8X 16 corresponding mode In this mode, the motion search and motion compensation to achieve better results. and then choose the 16X16 and the mode bit rate to a distortion optimization (RDO) algorithm to select the best mode, turn to step 213. step 207, if the value of the T-SAD than TV, description of the search and can not meet the requirements of the image quality of 16X 16 pixels in size, so you need then the target macroblock finer precision mode selection, ie 8X8, 4X8, 8X4, 4X4 mode, select one of the most appropriate; step 208, the mode selection of the target macroblock fine accuracy, the current macroblock further divided into four sub-block size 8X8 pixels; step 209, in the whole search on the basis of small-scale search to find their best matching block, the target sub-block and the best matching block to get the target sub-block partial SAD value (P-SAD); step 210 to determine the P- SAD value is smaller than the PV, then perform step 211, otherwise perform step 212; step 211, if the value is less than the P-SAD PV 8X8 pixel size of the search has to meet the image quality requirements, we only need then the mode selection of the coarse accuracy of the target sub-block, from the 8X8, 8X4, 4X8 mode, select one of the most appropriate; mode select for the target sub-block coarse accuracy were calculated in accordance with Figure 3 shows the target sub-block (8 X8 pixel size) in the upper half of the bottom half, left part and right part corresponding to the four SAD values were obtained in both modes of the up / down, left / right maximum SAD, the MAX8X4 and MAX4X8 then compare and select the mode the MAX8X4 and MAX4X8 smaller the corresponding instructions in this mode if the motion search and motion compensation to achieve better results. 8X8 and the mode and then choose to carry out the bitrate a distortion optimization (RDO) algorithm to select the best mode, turn to step 213. step 212, if the SAD of the P-value is greater than the PV 8X8 pixel size of the search does not meet the requirements of image quality, so only the target sub-block using the most sophisticated model, the 4X4 mode, go to step 213. steps 213,

the target macroblock inter mode.

the

Figure 4 the present invention to provide adaptive intra prediction mode selection algorithm flowchart. Figure fast intra mode selection algorithm includes the following steps: Step 401 based on image information or user input to determine the forecast threshold the VH; step 402 the image of each frame is divided into the basic coding unit, that is 16X16 pixel size macro block; to step 403. calculate the change in the intensity of the target macroblock internal image information, said in VT; step 404 to determine the value of VT is greater than VH, then perform step 405, otherwise perform step 408; step 405, if VT value is greater than VH and description of intense the macroblock internal image information changes, you need to fine the accuracy of forecasting models, forecasting models namely 4X4; step 406 if the fine accuracy of prediction models, namely 4X4 forecasting model, the purpose of calculating the current sub-block (4x4 pixels in size), the local direction vector P-DV; to step 407. the most likely candidate forecasting models drawn under the direction vector information, go to step 411; to step 408. If the value of VT is less than the VH that little change in the macroblock internal image information, you can use the rough accuracy of the forecasting model, ie, 16X16 prediction mode; to step 409. rough accuracy of the prediction model, that is 16X16 prediction mode of calculating the current macroblock (16X16 pixel size) global direction vector T-DV; step 410, the most likely candidate prediction mode information obtained under the direction vector; to step 411. the candidate forecasting model with the DC prediction mode (DC mode is the intra prediction 4x4 size a nine kinds of forecasting models, the predictive value of all pixels in the DC mode, the 4x4 block is the same) with one of bit-rate distortion optimization (RDO) algorithm to select the best mode. step 412. target macroblock intra prediction mode selection. discourse we can see from the above, the present invention can have the following effects: 1) The present invention is to select the intensity changes according to the macroblock internal image information of different fineness prediction can reduce the candidate patterns in the intra mode the number, thereby reducing the amount of computation.

2) of the present invention to the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference and as a division basis, the macroblock is divided into different fineness corresponds to

choose a different fineness prediction, thereby reducing the number of candidate patterns in the inter mode, thereby reducing the amount of computation.

3) The present invention also interframe macroblock motion direction information to reduce the number of candidate patterns in the inter mode.

4) of the present invention through the current macroblock direction vector information to reduce the number of candidate patterns in the intra mode.

5) of the present invention not only intra mode selection in the inter-frame mode selection, the present invention is applied to the H.264/AVC standard, can effectively reduce the 4x4 size of the number of candidate patterns is an effective video coding method to reduce the computational complexity and computation time. above is only preferred embodiments of the present invention, it should be pointed out that the person of ordinary skill for this technology, not out of principle of this invention, can also make a number of improvements and polish, these improvements and The polish should also be regarded as the scope of protection of the present invention. of Pit HI for intra mode selection, changes in the intensity of the internal image information according to the current macroblock to choose a different forecast of finesse in order to reduce the number of candidate patterns, and by "bit-rate distortion optimization" determine the selected mode; for the inter mode, the current frame and corresponding to the reference frame macroblock corresponding pixel absolute difference, and as divided in accordance with the macroblock is divided into different fineness and the corresponding select different fineness prediction to reduce the number of candidate patterns, and by the "bit rate is a distortion optimization" to determine the final selected model.

Cited by: CN101783951 A ; WO12068817 A1 ; WO12088806 A1 ;

PatentOrder

Publication number: CN101340581 A

Publication country: CHINA

Publication type: UNEXAMINED APPLIC. OPEN TO PUBLIC INSPECTION

Publication date: 20090107

Application number: CN200810041733

Application date: 20080815

Priority: CN200810041733320080815CN2008141733;

Assignee: SHANGHAI FULLHAN MICROELECTRONICS CO ;

Assignee[std]: SHANGHAI FULLHAN MICROELECTRON ;

Inventor: ZHU YUE ; CHEN XIAOCHUN ; LI XIAOFEI ;

Inventor[std]: YUE ZHU ; XIAOCHUN CHEN ; XIAOFEI LI ;

International class[8]: H04N19/593;

Family members: CN101340581 A CN101340581 B

Title: INTRA-FRAME PREDICTION MODE SELECTION APPARATUS BASED ON M ALGORITHM

Title: 一种基于M算法的帧内预测模式选择装置

Abstract:

The invention discloses a method for selecting an intraframe prediction mode based on an M algorithm, and a device thereof. The device comprises an intraframe prediction device and a block-mode cost accounting and block reconfiguration device which are connected in sequence. The device of the invention is characterized by also comprising a data management device which is used for storing buffer data of reconstructed images and managing information of adjacent pixels and paths, and a path cost accounting and prediction mode selecting device. The method for selecting the intraframe prediction mode, which is disclosed by the invention, realizes a multi-path global search of the intraframe prediction mode of the block so as to reduce the possibility that the search with the best mode of the intraframe prediction mode of the block falls into the local minimum.

Abstract:

本发明公开了一种基于M算法的帧内预测模式选择方法及其装置，该装置包括依次连接的帧内预测装置、块模式代价计算与块重构装置，其特征在于：还包括用于存储重构图像缓冲数据与管理相邻像素与路径信息的数据管理装置、路

The present invention discloses a method based on the M intra prediction mode selection method and apparatus, the apparatus comprises intra-forecast means are sequentially connected, the cost calculation block mode and block

径代价计算与预测模式选择装置。本发明的帧内预测模式选择方法通过应用M算法，实现块帧内预测模式的多路径全局搜索，从而降低块帧内预测模式最佳模式搜索陷入局部极小值的可能性。

reconstruction device, characterized by: further comprising means for storing reconstructed image data and management data buffer management unit pixels adjacent path information, path cost calculation and prediction mode selection means. Intra-prediction mode selection method of the present invention by applying the M algorithm, block intra prediction mode multipath global search, thereby reducing intra-prediction mode optimal block mode search into the possibility of local minima.

Claims:

1、一种基于M算法的帧内预测模式选择装置，包括依次连接的帧内预 测装置（2)、块模式代价计算与块重构装置（3),其特征在于：还包括用于 存储重构图像缓冲数据与管理相邻像素与路径信息的数据管理装置（1)和 路径代价计算与预测模式选择装置（4);其中， 数据管理装置（1)第一输入端与路径代价计算与预测模式选择装置（4）的输出端连接，接收块预测模式信息和源路径号，其第二偷入端与块模式代 价计算与块重构装置（3)的第三输出端相连，接收重构图像数据；其第一 输出端榆出相邻像素至帧内预测装置（2)的第一输入端，其第二输出端输 出相邻预测模式信息至块模式代价计算与块重构装置（3)的第二输入端， 第三输出端输出路径编号至路径代价计算与预测模式选择装置（4)的第三 输入端； 帧内预测装置（2)的第一愉入端与数椐管理装置（1)的第一输出端相 连，接收图像的相邻像素；其第一输出端输出经过计算的预测图像数据至与 块模式代价计算与块重构装置（3)的第一输入端； 块模式代价计算与块重构装置（3)的第一输入端与帧内预测装置（2) 的第一输出端相连，接收预测图像数据，其第二输入端与数据管理装置（1) 的第二输出端相连，接收图像的相邻预测模式信号，其第三输入端接初始编 码图像；块模式代价计算与块重构装置（3)的第一、第二输出端分别输出 模式代价数据、块预测模式信号至路径代价计算与预测模式选择装置（4) 的第一、第二输入端，其第三输出端输出重构图像至数据管理装置（1)的 第二输入端； 路径代价计算与预测模式选摔装置

1, M algorithm based on intra-forecast mode selecting apparatus, comprising sequentially connected intraframe prediction means (2), the cost calculation block mode and block reconstruction device (3), characterized in that: also includes the reconstructed image buffer for storing data and managing data management device pixels and path information (1) adjacent to and path cost calculation and prediction mode selection means (4); wherein the data management device (1) a first input end path cost calculation and prediction mode selection means (4) connected to the output, the reception block prediction mode information and the source path number, a second end and stole into the cost calculation block mode and block reconstruction device (3) in the third an output terminal connected to receive the reconstructed image data; a first output terminal of the adjacent pixels elm to the first input terminal intra prediction means (2), the second output terminal neighboring prediction mode information to calculate the cost of a block pattern a second input terminal block reconstruction device (3), the third output terminal path number to calculate the path cost prediction mode selection means (4) a third input terminal; intra prediction unit (2) first Discovery a first output terminal of the terminal management apparatus and

（4)第一、第二输入端分别与块模 式代价计算与块重构装置（3)的第一、第二输出端对应相连，分别接收模 式代价数据和块预测模式信号，其第三输入端与数据管理装置（1)的第三 输出端相连，接收路径编号数据；其第一输出端块预测模式与源路径号信息 至数据管理装置（1)的第一输入端。

2、如权利要求1所述的一种基于M算法的帧内预测模式选择装置，其 特征在于：所述的帧内预测装置（2)与块模式代价计算与块重构装置（3) 为多路并行连接。

3、如权利要求1所述的一种基于M算法的帧内预测模式选择装置，其 特征在于：所述的数据管理装置（1)存储M条路径的相邻像素与相邻预测 模式。

4、如权利要求1所述的一种基于M算法的帧内预测模式选择装置，其 特征在于：所述的数据管理装置（1)可采用寄存器交换结构、回溯结构或 二者的混合结构实现。

5、如权利要求1或4所述的一种基于M算法的帧内预测模式选择装置， 其特征在于：所述的采用寄存器交换结构的数椐管理装置（1),包括依次连 接的 相邻更新装置，采用寄存器组与选择器网络，用于存储及更新似条路 径的图像数据及相邻块预测模式数据； 相邻缓沖寄存器，具有M组寄存器，分别用于存储对应于M条路径的 来自块模式代价计算与块重构装置（3)第三输出端的重构图像数据，以及 来自路径代价计算与模式选择装置（4)第一输出端的块模式选择数据； 当前路径选择装置，用于在当前路径信息中选择当前路径及其对应的路 径数据；相邻像素与块预测模式选择装置，用于从当前路径中挑选出当前决相邻 像素信息与相邻块预测模式信息。

6、如权利要求1或2所述的一种基于M算法的帧内预测模式选择装置， 其特征在于：所述的帧内预测装置（2)，块模式代价计算与块重构装置（3) 为多路并行连接结构时，每一路帧内预测装置（2)及块模式代价计算与块 重构装置（3)分别具有各自对应的当前路径选择装置与相邻像素及块预测 模式选择装置。

the number noted in (1) is connected to receive the image of the neighboring pixels; a first output terminal thereof via the predicted image data to calculate the cost of computing the block mode and block reconstruction device (3) a first input terminal; block pattern and the cost calculation block reconstruction device (3), a first input terminal and the output terminal of a first intra prediction means (2) is connected to receive the predicted image data, a second data input terminal a second output terminal management apparatus (1) is connected to receive the image of the neighboring prediction mode signal, a third input connected to its initial encoded image; block pattern and the cost calculation block reconstruction device (3) first and second outputs end cost of data output mode, respectively, block prediction mode signal to the path cost calculation and prediction mode selection of the first and second input terminal means (4), the third output terminal to the reconstructed image data management device (1) first second input; path cost calculation and prediction mode selected fall means (4) first and second input terminals of the cost of computing block mode first, second output terminal block reconstruction device (3) is connected to the corresponding, respectively, to receive Consideration third output mode of data and the block prediction mode signal, a third input terminal thereof and the data management device (1) is connected to the reception path number data; a first output terminal and the source path prediction mode of block number information to the data management device (1) a first input terminal.

2, as claimed in claim 1, wherein a selection means based on intra-prediction mode M algorithm, characterized in that: said intra-forecast means (2) and the block mode Consideration Calculation and block reconstruction device (3) for the multiple parallel connections.

7、如权利要求1或4所述的一种基于M算法的帧内预测模式选择装置，其特征在于：所述的采用回溯结构的数据管理装置（1),包括依次连接路径 选择历史缓冲装置，回溯逻辑装置以及相邻缓冲RAM;其中，相邻缓冲RAM,用于存储对应于M条路径的来自块模式代价计算与块 重构装置（3)第三输出端的重构图像数据，以及来自路径代价计算与模式 选择装置（4)第一输出端的块模式选择数据；所述的路径选择历史缓冲装置用于存储和更新路径选择信息，并发送至 回溯逻辑装置；所述的回溯逻辑装置，用于回溯当前块数据的相邻块数据，得到当前块的块相邻像素及块相邻预测模式。

8、如权利要求7所述的一种基于M算法的帧内预测模式选择装置，其 特征在于：所述的回溯逻辑装置，具有一个或多个回溯逻辑单元，用于一次 或多次回溯访问相邻缓冲RAM得到当前块具有的一个或一个以上的相邻块 信息。

9、如权利要求7所述的一种基于M算法的帧内预测模式选择装置，其 特征在于：所述的采用回溯结构的数据管理装置（1)可通过复用方式实现 与多个帧内预测装置、锋模式代价计算与块重构装置之间的数据传输。

10、如权利要求1所述的一种基于M算法的帧内预测模式选择装置，其特征在于：所述的路径代价计算与模式选择装置（4)用于计算块内路径 代价或整宏块代价计算。

11、如权利要求1或10所述的一种基于M算法的帧内预测模式选择装 置，其特征在于：所述的路径代价计算与模式选择装置（4),包括依次连接 的排序装置、路径度量寄存器、最小路径度量选取装置；同时，路径号与块 模式寄存器的输入端与排序装置的一输出端相连，其中所述的排序装置，用于对似条路径的所有延伸路径度量进行排序，所述的路径度量寄存器，具有M组包含若干个寄存单元的寄存器，分 别用于存储M条路径的当前路径度量数据。

12、如权利要求11所述的一种基于M算法的帧内预测模式选择装置，其特征在于：所述的排序装置可采用冒泡排序算

3, as claimed in claim 1, wherein a selection means algorithm based on intra-prediction mode M, wherein: the data management device (1) of the M paths in storage adjacent pixel and the adjacent prediction modes.

4, as claimed in claim 1, wherein a selection means algorithm based on intra-prediction mode M, wherein: the data management device (1), wherein the switch fabric can register, backtracking or mixed fabric achieve both.

5, as claimed in one of claim 1 or 4, wherein the selecting means M algorithm based on intra-prediction mode, wherein: the number noted in the management of the exchange structure preclude the use of register means (1), comprising sequentially connected adjacent update means, using the register bank and a selector network, for storing and updating the image data and the like paths adjacent block prediction mode data; adjacent buffer register, having M groups register, respectively, for storing data corresponding to the M paths from block mode and block reconstruction cost of computing device (3) third output of the reconstructed image data, as well as the cost of computing from the path and mode selection means (4) first output terminal Block mode selection data; the current path selection means for selecting the current path and the corresponding path data in the current path information; adjacent pixel block prediction mode selection means for the selection of the current path from the current decision adjacent pixels information and prediction mode information of adjacent blocks.

6, such as a 1 or claim 2, wherein the selection means based on the algorithm of M intra prediction modes, characterized in that: said intra-forecast means (2), the block the cost calculation mode and block

法、插入排序算法及归并排 序算法对路径度量进行排序。

13、如权利要求1所述的一种基于M算法的帧内预测模式选择装置， 其特征在于：所述的排序装置可实现完全排序或部分排序。

14、如权利要求12所述的一种基于M算法的帧内预测模式选择装置， 其特征在于：所述的排序装置可采用比较网络或排序网络实现对数据进行排 序。

15、一种基于M算法的帧内预测模式选择方法，其特征在于：包括以 下步骤，A、生成当前预测块对应于M条路径的相邻像素与相邻预测模式信号； B、根据生成的相邻像素信息，计算出各种预测模式所对应的预测图像 数据； C、根据编码图像与生成的预测图像，按照各个不同预测模式进行重构 图像数据； D、根据相邻预测模式信息，以及编码图像与生成的预测图像，计算各 个预测模式代价； E、根据模式代价、块预测模式信号及路径编号，计算路径代价，选出M条路径代价最小的路径信息； F、根据选出的M条具有最小度量的路径对应的重构图像数据、块预测 模式以及源路径号数据更新各条路径相邻像素与块帧内预测模式信息； G、重复以上A至F步骤，直至各条路径均已达到终点，即沿似条路 径的计算都已对宏块内的所有块完成。此时选出M条路径中具有最小度量 的路径，并得到该宏块的帧内预测总代价； H、若还有对应于其他块大小的块帧内预测模式，则重复以上A至G过 程直至所有的块帧内预测总代价均已得到，然后将得到的块帧内预测总代价 与各宏块帧内预测模式代价相比较，选出具有最小代价的帧内预测模式，从 而确定宏块的最佳帧内预测模式。

16、如权利要求15所述的一种基于M算法的帧内预测模式选择方法， 其特征在于：所述的E步骤包括以下步骤， E1、保留对应于第“块运算开始时M个路径度量/^(V1,oh^TU Cv1,M-i), 并将每条路径的代价与来自块模式代价计算装置所对应的b个分支度量2相 力口； E2、将所获得的Mf个延伸后的路径度量排序，并得到M个最小路径度 量/;(s„，o)~/； 及对应的留存路径； E3、各新留存路径的源路径号，延伸分支所对应的块帧内预

reconstruction device (3) multi-channel parallel connection structure, each way intra prediction unit (2) and calculate the cost of block-mode and block reconstruction device (3) have respective current path selection means and adjacent pixels and the block prediction mode selection means.

Data management apparatus according to preclude the use of back structure (:

7, as claimed in one of claim 1 or 4, wherein the selecting means based on intra-prediction mode M algorithm, characterized in that 1), comprising sequentially connecting path selection history buffer means, and logic means backtracking adjacent buffer RAM; wherein adjacent buffer RAM, storing data corresponding to the M paths from block mode is used to calculate the cost of the block reconstruction device (3) reconstructed image data of the third output terminal, as well as the cost of computing from the path and mode selection means (4) first output data block mode selection; the path of the selection history buffer means for storing and updating routing information, and send back to logic means; said logic means backtracking, backtracking for the current block of data adjacent block data, obtained adjacent pixel blocks of the current block and the block adjacent to the prediction mode.

8, as claimed in claim 7, wherein a selection means algorithm based on intra-prediction mode M, wherein: said logic means backtracking, backtracking having one or more logical means for accessing one or more adjacent buffer RAM to give back the current block adjacent to a block having information for one or more than one.

9, as claimed in claim 7, wherein a selection means algorithm based on intra-prediction mode M, wherein: the data management apparatus back structure (1) can be adopted wherein be

测模式信号 以及被选中分支所对应的块重构图像数据被传送至数据管理装置（1)中， 并更新留存路径。

17、如权利要求15所述的一种基于M算法的帧内预测模式选择方法， 其特征在于：所述的步骤A中，采用回溯结构的数据管理装置（1)输出各 种模式下当前块的相邻像素与模式信息步骤如下： A1、从当前路径的当前块，利用来自路径选择历史缓冲的路径跳转信息 沿搜索树回溯至当前块的相邻块，并计算出其在相邻缓冲RAM中的地址； A2、按地址访问相邻缓冲RAM并得到当前块对应于一个或多个相邻块 的块相邻像素与块相邻预测模式； A3、相邻像素信息输出至帧内预测模块，相邻块预测模式输出至块模式 代价计算与块重构模块。

A4、重复以上过程直至当前块的所有相邻块的相邻像素与相邻块预测模 式均已被输出。

18、如权利要求15所述的一种基于M算法的帧内预测模式选择方法， 其特征在于：所述的步骤F中，采用回溯结构的数据管理装置更新路径的相 邻像素与块帧内预测模式信息步骤如下： FI、将块图像数据与块模式选择信息存入相邻缓冲RAM中，存储方式 取决于相邻缓冲RAM数据组织方式；可供选择的一种方式是将对应块数据 按当前的路径顺序依次存储； F2、将路径选择信息存入路径选择历史缓冲中，具体存储方式取决于相 邻缓冲RAM数据组织方式，可选择按当前的路径顺序依次存储。

19、如权利要求15所述的一种基于M算法的帧内预测模式选择方法， 其特征在于：所述的步骤A中，采用寄存器交换结构的数据管理装置（1）输出各种模式下当前块的相邻像素与模式信息步骤如下： A5、根据当前操作的路径，在相邻缓冲寄存器中选择当前计算的路径的 相邻像素信息与相邻块预测模式信息； A6、从当前路径的相邻像素信息与相邻块预测模式信息中挑选出当前计 算块的相邻像素信息与相邻块预测模式信息。

20、如权利要求15所述的一种基于M算法的帧内预测模式选择方法， 其特征在于：所述的步骤F中，采用寄存器交换结构的数据管理装置（1) 更新路径的相邻

achieved by multiplexing the plurality of intra prediction means calculate the cost front and block reconstruction mode data transmission between devices.

10, as claimed in claim 1, wherein a selection means M algorithm based on intra-prediction mode, wherein: the path of the cost calculation and mode selection means (4) in computing the cost of intra-block route or calculate the cost of the entire macro block.

11, as claimed in one of claim 1 or 10, wherein the selecting means M algorithm based on intra-prediction mode, wherein: the path of the cost calculation and mode selection means (4), including sequencing means connected in turn, the path metric register, minimum path metric selecting means; at the same time, the path number connected to a output of the input and sorting unit block mode register, sorting apparatus wherein, for like All extension path metric paths to sort the path metric register with M group contains a number of storage units of registers to store the current path M path metrics.

12, as claimed in claim 11, wherein a selection means M algorithm based on intra-prediction mode, wherein: said sorting means may be bubble sort algorithm, insertion sort Algorithm and merge sort algorithm to sort the path metric.

13, as claimed in claim 1, wherein a selection means M algorithm based on intra-prediction mode, wherein: said sorting means sorting can be completely or partially sorted.

14, as claimed in claim 12, wherein a selection means M algorithm based on intra-prediction mode, wherein: the sort means can preclude the use of comparative network or sorting networks sort the data.

像素与块帧内预测模式信息步骤如下：F3、根据各新留存路径的源路径号，将源路径号对应的路径数据存入各 新留存路径对应的寄存器中，寄存器数据组织方式决定路径数据和路经度量 的存储位置； F4、将对应各个新留存路径延伸分支的重构图像数据与块模式选摔信息 装入各新留存路径对应的寄存器中，寄存器数据组织方式决定路径数据和路 径度量的存储位置。

15, M-algorithm based on intra prediction mode selection method, characterized by: comprising the steps of, A, generates a predicted block corresponding to the current paths of adjacent pixels M and Phase o-prediction mode signal; B, according to the adjacent pixel information generated, calculate a variety of prediction modes corresponding predictive image data; C, according to the coding image and the predicted image generated in accordance with various prediction modes reconstructed image data; D, according to the neighboring prediction mode information, and encoded image and the generated prediction image, calculate the cost of each prediction mode; E, according to the cost model, block prediction mode signal and path number, path cost calculation, select the least costly path M path information; F, according to the selected M pieces having a minimum path metric corresponding to the reconstructed image data, prediction mode of block data updates and the source path number of each path the block of pixels neighboring intra prediction mode information; G, repeat A to F step until each path has reached the end, that is calculated along similar paths are for all blocks within the macro block is completed. At this election M paths having minimum metric path and get the macroblock intra prediction for a total consideration; H, if there are other blocks corresponding to the size of the block intra prediction mode, then repeat the above process A to G until all the blocks have obtained a total consideration of intra prediction and intra prediction block total consideration will be compared with the cost of intra-prediction mode macroblock, the intra-prediction mode selected with the minimum cost in order to determine the macro block The best intra prediction mode.

16, as claimed in claim 15, wherein a selection method based on intra-

prediction mode M algorithm, wherein: E step comprises the steps of, E1, corresponding to the retention The first "block operation at the beginning of the M path metrics / (V1, oh TU Cv1, Mi), and the cost of each path and the cost computing device block mode from the corresponding branch metric b 2 phase power outlet; E2 the Mf months after stretching the obtained path metrics sorted and get a minimum path metric M /; (s ", o) ~ /; and corresponding retained path; E3, the source path number of each new retained path extends branch block corresponding intra prediction mode signal and a block corresponding to the selected branch reconstructed image data is transmitted to the data management device (1), and updates the retained path.

17, as claimed in claim 15, wherein an M-algorithm-based intra prediction mode selection method, wherein: said step A, using the structure of the data management apparatus backtracking adjacent pixels of the current block pattern information in step (1) the output of various modes are as follows: A1, the current path from the current block, use the path from the path selection information history buffer jump back along the search tree adjacent to the current block block, and calculate the adjacent buffer at its address in the RAM; A2, by adjacent buffer RAM address access and obtain the current block corresponding to one or more adjacent blocks adjacent pixel blocks adjacent to the block prediction mode; A3, adjacent pixel information to the intra prediction module, the adjacent block prediction mode output to calculate the cost of block-mode and block reconstruction module.

A4, repeat the above process until all adjacent blocks of adjacent pixels of the current block and the adjacent block prediction modes are to be output.

18, as claimed in claim 15, wherein a selection method based on intra-

prediction mode M algorithm, characterized in that: said step F, preclude the updated data management means backtracking path structure adjacent pixel block intra prediction mode information following steps: FI, the block image data block mode selection information stored in adjacent buffer RAM, the memory depends on the adjacent buffer RAM data organization; alternative One way is to press the corresponding block number noted in the order of the current path storage; F2, the path selection information stored in the path selection history buffer, depending on the specific stored adjacent buffer RAM data organization, you can choose according to the current Path order of storage.

19, as claimed in claim 15, wherein an M-algorithm-based intra prediction mode selection method, wherein: said step A, preclude the use of register swap data structure adjacent pixels and mode information output step under various modes of the current block management means (1) are as follows: A5, according to the path of the current operation, select adjacent pixel information currently calculated path in the adjacent buffer register adjacent block prediction mode information; A6, selected adjacent pixel adjacent block prediction mode information and the information of the current block is calculated from the neighboring pixels adjacent block prediction mode information and the information of the current path.

20, as claimed in claim 15, wherein an M-algorithm-based intra prediction mode selection method, wherein: said step F, using the data management structure of register swap adjacent pixel block intra prediction mode information in step (1) update path is as follows: F3, according to the source path number retained for each new path, the path number corresponding to the source data into the path of the new and retained path corresponding register

register data organization and storage location determines the path data path metrics; F4, retained path corresponding to each new branch of the reconstructed image data and block mode extends fall election information is loaded to the new retained path corresponding register, the register data organization decided to route data and storage location path metric.

Description:

一种基于M算法的帧内预测模式选摔装置 技术领域 本发明属于数字视频处理领域，涉及一种帧内预测模式选择方法与实现 装置，特别是一种基于M算法的帧内预测模式选择方法与实现装置。

背景技术 随着音视频产业的不断发展，国际上对于音视频编解码技术的要求也越 来越高。视频编码技术演进至今，已出现了以MPEG-4AVC/H.264为代表的视频编码技术国际标准，而我国则推出了音视频编码标准AVS（Advanced Audio-Video Coding Standard in Information Technology,信息技术先进音视 频编码）。这些音视频编码标准能在保持视频主观质量的条件下，对信息量极大的视频信号进行高效压缩，极大降低存储空间和网络带宽要求。

作为视频编码的重要组成部分，帧内编码通过利用图像在空间上的自相 关性，除去视频中的信息冗余。为了进一步的提高帧内编码效率，各类先进 视频编码标准中进一步的使用了帧内预测技术。帧内预测技术利用相邻像素 的相关性，通过当前像素块的左边和上边已编码重建的像素对待编码像素进 行预测，并只对实际值与预测值的差值进行编码，从而降低了编码后的比特 数，提高编码效率。

H.264与AVS的帧内预测的基本思路是从不同方向计算像素亮度梯度， 并计算具有最小代价的方向作为最佳的预测方向。现有的帧内预测模式选择 方法将帧内预测分为两类，一类为整宏块帧内预测，宏块内部各像素点预测 值不直接相关，例如H.264的16x16亮度帧内预测；一类为块帧内预测，例 如AVS的8x8亮

M algorithm based on prediction mode selected wrestling TECHNICAL FIELD The present invention belongs to the field of digital video processing, relates to a method of intra prediction mode selection and implementation of equipment, M algorithm is based on the particular frame prediction mode selection methods and implement devices.

BACKGROUND With the continuous development of audio and video industry, the international technology for audio and video codec requirements are increasingly high. Video encoding technology evolution so far, has appeared to MPEG-4AVC / H.264 video coding represented international standard technology, while China introduced audio and video coding standard AVS (Advanced Audio-Video Coding Standard in Information Technology, advanced IT audio and video coding). These audio and video coding standard can be maintained in the video subjective quality conditions, the maximum amount of information for efficient video signal compression, which greatly reduces storage space and network bandwidth requirements.

As an important part of video encoding, intra-coded images by using spatial autocorrelation of removing redundant video information. In order to further improve the efficiency of intra coding, advanced video coding standard in all kinds of further use of intra prediction

度帧内预测，宏块被分为多个块并逐个进行预测，前块的 预测结果影响后块的预测。整宏块预测模式的代价通过宏块的相邻像素进行 整宏块预测计算得到。而块帧内预测需要从宏块内的第一个待预测块开始， 依次计算每一个块代价最小的模式，重构当前块，更新重构数据缓冲并在下 一个块上重复此步骤直至宏块内所有块的块帧内预测模式均选择完成，计算 顺序如图1所示，最后利用各个块的块帧内预测模式代价计算此块帧内预测 模式的总代价。在完成各种块帧内预测模式的总代价的计算后，将其与各个 整宏块预测模式的代价傲比较，代价最小的预测模式为最佳的帧内预测模式。

如图2所示，现有的块帧内预测模式选择装置由四部分组成，包括重构 数据缓冲与相邻管理装置，帧内预测装置，块模式代价计算与块重构装置以 及模式代价计算与模式选择装置。其中，重构数据缓冲与相邻管理装置记录 之前的编码过程中重构的图像数据以及所选择的块预测模式，生成当前预测 块(4x4块或8x8块)的相邻像素与相邻模式，并将前者传送至帧内预测装置， 后者传送至块模式代价计算与块重构装置。帧内预测装置根据相邻像素信 息，计算出当前块可用的各种预测模式下预测图像数据，并将其传送至块模 式代价计算与块重构装置。

块模式代价计算与块重构装置根据待编码图像以及来自帧内预测装置 的预测图像计算各个帧内预测模式代价，并分别按各个预测模式重构图像。

将重构图像传送至重构数据缓冲与相邻管理装置，将块模式代价传送至模式 代价计算与模式选择装置。

模式代价计算与模式选择装置在块级与宏块级两个层次上工作。在依次 计算各个块的块帧内预测模式的过程中，首先将各个块的6个块帧内预测模 式做比较，求出最小值并据此决定当前块最佳的块帧内预测模式。当各个块 的块帧内预测模式均已确定后，利用各个块帧内预测模式代价计算块帧内预 测模式的总代价。在完成各种块帧内预测模式的总代价的计算后，将其与各 个整宏块预测

techniques. After intra-prediction technology utilizes correlation between adjacent pixels, treat coded pixel predicted by the current left and top pixel block coded reconstruction of pixels, and only the difference between the actual and predicted values are encoded, thereby reducing the coding The number of bits, to improve the coding efficiency.

The basic idea intra prediction for H.264 and AVS are calculated from different directions pixel brightness gradient and direction with the minimum cost of computing as the best prediction direction. Existing intra-prediction mode selection method of intra prediction into two categories, one for the whole macroblock intra prediction, macroblock internal predictive value of each pixel is not directly related, such as 16x16 luminance intra prediction of H.264; one for the intra prediction block, for example, the AVS intra 8x8 luma prediction, a macroblock is divided into a plurality of blocks, and one by one prediction, prediction of the impact of predicted block is the previous block. The cost of the whole macroblock prediction mode by neighboring pixels macroblock whole macroblock prediction calculated. The block intra prediction macroblock needs from the first start of the block to be predicted, followed by calculation of the minimum cost of each block mode, the reconstruction of the current block, update reconstruct the data buffer on the next block and repeat this step until the macro block within a block intra prediction mode selection is completed all of the blocks are calculated in the order shown in Figure 1. Finally, each block of the block intra prediction mode to calculate the total cost of the cost of intra-prediction mode of the block. After completing the calculation of the total cost of the various block intra prediction mode, which is proud of the whole compared with the cost of each macroblock prediction mode, the

模式的代价做比较，代价最小的预测模式为最佳的帧内预测模 式。

在现有的方法与实现装置中，预测模式的最优化通过逐块搜索当前块的 最佳预测模式实现。然而由于帧内预测依赖于已编码与重构的数据，因而其 当前块预测的模式代价依赖于之前块的预测模式。这意味着现有的单路径搜 索方式虽然在搜索过程中各块都选择了局部极小代价，但整体代价却不一定 是全局最小值，导致了性能损失。

本发明建立于现有的帧内预测模式选择方法与实现装置基础之上，通过 应用M算法，实现块帧内预测模式的多路径全局搜索，从而降低块帧内预 测模式最佳模式搜索陷入局部极小值的可能性。

发明内容 本发明解决的技术问题是，降低宏块内部块帧内预测搜索陷入局部极值 的概率，本发明提供了一种实现块帧内预测模式的方法及其装置，采用多路径全局搜索手段，提高帧内预测模式选择方法选择到块预测代价的全局最小 的概率，从而提高编码性能。

为实现以上的发明目的，本发明是通过以下技术方案实现： 一种基于M算法的帧内预测模式选择装置，包括依次连接的帧内预测 装置2、块模式代价计算与块重构装置3,其特征在于：还包括用于存储重 构图像缓冲数据与管理相邻像素与路径信息的数据管理装置1和路径代价计 算与预测模式选择装置4;其中，数据管理装置1第一输入端与路径代价计 算与预测模式选择装置4的输出端连接，接收块预测模式信息和源路径号， 其第二输入端与块模式代价计算与块重构装置3的第三输出端相连，接收重 构图像数据；其第一输出端输出相邻像素至帧内预测装置2的第一输入端， 其第二输出端输出相邻预测模式信息至块模式代价计算与块重构装置3的第 二输入端，第三输出端输出路径编号至路径代价计算与预测模式选择装置4 的第三输入端；帧内预测装置2的第一输入端与数据管理装置1的第一输出 端相连，接收图像的相邻像素；其第一输出端输出经过计算的预测图像数据 至与块模式代价计算与块重构装置3的第一输入端；块模式代价计算与块重 构装置3的第一愉入端与帧内预测装置2的第一输出端相连，接收预

minimum cost of a prediction mode for the best intra-prediction mode.

2, the existing block intra prediction mode selection means calculates consists of four parts, including the reconstruction of the data buffer and the adjacent management unit, an intra prediction unit block mode as well as the costs and block reconstruction device the cost of computing mode and mode selection means. Wherein the encoding process to reconstruct the data buffer and the recording management apparatus before the adjacent reconstructed image data block and the selected prediction mode, to generate the current prediction block (4x4 block or 8x8 block) and the adjacent neighboring pixels mode, and the former transferred to the intra prediction unit, which is transmitted to the block mode and block reconstruction cost of computing devices. Intra prediction apparatus according to the information of adjacent pixels is calculated at the current block predicted image data of various prediction modes available, and transfers it to the cost calculation block mode and block reconstruction device.

Block mode and block reconstruction cost of computing means calculates the cost of each intra prediction mode in accordance with an image to be coded and a prediction image from the intra prediction unit, respectively, and the reconstructed image by each prediction mode.

The reconstructed image is transferred to reconstruct the data buffer and the adjacent management unit, the cost of the transport block mode to mode and mode selection means the cost of computing. mode and mode selection means for calculating the cost of work at the block level and the macroblock level two levels. Sequentially calculating for each block a block intra prediction mode process, first, the blocks of each block 6 compares the intra prediction

测图像 数据，其第二输入端与数据管理装置1的第二输出端相连，接收图像的相邻 预测模式信号，其第三输入端接初始编码图像；块模式代价计算与块重构装置3的第一、第二输出端分别输出模式代价数据、块预测模式信号至路径代 价计算与预测模式选择装置4的第一、第二输入端，其第三输出端输出重构 图像至数据管理装置1的第二偷入端；路径代价计算与预测模式选择装置4 第一、第二输入端分别与块模式代价计算与块重构装置3的第一、第二输出 端对应相连，分别接收模式代价数据和块预测模式信号，其第三输入端与数 据管理装置1的第三输出端相连，接收路径编号数据；其第一输出端块预测 模式与源路径号信息至数据管理装置1的第一输入端。

所述的帧内预测装置2与块模式代价计算与块重构装置3为多路并行连 接。所述的数据管理装置1存储M条路径的相邻像素与相邻预测模式。数 据管理装置1可采用寄存器交换结构、回溯结构或二者的混合结构实现。

所述的采用寄存器交换结构的数据管理装置1，包括依次连接的 相邻更新装置，采用寄存器组与选择器网络，用于存储及更新似条路 径的图像数据及相邻块预测模式数据；相邻缓沖寄存器，具有似组寄存器， 分别用于存储对应于M条路径的来自块模式代价计算与块重构装置3第三 输出端的重构图像数据，以及来自路径代价计算与模式选择装置4第一输出 端的块模式选择数据；当前路径选择装置，用于在当前路径信息中逸择当前 路径及其对应的路径数据；相邻像素与块预测模式选择装置，用于从当前路 径中挑选出当前块相邻像素信息与相邻块预测模式信息。

所述的帧内预测装置2,块模式代价计算与块重构装置3为多路并行连 接结构时，每一路帧内预测装置2及块模式代价计算与块重构装置3分别具 有各自对应的当前路径选择装置与相邻像素及块预测模式选择装置。

所述的采用回溯结构的数据管理装置1,包括依次连接路径选择历史缓 沖装置，回溯逻辑装置以及相邻缓沖RAM;其中，相邻缓沖RAM,用于存储对应于M条路径的来自块模式代价计算与块 重构装置3

mode, the minimum value is obtained and accordingly determines the current best intra prediction block mode block. When the block intra prediction mode of each block have been determined using the intra-prediction mode of each block Consideration The total consideration for the block intra prediction mode calculations. After completing the calculation of the total cost of the various block intra prediction mode, which compares with a cost of whole macroblock prediction mode, the minimum cost of a prediction mode for the best intra-prediction mode.

In the conventional method and apparatus to realize the prediction mode optimized by the best prediction mode of the current block-by-block search for implementation. However, due to the intra prediction and reconstruction depend on coded data, so its current block prediction mode prediction mode depends on the cost of the block before. This means that existing single-path search approach, while during the search have chosen a local minimum cost of each block, but overall the price is not necessarily a global minimum, resulting in a loss of performance.

The present invention is built on existing intra-prediction mode selection method and apparatus based on the realization, through the application of M algorithm, block intra prediction mode multipath global search, thereby reducing intra-prediction mode optimal mode block Search into the possibility of local minima.

Technical problems

The present invention solves is to reduce the intra-prediction macroblock search internal block into local extreme probability, the present invention provides a method and an apparatus for realizing block intra prediction mode, multi-path Global search for means to improve the intra-prediction mode

第三输出端的重构图像数据，以及来自路径代价计算与模式选择 装置4第一输出端的块模式选择数据； 所述的路径选择历史缓冲装置用于存储和更新路径选择信息，并发送至 回溯逻辑装置；所述的回溯逻辑装置，用于回溯当前块数据的相邻块数据， 得到当前块的块相邻像素及块相邻预测模式。

所述的回溯逻辑装置，具有一个或多个回溯逻辑单元，用于一次或多次 回溯访问相邻缓冲RAM得到当前块具有的一个或一个以上的相邻块信息。

采用回溯结构的数据管理装置1可通过复用方式实现与多个帧内预测装置、 块模式代价计算与块重构装置之间的数据传输。

所述的路径代价计算与模式选择装置4用于计算块内路径代价或整宏块 代价计算。所述的路径代价计算与模式选择装置4,包括依次连接的排序装 置、路径度量寄存器、最小路径度量选取装置；同时，路径号与块模式寄存 器的输入端与排序装置的一输出端相连，其中所述的排序装置，用于对M条 路径的所有延伸路径度量进行排序，所述的路径度量寄存器，具有M组包 含若干个寄存单元的寄存器，分别用于存储M条路径的当前路径度量数据。

所述的排序装置可采用冒泡排序算法、插入排序算法及归并排序算法对路径 度量进行排序。排序装置可实现完全排序或部分排序，可采用比较网络或排 序网络实现对数据进行排序。

一种基于M算法的帧内预测模式选择方法，其特征在于：包括以下步 骤， A、生成当前预测块对应于M条路径的相邻像素与相邻预测模式信号； B、根据生成的相邻像素信息，计算出各种预测模式所对应的预测图像 数据； C、根据编码图像与生成的预测图像，按照各个不同预测模式进行重构 图像数据； D、根据相邻预测模式信息，以及编码图像与生成的预测图像，计算各 个预测模式代价； E、根据模式代价、块预测模式信号及路径编号，计算路径代价，选出 M条路径代价最d、的路径信息； F、根据选出的M条具有最小度量的路径对应的重构图像数据、块预测 模式以及源路径

selection method selection to block prediction of the probability of a global minimum cost, thereby improving coding performance.

To achieve the above object of the invention, the present invention is realized by the following technical scheme: An Algorithm Based M intra-forecast mode selecting apparatus, comprising sequentially connected intraframe prediction means 2, the cost calculation block mode and block reconstruction means 3, characterized in that: further comprising a buffer for storing reconstructed image data and management data management apparatus and the route information of one pixel and adjacent path cost calculation and prediction mode selection device 4; wherein the first input data management apparatus 1 end path cost calculation and prediction mode selection means 4 is connected to the output of the receiving block prediction mode information and the source path number, a second input terminal block mode block reconstruction cost of computing and third output means 3 is connected to the receiver number noted in the reconstructed image; a first output terminal adjacent pixels to the intra prediction unit 2 of the first input terminal and a second output terminal adjacent to the block mode prediction mode information to calculate the cost and block reconstruction device 3 a second input terminal, a third output terminal path number to path cost calculation and prediction mode selection means 4 of the third input terminal; a first intra prediction means with the output of the first input terminal of a data management apparatus 2 is connected, receiving an image of adjacent pixels; a first output terminal through predictive image data to calculate the cost of computing and block mode and block reconstruction first input device 3; the price calculation block mode and block reconstruction device of the first 3 Discovery into the

号数据更新各条路径相邻像素与块帧内预测模式信息； G、重复以上A至F步骤，直至各条路径均已达到终点，即沿Af条路 径的计算都已对宏块内的所有块完成。此时选出M条路径中具有最小度量 的路径，并得到该宏块的帧内预测总代价。

H、若还有对应于其他块大小的块帧内预测模式，则重复以上A至G过 程直至所有的块帧内预测总代价均已得到，然后将得到的块帧内预测总代价 与各宏块帧内预测模式代价相比较，选出具有最小代价的帧内预测模式，从 而确定宏块的最佳帧内预测模式。

所述的E步骤包括以下步骤， E1、保留对应于第"块运算开始时M个路径度量OaGh/^1 CV1,M-0， 并将每条路径的代价与来自块模式代价计算装置所对应的b个分支度量A相 力口； E2、将所获得的Mf个延伸后的路径度量排序，并得到M个最小路径度 量/;(s„，0)∽/; (S„，M_7)及对应的留存路径； E3、各新留存路径的源路径号，延伸分支所对应的块帧内预测模式信号 以及被选中分支所对应的块重构图像数据被传送至数椐管理装置1中，并更 新留存路径。

所述的步骤A中，采用回溯结构的数据管理装置1输出各种模式下当前 块的相邻像素与模式信息步骤如下： A1、从当前路径的当前块，利用来自路径选择历史缓沖的路径跳转信息 沿搜索树回溯至当前块的相邻块，并计算出其在相邻缓沖RAM中的地址； A2、按地址访问相邻缓沖RAM并得到当前块对应于一个或多个相邻块 的块相邻像素与块相邻预测模式； A3、相邻像素信息输出至帧内预测模块，相邻块预测模式输出至块模式 代价计算与块重构模块。

A4、重复以上过程直至当前块的所有相邻块的相邻像素与相邻块预测模 式均已被输出。

所述的步骤F中，采用回溯结构的数据管理装置更新路径的相邻像素与 块帧内预测模式信息步骤如下： FI、将块图像数据与块模式选择信息存入相邻缓冲RAM中，存储方式 取决于相邻缓沖RAM数据组织方式；可供选择的一种方式是将对应块数据 按当前的路径顺序依次存储；

end of the intra-forecast apparatus 2 is connected to a first output terminal, receiving the predicted image data, a second input terminal and the output terminal of the second number noted in the management apparatus 1 is connected to receive the image of the neighboring prediction mode signal, its first Three initial encoding of the image input terminal; block pattern block reconstruction cost of computing the first and second output terminals, respectively, the cost of the output device 3 mode data block to the prediction mode signal path cost calculation and prediction mode selection device 4 first, a second input terminal, a third output terminal of the reconstructed image to the end of the second steal data management device 1; path cost calculation and prediction mode selection means 4 first and second input terminals of the cost of computing and block mode block a first, a second output terminal connected to a corresponding reconstitution device 3, the cost of data receive mode and the block prediction mode signal, respectively, a third output and its third input terminal is connected to a data management apparatus, the reception path number data; its an output block prediction mode and source path number information to the first input-side data management device 1.

Intra-prediction unit

2 the cost of computing and block mode and block reconstruction device 3 multiple parallel connections. M neighboring pixels stored in said data path management apparatus adjacent prediction mode. Data management apparatus 1 can register exchange architecture, backtracking or mixed fabric achieve both.

Image data Bian

of the register with the data management apparatus 1 switch fabric comprising adjacent update means

F2、将路径选择信息存入路径选择历史缓冲中，具体存储方式取决于相 邻缓冲RAM数据组织方式，可选择按当前的路径顺序依次存储。

所述的步骤A中，采用寄存器交换结构的数据管理装置1输出各种模式 下当前块的相邻像素与模式信息步骤如下： A5、根据当前操作的路径，在相邻缓冲寄存器中选择选择当前计算的路 径的相邻像素信息与相邻块预测模式信息； A6、从当前路径的相邻像素信息与相邻块预测模式信息中挑选出当前计 算块的相邻像素信息与相邻块预测模式信息。

所述的步骤F中，采用寄存器交换结构的数据管理装置更新路径的相邻 像素与块帧内预测模式信息步骤如下： F3、根据各新留存路径的源路径号，将源路径号对应的路径数据存入各 新留存路径对应的寄存器中，寄存器数据组织方式决定路径数据和路经度量 的存储位置； F4、将对应各个新留存路径延伸分支的重构图像数据与块模式选择信息 装入各新留存路径对应的寄存器中，寄存器数据组织方式决定路径数据和路 径度量的存储位置。

附图说明 图1是H.2644x4块帧内预测模式的块计算顺序示意图； 图2是现有块帧内预测模式选择装置结构示意图； 图3是现有亮度块帧内预测模式选择方法计算示意图，AVS 8x8亮度块模式 帧内预测； 图4是本发明的帧内预测模式选择方法计算示意图，AVS 8x8亮度块模式帧 内预测，其中M=2; 图5是本发明的串行路径帧内模式选择装置结构示意图；图6是本发明的并行路径帧内模式选择装置结构示意图； 图7是本发明中采用寄存器交换结构的数据管理装置结构示意图； 图8是本发明中采用回溯结构的数据管理装置结构示意图； 图9是现有模式代价计算与模式选择装置中的路径级部分结构示意图； 图10是现有重构数据缓冲与相邻管理装置结构示意图； 图11是本发明中的数据管理装置中的路径管理装置结构示意图； 图12是现有块模式代价计算与模式选择装置结构示意图；图13是本发明中路径代价计算与模式选择装置结构示意图。

具体实施方式 以下结合附图和具体实施例对本发明在H.264实时编码器上应用所

sequentially connected, using the register bank selector network, for storing and updating like paths and the adjacent block prediction pattern data; adjacent buffer register, having similar set of registers for storing respectively corresponding to the M paths from the cost calculation block mode data block reconstructed image reconstructed three third output terminal means, and from the path cost calculation and mode selecting means 4 first output mode selection data block; the current path selection means for in the current path information Yat select the current path and the corresponding path data; adjacent pixel block prediction mode selection means for the current path selected pixel information and prediction mode information of adjacent blocks adjacent to the current block.

When the intra-prediction unit

The two block models the cost of computing and block reconstruction device 3 multi-channel parallel connection structure, mode 2, and the cost of each road block intra prediction means calculates the block reconstruction device 3 respectively the respective current path selection means and the adjacent pixels and the block prediction mode selection means.

Bian

with the configuration of the data management apparatus 1 back, comprising sequentially connecting path selection history buffer means, and logic means backtracking adjacent buffer RAM; wherein adjacent buffer RAM, for storing data corresponding to the M Block mode from the cost calculation and block the path of the reconstructed image data reconstructed 3 third output means, and from the path cost calculation and mode selection means first output terminal 4 block mode selection data; the path of the history buffer device selection Choose to store and update path information

述帧 内模式选择方法流程与实现装置结构。

如图3所示,若将自首块开始各块可能的块帧内预测模式视作网格节点， 那么块帧内预测模式选择的历史可以看做穿过网格的路径，出发自节点的分 支即为块预测模式代价，模式选择方法即为在树中搜索具有最小度量厂即 块帧内预测模式的总代价最小路径的操作。现有的块帧内预测模式选择方法 在搜索过程的每一步中选出当前最佳路径的6条个延伸分支中，分支度量 即块帧内预测模式代价最小的路径。并将其附加于当前最佳路径做为下一步 骤的初始最佳路径。其中6为节点分支数，即为某块所有可能的帧内预测模 式数。例如对于H.264的4x4帧内预测模式，对于AVS的8x8帧内预 测模式，Z>=5(参考图3)。可见每一步骤中都只保留了当前局部最佳的分支， 并去弃了其余所有的分支。此过程极易于陷入局部极小。

本发明建立于现有的帧内预测模式选择方法与实现装置基础之上，通过 应用M算法，实现块帧内预测模式的多路径全局搜索，从而降低块帧内预 测模式最佳模式搜索陷入局部极小值的可能性。如图4,与现有的块帧内预 测模式选择方法在仅保留一条路径的傲法不同，所述帧内预测模式选择方法 在块n的模式选择中除了保留1条代价最小的路径^以外，还额外的保留 M-I条代价次小的路径、 ，/e[I,M-I]。在每一步骤中，延伸M条路径产生 bM条路径,分别计算每条延伸后路径的代价并对其排序，选出具有最小代 价的M条路径作为下一步的最佳路径。

如图5,所述帧内预测模式选择装置主要由4部分组成，包括数据管理 装置，帧内预测装置，块模式代价计算与块重构装置以及路径代价计算与模 式选择装置。其中，帧内预测装置与块模式代价计算与块重构装置与现有帧 内预测模式选择装置相同如图5所示。考虑到所述帧内预测模式选择方法需 要计算更多的路径，因此可以添加更多相同的帧内预测装置，块模式代价计 算与块重构装置以并行计算多条路径的方式提高速度，其结构如图6所示。

and send it to backtrack logic means; backtracking logic means for backtracking current data block adjacent blocks to obtain a block of the current block and the block adjacent pixel adjacent prediction mode.

Logic means backtracking

, wherein the back having one or more logical units, for access to one or more adjacent buffer RAM to give back the current block adjacent to a block having information for one or more than one. preclude the use of data management device back structure 1 can be achieved with a plurality of intra prediction computing devices, block mode block reconstruction costs and the number noted in the transmission between devices by multiplexing scheme. path of the cost calculation and mode selection means 4 is used within the computing block path cost or the cost of the entire macro-block calculations. The path cost calculation and the mode selection means 4, including sorting means connected in turn, the path metric register, minimum path metric selecting means; the same time, the path number connected to a output of the input and sorting unit block mode register, wherein said sorting means for extending the path of all M paths are sorted metric, the path metric register, having M groups comprising a plurality of registration unit registers, respectively, for the current path metric storing M data paths.

Sequencing means

bubble sort algorithm can be described, and the insertion sort algorithm merge sort algorithm to sort the path metric. Sorting device can be completely or partially sorted order, may preclude the use of comparative network or networks sort to sort the data.

M algorithm based intra prediction mode selection method, characterized by: comprising the steps of, A, to

数据管理装置同时管理M条预测路径，并根据来自路径代价计算与模 式选择装置的路径选择信息，控制M条路径更新为M条最佳的延伸后路径。

数据管理装置可采用寄存器交换结构，其装置结构如图7所示；也可采用回 溯结构，如图8所示，或此二者的混合结构管理路径的存储与更新。数据管 理装置生成当前预测块对应于M条路径的相邻像素与相邻预测模式，并将 其传送至各个帧内预测装置与块模式代价计算与块重构装置。

如图7所示，寄存器交换结构数据管理装置由4个装置组成，包括相邻 更新，相邻缓沖寄存器，当前路径选择，相邻像素与块预测模式选择。其中 相邻更新装置与相邻缓沖寄存器结合。相邻缓冲寄存器存储M条路径的块 图像数据/块模式选择的相邻信息。相邻更新装置接收图像数据与相邻块预测 模式的初始化数据并初始化相邻缓沖寄存器中M条路径相关的相邻信息， 接收来自块模式代价计算与块重构装置的更新块图像数据、来自路径代价计 算与模式选择装置的块模式选择数据与路径选择数据并更新相邻缓沖寄存 器中M条路径相关的相邻信息。当前路径选择从相邻缓冲寄存器中选出对 应当前路径的数据，并将其传送至相邻像素/块预测模式选择装置中。相邻像 素/块预测模式选择装置从当前路径数据中选出当前块对应的相邻像素信息与相邻块预测模式。相邻像素信息被发送至帧内预测模块，相邻块预测模式 被发送至块模式代价计算与块重构模块。若具有多个帧内预测装置/块模式代 价计算与块重构装置并行计算多条路径，则每个帧内预测装置/块模式代价计 算与块重构装置可具有各自对应的当前路径选择装置与相邻像素与块预测 模式选择装置，或者共用较少数量的当前路径选摔装置与相邻像素与块预测 模式选择装置。

如图8所示，回溯结构数据管理装置由3个装置组成，包括路径选择历 史缓沖，回溯逻辑以及相邻缓冲RAM。其中相邻缓沖RAM存储似条路径 的块图像数据/块模式选择的相邻信息,在工作开始时按照图像数据与相邻块 预测模式的初始化数据初始化，接收来自块模式代价计算与块重构装置的更 新块图像数据、来自

generate the prediction mode of the current prediction block adjacent pixel signals corresponding to the M paths adjacent; B, based on the generated adjacent pixel information, calculate a variety of prediction modes corresponding predictive image data; C, according to the coding image and the predicted image generated in accordance with various prediction modes reconstructed image data; D, based on the neighboring prediction mode information, and encoded image and the predicted image generated by calculating the cost of each prediction mode; E, according to the cost model, block prediction mode signal and path number, calculate the path cost of paths chosen M costs the least and d, the path information; F, depending on the options The strip has a minimum metric M path corresponding to reconstructed image data, prediction mode of block data updates and the source path number of each path the block of pixels neighboring intra prediction mode information; G, repeat the above step A to F, until each paths have reached the end, that is, along Af paths are calculated for all the blocks within the macro block is completed. At this election M paths having minimum metric path and get the total cost of intra prediction of the macroblock.

H, if there are other blocks corresponding to the size of the block intra prediction mode, then repeat the above process A to G until all the blocks have been obtained intra prediction for a total consideration, then the resulting total cost of intra prediction block Compared with the cost of intra-prediction mode macroblock, select an intra prediction mode having the smallest cost to determine the optimum intra-prediction mode macroblock.

Step E

comprises the steps of, E1, corresponding to the retained "operation begins at block M path

路径代价计算与模式选择装置的块模式选择数据并存入 RAM中。路径选择历史缓冲记录来自路径代价计算与模式选择装置的路径 选择信息，供回溯逻辑使用。回溯逻辑从当前路径的当前块，利用来自路径 选择历史缓冲的路径跳转信息沿搜索树回溯至当前块的相邻块，并计算出其 在相邻缓冲RAM中的地址，按地址访问相邻缓冲RAM并得到当前块的块 相邻像素/块相邻预测模式。相邻像素信息被发送至帧内预测模块，相邻块预测模式被发送至块模式代价计算与块重构模块。一般的，当前块具有超过1 个以上的相邻块，可以通过多次回溯逐个访问的方式实现，亦可通过特定的 相邻缓冲RAM数据组织方式减少回溯与相邻缓冲RAM访问次数。回溯结 构数据管理装置适合但不限制于配合单个帧内预测装置与块模式代价计算 与块重构装置以串行方式计算多条路径。若具有多个帧内预测装置/块模式代 价计算与块重构装置则可通过时分等方式复用。回溯逻辑可以使用但不限于 直接回溯，寄存器交换等结构。

路径代价计算与模式选择装置在路径级与宏块级两个层次上工作。在块 级上，路径代价计算与模式选择装置保留对应于第《个块计算开始时待更新 的M个源路径度量/^(s W，。)∽/;^ (S&m),并将每条路径的代价与来自帧 内预测装置和块模式代价计算装置所对应的b个分支度量义相加，将所获得 的办M个延伸后路径的度量排序，并得到M个更新的最小路径度量/;(s„,o：)∽ m {s^M.x),其对应延伸后路径被保留，其余延伸路径被丢弃。新的M留存 路径延伸自当前块开始计算时不超过似条的源路径，余下的不超过M-l条 源路径没有延伸路径留存而被去弃，请参考图4,路径∽在计算中被丢弃， 留存路径、。与~全部延伸自各新留存路径的源路径号，延伸分支所对 应的块帧内预测模式以及被选中分支所对应的块重构图像数据被传送至数 据管理装置中，并更新留存路径。当各条路径均已达到终点，即沿似条路 径的计算都已对宏块内的所有块完成后，此时具有最小度量的路径被选出。

其路径度量用于计算此块帧内预测模式总代价。此后路径代价计算与模式选 择装置完成与模式代价计算与模式选择装置相同的工作：将各宏块帧内预测 模式

metrics OaGh / 1 CV1, M-0, and the calculation of the cost and the cost of each path from block mode It means the corresponding branch metric b A phase force mouth; E2, the one extending Mf after sorting path metric obtained, and with a minimum path metric M /; (s '', 0) ~ /; (S'', M_7) and retained the corresponding path; E3, the source path number of new and retained path extending branch of the corresponding block intra prediction mode signal and corresponding to the selected branch block reconstructed image data is transmitted to the number noted in the management apparatus 1 and updates the retained path.

1 output under various modes of adjacent pixels of the current block pattern information in step

said step A, using retrospective data management device structure is as follows: A1, from the current block of the current path, the use of relief from the path selection history Chong jump information along the path of the search tree back to an adjacent block of the current block and calculates its address in the adjacent buffer in RAM; A2, adjacent buffer RAM access by the address and get the current block corresponding to one or a plurality of adjacent blocks adjacent pixel blocks adjacent to the block prediction mode; A3, adjacent pixel information to the intra prediction module, the adjacent block prediction mode output to calculate the cost of block-mode and block reconstruction module.

A4, repeat the above process until all adjacent blocks of adjacent pixels of the current block and the adjacent block prediction modes are to be output.

Step F

described, preclude the use of the adjacent pixel block intra prediction mode information data management means updates Step back path

代价相比较，选出具有最小代价的帧内模式，从而确定宏块的最佳帧内 预测模式。

路径代价计算与模式选择装置路径级部分见图9，由排序或部分排序装 置、路径度量寄存器、最小路径度量选取以及路径号与块模式寄存器构成。

路径度量寄存器存储M条路径的当前路径度量。来自块模式代价计算与块 重构的一个或多个模式代价值与所对应路径的当前路径度量相加生成延伸 路径度量值，并送入排序/部分排序装置。排序或部分排序装置将M条路径 的全部Z)M个延伸路径度量值排序或部分排序，得到最小的M个最佳度量值。

在应用排序装置的情况下此M个值为有序的，在应用部分排序装置的情况 下此M个值可能为无序的。对应的，数据管理装置中的路径管理可但不限 于按照路径度量有序或无序。排序或部分排序装置排序中选择的结果传送至 路径号与块预测模式寄存器，从输入数据的源路径号与块预测模式中选择出 与此M个最佳度量值相对应的源路径号与块预测模式，并传送至数据管理 装置。最小路径度量选取装置待全部路径全部块计算完成后从M个最佳度 量值中选出最小值用于此块预测模式的总开销计算，当采用排序装置实现时 可以无需使用最小路径度量选取装置。排序或部分排序装置可采用但不限于 冒泡排序、插入排序及归并排序等各类排序算法，可采用但不限于比较网络 或排序网络或其他的排序实现方式。

所述H.264实时编码器采用基于差平方和(SSD)的率失真率优化(RDO) 模式选择，代价为/ = 5^ + ；1.及她，其中;I为拉格朗日系数。如图3所示，应 用串行的路径计算结构，路径数M=3。块预测模式为4x4帧内预测。如图5 所示，数据管理装置同时管理3条预测路径，并根据来自路径代价计算与模 式选择装置的路径选择信息，控制3条路径更新为3条最佳的延伸后路径。

视不同情况，数据管理装置可以采用如图7所示的寄存器交换结构，以及如 图8所示的回溯结构或二者的混合结构管理路径的存储与更新。数据管理装 置生成

structure follows: FI, the block of image data and block mode selection information is stored in adjacent buffer RAM, the memory depends on the adjacent buffer RAM data organization; a way to choose is to block data corresponding to the current path by order of storage; F2, the path selection information stored in the path selection history buffer, depending on the specific storage adjacent buffer RAM data organization, you can choose according to the current path in the order storage. according to step A, using the data management apparatus registers the switch fabric at an output of the various modes and the mode information of adjacent pixels of the current block in step follows: A5, according to the path of the current operation in the adjacent buffer register Select Select the currently calculated route information of neighboring pixels adjacent block prediction mode information; A6, the selection information from the adjacent pixel adjacent block prediction mode information of the current path in the adjacent pixel information of the current calculating block and the adjacent block prediction mode information.

Step F

said, using the data management means updates the path register swap structure adjacent pixel block intra prediction mode information as follows: F3, according to the source path number retained for each new route, the source route No. corresponding path data into the new and retained path corresponding register, the register data organization and storage location determines the path data path metrics; F4, corresponding to each new branch retained path reconstructed image data and extended block mode selection information Load all new and retained path corresponding register, the register data organization decided to route data storage location and path metrics.

当前预测块对应于M条路径的相邻像素，并将其传送至帧内预测装 置，帧内预测装置根据相邻像素信息，计算出各种模式所对应的预测图像数 据，并将其传送至块模式代价计算与块重构装置。

块模式代价计算与块重构装置根据编码图像与来自帧内预测装置的预 测图像以及外来的帧间预测数据计算各个帧内与帧间预测模式代价，并分别 按各个预测模式重构图像。将重构图像传送至数据管理装置，将块模式代价 传送至模式代价计算与模式选择装置。

路径代价计算与模式选择装置在路径级与宏块级两个层次上工作。在块 级上，路径代价计算与模式选择装置保留对应于第n块运算开始时3个路径 度量/^1 (Ho)∽rnA 并将每条路径的代价与来自块模式代价计算装 置所对应的9个分支度量A相加，将所获得的27个延伸后的路径度量排序, 并得到3个最小路径度量尸办一∽/; (s„，2)及对应的留存路径。各新留存路径 的源路径号，延伸分支所对应的块帧内预测模式以及被选中分支所对应的块 重构图像数椐被传送至数据管理装置中，并更新留存路径。当各条路径均已 达到终点，即沿3条路径的计算都已对宏块内的所有块完成后，此时具有最 小度量的路径被选出。其路径度量用于计算此块帧内预测模式总代价，将各 宏块帧内预测模式代价相比较，选出具有最小代价的帧内模式，从而确定宏 块的最佳帧内预测模式。

如图10所示，为原设计的重构数据缓沖与相邻管理装置，相邻像素与 相邻预测模式信息保存于384bit的寄存器中；相邻更新逻辑完成开始工作前 相邻信息的初始化工作以及各块计算后更新相邻像素数据；相邻选择装置从 寄存器中选出当前块的相邻像素数据以及相邻块预测模式。前者发送至帧内 预测装置，后者发送至块模式代价计算与块重构装置。

如图11所示为所述的数据管理装置中的路径级管理装置，由于需要同 时管理3条路径，因此其具有三组寄存器与相邻更新装置，存储的三条路径 按其度量排序，即有对于"e[o,i5], r„(Sll0vr„(Sf^r»增加的交换结 构根据来自路径代价计算与模式选择装置的路径选择信息控制路径更新。增 加的路径选择装置从三条路径

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 is a block H.2644x4 calculation block intra prediction mode order schematic view; FIG. 2 is a conventional block schematic structural view of an intra-forecast mode selecting means; FIG. 3 is a conventional luminance intra-prediction mode selection block The method of calculating a schematic, AVS 8x8 luma block intra prediction mode; FIG. 4 is intra-forecast mode selecting method of the present invention to calculate a schematic, AVS 8x8 luma block intra prediction mode, where M = 2; FIG. 5 of the present invention is a serial Schematic path intra mode selection means; Figure 6 is a parallel path to the present invention, means the intra-mode selection structure diagram; Figure 7 is preclude the use of the present invention, the data management device configuration register swap schematic structure; Fig. 8 is the present invention Bian schematic data management device structure back structure; FIG. 9 is the cost of the existing model calculation and mode selection means the path stage partial structure diagram; Figure 10 is a conventional reconfigurable data buffer management unit and the adjacent structural diagram; FIG. 11 is path management structure diagram apparatus of the present invention, a data management device; Figure 12 is an existing block model to calculate the cost structure diagram with mode selection means; Fig. 13 is the present invention, path cost calculation structure diagram with mode selection means.

The following specific embodiments with the accompanying drawings and embodiments of the present invention is applied on the frame H.264 real-time encoder mode selection process flow within the device structure and implementation of the specific implementation.

中选出当前处理的路径提供至相邻选择，并生 成块相邻输出。当前路径选择信息表示步骤"中路径知_/_分别延伸自路径S^1V Sn.u/ S„.1;2中哪一条，交换结构将对应的原始路径交换至延伸后路径的存储位置，并由更新逻辑完成路径延伸。路径代价计算与模式选择装置 保证S1VS1/SI，2—定是延伸自S0，0，因此对路径So/Soj的初始化是不必要的。

如图13所示，所述路径代价计算与模式选择装置使用优化后的冒泡排 序，在逐次计算当前块对应于各路径各模式的代价过程中完成排序操作。由 于需要逐次各条路径，因此除了记录块预测模式以外还需记录源路径编号。

所述源路径编号即当前路径是延伸自哪条路径，三个源路径编号组成路径选 择信息。

由一个宏块处理开始，所述帧内预测模式选择方法具有以下全局流程： 1.首先完成初始化操作：由数据管理装置根据来自外部行缓沖的宏块相邻 数据初始化路径So。由路径代价计算与模式选择装置初始化各路径度量， 厂。(V)初始化为0或宏块基本语法元素代价，厂。(ㄟ)与厂(ㄟ,2)初始化为 大于尸„上界的正值。

2.初始化完成后开始依图1所示顺序依次计算各个块的预测模式总代价， 此步可细分为以下步驟： A.由路径代价计算与模式选择装置初始化各源路径号为0,初始化朴对 应排序中间寄存器为0,其余排序中间寄存器为大于/；上界的正值。

B.除第一块仅需计算路径补的9种4x4块帧内预测模式外，其余块均需 计算Ws2各9种共27个4x4块帧内预测模式。计算顺序不对计算 结果产生影响。

•数据管理装置依次输出3条路径9种模式下当前块的相邻像素与 模式信息。具体来说，通办于当前路径选择装置设定当前路径， 对相邻像素与块预测模式选择装置设定当前块，从相邻缓冲寄存 器中选出恰当的相邻数据。

参帧内预测装置计算对应的预测值。

3, if the surrender block starts each block may be regarded as intra-prediction mode of block grid nodes, block intra prediction mode selection history Imitation can see a path through the grid, starting from branch node is the cost of block prediction mode, the mode selection method is the search in the tree with the smallest measure of plant that is operating block intra prediction mode total cost of the minimum paths. Existing block intra prediction mode selection method to elect the current best path at every step of the search process in the six months extension branch, the branch metric block intra prediction mode that is the least costly path. And attach it as the best path to the next step of the initial to the current best path. Six of the node number of branches, is a piece of all the possible intra prediction mode number. For example, the intra 4x4 prediction mode of H.264, AVS for the intra 8x8 prediction mode, Z and gt; = 5 (refer to FIG. 3). Visible in each step only retained the best branch of the current local, and all the rest to abandon the branch. This process is extremely easy to fall into local minima.

The present invention is built on existing intra-prediction mode selection method and apparatus based on the realization, through the application of M algorithm, block intra prediction mode multipath global search, thereby reducing intra-prediction mode optimal mode block Search into the possibility of local minima. 4, and the existing block intra prediction mode selection method to retain only a path different proud method as shown, the intra-prediction mode selection method in a block n mode selection in addition to retaining a minimum cost path outside, but also additional costs of retention times MI strip small path,, / e [I, MI]. At each step, the extension M paths produce bM paths calculate the cost of extending the path and after each of its sort, elected on behalf of M paths with

参块模式代价计算与块重构装置完成对应的块重构，并根据基于 SSD的RDO准则计算代价。

参路径代价计算与模式选择装置对各个路径度量值进行排序。对于 本实施例，将数据逐个输入则比较网络即可自动完成排序。

重复以上过程直至当前块的全部路径与模式均计算完毕。

C.数据管理装置完成路径更新。具体而言，根据来自路径代价计算与模 式选择装置三条留存路径的源路径号完成路径交换，再根据每条路 径对应的的源路径号与块预测模式选出之前由块模式代价计算与块 重构装置产生的重构结果，更新各条路径的相邻像素与相邻预测信 息 O 以上步骤ABC重复直至全部16个4x4块均已计算完成。

3.数据管理装置中，、 ，所对应的块帧内预测模式历史即为4x4块帧内预测 的最佳模式，共包含16个块预测模式。数据管理装置中路径度量r16(V()) 为其代价。

4.执行原预测模式选择步骤的其余部分。具体而言，将4x4块帧内预测最 佳模式代价其加上宏块的其余基本语法元素代价，若步骤1中已加入则 略过，与16x16帧内预测的4种模式代价以及帧间预测的各模式代价做 比较，并确定最佳的宏块预测模式。

以上过程仅针对此实施例，当各组件装置采用不同结构时，整体步骤不 变但步骤细节可能出现变化。例如采用回溯结构重构数据管理装置时，输出 各种模式下当前块的相邻像素与模式信息步骤如下： 1.回溯逻辑从当前路径的当前块，利用来自路径选择历史缓冲的路径跳转信息沿搜索树回溯至当前块的相邻块，并计算出其在相邻緩沖RAM中 的地址。

2.按地址访问相邻缓沖RAM并得到当前块对应于一个或多个相邻块的块 相邻像素与块相邻预测模式。

3.将相邻像素信息输出至帧内预测模块，相邻块预测模式输出至块模式代 价计算与块重构模块。

the smallest price as the best path to the next step.

Figure 5, the intra-prediction mode selection device is mainly composed of four parts, including data management device, an intra prediction unit block mode and block reconstruction cost of computing device and path cost calculation and mode selection means. Wherein, intra prediction block mode means the cost calculation block reconstruction means selects the intra-prediction mode with the prior frame same apparatus as shown in FIG. Taking into account the intra prediction mode selection method need to be calculated more paths, it is possible to add more of the same intraframe prediction means, the cost calculation block mode and block reconstruction device for computing a plurality of parallel paths ways to improve the speed, which structure shown in Figure 6. data management device manages the M predicted path, and the path from the path selection information based on the cost of computing and mode selection means, control M path update after extending the best M paths. data management device can register exchange architecture, which means the structure shown in Figure 7; also can be traced back to the structure, as shown in the storage and updating, or hybrid structure to manage paths 8 This two. Data management means generates a prediction block corresponding to the current paths of the M neighboring pixels adjacent prediction mode, and transmits it to the respective intra prediction block mode means the cost calculation block reconstruction device.

7, register swap configuration data management apparatus 4 by the apparatus, including the adjacent updating, adjacent buffer register, the current path selection, neighboring pixel block prediction mode selection. Wherein adjacent update means and the adjacent buffer register

重复以上过程直至当前块的所有相邻块的相邻像素与相邻块预测模式 均已被愉出。

采用回溯结构数据管理装置时，完成路径更新步骤如下： 1.将块图像数据与块模式选择信息存入相邻缓冲RAM,具体存储方式取决 于相邻缓冲RAM数据组织方式，本专利可供选择的一种方式是将对应 块数据按当前的路径顺序依次存储。

2.按路径选择信息存入路径选择历史缓沖中，具体存储方式取决于相邻缓 沖寄存器数据组织方式，本专利可供选择的一种方式是将M条延伸路径 的源路径号按当前的路径顺序依次存储。

当在路径代价计算与模式选择装置中采用不同于本实现例的排序或部 分排序算法以及实现结构时，可能会对排序或部分排序操作的中间结果产生 影响，可能会对排序/部分排序操作所需的时间与硬件开销产生影响，但不影 响当前块计算完成后留存路径与其度量值的选择结果，若采用部分排序的情 况下其顺序可能改变，因而不会对整体步骤产生影响，除了以下的情况：采 用部分排序的情况下在全局流程步骤3中需要对留存的M个路径度量值进 行比较以确定最小路径度量，此步骤在采用排序时不必要。

以上所述仅为本发明的较佳实施例而已，并不用于限制本发明。凡在本 发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在 本发明的保护范围之内。

combination. Adjacent blocks of information stored in the image data buffer path adjacent M / block mode selection. Adjacent device receives the image data and the adjacent block prediction mode initialization data update and initialize adjacent buffer register adjacent to M path associated information received from the block model to calculate the cost of updating the block image data block reconstruction device from the path cost calculation and mode selection means for selecting data block mode and routing data and update the adjacent buffer register adjacent to M path associated information. The current path selection should be elected on the former path of the data from the adjacent buffer register, and transferred to the adjacent pixel / block prediction mode selection device. Adjacent pixel / block prediction mode selection means select the corresponding neighboring pixels of the current block and the adjacent block prediction mode information from the current path data. Adjacent pixel information is transmitted to the intra prediction module, the adjacent block prediction mode is transmitted to the calculation of the cost of a block pattern of the block reconstruction module. If the device has a plurality of intra prediction / block mode block reconstruction unit cost of computing and parallel computing multiple paths, each intra-prediction unit / block mode and block reconstruction cost of computing device may have respective current path selection means adjacent pixel block prediction mode selection means, or a small number of common current path selection means and the adjacent pixels fall block prediction mode selection means. shown in Figure 8, the structure of the data management apparatus from the back three means, including path selection history buffer, back buffer, and adjacent logic RAM. Wherein adjacent adjacent buffer RAM to store information like the path of the block image data / block mode is

selected, at work started in accordance with the initialization data to initialize the image data and the adjacent block prediction mode, to receive the price calculated from the block mode and block weight Update block image data structure means, the cost of computing from the path and mode selection means for selecting data block mode and stored in RAM. History buffer recording path selection from the path cost calculation mode selection means and path selection information for backtracking logical to use. Backtracking logic current path from the current block, use the path from the path selection information history buffer along the search tree jump back to an adjacent block of the current block and calculates its address in the adjacent buffer in RAM, according to address access phase o buffer RAM and get blocks of the current block adjacent pixel / block adjacent to the prediction mode. Adjacent pixel information is transmitted to the intra prediction module, adjacent block prediction mode is transmitted to the calculation of the cost of a block pattern of the block reconstruction module. In general, the current block having more than one or more adjacent blocks, backtracking can be individually accessed by multiple ways, and also reduce backtracking adjacent buffer RAM access times through specific adjacent buffer RAM data organization. Retrospective data structure management means are adapted, but not limited to devices with a single frame prediction mode and the cost calculation block and block reconstruction means serially computing a plurality of paths. If the device has a plurality of intra prediction / block mode Generation Rate Calculation and block reconstruction device, etc. can be multiplexed by time division. Backtracking logic can be used, but not limited to direct backtracking, register exchange and other structures. path cost calculation and mode selection means work on the path level and the

macroblock level two levels. At the block level, path cost calculation and mode selection means to retain corresponding to the "block calculated from the beginning to be updated M source path metric / (s W,.) ~ /; (S and m), and the cost of each path and the computing device and intra prediction block mode means the cost from the corresponding branch metric b righteousness sum obtained will do sort metric M rear extension path, and has been updated M smallest path Metric /; (s ", o:) ~ rn {s Mx), after extending its corresponding path is retained, and the rest is discarded path extending new M retained path extending from the start to the current block no more than a source like strip. path, remaining no more than MI article source path does not extend path retained is to give up, please 4, route - is discarded in the calculations, retained path with reference to FIG. and ~ all extending from the new and retained path source path number, extension corresponding to the branch block and a block to be intra prediction mode corresponding to the selected branch reconstructed image data is transmitted to the data management apparatus, and updates the retained path when the respective paths have reached the end point, i.e., along similar paths calculated have been completed for all the blocks after the macro block, this time with a minimum metric path is selected.

This block its path metric for intra prediction mode to calculate the total consideration. Best of each macroblock intra prediction mode compared to the expense of elected intra mode with the minimum cost in order to determine the macro block: After this path cost calculation and mode selection means for calculating the cost to complete the pattern and mode selection means the same work intra prediction mode. path cost calculation and mode selection means path level section shown in Figure 9, by a sort or partial ordering means, path metric register, minimum

path metric and a route number and choose Block mode register.

The path metric register stores M paths of current path metrics. The current path metric block mode from consideration on behalf of one or a plurality of modes corresponding to the path calculation value and the reconstructed block is generated by adding extension path metric, and fed sorting / partial ordering means. Ordering or partial ordering means all paths Z M) M an extension path metric ordering or partial ordering, give the smallest M best metric.

In the case of the application of this sort means of M is ordered, in the case of the application of this section sorting means M values may be unordered. Corresponding to the data management device path management can be but is not limited to follow the path metric ordered or unordered. Selected results of sort or partially transferred to the sorting means sorting path number and block prediction mode register, selected with this M best metric corresponding source path from the source path number and block number of the input data block prediction mode prediction mode, and transmitted to the data management device. The minimum path metric selecting means until all of the blocks all paths to calculate the total cost of the minimum value for the calculation of this block prediction mode after the completion of the M selected from the best metric, it is possible without the use of a minimum path metric selecting means when using the sequencing means to achieve. Sort or partial sequencing means may be, but are not limited to bubble sort, insertion sort and merge sort and other sorting algorithms, can preclude the use of, but not limited to compare the network or the network or other sort sort implementations.

The H.264 encoder preclude the use of real-time based on the difference of

squares (SSD) rate distortion optimization (RDO) mode selection, price for / = 5; 1 and she, in which; I is La Long day factor. As shown in Figure 3, you should calculate the path of a serial structure, the number of paths M = 3. Block prediction mode is the intra 4x4 prediction. As shown in Figure 5, the data management apparatus manages three predicted path, and the path from the path selection information according to the cost calculation and mode selection means, the control path 3 extending updated after three best path. depending on the circumstances, the data management device may preclude the exchange register with the structure shown in the figure, as well as storage and updating back structure or a mixed structure of both management path as shown in Figure 8. The data management apparatus generates a predicted block corresponding to the current paths of M adjacent pixels, and transferred to the intra prediction apparatus, the intra prediction apparatus according to the adjacent pixel information to calculate the various modes corresponding to the predictive image data, and transfer it to a block mode and block reconstruction cost of computing devices. block pattern and the cost calculation means for calculating the respective reconstructed intra block and inter prediction modes based on the cost of coding the prediction image and the prediction image from the intra-device and inter prediction external data, and reconstructed prediction modes separately by each image. The reconstructed image is transmitted to a data management device, the cost of the transport block mode to mode and mode selection means the cost of computing. path cost calculation and mode selection means work on the path level and the macroblock level two levels. At the block level, path cost calculation and mode selection means to retain the first n block operation began three paths corresponding to the metric / 1 (Ho) ~rnA and the cost of

each path and the cost computing device block mode from the corresponding A nine branch metrics added 27 after extending the obtained path metrics sorted and get three minimum path metric corpse do a ~ /; (s '', 2) and the corresponding path of the new and retained retained path. source path number, extending the corresponding branch block intra prediction mode is selected and the corresponding branch block number noted in the reconstructed image is transferred to the data management apparatus, and update the retained path. When have reached the end of each path, That along three paths have been calculated for all the blocks after the completion of the macro block, this time with a minimum metric path is selected in its path metric used to calculate this block total cost of intra-prediction mode, each macro block frame comparing the cost of intra-prediction mode, an intra mode with the minimum cost selected in order to determine the best intra prediction mode for the macroblock.

10, for the reconstruction of the original design of the data buffer management means adjacent, neighboring pixels adjacent 384bit prediction mode information stored in the register; adjacent to work is completed before the update logic adjacent After initialization information and update each block adjacent pixel data computing; adjacent pixel data selection means select adjacent and neighboring block prediction mode of the current block from the register. The former is sent to the intra prediction unit, which is sent to the block mode and block reconstruction cost of computing devices.

FIG. 11 shows the data management apparatus according to the path management device level, because of the need to manage three paths, so the three sets of registers having three paths adjacent update means for

storing sorted according to their metric, that the "e [o, i5], r" (SIl0vr "(Sf r□ added switch fabric control path selection information updated according to the price calculated from the path and mode selection means path. The path selection means increased from three path of the current process provides the path to an adjacent selected choices and building blocks adjacent output current path selection information represents a step "in the path known _ / _ respectively extending from the path S 1V Sn.u / S".1.; 2 in which one switch fabric corresponding original path after switching to the storage location path extending by update logic complete path extends path cost calculation and mode selection means to ensure S1VS1 / Sl, 2- given an extension from S0,0 Therefore the initialization path So / Soj is unnecessary.

13, calculating the path cost and mode selection means using the optimized bubble sort, complete sequencing operation in successive computational cost of the current block pattern corresponding to the paths of the process. Due to the need of each successive path, so in addition to recording block prediction mode number needed to record the source path.

The source path is the number that is the current path extending from which path, the path number three source composition path selection information. consists of a macro block process is started, the intra-prediction mode selection method has the following global process: 1. First complete the initialization operation: data initialization path by the data management apparatus according to the macro block adjacent to the buffer from the outside row of So. Select the path cost computing device initialization and mode of each path metric, plant. (V) is initialized to 0 or macroblock syntax elements basic price, factory. (~) And

plant (~ 2) is initialized to be greater than dead "positive on the sector.

.

2 After initialization start in sequence shown in Figure 1 followed by calculation of the prediction mode of each block the total consideration, this step can be broken down into the following steps: A. Select the device to initialize each source path resolution by the path cost calculation and mode 0, initialize the corresponding sort Park middle register is 0, and the remaining intermediate register sorted greater than /; a time when the community.

B. In addition to the first one just calculated path completion of nine kinds of 4x4 block intra prediction mode, the remaining blocks are required to calculate the total of nine kinds Ws2 27 each 4x4 block intra prediction mode. Calculation order not affect the results.

□ sequentially output data management apparatus and mode information of adjacent pixels path 9 under three modes of the current block. Specifically, through the Office of the selection means setting the current path in the current path of the neighboring pixels and block prediction mode selection means setting the current block, select the appropriate data from the adjacent adjacent buffer register.

Senate intra prediction means calculates the predicted value of the corresponding. parameter block mode block reconstruction unit cost of computing and the corresponding block reconstruction is completed, and based on the calculated cost of SSD-based RDO guidelines.

Senate path cost calculation and mode selection means for each path metric sorted. For the present embodiment, the data is clocked comparator network can be done automatically sorted.

Repeat the process until all the current path and pattern blocks are been calculated.

C. Data management device complete path update. Specifically, according to the price calculated from the path and mode selection means complete path source Path number three retained path exchange, then the cost of computing by the block model prior to each path according to the source path number corresponding to the prediction mode selected with the block and block weight Reconstruction results structure means for generating, updating each path adjacent pixels and the adjacent prediction information O ABC Repeat the above steps until all 16 4x4 blocks have been calculated.

3.,,, Data management apparatus corresponding block is the intra 4x4 prediction mode history best mode of intra prediction block, it contains a total of 16 block prediction mode. Data management apparatus path metric r16 (V ()) for its price.

4. Implementation of the original prediction mode selection step of the rest. Specifically, the remaining elements of the cost of basic grammar will best 4x4 block intra prediction mode plus the cost of its macro-block, if step 1 has been added to the skip, with four 16x16 intra prediction modes as well as the cost of inter prediction compare the cost of each mode of imitation, and determine the best macroblock prediction mode.

The above process only for this embodiment, when each component means preclude the use of different structures, overall steps do not change but the steps details may change. For example the use of retrospective data management apparatus Restructuring adjacent pixel mode information output step under various modes of the current block is as follows: 1. retrospective logic of the current path from the current

block, use the path from the path selection history buffer jump information along the search tree back to an adjacent block of the current block and calculates its address in the adjacent buffer RAM.

2. By adjacent buffer RAM address to access and obtain the current block corresponding to one or more adjacent blocks adjacent pixel blocks adjacent to the block prediction mode.

3. The adjacent pixel information to the intra prediction module, the adjacent block prediction mode to block mode output price calculation and generation of block reconstruction module.

Repeat this process until all adjacent blocks adjacent pixels adjacent block prediction mode of the current block have been out of Discovery.

When using retrospective data management device structure, complete the path update as follows: 1. Place the block of image data and block mode selection information stored in adjacent buffer RAM, depending on the specific storage adjacent buffer RAM data organization, this patent An alternative way is to press the corresponding block data is sequentially stored in the path of the current order.

2. Press path selection information stored in the path selection history buffer, depending on the specific stored adjacent buffer register data organization, an alternative embodiment of the present patent is the source path extending paths of M No. At current path in the order storage.

When the path cost calculation and mode selection means preclude the realization of this example different from ordering or partial ordering algorithms and implementation structure, intermediate results could have an impact ordering or partial ordering operation may have to sort / the time

required to produce part of the sorting operation and hardware overhead influence, but does not affect the calculation after the completion of the current block retained its path metric selection result of the case when using some sort of order may change, and thus will not have an impact on the overall steps except the following cases: Under the sort of conditions preclude the use of part of the global process in step 3 need to be retained in the M path metrics are compared to determine a minimum path metric, this step is not necessary when using sorted.

The foregoing is only preferred embodiments of the present invention but not to limit the invention. Any modification within the spirit and principles of the present invention, made, equivalent replacement, or improvement should be included within the scope of the present invention.

Cited by: CN103220511 A ;

PatentOrder

Publication number: JP2004128749 A2
Publication country: JAPAN
Publication type: APPLICATION
Publication date: 20040422
Application number: JP20020288015
Application date: 20020930
Priority: JP20020288015 20020930 JP20020288015;
Assignee[std]: TOSHIBA CORP ;
Inventor[std]: NAKAJO TAKESHI ; KOTO SHINICHIRO ; KIKUCHI YOSHIHIRO ;
International class[1-7]: H04N7/32 ; H03M7/36 ;
International class[8]: H04N7/32; H03M7/36;
Family members: JP2004128749 A2 JP3940657 B2
Title: MOVING IMAGE CODING METHOD AND DECODING METHOD
Title: 動画像符号化方法と装置及び動画像復号化方法と装置

Abstract:
PROBLEM TO BE SOLVED: To provide a moving image coding method which realizes much higher coding efficiency by improving prediction efficiency while suppressing the increase of overheads for coding. SOLUTION: One prediction mode is selected from among a plurality of the prediction modes comprising the combination of block shapes and motion compensation prediction from single or a plurality of reference frames for every prediction object block of input image signals. The motion compensation prediction of the prediction object block following the selected prediction mode is executed. At least one table among a plurality of prediction mode tables, in which a plurality of the prediction modes and a plurality of prediction mode indexes are made to correspond to each other for every coding object frame or for every prescribed region in the frame, is referred. Then, the prediction mode index corresponding to the selected prediction mode is coded.

Abstract:

【課題】符号化のオーバヘッドの増加を抑えつつ予測効率を向上させて、より高い符号化効率を実現できる動画像符号化方法を提供する。【解決手段】入力画像信号の予測対象ブロック毎にブロック形状と単一または複数の参照フレームからの動き補償予測との組み合わせからなる複数の予測モードから一つの予測モードを選択し、選択された予測モードに従って前記予測対象ブロックの動き補償予測を行い、符号化対象フレーム毎または該フレーム内の所定領域毎に複数の予測モードと予測モードインデックスを対応付けた複数の予測モードテーブルのうちの少なくとも一つのテーブルを参照して、選択された予測モードに対応する予測モードインデックスを符号化する。【選択図】図2

improve efficiency while suppressing the increase in the predicted coding overhead problems [], to provide a video coding method can achieve higher coding efficiency. Choose one from prediction mode for the combination of multiple prediction modes and motion-compensated prediction from a single or multiple reference frames for each block and the block shape of the input image signal and the predicted [resolved] means to predict the selected performed above the predicted motion-compensated prediction block according to the mode, at least one of multiple prediction mode table ???Keta mode prediction mode and prediction of multiple indices for each said predetermined region in the frame to be encoded for each frame or Refer to the

table to encode the prediction mode corresponding to the index of the selected prediction mode. Figure 2 selected] 2004-2010

Claims:

1. 入力画像信号の符号化対象フレーム内の複数の画素からなる予測対象ブロック毎に動き補償予測を含む符号化処理を行う動画像符号化方法において、 前記予測対象ブロック毎に、予測対象ブロック形状と単一または複数の参照フレームからの動き補償予測との組み合わせからなる複数の予測モードから一つの予測モードを選択するモード選択ステップと、 選択された予測モードに従って前記予測対象ブロックの動き補償予測を行うステップと、 前記符号化対象フレーム毎または該フレーム内の所定領域毎に、前記複数の予測モードと予測モードインデックスを対応付けた複数の予測モードテーブルのうちの少なくとも一つの予測モードテーブルを参照して、選択された予測モードに対応する予測モードインデックスを符号化するステップとを具備する動画像符号化方法。

2. 前記複数の予測モードテーブルは、前記複数の参照フレームをそれぞれ示すために同一の参照フレームに対してそれぞれ異なる値の参照フレームインデックスを付与した第1及び第2の参照リストの両方により前記動き補償予測に用いる参照フレームを規定する第1の予測モードテーブルと、前記第1及び第2の参照リストの一方のみにより前記動き補償予測に用いる参照フレームを規定する第2の予測モードテーブルとを含む請求項1記載の動画像符号化方法。

3. 前記符号化対象フレーム毎または該フレーム内の所定領域毎に、前記複数の予測モードテーブルから一つの予測モードテーブルを選択するテーブル選択ステップをさらに具備し、 前記予測モードインデックスを符号化するステップは、選択された予測モードテーブルを参照する請求項1記載の動画像符号化方法。

4. 前記選択された予測モードテーブルを示すテーブルインデックスを符号化するステップをさらに具備する請求項3記載の動画像符号化方法。

5. 前記テーブル選択ステップは、前記複数の参照フレームが前記符号化対象フレームに対して時間的に過去及び未来の両方のフレームであるか、あるいは前記符号化対象フレームに対して時間的に過去及び未来のいずれか一方のみのフレームであるかに応じて、該テーブル選択ステップが選択する前記一つの予測モードテーブルを決定する請求項3記載の動画像符号化方法。

6. 復号化対象フレーム内の複数の画素からなる予測対象ブロック毎に動き補償予測を含む符号化処理

1. A method for processing video coding, including a motion compensated predictive coding for each block consisting of a plurality of pixels being predicted to be encoded in a frame of input image signals, each said block being forecast, predict mode selection step of selecting a single prediction mode from a plurality of prediction modes and motion-compensated prediction from a combination of single or multiple reference frames from the target block shape, wherein the predicted motion block prediction mode selected in accordance with and performing a compensation prediction, each said predetermined region or in the frame to be encoded for each frame, wherein at least one prediction mode table of multiple prediction mode table index ???Keta mode the plurality of prediction modes and prediction See a video coding method comprising the steps of encoding the prediction mode index corresponding to the selected prediction mode.

2. mode, the plurality of prediction tables, lists both the first and second reference frame granted reference index for different values ??of the same reference frame respectively for the plurality of reference frames The first prediction mode table and defining the reference frame used by said motion-compensated prediction, a second prediction mode table defines the reference frame used for motion compensation prediction, only one in the list above by said first and second reference video coding method of claim 1, including a.

3. every said predetermined region or in the frame to be encoded for each frame, wherein the table further comprising a selection step of selecting a prediction mode table from one of the plurality of prediction mode table, said coding mode prediction index steps to do the video encoding method of claim 1 to refer to the selected prediction mode table.

4. video coding method of claim 3 further comprising the step of encoding an index table indicating said selected prediction mode table.

5. said table selection step, whether the frame of both past and future frames to be encoded in time to said plurality of reference frames, time frames to be encoded or above Depending on whether

を行う動画像復号化方法において、 前記予測対象ブロック毎に符号化された予測モードインデックスを復号化するステップと、 予測対象ブロック形状と単一または複数の参照フレームからの動き補償予測との組み合わせからなる複数の予測モードと予測モードインデックスを対応付けた複数の予測モードテーブルのうちの少なくとも一つの予測モードテーブルを参照して、復号化された予測モードインデックスに対応する予測モードを選択するモード選択ステップと、選択された予測モードに従って前記予測対象ブロックの動き補償予測を行うステップと を具備する動画像復号化方法。

7. 前記複数の予測モードテーブルは、前記複数の参照フレームをそれぞれ示すために同一の参照フレームに対してそれぞれ異なる値の参照フレームインデックスを付与した第 1 及び第 2 の参照リストの両方により前記動き補償予測に用いる参照フレームを規定する第 1 の予測モードテーブルと、前記第 1 及び第 2 の参照リストの一方のみにより前記動き補償予測に用いる参照フレームを規定する第 2 の予測モードテーブルとを含む請求項 6 記載の動画像復号化方法。

8. 前記符号化対象フレーム毎または該フレーム内の所定領域毎に、前記複数の予測モードテーブルから一つの予測モードテーブルを選択するテーブル選択ステップをさらに具備し、 前記モード選択ステップは、選択された予測モードテーブルを参照する請求項 6 記載の動画像復号化方法。

9. 符号化されたテーブルインデックスを復号化するステップをさらに具備し、 前記テーブル選択ステップは、復号化されたテーブルインデックスに従って一つの予測モードテーブルを選択する請求項 8 記載の動画像符号化方法。

10. 前記テーブル選択ステップは、前記複数の参照フレームが前記復号化対象フレームに対して時間的に過去及び未来の両方のフレームであるか、あるいは前記復号化対象フレームに対して時間的に過去及び未来のいずれか一方のみのフレームであるかに応じて、該テーブル選択ステップが選択する前記一つの予測モードテーブルを決定する請求項 8 記載の動画像符号化方法。

only one frame of the past and future, video coding method of claim 3, wherein a prediction table to determine the mode selected by said selection step of said table.

6. In the video decoding method of performing motion compensation predictive encoding process including prediction for each block to consist of a plurality of pixels in the frame being decoded, encoded prediction mode for each block to be above forecast a step of decoding the index of multiple predictions of the mode table index ???Keta modes predicted from a combination of a plurality of prediction modes and motion-compensated prediction from a single or multiple reference frame prediction block to the shape and see at least one prediction mode tables, and mode selection step to select the prediction mode corresponding to the decoded prediction index mode, the step of performing the predicted motion-compensated prediction block according to said selected prediction mode video decoding method comprising.

7. mode, the plurality of prediction tables, lists both the first and second reference frame granted reference index for different values ??of the same reference frame respectively for the plurality of reference frames The first prediction mode table and defining the reference frame used by said motion-compensated prediction, a second prediction mode table defines the reference frame used for motion compensation prediction, only one in the list above by said first and second reference video decoding method of claim 6, including a.

8. every said predetermined region or in the frame to be encoded for each frame, wherein the table further comprising a selection step of selecting a single prediction mode table prediction table from the plurality of modes, mode selection step, said video decoding method of claim 6, refer to the selected prediction mode table.

9. further comprising the step of decrypting the encoded index table, said table selection step, the video coding of claim 8, a prediction mode selection table according to the table of decoded index How do.

10. said table selection step, whether the frames of both the past and future time frame applies to decrypt said plurality of reference frames, time frames for the target or above decryption Depending on whether only one frame of the past and future, video coding method of claim 8 prediction mode table to determine said one of said table selection step is to choose.

Description:

[０００１][発明の属する技術の分野] 本発明は、動画像の圧縮符号化及び復号化方法に係り、特に複数の参照フレームを利用して動き補償予測を行う動画像符号化方法及び復号化方法に関する。

[０００２][従来の技術] 動画像の圧縮符号化技術として、ＭＰＥＧ（Ｍｏｔｉｏｎ Ｐｉｃｔｕｒｅ Ｅｘｐｅｒｔｓ Ｇｒｏｕｐｅ）方式が実用化されている。ＭＰＥＧ方式は、基本的に動き補償予測とＤＣＴ変換及び可変長符号化を組み合わせて符号化を行う方式であり、ＭＰＥＧ１、ＭＰＥＧ２及びＭＰＥＧ４が規格化されている。

[０００３]動き補償予測では、入力画像信号を複数の画素からなる画素ブロック（予測対象ブロックという）に分割し、予測対象ブロック毎に参照フレームからの動きを示す動きベクトルを求めて動き補償を行い、予測画像信号を生成する。予測対象ブロックの形状は符号化方式によって異なり、例えばＭＰＥＧ１では１６ｘ１６画素、ＭＰＥＧ２では１６ｘ１６画素または１６ｘ８画素、ＭＰＥＧ４では１６ｘ１６画素または８ｘ８画素にそれぞれ制限されている。

[０００４][発明が解決しようとする課題] 動画像符号化においては、符号化効率のより一層の向上が要求されている。このためには選択可能な予測対象ブロックの形状を増やしたり、動き補償予測に用いる参照フレームの選択肢を増やしたりすることで、予測効率を上げればよい。すなわち、予測対象ブロック毎に選択可能な予測モードの数を増やして、より予測効率の高い予測モードを選択することで、符号化効率の向上を図ることができる。しかしながら、予測対象ブロック毎に様々な予測モードを選択可能にすると、予測モードの種類の増加に伴い、予測モード情報を符号化するオーバーヘッドの符号量が増大するという問題がある。

[０００５]本発明は、符号化のオーバヘッドの増加を抑えつつ予測効率を向上させて、より高い符号化効率を実現できる動画像符号化方法及び復号化方法を提供することを目的とする。

[０００６][課題を解決するための手段] 上記課題を解決するため、本発明は入力画像信号の符号化対象フレーム内の複数の画素からなる予測対象ブロック毎に動き補償予測を含む符号化処理を行う動画像符号化方法において、予測対象ブロック毎に、予測対象ブロック形状と単一または複数の参照フレームからの動き補償予測との組み合わせからなる複数の予測モードから一つの予測モードを選択し、選択された予測モードに従って予測対象ブロックの動き補償予測を行い、符号化対象フレーム毎または該フレ

[0001] [which the present technical field The present invention relates to a method of encoding and decoding of video compression, video coding method performing motion compensation prediction using multiple reference frames, especially and on how to decrypt.

[0002] [ART] video compression coding technology, MPEG (Motion Picture Experts Groupe) scheme has been put into practical use. MPEG format is a combination of methods to encode variable-length coding and motion-compensated prediction and DCT transform Basically, MPEG1, MPEG4 has been standardized and MPEG2.

[0003] in the motion-compensated prediction, the pixel block consisting of the input image signal a plurality of pixels (a block to be predicted) is divided into motion asking for a motion vector indicating the movement of each block to be predicted from the reference frame compensation is performed to generate a predictive image signal. Unlike the predicted shape of the block coding scheme, 16x16 pixels in MPEG1, for example, 16x8 pixels or 16x16 pixels in MPEG2, which is limited in each 16x16 pixels or 8x8 pixels MPEG4.

[0004] [Problems to be Solved by the Invention In video coding, which is required to further improve the coding efficiency. To do this you can increase the shape of the predicted block can be selected, or by increasing the choice of reference frame used for motion-compensated prediction, raising the efficiency may be expected. In other words, increasing the number of prediction modes can be selected for each block to be predicted, the prediction mode by selecting more efficient forecast, it is possible to improve the coding efficiency. However, you can choose a different prediction modes for each block to be predicted, with the increasing variety of prediction mode, there is a problem that increases the amount of overhead sign to encode the prediction mode information.

[0005] The present invention, the prediction efficiency is improved while suppressing an increase in encoding overhead, and aims to provide a method for video coding and decoding method can achieve higher coding efficiency be.

[0006] [Means for Solving the Problems] To solve

ーム内の所定領域毎に、複数の予測モードと予測モードインデックスを対応付けた複数の予測モードテーブルのうちの少なくとも一つの予測モードテーブルを参照して、選択された予測モードに対応する予測モードインデックスを符号化する。

[０００７]ここで、複数の予測モードテーブルは、複数の参照フレームをそれぞれ示すために同一の参照フレームに対してそれぞれ異なる値の参照フレームインデックスを付与した第１及び第２の参照リストの両方により動き補償予測に用いる参照フレームを規定する第１の予測モードテーブルと、第１及び第２の参照リストの一方のみにより動き補償予測に用いる参照フレームを規定する第２の予測モードテーブルとを含む。

[０００８]予測モードインデックスの符号化のために、符号化対象フレーム毎または該フレーム内の所定領域毎に複数の予測モードテーブルから一つの予測モードテーブルを選択し、選択した予測モードテーブルを参照して予測モードインデックスを符号化してもよい。

[０００９]動き補償予測に際しては、さらに選択された予測モードテーブルを示すテーブルインデックスを符号化してもよい。予測モードテーブルの選択は、複数の参照フレームが符号化対象フレームに対して時間的に過去及び未来の両方のフレームであるか、あるいは符号化対象フレームに対して時間的に過去及び未来のいずれか一方のみのフレームであるかに応じて、該選択する一つの予測モードテーブルを決定することもできる。

[００１０]また、本発明は復号化対象フレーム内の複数の画素からなる予測対象ブロック毎に動き補償予測を含む符号化処理を行う動画像復号化方法において、予測対象ブロック毎に符号化された予測モードインデックスを復号化し、予測対象ブロック形状と単一または複数の参照フレームからの動き補償予測との組み合わせからなる複数の予測モードと予測モードインデックスを対応付けた複数の予測モードテーブルのうちの少なくとも一つの予測モードテーブルを参照して、復号化された予測モードインデックスに対応する予測モードを選択し、選択された予測モードに従って予測対象ブロックの動き補償予測を行う。

[００１１]を具備する動画像復号化方法。

[００１２]ここで、複数の予測モードテーブルは、複数の参照フレームをそれぞれ示すために同一の参照フレームに対してそれぞれ異なる値の参照フレームインデックスを付与した第１及び第２の参照リストの両方により動き補償予測に用いる参照フレームを規定する第１の予測モードテーブルと、第１及び第２の参照リストの一方のみにより動き補償予測に用いる参照フレームを規定する第２の予測モードテー

the above problems, the present invention includes the predicted motion-compensated prediction for each block consisting of a plurality of pixels in the frame to be encoded in the input image signal The video coding method to handle the encoding, each block being forecast, a prediction mode from a plurality of prediction modes and a combination of motion-compensated prediction from a single or multiple reference frames and the predicted block shape Select and perform the predicted motion-compensated prediction block selected according to the prediction mode, each frame in said predetermined area to be encoded for each frame or, more predictable mode mode ???Keta prediction mode and prediction of multiple indexes see at least one prediction mode table of the table to encode the prediction mode index corresponding to the selected prediction mode.

[0007] In this mode, multiple prediction tables are listed first and second reference frame granted reference index for different values ??of the same reference frame for multiple reference frames respectively The first prediction mode table and defining the reference frame used by both the motion-compensated prediction, the second prediction mode table and defining the reference frame used for motion-compensated prediction by only one of the first and second reference list including.

[0008] The predictive index for the encoding mode, select the mode from the mode table prediction table one more forecast for every said predetermined region or in the frame to be encoded for each frame, the selected prediction mode table coding mode may be predicted with reference to the index.

[0009] When motion-compensated prediction, the table may be coded index of the selected prediction mode table further. Prediction mode selection table, or frame of both past and future time frames to be encoded for multiple reference frames, one of the past and future frames to be encoded in terms of time or for Depending on whether only one frame can also decide to choose one mode of said prediction table.

[0010] The present invention is in the video decoding method of performing motion compensation predictive encoding process including prediction for each block to consist of a plurality of pixels in the frame being decoded, target predictive coding for each block decodes the prediction mode-ized index of multiple prediction mode table index ???Keta modes predicted from a combination of a plurality of

ブルとを含む。

[００１３]予測モードインデックスの符号化のために、符号化対象フレーム毎または該フレーム内の所定領域毎に、複数の予測モードテーブルから一つの予測モードテーブルを選択し、選択された予測モードテーブルを参照して予測モードインデックスを符号化してもよい。

[００１４]動き補償予測に際しては、符号化されたテーブルインデックスを復号化し、復号化されたテーブルインデックスに従って一つの予測モードテーブルを選択するようにしてもよい。予測モードテーブルの選択は、複数の参照フレームが復号化対象フレームに対して時間的に過去及び未来の両方のフレームであるか、あるいは復号化対象フレームに対して時間的に過去及び未来のいずれか一方のみのフレームであるかに応じて、選択する一つの予測モードテーブルを決定することもできる。

[００１５]各予測モードテーブルとして、取り得る予測モードのサブセット、参照フレームの構成により発生しない予測モードを削減したもの、あるいは予測モードの並べ替えを行ったテーブルを用い、符号化対象フレーム毎あるいは符号化対象フレームの領域毎に、予測モードの選択状況に応じて適切な予測モードテーブルを選択することで、予測モードインデックスを符号化するための符号量を削減することが可能となる。

[００１６][発明の実施の形態]以下、図面を参照して本発明の実施の形態を説明する。（符号化側の構成）図１に、本発明の一実施形態に係る動画像符号化方法を実施する装置の構成を示す。図１に示す動画像符号化装置は、ハードウェアで実現してもよいし、コンピュータを用いてソフトウェアにより実行してもよい。一部の処理をハードウェアで実現し、他の処理をソフトウェアにより行ってもよい。

[００１７]図１において、入力画像信号（動画像信号）１００は、符号化部１０１に入力される。符号化部１０１は減算部１０２、直交変換部１０３、量子化部１０４、逆量子化部１０５、逆直交変換部１０６、加算部１０７、フレームメモリ／予測画像作成部１０８、モードテーブル選択部１０９、モード選択部１１０及び可変長符号化部１１６を有する。

[００１８]フレームメモリ／予測画像作成部１０８は、複数のフレームメモリと動きベクトル検出機能及び動き補償予測機能を有する。フレームメモリ／予測画像作成部１０８では、まず入力画像信号１００の複数に分割された複数の画素からなる画素ブロック（以下、これを予測対象ブロックという）毎に、フレームメモリに記憶されている複数の参照フレームを用いて、動きベクトルの検出が行われる。一方、モード選択部１１０では、複数の予測モードの中か

prediction modes and motion-compensated prediction from a single or multiple reference frame prediction block to the shape and see at least one prediction mode table, select the prediction mode corresponding to the decoded prediction index mode, do the predicted motion-compensated prediction block according to the prediction mode is selected.

[0011] video decoding method comprising the.

[0012] In this mode, multiple prediction tables are listed first and second reference frame granted reference index for different values ??of the same reference frame for multiple reference frames respectively The first prediction mode table and defining the reference frame used by both the motion-compensated prediction, the second prediction mode table and defining the reference frame used for motion-compensated prediction by only one of the first and second reference list including.

[0013] The predictive index for the encoding mode, each said predetermined region in the frame to be encoded for each frame, or to select the mode from the mode table prediction table one more forecast, prediction of the selected may encode the prediction mode, see mode table index.

[0014] When motion-compensated prediction, decodes the encoded index table may have to select a prediction mode table according to the decrypted index table. Prediction mode selection table, or frame of both past and future time frames being decoded to a plurality of reference frames, one of the past and future in time to decrypt the target frame or Depending on whether only one frame can also be determined to select a prediction mode table.

[0015] As each prediction mode table, a subset of the possible prediction mode, prediction mode was not caused by the reduced configuration of the reference frame used to make the sort of prediction mode table or frame to be encoded every region of the frame to be encoded or every prediction table by selecting the appropriate mode depending on the circumstances of the prediction mode selection, it is possible to reduce the amount of codes to encode the prediction mode index.

[0016] [Embodiment of the Invention The following description of embodiments of the present invention with reference to the drawings. (Structure of the coding) in Figure 1 shows the

らの最適な一つの予測モードが選択される。こうして検出された動きベクトル及び選択された予測モードを用いて、フレームメモリ／予測画像作成部１０８により一つまたは複数の参照フレームに対して動き補償が行われ、予測画像信号が生成される。

[００１９]フレームメモリ／予測画像作成部１０８により生成された予測画像信号は減算部１０２に入力され、ここで入力画像信号１００との差である予測誤差信号が求められる。この予測画像信号に対して、直交変換部１０３により直交変換（例えば、ＤＣＴ変換）が施され、直交変換係数が生成される。直交変換係数は、量子化部１０４により量子化される。

[００２０]量子化された直交変換係数１１１は、モード選択部１１０から出力されるモード情報１１２、モードテーブル１０９から出力されるモードテーブル情報１１３、フレームメモリ／予測画像作成部１０８から出力される動きベクトル情報１１４及び参照フレーム／予測係数情報１１５などの動き補償予測方法に関わる情報と共に、可変長符号化部１１６により可変長符号化される。以下の説明では、参照フレーム／予測係数情報１１５を参照フレーム情報と予測係数情報に分けて述べる場合もある。モード情報１１２、モードテーブル情報１１３、動きベクトル情報１１４及び参照フレーム／予測係数情報１１５については、後に詳しく説明する。

[００２１]可変長符号化部１１６によって生成される上記の各情報１１１～１１５に対応する可変長符号のデータは、多重化部１１８によって多重化され、出力バッファ１１９を介して符号化データ１２０が出力される。符号化データ１２０は、図示しない蓄積系または伝送系へ送出される。

[００２２]量子化された直交変換係数１１１は逆量子化部１０５にも入力され、逆量子化部１０５及び逆直交変換部１０６によって量子化部１０４及び直交変換部１０３と逆の処理を受けることにより、予測誤差信号が再生される。再生された予測画像信号は加算部１０７に入力され、フレームメモリ／予測画像生成部１０８により生成された予測画像信号と加算されることにより、局部復号化画像信号が生成される。局部復号化画像信号は、参照フレームとしてフレームメモリ／予測画像作成部１０８内のフレームメモリに保存される。

[００２３]出力バッファ１１９からは、出力バッファ１１９に一時保持される符号化データの発生符号量の情報が符号化制御部１１７にフィードバックされる。これによって、符号化制御部１１７は量子化部１０３及び逆量子化部１０５量子化特性（量子化ステップ幅）の制御により発生符号量の制御を行うレート制御と、フレームメモリ／予測画像作成部１０８におけるフレーム間予測構造の制御を始めと

configuration of a video coding method to implement according to one embodiment of the invention. Video encoding apparatus shown in Figure 1, may be implemented in hardware, software may be executed by using a computer. Implemented in hardware and some processing may be carried out by other software processes.

[0017] In Figure 1, the input image signal (video signal) 100 is input to coding section 101. Coding section 101 is the subtraction unit 102, an orthogonal transformation unit 103, quantizer 104, inverse quantization unit 105, inverse orthogonal transformation unit 106, adder 107, frame memory unit 108 creates predictive picture / mode selection table 109, with a variable-length coding section 110 and section 116 mode selection.

[0018] 108 frame predictive image generation unit memory / prediction and motion compensation has a function of motion vector detection and multiple frame memory. The 108 creates predictive image frame memory / pixel block consisting of a plurality of pixels divided into a plurality of signal 100 input image first (hereinafter referred to as block to predict it) each, have more than one stored in frame memory using a reference frame, motion vector detection is carried out. In contrast, the mode selection unit 110, a prediction mode is selected from among the best prediction of multiple modes. Selected using the prediction mode and motion vector thus detected is performed on the reference frame motion compensation section 108 by creating one or more predictive image frame memory / prediction picture signal is generated.

[0019] The image signal generated by the prediction unit 108 creates predictive picture frame memory / subtraction unit 102 is input, the prediction error signal obtained is the difference between the input image signal 100 here. Predictive image signal, the orthogonal transform section 103 by the orthogonal transformation (eg, converting DCT) is applied, the orthogonal transform coefficients are generated. Orthogonal transform coefficients are quantized by the quantization unit 104.

[0020] 111 the quantized orthogonal transform coefficients 112, mode information is output from the mode select section 110, 113 table mode information is output from the mode table 109, the prediction image generating unit 108 outputs the frame memory / motion-compensated prediction with information relating to how such a frame prediction factor information 115 / 114 and motion vector information is information, which is

する、符号化部１０１に対する各種の制御を行う。

[００２４]モードテーブル選択部１０９は、入力画像信号１００の符号化対象フレーム毎または符号化対象フレーム内の複数の画素ブロックから構成される領域（以下、スライスと呼ぶ）毎に、複数の予測モードテーブルの中から符号化効率をより高くする一つの予測モードテーブルを選択する。予測モードテーブルは、後述するように予測対象ブロックの形状と第１または第２の動き補償予測モードとの組み合わせと予測モードインデックスを対応付けたテーブルである。

[００２５]モードテーブル選択部１０９によって選択された予測モードテーブルを示すモードテーブル情報１１３は、可変長符号化部１１６によりインデックス（予測モードテーブルインデックス）として符号化される。他の実施形態として、参照フレームの構成（動き補償予測構造）に応じて予測モードテーブルが自動的に一意に決定される構成としてもよい。その場合は、予測モードテーブルインデックスの符号化は不要である。いずれの場合においても、復号化時には符号化時に使用したのと同一の予測モードテーブルが選択され、予測モード情報の復号化が行われる。

[００２６]モード選択部１１０は、入力画像信号１００の予測対象ブロック毎に、モードテーブル選択部１０９によって選択された予測モードテーブルを参照して最適な予測モードを選択すると共に、選択した予測モードを示すモード情報１１２を出力する。ここで、予測モードとは後述するように、予測対象ブロックの形状と動き補償予測モードの組み合わせである。モード選択部１１０から出力される選択した予測モードを示すモード情報１１２は、可変長符号化部１１６により予測モードインデックスとして符号化される。

[００２７]（符号化処理手順） 次に、本実施形態における動き補償予測を含む符号化処理手順について説明する。 まず、最小符号化コストＭｉｎＣｏｓｔの初期値に最大値を設定（ステップＳ１０）、予測ブロックの形状モードを示すインデックスＢｌｏｃｋＴｙｐｅに０を設定（ステップＳ１１）、参照フレームを識別するインデックスＲｅｆＦｒａｍｅに０を設定する（ステップＳ１２）。

[００２８]次に、入力画像信号１００の予測対象ブロック毎に、設定されたブロック形状及び参照フレームを用いて動きベクトルの検出を行う（ステップＳ１３）。すなわち、周知のブロックマッチングを用いて、入力画像信号１００である符号化対象フレームの予測対象ブロックと最も相関の高い複数の画素から構成される領域（参照ブロック）を参照フレームから探索し、予測対象ブロックと参照ブロックとをつなぐベクトルを動きベクトルとして検出する。

encoded by variable length coding unit 116 variable length. In the following description, sometimes discussed separately in the reference frame information and the best predictive factor of 115 for the frame prediction factor information / reference. Mode information 112, mode information table 113, 115 for the frame prediction factor information / reference information 114 and motion vector will be described in detail later.

[0021] The variable-length code data corresponding to 111-115 of the above information generated by the variable-length coding unit 116 are multiplexed by multiplexing section 118, 119 through the buffer output encoding data 120 is output. Encoded data 120 is transmitted to the system or transmission system (not shown) ??.

[0022] 111 the quantized orthogonal transform coefficients are input to the inverse quantization unit 105, an orthogonal transform processing section 103 and the reverse quantization unit 104 inverse orthogonal transformation section 105 and section 106, inverse quantization by receiving a prediction error signal is played. The reconstructed predicted image signal is input to the summing unit 107, is added by the predictive image signal generated by the predictive image generation unit 108 frame memory / decoded image signal is generated locally. The local decoded image signal is stored in frame memory in the frame memory unit 108 creates predictive picture / a reference frame.

[0023] The output buffer 119 is fed to the encoding control unit 117 of information generation code amount of coded data is temporarily held in output buffer 119. This coding control unit 117 is quantized properties 105 103 quantization and inverse quantization (quantization step size) and rate control for controlling the code amount generated by the control unit 108 creates predictive picture frame memory / and other control structures in the inter-frame prediction, and control a variety of coding for 101.

[0024] 109 the mode select table space is composed of blocks of pixels to be coded in multiple frames to be encoded for each frame or input image signal 100 (hereinafter referred to as slices) each, more One prediction table to select the mode to higher coding efficiency predicted from the mode table. Prediction mode table is a table index ???Keta mode prediction mode and motion-compensated prediction in combination with first or second block to predict the shape and as described below.

[００２９]次に、検出された動きベクトルを用いて、設定された参照フレームの各々から参照ブロックを切り出し、切り出された参照ブロックに対し線形予測演算を施して、予測画像信号を生成する（ステップＳ１４）。参照フレームが２つ選択された予測対象ブロックに対しては、各々の参照フレームから一つずつ、計２個の参照ブロックが切り出されることになる。

[００３０]次に、切り出された参照ブロックに対し線形予測演算を施して、予測画像信号を生成する（ステップＳ１４）。

[００３１]ステップＳｉ４においては、参照ブロックが一つの場合の予測画像信号は次式（１）または（２）に従って計算され、参照ブロックが２つの場合の予測画像信号は次式（３）に従って計算される。Ｐ（ｘ，ｙ）＝Ｗ０ｘＲ（ｒｅｆ＿ｉｄｘ＿ｉ０，ｘ＋ｍｖ０．ｘ，ｙ＋ｍｖ０．ｙ）＋Ｃ（１）Ｐ（ｘ，ｙ）＝Ｗ１ｘＲ（ｒｅｆ＿ｉｄｘ＿ｉ１，ｘ＋ｍｖ１．ｘ，ｙ＋ｍｖ１．ｙ）＋Ｃ（２）Ｐ（ｘ，ｙ）＝Ｗ０ｘＲ（ｒｅｆ＿ｉｄｘ＿ｉ０，ｘ＋ｍｖ０．ｘ，ｙ＋ｍｖ０．ｙ）＋Ｗ１ｘＲ（ｒｅｆ＿ｉｄｘ＿ｉ１，ｘ＋ｍｖ１．ｘ，ｙ＋ｍｖ１．ｙ）＋Ｃ（３）ここで、Ｐ（ｘ，ｙ）は画素位置（ｘ，ｙ）の予測画像信号、Ｒ（ａ，ｂ，ｃ）はａ番目の参照フレームの画素位置（ｂ，ｃ）の画像信号（画素信号）、Ｗ０，Ｗ２，Ｃは線形予測係数、ｒｅｆ＿ｉｄｘ＿ｉ０及びｒｅｆ＿ｉｄｘ＿ｉ１は参照フレームインデックス、（ｍｖ０．ｘ，ｍｖ０．ｙ）及び（ｍｖ１．ｘ，ｍｖ１．ｙ）は動きベクトルのｘ方向成分及びｙ方向成分をそれぞれ表す。

[００３２]次に、予測画像信号と符号化対象ブロックの信号から予測誤差信号を生成し（ステップＳ１５）、該予測モードにおける符号化コスト計算を行う（ステップＳ１６）。符号化コストがそれまでの最小コストＭｉｎＣｏｓｔを下回れば、最適なブロック形状のインデックスＢｅｓｔＢｌｏｃｋＴｙｐｅと、最適な参照フレームのインデックスＢｅｓｔＲｅｆＦｒａｍｅの更新を行う（ステップＳ１７，Ｓ１８）。

[００３３]次に、参照フレームのインデックスを更新して（ステップＳ１９）、全ての参照フレームに対し再度動き検出及びコスト計算を繰り返す（ステップＳ２０）。１つのブロック形状で各々の参照フレームを用いた予測のコスト計算を行った後、予測対象ブロックのブロック形状を示すインデックスＢｌｏｃｋＴｙｐｅの更新を行い（ステップＳ２１）、再度各々の参照フレームを用いた予測のコスト計算を全ブロック形状について行う（ステップＳ２２）。

[００３４]ステップＳ２２の処理が終了した時点で、その符号化ブロックに対する最適なブロック形状

[0025] 113 prediction mode information table table showing the mode selected by mode selection table 109 is indexed by the variable-length coding section 116 (index prediction mode table) is encoded as. As another embodiment, the configuration of the reference frame (motion-compensated prediction structure) may be a unique configuration mode is automatically determined according to the prediction table. In that case, the coding mode prediction table index is not required. In either case, when the decryption is the same prediction mode is selected the table that was used during encoding, decryption is performed on the prediction mode information.

[0026] 110 is the selection mode, each block of the predicted input image signal 100, along with the best prediction mode selection table refer to the prediction mode selected by mode selection table 109, the selected outputs a prediction mode, the mode information 112. Here, the prediction mode as described below, is a combination of motion compensation prediction mode and the predicted shape of the block. Prediction mode information 112 indicates the selected mode is outputted from the mode select section 110 is encoded as predicted by the mode index variable length coding unit 116.

[0027] (Coding Tasks) The following describes the procedures, including the motion compensated predictive coding process in this embodiment. First, set the maximum initial value of the minimum cost of coding MinCost (step S10), the index is set to 0 indicates the mode shape of the predicted block BlockType (step S11), set to 0 to identify the reference frame index RefFrame (step S12).

[0028] Next, the predicted block for each input image signal 100, the detection of motion vectors for the shape and frame of reference blocks set (step S13). That is, using the well-known block matching, the area that comprises the most highly correlated with multiple pixels in the frame prediction block to be encoded input image signal 100 (reference block) to explore the reference frame, the predicted detect a motion vector and reference vector connecting the blocks and blocks.

[0029] Next, using the motion vector detected by cutting out a reference block from each of the reference frame was set, subjected to linear prediction reference block operations cut, generating a predicted image signal (step S14). Prediction block to be selected for the two reference frames, one for each frame of

ＢｅｓｔＢｌｏｃｋＴｙｐｅと、最適な参照フレームＢｅｓｔＲｅｆＦｒａｍｅが決定される。動きベクトル、ブロック形状及び参照フレームの識別情報は、通常の予測モードでは符号化ブロック毎にヘッダ情報として符号化される（ステップＳ２３）。

[００３５]すなわち、式（１）～（３）のうちのいずれの計算式（予測モード）で予測画像信号を生成したかを示すモード情報１１２、ステップＳ１１で選択された参照フレームを示す参照フレーム番号（参照フレームインデックス）ｒｅｆ＿ｉｄｘ＿ｌ０，ｒｅｆ＿ｉｄｘ＿ｌ１と線形予測係数Ｗ０，Ｗ１，Ｃを含む参照フレーム／予測係数情報１１５、各参照フレームに対応する動きベクトル（ｍｖ０．ｘ，ｍｖ０．ｙ），（ｍｖ１．ｘ，ｍｖ１．ｙ）を示す動きベクトル情報１１４が直交変換係数情報１１１と共に可変長符号化され、多重化部１１８によって符号化データ１２０に多重化される。

[００３６]なお、後述する予測モードテーブルの選択を行うステップは，図２のフローチャートでは、示されていないが、例えば図２の最初に予測モードテーブル選択ステップを付加してもよい。

[００３７]（動き補償予測方法について）次に、図３～図５を用いて本実施形態における具体的な動き補償予測方法について説明する。本実施形態における動き補償予測予測方法は、時間的に過去の複数の参照フレームのみからの予測を行う方法と、時間的に過去及び未来の複数フレームからの予測を行う方法に大別される。

[００３８]図３は、時間的に過去の複数の参照フレームのみからの予測を行う例であり、フレーム２００は符号化対象フレーム、フレーム２０１，２０２，２０３はそれぞれ参照フレームを示している。ブロック２０４，２０５，２０６は、それぞれ符号化対象の画素ブロックを示している。画素ブロック２０４に対しては、参照フレームがフレーム２０１であることを示す参照フレーム番号の情報と、動きベクトル２０７を用いた一つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。同様に、画素ブロック２０５に対しては、参照フレームがフレーム２０３であることを示す参照フレーム情報と、動きベクトル２０８を用いた一つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。一方、画素ブロック２０６に対しては、参照フレームがフレーム２０１及び２０２であることを示す参照フレーム情報と、動きベクトル２０９及び２１０を用いた２つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。

[００３９]図４は、時間的に過去及び未来の複数フレームからの予測を行う例であり、フレーム２２０は符号化対象フレーム、フレーム２２１，２２

reference, the reference block will be cut out of a total of two.

[0030] Next, operations subjected to linear prediction reference block was excised, generating a predicted image signal (step S14).

[0031] In step S14, the prediction picture signal in the case of one reference block following equation (1) or (2) is calculated according to image signals for the two prediction reference block following equation (3) are calculated according to. P (x, y) = W0xR (ref_idx_l0, x mv0.x, y mv0.y) C (1) P (x, y) = W1xR (ref_idx_l1, x mv1.x, y mv1.y) C (2) P (x, y) = W0xR (ref_idx_l0, x mv0.x, y mv0.y) W1xR (ref_idx_l1, x mv1.x, y mv1.y) C (3) where, P (x, y) is the pixel position (x, y) image signal predicted, R (a, b, c) is the pixel position of the reference frame of the second a (b, c) the image signal (pixel signal), W0, W2, C is the coefficient of linear prediction, ref_idx_l1 and ref_idx_l0 is reference frame index, (mv0.x, mv0.y) and (mv1.x, mv1.y) respectively represent the x-direction component and the y-direction component of the motion vector.

[0032] Next, the prediction error signal to generate a coded signal from a target block and predicted image signal (step S15), to calculate the cost of encoding the prediction mode (step S16). Cost is below the previous minimum MinCost coding cost index and a block shape BestBlockType best to update the reference frame index best BestRefFrame (steps S17, S18).

[0033] Next, update the index of the reference frame (Step S19), repeats the motion again and costing for all reference frames (step S20). After the cost calculations for each frame of reference for prediction in the form of a single block, showing you to update the index BlockType prediction block shape of the target block (step S21), the prediction for each frame of reference re- block shape for all cost calculations performed (step S22).

[0034] at the end of the process in step S22, and the best block to block BestBlockType coded form that is determined BestRefFrame optimal reference frame. Motion vectors and reference frame identification information of a block shape, the prediction mode is usually encoded as header information encoded in each block (step S23).

[0035] That is, (1) to (3) Any calculation of the (prediction mode), the mode information 112 generated the predicted image signal in a

２，２２３はそれぞれ参照フレームを示している。ブロック２２４，２２５，２２６は、それぞれ符号化対象の画素ブロックを示している。画素ブロック２２４に対しては、参照フレームが前方参照フレーム２２１であることを示す参照フレーム情報と、動きベクトル２２７を用いた一つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。画素ブロック２２５に対しては、参照フレームが後方参照フレーム２２３であることを示す参照フレーム情報と、動きベクトル２２８を用いた一つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。一方、画素ブロック２２６に対しては、参照フレームが前方参照フレーム２２２及び後方参照フレーム２２１であることを示す参照フレーム情報と、動きベクトル２２９及び２３０を用いた２つの参照フレームからの動き補償予測であることを示す予測モード情報が符号化される。

[００４０]さらに、本実施形態では動き補償予測を行う単位である予測対象ブロックの形状が一定でなく、例えば１６画素x１６ラインの画素ブロック（以下、これをマクロブロックという）単位あるいはマクロブロックをさらに分割した微小ブロック単位に動き補償予測を行う。ここで予測対象ブロックのブロック形状とは、上記マクロブロックの形状を１６画素x１６ラインで表しているように、ブロックの形（正方形または長方形）のみでなく、大きさも表す。一般的に、画素ブロックの形状はm画素（画像の水平方向の画素数）xnライン（画像の垂直方向のライン数）で表されるが、以後はライン数nを画像の垂直方向の画素数とみなして、mxn画素の表記形式で予測対象ブロックの形状を表すものとする。予測対象ブロックの形状はマクロブロック単位に変更可能であり、予測対象ブロックの形状の情報は上述した予測モード情報と併せて符号化される。

[００４１]具体的には、図５に示すように１６x１６画素のマクロブロックを一つの予測対象ブロックとみなす（ａ）、マクロブロックを１６x８画素または８x１６画素の２つの予測対象ブロックに分割する（ｂ）（ｃ）、マクロブロックを８x８画素の４つの予測対象ブロックに分割する（ｄ）、マクロブロックを８x８画素の４つの画素ブロックに分割した後、さらに各８x８画素ブロック毎に８x８画素の予測対象ブロック、８x４画素または４x８画素の２つの予測対象ブロック、４x４画素の４つの予測対象ブロックのいずれかに分割する（ｅ）、といった各種のブロック形状を選択することが可能である。ここで、図５（ｅ）の８x８画素以下のブロックをサブマクロブロックと呼ぶことにする。

[００４２]従来の動画像符号化方式における予測対象ブロックの形状は、ＭＰＥＧ１では１６x１６画素、ＭＰＥＧ２では１６x１６画素または１６x８画素、ＭＰＥＧ４では１６x１６画素または８x８画素にそれぞれ制限されている。これに対し、本

reference frame that is selected in step S11 numbers indicate the reference frame (reference frame index) ref_idx_l0, W0 and linear prediction coefficients ref_idx_l1, W1, 115 frame prediction factor information / reference that contains C, and each motion vector corresponding to the reference frame (mv0.x, mv0.y), (mv1.x, mv1.y) together with the variable length coding the orthogonal transform coefficient information 111 information 114 indicating the motion vector are multiplexed into encoded data 120 by the multiplexing unit 118.

[0036] In addition, performing a prediction mode selection table below is a flow chart of Figure 2, but not shown, may be added to the first step prediction mode selection table in Figure 2, for example.

[0037] (how motion-compensated prediction), then describe how a specific motion-compensated prediction used in the embodiment in Figure 3 to 5. Prediction motion-compensated prediction method in this embodiment, the only way to make predictions from several past reference frames in time, be divided into multiple frames to make predictions from a time past and future.

[0038] 3 figure is an example of a prediction from multiple reference frames in the past only in terms of time frame to be encoded 200 frames, the frame will show the reference frame, respectively 201,202,203 The. Blocks 204,205,206, which represents the block to be coded for each pixel. For pixel block 204, the reference frame number information indicating that the reference frames 201 frames, encoded prediction mode information indicating that the motion-compensated prediction from one reference frame using motion vectors 207 imposed. Similarly, for the pixel block 205, information indicating that the frame reference frame reference frame 203, the prediction mode information indicating that the motion-compensated prediction from one reference frame using motion vectors 208 be encoded. Meanwhile, the pixel block 206, the information frame of reference frame 201 and 202 indicate that the reference frame, indicating that the predicted motion-compensated prediction from two reference frames and 210 for motion vector 209 mode information is encoded.

[0039] 4 figure is an example of multiple frames to predict the future from the past and the time frame to be encoded 220 frames, the frame is showing reference frames 221,222,223 respectively. Blocks 224,225,226, which represents the block to be coded for each pixel.

実施形態では上述のように１６ｘ１６画素から４ｘ４画素ブロックまでにわたって、予測対象ブロックの形状の種類がより拡張されている。

[００４３]（動き補償予測に関わる情報の符号化方法について） 次に、本実施形態における動き補償予測に関わる各種の情報の具体的な符号化方法について説明する。本実施形態における符号化すべき動き補償予測に関わる情報は、▲１▼参照フレーム番号、▲２▼予測係数情報、▲３▼動きベクトル情報、▲４▼予測対象ブロックの形状を示す情報、▲５▼一つの参照フレームの予測かあるいは２つの参照フレームからの線形予測かを示す情報である。▲１▼▲２▼及び▲３▼については、参照フレーム／予測係数情報１１５及び動きベクトル情報１１４としてそれぞれ独立に符号化する。▲４▼と▲５▼については、両者の組み合わせを表すモード情報１１２（予測モードインデックス）として符号化する。以下、▲１▼▲４▼▲５▼の各情報の符号化方法について述べる。

[００４４]（ａ）参照フレーム番号の符号化方法： まず、前記▲１▼の参照フレーム番号の符号化方法について説明する。符号化対象フレーム毎に、動き補償予測に使用される参照フレームを区別するためのインデックステーブル（以下、参照フレームインデックステーブルという）を２セットを用意して、それぞれ異なる番号付けの定義を行う。参照フレームの番号付けについては、予め定義された規則に従って自動的に設定してもよいし、符号化対象フレーム毎に明示的に番号付けを設定する情報を符号化してもよい。以下、フレーム間予測構造と参照フレームインデックステーブルの例を示す図６及び図７を用いて、参照フレーム番号の符号化方法の具体例について述べる。

[００４５]図６は、既に符号化及び局部復号化された過去の４フレームを参照フレームとするフレーム間予測構造の例を示しており、表示順が「４」のフレームを現フレームである符号化対象フレームとし、表示順が「０」～「３」のフレームを参照フレームとしている。この例では、表示順と符号化順は同一である。参照フレームインデックステーブルとしては、参照リスト０と参照リスト１の２セットが用意される。これによれば、例えば符号化対象フレームの直前に符号化された表示順が「３」のフレームは、参照リスト０の「０」番目のフレーム、または参照リスト１の「１」番目のフレームが参照フレームとして用いることを示している。

[００４６]図７は、表示順と符号化順が異なるフレーム間予測構造の例である。実線で表したフレームが符号化対象フレームの場合は、過去の実線で表した複数のフレームを参照フレームとして用いる。破線で表したフレームが符号化対象フレームの場合は、既に符号化済みの実線で表した未来の１フレームと、実線で表した過去の複数フレームを参照フレ

For pixel block 224, and reference frame information indicating that the forward reference frame 221 is the reference frame prediction encoding mode information indicating that the motion-compensated prediction from one reference frame using motion vectors 227 imposed. For pixel block 225, and reference frame information indicating that the backward reference frame reference frame 223, encoded prediction mode information indicating that the motion-compensated prediction from one reference frame using motion vectors 228 imposed. Meanwhile, the pixel block 226, and reference frame information indicating that the backward reference frame 221 and frame 222 is forward reference frame of reference, the motion-compensated prediction from two reference frames and 230 for motion vector 229 The encoded prediction mode information indicating that.

[0040] Further, in this embodiment is not certain that the predicted shape of the block unit to perform motion-compensated prediction block of pixels 16 pixels, for example the x16 line (hereinafter referred to as the macro block it), or macro units performing motion compensation prediction unit is further divided into blocks small blocks. The predicted shape of the block and the block here, as in the x16 line represents the shape of the macro block 16 pixels above the block shape (square or rectangular) as well as representing the magnitude. In general, the shape of the pixel block of pixels m (image horizontal pixel number) xn lines (lines of vertical image) is represented by the subsequent vertical pixels of the image line number n considered to represent the shape of the block and the predicted pixel representation in the form mxn. The shape of the block to be predicted macroblock is possible to change the information in the shape of the predicted block is encoded along with the above prediction mode information.

[0041] Specifically, the predicted one as a block of the macroblock of 16x16 pixels, as shown in Figure 5 (a), divided into two blocks of the predicted pixel 16x8 or 8x16 pixels in a macroblock to (b) (c), into blocks the predicted four-pixel 8x8 macroblocks (d), after splitting into blocks of pixels of four pixels 8x8 macro block pixel 8x8 each pixel block 8x8 each addition block to be predicted, the predicted two blocks of pixels or 4x8 pixels 8x4, divided into one of the predicted four-pixel blocks 4x4 (e), it is possible to select a variety of shapes and blocks. Here, in Figure 5 (e) will be referred to the following block and sub-macroblock of 8x8 pixels.

ームとして用いる。例えば、破線で表したフレームのうち表示順が「８」のフレームが符号化対象フレームの場合は、表示順が「９」（符号化順が「７」）の未来の１フレームと、表示順が「０」，「３」及び「６」の過去の３フレームをそれぞれ参照フレームとして用いる。

[００４７]図７の例においても、参照フレームインデックステーブルは２セット用意されており、それぞれ異なる番号付けが行われる。例えば、参照リスト１の「０」番の参照フレームは表示順が「９」のフレームを示し、参照リスト０の「０」番の参照フレームは表示順が「６」のフレームを示す。

[００４８]参照フレームインデックスは、参照フレームインデックステーブル（参照リスト０及び参照リスト１）によって示される参照フレーム番号を表すインデックスである。参照リスト０及び参照リスト１中にそれぞれ参照フレームが２フレーム以上あった場合、参照フレームインデックスは、参照リスト０中の参照フレームを示すインデックスｒｅｆ＿ｉｄｘ＿ｌ０、参照リスト１中の参照フレームを示すインデックスｒｅｆ＿ｉｄｘ＿ｌ１として、予測対象ブロック単位に符号化される。

[００４９]動き補償予測を一つの参照フレームを用いて行う予測対象ブロックでは、参照フレームインデックスとしてｒｅｆ＿ｉｄｘ＿ｌ０またはｒｅｆ＿ｉｄｘ＿ｌ１のいずれか一方が符号化され、動き補償予測を２つの参照フレームの線形和で行う予測対象ブロックでは、参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０及びｒｅｆ＿ｉｄｘ＿ｌ１の両方が符号化される。ただし、各参照リスト０及び１にそれぞれ含まれる参照フレーム数が２フレーム以上でなければ、明示的に参照フレームインデックスを送る必要が無いため、このような参照フレームインデックスの符号化は行われない。

[００５０]一方、参照リスト０及び参照リスト１にそれぞれ含まれる参照フレーム数を示す情報は、スライス毎にヘッダ情報（スライスヘッダ）として、それぞれｎｕｍ＿ｒｅｆ＿ｉｄｘ＿ｌ０＿ａｃｔｉｖｅ＿ｍｉｎｕｓ１及びｎｕｍ＿ｒｅｆ＿ｉｄｘ＿ｌ１＿ａｃｔｉｖｅ＿ｍｉｎｕｓ１のように符号化される。

[００５１]ここで、図８～図１１を用いて符号化データ１１０の各ヘッダ情報のデータ構造について説明しておく。図８にスライスヘッダのデータ構造の一部を示し、図９にマクロブロックヘッダのデータ構造の一部を示し、図１０及び図１１に予測対象ブロックヘッダ及びサブブロックヘッダのデータ構造を示す。

[００５２]図８のスライスヘッダでは、後述するマクロブロックタイプ情報の予測モードテーブルに関する情報ｍｂ＿ｔｙｐｅ＿ｔａｂｌｅ＿ｉｄｘと、

[0042] The shape of the block to be predicted in conventional video coding scheme is 16x16 pixels in MPEG1, 16x8 pixels or 16x16 pixels in MPEG2, each of which is limited to 16x16 pixels or 8x8 pixels in MPEG4. By contrast, in this embodiment over the 4x4 pixels to 16x16 pixel blocks as described above, are more advanced type of the predicted shape of the block.

[0043] (For information related to the encoding method of motion-compensated prediction), then describe how a specific encoding of information is related to the motion-compensated prediction in this embodiment. Information relating to the motion-compensated prediction is to be encoded in this embodiment, the reference frame number ? 1 ?, information prediction coefficient ? 2 ?, motion vector information ? 3 ?, information indicating the shape of the predicted block ? 4 ?, ? 5 LPC is information indicating whether the reference frame or two from the reference frame or a prediction ?. For ? 3 ? ? 1 ? ? 2 ? and will be independently coded information 114 as information 115 and motion vector prediction coefficients frames / see. ? 4 ? and ? 5 ? for the information 112 represents a combination of both modes (predictive index mode) is coded into. The following describes a method of encoding information for each ? 1 ? ? 4 ? ? 5 ?.

[0044] (a) Reference frame number encoding method: First, the number encoding method will be described ? 1 ? frame of reference of the above. To be encoded for each frame, the index table to distinguish the reference frame used in the motion-compensated prediction (hereinafter referred to reference frame index tables) to provide two sets, to define the different numbering. For the numbering of the reference frame, you may be automatically set according to predefined rules, the information may be coded to explicitly set the numbering to be encoded for each frame. Below, using 6 and 7 shows an example of a frame structure and inter-frame prediction reference index table describes examples of how information encoded frame number.

[0045] 6 Figure shows an example of the structure and inter-frame prediction and frame of reference to the last frame was encoded and already locally decoded, the display order of "4" frame to the current frame and a frame to be encoded, the display order "0" to "3" as a reference frame to frame. In this example, the display order and coding order is the same. Index table as reference frames, there will be two sets of reference list 0 and list 1 reference. According to this, the display order encoded immediately

参照フレーム数を示す情報ｎｕｍ＿ｒｅｆ＿ｉｄｘ＿ｌ０＿ａｃｔｉｖｅ＿ｍｉｎｕｓ１，ｎｕｍ＿ｒｅｆ＿ｉｄｘ＿ｌ１＿ａｃｔｉｖｅ＿ｍｉｎｕｓ１等が符号化される。図９のマクロブロックヘッダでは、予測対象ブロックの形状が８ｘ８画素より大きい場合とそうでない場合とでデータ構造が異なる。前者の場合は、図１０の予測対象ブロックヘッダのデータ構造に従って、参照フレームインデックス（ｒｅｆ＿ｉｄｘ＿ｌ０，ｒｅｆ＿ｉｄｘ＿ｌ１）と動きベクトル情報（ｍｖｄ＿ｌ０，ｍｖｄ＿ｌ１）が符号化される。後者の場合は、図１１のサブブロックヘッダのデータ構造に従って、参照フレームインデックス及び動きベクトル情報が符号化される。

[００５３]（ｂ）予測対象ブロックの形状を示す情報と、一つの参照フレームからの予測かあるいは２つの参照フレームからの線形和による予測かを示す情報の符号化方法：前記▲４▼の予測対象ブロックの形状を示す情報と、前記▲５▼の一つの参照フレームの予測かあるいは２つの参照フレームからの線形和による予測かを示す情報は、図９中に示されるマクロブロックタイプｍｂ＿ｔｙｐｅ及び図１１中に示されるサブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅを表すインデックス（これを予測モードインデックスという）を用いて符号化される。

[００５４]マクロブロックタイプｍｂ＿ｔｙｐｅを表す予測モードインデックスは、表１に示す予測モードテーブルと表２に示す予測モードテーブルのいずれか一方を用いて符号化される。

[００５５][表１]

[００５６][表２]

[００５７]表１及び表２に示す予測モードテーブルは、マクロブロックタイプ値Ｖａｌｕｅ ｏｆｍｂ＿＿ｔｙｐｅ、マクロブロックタイプ名Ｍａｃｒｏｂｌｏｃｋ ｔｙｐｅ ｍｂ＿ｔｙｐｅ ｎａｍｅ、マクロブロック分割数ｎｕｍ＿ｍｂ＿ｐａｒｔｉｔｉｏｎ（）、２種類のマクロブロック分割予測モードｍｂ＿ｐａｒｔｉｔｉｏｎ＿ｐｒｅｄ＿ｍｏｄｅ（，１），ｍｂ＿ｐａｒｔｉｔｉｏｎ＿ｐｒｅｄ＿ｍｏｄｅ（，２）及びサブブロック数ｎｕｍ＿ｓｕｂ＿ｂｌｏｃｋｓ（）の各項目を対応付けている。

[００５８]ここで、マクロブロックタイプ値Ｖａｌｕｅ ｏｆ ｍｂ＿ｔｙｐｅが予測モードインデックスに相当する。マクロブロックタイプ名Ｍａｃｒｏｂｌｏｃｋ ｔｙｐｅ ｍｂ＿ｔｙｐｅ ｎａｍｅは、予測対象ブロックの予測モード及び予測対象ブロックの形状を表しており、Ｄｉｒｅｃｔはダイレクトモード符号化モードを意味し、Ｐｒｅｄは一つの参照フレームからの動き補償予測、ＢｉＰｒｅｄは２つの参照フレームからの動き補償予測をそれぞれ意味

before the frame to be encoded, for example, "3" frame, the list 0 reference "0" th frame, the list 1 reference or "1" th frame have shown to be used as reference frame.

[0046] 7 shows an example of inter-frame prediction structure different coding order and display order. If the frame to be encoded is represented by a solid line frame, multiple frames are used as reference frames in the past represented by solid lines. If the frame is a frame to be encoded is represented by dashed and solid lines in one frame of the future representation already encoded, using multiple frames as reference frames in the past represented by the solid line. For example, the display order of frames represented by a dashed line, "8" frame to be encoded frames, the display order "9" (coded order "7") and one frame of the future, the display order "0", "3" and "6" is used as a reference frame of the past three frames, respectively.

[0047] In the example in Figure 7, the reference frame index table are two sets is performed with different numbers. For example, see Listing 1: "0" reference frame of the display order number is "9" represents a frame of reference list 0 "0" reference frame of the display order number is "6" indicates the frame.

[0048] The reference frame index, reference frame index table (see list 0 and list 1 reference) is an index representing the frame number indicated by the reference. If more than two frames each in the list 1 reference frame and a reference list 0 reference frame reference index, an index indicating the reference frame ref_idx_l0 in the list 0 reference frame of reference as the index ref_idx_l1 in Listing 1 shows the reference blocks are encoded into being forecast.

[0049] In the block to be predicted to do by using a reference frame of one motion-compensated prediction is encoded either ref_idx_l1 or ref_idx_l0 as a frame reference index, a linear sum of the reference frame of the two motion-compensated prediction done in the predicted block is encoded and ref_idx_l1 both the reference frame index ref_idx_l0. However, the number of frames must be at least two reference frames are included for each reference list 0 and 1, respectively, since there is no need to send an explicit reference frame index, index of encoding frame is no such reference.

[0050] On the other hand, contains information indicating the number of each reference frame list 0 and list 1 reference reference, the header information for each slice (the slice header), as is

する。ＬＯは、動き補償予測に用いる参照フレームが参照リスト０の参照フレームであることを意味する。ｉ６ｘｉ６，　１６ｘ８，　８ｘｉ６，　８ｘ８は、図５（ａ）（ｂ）（ｃ）（ｄ）で説明した予測対象ブロックの形状を意味する。なお、ダイレクト符号化モードは例えば双方向フレーム間予測符号化モード（いわゆるＢピクチャの符号化モード）であり、最も時間的に近いフレーム内符号化フレーム（Ｉピクチャ）、またはフレーム間予測符号化フレーム（Ｐピクチャ）のフレーム上の空間的に同じ位置のマクロブロックを予測元として、動き補償予測情報を生成するモードである。

[００５９]マクロブロック分割数ｎｕｍ＿ｍｂ＿ｐａｒｔｉｔｉｏｎ（）は、予測対象ブロックがマクロブロックをいくつ分割しているかを表し、予測対象ブロックの形状に対応する。マクロブロック分割予測モードｍｂ＿ｐａｒｔｉｔｉｏｎ＿ｐｒｅｄ＿ｍｏｄｅ（，１），ｍｂ＿ｐａｒｔｉｔｉｏｎ＿ｐｒｅｄ＿ｍｏｄｅ（，２）は、予測対象ブロックの予測モードを表し、使用している記号の意味は上述の通りである。サブブロック数ｎｕｍ＿ｓｕｂ＿ｂｌｏｃｋｓ（）は、予測対象ブロックに含まれるサブブロックの数を表す。

[００６０]一方、サブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅを示す予測モードインデックスは、表３に示す予測モードテーブルと表４に示す予測モードテーブルのいずれか一方を用いて符号化される。

[００６１][表３]

[００６２][表４]

[００６３]表３及び表４に示す予測モードテーブルは、サブマクロブロックタイプ値Ｖａｌｕｅ　ｏｆ　ｓｕｂ＿ｍｂ＿ｔｙｐｅ、サブマクロブロックタイプ名Ｎａｍｅ　ｏｆ　ｓｕｂ＿ｍｂ＿ｔｙｐｅ、サブマクロブロック分割数ｎｕｍ＿ｓｕｂ＿ｍｂ＿ｐａｒｔｉｔｉｏｎ（）、サブマクロブロック予測モードｓｕｂ＿ｍｂ＿ｐｒｅｄ＿ｍｏｄｅ（）、サブマクロブロックイントラ分割数ｎｕｍ＿ｓｕｂ＿ｍｂ＿ｉｎｔｒａ＿ｐａｒｔｉｔｉｏｎ（）及びサブブロック数ｎｕｍ＿ｓｕｂ＿ｂｌｏｃｋｓ（）を対応付けている。

[００６４]ここで、サブマクロブロックタイプ値Ｖａｌｕｅ　ｏｆ　ｓｕｂ＿ｍｂ＿ｔｙｐｅが予測モードインデックスに相当する。サブマクロブロックタイプ名Ｎａｍｅ　ｏｆ　ｓｕｂ＿ｍｂ＿ｔｙｐｅは、予測対象ブロックの予測モード及び予測対象ブロックの形状を表し、Ｄｉｒｅｃｔはダイレクトモード符号化モードを意味し、Ｐｒｅｄは一つの参照フレームからの動き補償予測、ＢｉＰｒｅｄは２つの参照フレームからの動き補償予測、Ｉｎｔｒａはイントラ予測をそれぞれ意味する。ＬＯ，　Ｌ１は、それ

encoded as num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 respectively.

[0051] Here, you describe the data structure of encoded data 110 the header information for each of Figures 8 through 11. Illustrates some of the data structure of a slice header Figure 8 illustrates some of the macro-block header data structure of Figure 9 shows the data structure of the predicted sub-block header and the header block 10 and 11.

[0052] Figure 8 is the slice header, and information about the mode table mb_type_table_idx prediction macroblock type information later, num_ref_idx_l0_active_minus1 information indicating the number of reference frames is encoded as num_ref_idx_l1_active_minus1. Macroblock header in Figure 9, with a different data structure, otherwise the shape of 8x8 pixels is greater than the predicted block. In the former case, according to the predicted structure of the data block header 10, the reference frame index (ref_idx_l0, ref_idx_l1) and motion vector information (mvd_l0, mvd_l1) will be encoded. In the latter case, according to the sub-block header data structure 11, which is encoded with reference frame index and motion vector information.

[0053] (b) information indicating the shape of the predicted block, information indicating the encoding method of prediction from the linear sum of two reference frames or reference frames or a prediction from: ? 4 above information indicating the shape of the block being predicted ?, information indicating whether the predictions of the linear sum of the reference frame or two or predict the reference frame of one of the ? 5 ? wherein the macroblock type shown in Figure 9. index of the sub-macroblock types and 11 are shown in sub_mb_type mb_type (predictive index of this mode) is coded using.

[0054] represents the prediction mode macroblock type mb_type index is encoded using either prediction mode table shown in Table 2 and the prediction mode table shown in Table 1.

[0055] [1 table]

[0056] [2 tables]

[0057] prediction mode table shown in Table 1 and Table 2, Value ofmb_type the macroblock type, Macroblock type mb_type name macroblock type name, num_mb_partition macroblock division number (), divide the macroblock

ぞれ動き補償予測に参照リスト０及び参照リスト１の参照フレームを用いることを意味する。８ｘ８，８ｘ４，４ｘ８，４ｘ４は、図５（ｅ）で説明した予測対象ブロックの形状を意味する。

[００６５]サブマクロブロック分割数ｎｕｍ＿ｓｕｂ＿ｍｂ＿ｐａｒｔｉｔｉｏｎ（）は、予測対象ブロックがいくつのサブブロックに分割されているかを表し、予測対象ブロックの形状に対応する。サブマクロブロック予測モードｓｕｂ＿ｍｂ＿ｐｒｅｄ＿ｍｏｄｅ（）は、予測対象ブロックの予測モードを表し、使用している記号の意味は上述の通りである。サブブロックイントラ分割数ｎｕｍ＿ｓｕｂ＿ｍｂ＿ｉｎｔｒａ＿ｐａｒｔｉｔｉｏｎ（）は、イントラ予測対象ブロックに含まれるサブブロックの数を表し、サブブロック数ｎｕｍ＿ｓｕｂ＿ｂｌｏｃｋｓ（）は、予測対象ブロックに含まれるサブブロックの数を表す。

[００６６]（予測モードテーブルを区別する方法について）上述したように、本実施形態では予測モードインデックスは表１～表４の予測モードテーブルのいずれか用いて符号化される。従って、予測モードインデックスの符号化にいずれの予測モードテーブルを用いたかを復号化側で区別できるようにする必要がある。この方法として、以下の二つの態様がある。

[００６７]（１）第１の態様では、予測モードインデックスを表１～表４の予測モードテーブルのいずれか用いて符号化したかを示す情報がマクロブロックタイプテーブルインデックスｍｂ＿ｔｙｐｅ＿ｔａｂｌｅ＿ｉｄｘとして符号化され、これがスライスヘッダに付加される。

[００６８]（２）第２の態様では、予測モードインデックスの符号化に表１～表４の予測モードテーブルのいずれか用いるかを符号化対象フレーム毎にフレーム間予測構造に基づいて、すなわち動き補償予測に用いる参照フレームの構成に基づいて、自動決定して選択する。

[００６９]具体的には、例えば図６の表示順が「４」のフレームが符号化対象フレームである場合のように、符号化対象フレームに対して時間的に過去の複数のフレームのみを参照フレームとする場合は、表２または表４に示す予測モードテーブルを選択する。また、例えば図７の表示順が「８」のフレームが符号化対象フレームである場合のように、符号化対象フレームに対して時間的に未来及び過去の複数のフレームを参照フレームとする場合は、表１または表３に示す予測モードテーブルを選択する。このように使用する予測モードテーブルを参照フレームの構成に基づいて自動決定で選択する場合には、予測モードテーブルを示すテーブルインデックスを明示的に符号化する必要はない。

prediction mode of two mb_partition_pred_mode (, 1), mb_partition_pred_mode (, 2) num_sub_blocks number of sub-blocks and () are associated with each item.

[0058] Here, the index corresponding to the mode predicted macroblock type the Value of mb_type. Macroblock type mb_type name name of the macro block type, which represents the shape of the predicted block and the prediction mode block to be predicted. Direct means the direct mode coding mode, Pred is motion-compensated prediction from one reference frame, BiPred means of motion-compensated prediction from each of the two reference frames. The L0, which means that the list 0 reference frame of reference for predicting the motion-compensated reference frame. 16x16, 16x8, 8x16, 8x8, Figure 5 (a) (b) (c) (d) refers to the predicted shape of the blocks described. In addition, the coding mode for direct bi-directional inter-frame prediction coding mode, for example (the so-called B-picture coding mode), and intra-frame coded frame close to the most time (I picture), or inter-frame prediction coded frame (P-picture) as a source to predict the macroblock of the same spatial location on the frame, a motion-compensated prediction mode information is generated.

[0059] num_mb_partition macroblock division number () denotes whether the couple to split the target macroblock prediction block corresponding to the predicted shape of the block. Mb_partition_pred_mode prediction macroblock mode split (, 1), mb_partition_pred_mode (, 2) represents the prediction mode of the predicted block, the meaning of the symbols used are as described above. Num_sub_blocks number of sub-blocks () represents the number of sub-blocks in the block to be predicted.

[0060] On the other hand, shows the sub-macroblock prediction mode index sub_mb_type type is encoded using either prediction mode table shown in Table 4 prediction mode table shown in Table 3.

[0061] [3 tables]

[0062] [4 tables]

[0063] prediction mode table shown in Table 3 and Table 4, Value of sub_mb_type the sub-macroblock type, Name of sub_mb_type sub-macroblock type, num_sub_mb_partition macroblock sub-division number (), sub-macroblock prediction mode sub_mb_pred_mode (), num_sub_mb_intra_partition intra macroblock

[００７０]（予測モードテーブルについて）次に、表１～表４に示した予測モードテーブルの特徴的な構成について説明する。予測モードインデックス（マクロブロックタイプｍｂ＿ｔｙｐｅ及びサブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅ）として符号化される情報は、前述したように（１）予測対象ブロックの形状（１６ｘ１６画素、８ｘ１６画素など）を示す情報と、（２）参照リスト０の参照フレームからの動き補償予測（Ｐｒｅｄ＿Ｌ０）か、参照リスト１の参照フレームからの動き補償予測（Ｐｒｅｄ＿Ｌ１）か、あるいは参照リスト０の参照フレームと参照リスト１の参照フレームとの線形和による動き補償予測（ＢｉＰｒｅｄ）かを区別する情報である。これらの情報に応じて、図１０及び図１１に示したヘッダデータのデータ構造に示されるように、参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０，ｒｅｆ＿ｉｄｘ＿ｌ１の符号化を行うか否かが決定される。

[００７１]図６及び図７で示したように、一つの参照フレームからの動き補償予測において参照フレームを特定する方法としては、同一の参照フレームを特定する場合でも、参照リスト０の参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０を用いる方法と、参照リスト１の参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ１を用いる方法の２通りがある。この２通りの方法を可能とするものが、表１及び表３の予測モードテーブルである。

[００７２]一方、一つの参照フレームからの動き補償予測において参照リスト０の参照フレームインデックスのみを用いることに制限されたものが、表２及び表４の予測モードテーブルである。つまり、表１及び表３の予測モードテーブルでは、参照リスト０の参照フレームからの動き補償予測であるＰｒｅｄ＿Ｌ０と、参照リスト１の参照フレームからの動き補償予測であるＰｒｅｄ＿Ｌ１という２つの予測モードが存在するが、表２及び表４の予測モードテーブルではＰｒｅｄ＿Ｌ０のみ存在し、Ｐｒｅｄ＿Ｌ１は存在しない。

[００７３]マクロブロックタイプｍｂ＿ｔｙｐｅ、サブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅ及び参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０，ｒｅｆ＿ｉｄｘ＿ｌ１などのインデックス情報は、それぞれ可変長符号化等のエントロピー符号化が施される。従って、これらのインデックス情報の値には発生頻度に応じて適切な符号長が割り当てられる。すなわち、発生頻度の高い値には短いコード長、発生頻度の低い値には長い符号長がそれぞれ割り当てられ、これにより符号化効率（圧縮率）を高めることが可能となる。

[００７４]今、各インデックス情報ｍｂ＿ｔｙｐｅ，ｓｕｂ＿ｍｂ＿ｔｙｐｅ及びｒｅｆ＿ｉｄｘ＿ｌ０，ｒｅｆ＿ｉｄｘ＿ｌ１には、それぞれの値が小さいほど短い符号長が割り当てられると仮定する。例えば、図６の例で表示順が「４」の

sub-division number () num_sub_blocks and a number of sub-blocks () is mapped to.

[0064] Here, the index corresponding to the mode predicted value Value of sub_mb_type sub-macroblock type. Name of sub_mb_type sub-macroblock type represents the shape of the predicted block and the prediction mode block to be predicted, Direct means the direct mode coding mode, Pred is a motion-compensated prediction from one reference frame, BiPred is motion-compensated prediction from one reference frame 2, Intra means intra-prediction, respectively. LO, L1, which means use the frame of reference list 0 and list 1 reference motion-compensated prediction in each reference. 8x8, 8x4, 4x8, 4x4, Figure 5 (e) refers to the predicted shape of the blocks described.

[0065] num_sub_mb_partition macroblock sub-division number () represents what has been divided into several sub-blocks of the predicted block, the block corresponding to the predicted shape. Sub_mb_pred_mode sub-macroblock prediction mode () denotes a prediction mode of the predicted block, the meaning of the symbols used are as described above. Intra-sub-block division number num_sub_mb_intra_partition () denotes the number of sub-blocks in the block to be intra-prediction, num_sub_blocks number of sub-blocks () represents the number of sub-blocks in the block to be predicted.

[0066] (how to distinguish between the prediction mode table), as described above, in this embodiment mode, the index is predictive coding mode prediction table using one of Tables 1 through 4. Therefore, it is necessary to be able to distinguish whether the decrypted using any prediction mode table to encode the prediction mode index. This way, there are the following two aspects.

[0067] (1) In a first aspect is encoded as a macro block type information indicating whether the table index mb_type_table_idx encoded prediction mode table for one of Tables 1 to 4 index prediction mode are to be added to the slice headers.

[0068] (2) In a second aspect is based on the predicted structure for each frame between frames to be encoded or predictions for any of the mode table in the Tables 1 to 4 coding mode prediction index Based on the configuration of the reference frame used for motion-compensated prediction, ie, automatic selection and decision.

フレームを符号化する際に、表示順が「３」及び「２」のフレームが参照フレームとして用いられる確率がそれぞれ５０％ずつであり、その他のフレームは参照フレームとして用いられない場合を考える。

[００７５]このような場合、表１または表３の予測モードテーブルを用い、予測モードインデックスとしてマクロブロックタイプｍｂ＿ｔｙｐｅ及びサブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅを符号化する。表示順が「３」のフレームを参照フレームとする予測対象ブロックでは、参照フレームを参照リスト０の０番のフレームとし、参照フレームインデックスとしてｒｅｆ＿ｉｄｘ＿ｌ０＝０を符号化する。表示順が「２」のフレームを参照フレームとする予測対象ブロックでは、参照フレームを参照リスト１の０番のフレームとし、参照フレームインデックスとしてｒｅｆ＿ｉｄｘ＿ｌ１＝０を符号化する。このようにすることで、参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０及びｒｅｆ＿ｉｄｘ＿ｌ１を符号化するための符号量を最低限に抑えることが可能となる。

[００７６]一方、図６の例で表示順が「４」のフレームが符号化対象フレームである場合に、フレーム間距離が符号化対象フレームに近い参照フレームほど選択される確率が高い場合を考える。このような場合は、表２または表４の予測モードテーブルを用い、予測モードインデックスとしてマクロブロックタイプｍｂ＿ｔｙｐｅ及びサブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅを符号化する。表示順が「３」のフレームを参照フレームとする予測対象ブロックでは、参照フレームを参照リスト０の０番のフレームとし、参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０＝０を符号化する。表示順が「２」のフレームを参照フレームとする予測対象ブロックでは、参照フレームを参照リスト０の１番のフレームとし、参照フレームインデックスｒｅｆ＿ｉｄｘ＿ｌ０＝１を符号化する。

[００７７]表２または表４の予測モードテーブルを用いると、表１または表３の予測モードテーブルと比べて予測モードの総数が少ないため、予測モードインデックスであるマクロブロックタイプｍｂ＿ｔｙｐｅ及びサブマクロブロックタイプｓｕｂ＿ｍｂ＿ｔｙｐｅの平均的な符号長を短くすることが可能となり、これらのインデックス情報の符号化効率を向上させることが可能となる。

[００７８]このように予測モード及び使用する参照フレームの選択頻度に応じて、マクロブロックタイプｍｂ＿ｔｙｐｅ及びサブマクロブロックｓｕｂ＿ｍｂ＿ｔｙｐｅの予測モードテーブルを切り替えることで、符号化すべきヘッダ情報の符号化効率を向上させることが可能となる。

[００７９]図６のように過去からの予測に限定され

[0069] More specifically, for example, the display order of 6 '4 "frame to be encoded as if the frame of the past several frames for the frame to be encoded in time If your only frame of reference, select a prediction mode table shown in Table 2 or Table 4. In addition, the display order of 7 for example, "8" as if the frame is a frame to be encoded, if the previous frame and multiple frames of reference in time for future frames to be encoded is to select the prediction mode table shown in Table 1 or Table. If you are determined automatically based on the configuration selected in the reference frame prediction mode table used in this way is not necessary to explicitly encode the prediction mode table shown in the table has no index.

[0070] (table prediction modes) Next, the characteristic describes the configuration of the prediction mode table shown in Tables 1 through 4. Index prediction mode (sub-macroblock type and macroblock type sub_mb_type mb_type) information that is coded as is, as previously described (1) the predicted shape of the block pixels (16x16, pixels, etc. 8x16) indicating the information, (2) motion-compensated prediction from list 0 reference frame of reference (Pred_L0) or motion-compensated prediction from list 1 reference frame of reference (Pred_L1), or a list 1 reference frame and a linear reference frame or reference list 0 reference And by the motion-compensated prediction (BiPred) is to distinguish whether information. Depending on the information, as shown in the data structure of header data are shown in Figures 10 and 11, reference frame index ref_idx_l0, it is determined whether or not the encoding ref_idx_l1.

[0071] As shown in Figures 6 and 7, as a way to identify the reference frames in motion-compensated prediction from one reference frame, even when given the same reference frame, the list 0 reference method using a reference frame index and ref_idx_l0, there are two methods using a reference frame index ref_idx_l1 see Listing 1. What are two ways to enable this, the prediction mode table of Table 1 and Table 3.

[0072] In contrast, those restricted to using only the reference list 0 reference frame index in the motion-compensated prediction from one reference frame, the prediction mode table of Table 2 and Table 4. That is, the prediction mode table in Table 1 and Table 3, and a motion-compensated prediction from Pred_L0 reference list 0 reference frame, there are two prediction modes Pred_L1 a motion-compensated prediction from list 1 reference frame of reference but, in

る場合、一般的な画像では符号化対象フレームに時間的に近いフレームほど符号化対象フレームとの相関が強いため、時間的に近い参照フレームが選択される頻度が高くなる。従って、表２及び表４の参照リスト１による１フレーム予測を除いた予測モードテーブルを強制的に用いる構成としてもよい。

[００８０]図７の例のように過去及び未来に参照フレームが存在する場合は、一般に過去からの予測と未来からの予測が五分五分に行われる可能性が高いため、過去からインデックス付けを行う参照リスト０と未来からインデックス付けを行う参照リスト１とを使い分けることが可能な表１及び表３の予測モードテーブルを強制的に用いる構成としてもよい。

[００８１]このように符号化対象フレームと参照フレームの時間的位置関係によって、使用する予測モードテーブルを一意に決定付けることで、テーブル切り替えを示すインデックスを符号化する必要がなくなり、その分だけ符号化効率を向上させることが可能となる。

[００８２]（復号化側の構成）図１２に、本実施形態に係る動画像復号化装置の構成を示す。図１２に示す動画像復号化装置は、ハードウェアで実現してもよいし、コンピュータを用いてソフトウェアにより実行してもよい。一部の処理をハードウェアで実現し、他の処理をソフトウェアにより行ってもよい。本実施形態の動画像復号化装置は、図１に示した動画像符号化装置に対応した構成となっている。すなわち、図１２の動画像復号化装置には、図１に示した動画像符号化装置から出力される符号化データ１２０が伝送系または蓄積系を経て、符号化データ３００として入力される。

[００８３]動画像復号化装置に入力された符号化データ３００は、入力バッファ３０１及び多重化分離部３０２を経て可変長復号化部３０３に入力される。可変長復号化部３０３では、符号化データ３００に多重化されている可変長符号の復号化が行われ、量子化された直交変換係数情報３１１、モード情報３１２、モードテーブル情報３１３、動きベクトル情報３１４及び参照フレーム／予測係数情報３１５が出力される。

[００８４]量子化された直交変換係数情報３１１は、逆量子化部３０４により逆量子化され、さらに逆直交変換部３０５によって逆直交変換（例えば、逆ＤＣＴ変換）が施されることにより復号化され、予測誤差信号が再生される。予測誤差信号は、加算部３０６によりフレームメモリ／予測画像作成部３０８によって作成された予測画像信号と加算され、再生画像信号３１０が生成される。

[００８５]フレームメモリ予測画像生成部３０８では、モード情報３１２、モードテーブル情報３１３、動きベクトル情報３１４及び参照フレーム／予

prediction mode table shown in Table 4 and Table 2 is only present Pred_L0, Pred_L1 does not exist.

[0073] macro-block type mb_type, reference frame index and the sub-macroblock type ref_idx_l0 sub_mb_type, such as indexing information ref_idx_l1 are subjected to entropy coding such as variable length encoding respectively. Therefore, the value of the index information of these codes are assigned an appropriate length depending on the frequency of occurrence. That is, the higher incidence of short code length, the lower the frequency is assigned a long code length, respectively, which coding efficiency (compression) and can be enhanced.

[0074] Now, mb_type information for each index, ref_idx_l0 and sub_mb_type, to ref_idx_l1 is assumed that the short code length is assigned a smaller value of each. For example, the first example shown in Figure 6, "4" when encoding a frame, the display order "3" and "2" is a 50 percent chance of frames used as reference frames, etc. consider the case frame is not used as reference frame.

[0075] In such a case, using the prediction mode table of Table 1 or Table 3, which encodes the sub-macroblock type and macroblock type mb_type sub_mb_type index prediction modes. Sort the "3" block and the predicted frame is a frame of reference, and the list 0 reference frame 0 of the reference frame that encodes ref_idx_l0 = 0 as a reference frame index. Display order is "2" block and the predicted frame of reference frame, frame # 0 and list 1 reference frame of reference to encode the reference index as a frame ref_idx_l1 = 0. By this way, it becomes possible to minimize the amount of codes for encoding the frame index and ref_idx_l1 ref_idx_l0 reference.

[0076] Meanwhile, the order shown in the example in Figure 6, "4" if the frame to be encoded frames, more likely a reference frame is chosen close to the distance between the frames to be encoded frame consider the case. In such a case, using the prediction mode table shown in Table 4 or Table 2 encoding a sub-macroblock type and macroblock type mb_type sub_mb_type index prediction modes. Sort the "3" block and the predicted frame is a frame of reference, and the list 0 reference frame 0 of the reference frame to encode a reference frame index ref_idx_l0 = 0. Display order is "2" block and the predicted frame is a frame of reference, and the list 0 reference frame of the first of the reference frame to encode a reference frame index ref_idx_l0 = 1.

測係数情報３１５を用いて予測対象ブロックの画像信号を生成する。予測モードは、モードテーブル情報３１３に従って予測モードテーブル３０９の中から適切なテーブルを選択し、受信したモード情報３１２の解読を行うことで決定される。モードテーブル情報３１３が明示的に符号化されていない場合は、参照フレームの時間的な構成、すなわち過去からのみの予測か、あるいは未来にも参照フレームがあるかに応じて予測モードテーブル３０９の切り替えを行う。

[００８６]上述した本発明の符号化及び復号化の処理は、前述したようにハードウェアにより実現してもよいし、処理の一部または全部をコンピュータを用いてソフトウェアにより実行することも可能である。従って、本発明によると上述した動画像符号化及び復号化処理をコンピュータに行わせるための動画像符号化プログラム及び動画像復号化プログラムを提供することも可能である。また、実施形態で説明した予測モードテーブルは、ＲＯＭのようなメモリ上に物理的に記憶されていてもよいが、動画像符号化処理プログラム及び復号化処理プログラム中にいわゆる論理テーブルとして組み込まれてもよい。

[００８７][発明の効果]以上説明したように、本発明によれば予測効率を落とすことなく符号化オーバヘッドの増加を抑えて、より符号化効率の高い動画像の符号化及び復号化を行うことが可能となる。[図面の簡単な説明][図１]本発明の一実施形態に係る動画像符号化装置の構成を示すブロック図 [図２]同実施形態における動き補償予測の手順を示すフローチャート [図３]同実施形態における予測モードの例を示す図 [図４]同実施形態における予測モードの例を示す図 [図５]同実施形態における予測対象ブロックの形状を説明する図 [図６]同実施形態におけるフレーム間予測構造及び参照フレームインデックスの例を示す図 [図７]同実施形態におけるフレーム間予測構造及び参照フレームインデックスの例を示す図 [図８]同実施形態におけるスライスヘッダのデータ構造を示す図 [図９]同実施形態におけるマクロブロックヘッダのデータ構造を示す図 [図１０]同実施形態における予測対象ブロックヘッダのデータ構造を示す図 [図１１]同実施形態における予測サブブロックヘッダのデータ構造を示す図 [図１２]同実施形態に係る動画像復号化装置の構成を示すブロック図 [符号の説明]１００…入力画像信号 １０１…符号化部 １０２…減算部 １０３…直交変換部 １０４…量子化部 １０５…逆量子化部 １０６…逆直交変換部 １０８…フレームメモリ／予測画像作成部 １０９…モードテーブル選択部 １１０…モード選択部 １１１…直交変換係数情報 １１２…モード情報 １１３…モードテーブル情報 １１４…動きベクトル情報 １１５…参照フレーム／予測係数情報 １１６…可変長符号化部 １１７…符号化制御部 １１８…多重化部 １１９…出力バッファ １２０…符号化データ ３００…符号化データ ３０１…入力バッファ ３０２…多重化分離部 ３０３…可変長復号化部 ３０４…逆量子化部 ３

[0077] With the prediction mode table of Table 2 or Table 4, less than the total number of prediction modes and prediction mode table of Table 1 or Table 3, the sub-index mode and a prediction macroblock type mb_type becomes possible to shorten the average code length of the macro block type sub_mb_type, which can improve the coding efficiency of the information these indexes.

[0078] Depending on the frequency and choice of reference frame prediction mode to be used this way, by switching the mode table prediction macroblock type and sub-macroblock sub_mb_type mb_type, the coding efficiency of the header information to be coded and can improve.

[0079] If the prediction is limited to the past as in Figure 6, the general picture is strongly correlated with the frame to be encoded so close to the time frame to be encoded in a frame, temporal increase the frequency of the reference frame is chosen close to. Therefore, the configuration may be used to force the frame prediction mode, prediction table with the list except for one reference in Table 2 and Table 4.

[0080] If there is a reference to the past and future frames as shown in Figure 7 are likely to be done in order to predict the future upsides from the past to predict and generally last may be configured to force the mode table for prediction of Table 1 and Table 3 can be sure you use a reference list from the future to do indexing and indexing is performed from the reference list 0.

[0081] The relative position of the reference frame and time frame to be encoded in this way, by placing a table uniquely determine the prediction mode to be used, eliminating the need to encode the index of the switching table, the which can only improve the coding efficiency minutes.

[0082] (Configuring the decryption side) Figure 12 shows the configuration of the moving image decoding apparatus according to this embodiment. Video decoding apparatus shown in Figure 12, may be realized in hardware, software may be executed by using a computer. Implemented in hardware and some processing may be carried out by other software processes. Video decoding apparatus of the present embodiment, it is configured and compatible with the video encoding apparatus shown in Figure 1. That is, the moving picture decoding device 12, the transmission system or storage system via a coded data 120 is output from the video encoding apparatus shown in Figure 1, are input as coded

０５…逆直交変換部　３０８…フレームメモリ／予測画像作成部　３０９…モードテーブル切替部　３１０…再生画像信号　３１１…直交変換係数情報　３１２…モード情報　３１３…モードテーブル情報　３１４…動きベクトル情報　３１５…参照フレーム／予測係数情報

data 300.

[0083] The encoded data 300 input to the video decoding device is input to variable-length decoding unit 303 through the multiplexing separation unit 302 and input buffer 301. Variable-length decoding unit 303, variable length code decoding is performed that is multiplexed into encoded data 300, 311 quantized orthogonal transform coefficient information 312, mode information, mode information table 313, motion vector information prediction coefficient information is output 315 frames / 314 and see.

[0084] 311 information that is quantized orthogonal transform coefficients are inverse quantized by inverse quantizer 304, inverse orthogonal transform by the inverse orthogonal transformation unit 305 and (for example, the inverse DCT transform) can be subjected to is decrypted by the prediction error signal is reproduced. Prediction error signal is summed with the predicted image signal that is created by section 308 creates predictive picture frame memory / by adding section 306, the reproduced image signal 310 is generated.

[0085] The prediction image generation unit 308 frame memory 312, mode information, mode information table 313 to produce an image signal of the block to be predicted using frame prediction factor information 315 / 314 motion vector information and reference. Prediction mode, select the appropriate table from the mode table 309 according to the prediction mode information table 313 is determined by performing the decoding of the received mode information 312. If it is not encoded explicitly 313 mode information table is composed of temporal reference frames, or predicted only from the past, ie, switching mode table 309 predictions depending on whether there is a frame of reference to the future or do.

[0086] encoding and decoding processing of the present invention described above, you may be implemented by hardware, as mentioned earlier, you can run by using computer software in whole or in part of the It is also possible. Therefore, it is also possible to provide a decryption program and a video encoding program for causing a computer video of a video encoding and decoding process according to the invention described above. Also, the mode prediction table described embodiments, but may be stored physically in the memory like a ROM, included as a logical table called during program processing and decoding processing program Video Coding MAY.

[0087] [Effects of the Invention] As described above, by suppressing the increase in encoding overhead without degrading the prediction efficiency according to the invention, video encoding and decoding of more efficient coding it is possible to do that. [Brief Description of Drawings] [Figure 1] Block diagram showing the configuration of the video encoding device according to one embodiment of the invention [2] a flowchart showing the steps in the embodiment of motion compensated prediction [3] shows an example of a prediction mode in the embodiment [4] shows an example of a prediction mode in the embodiment [Figure 5] the shape of a block diagram illustrating the embodiment of the predicted [6] the embodiment shows an example of a frame structure and the inter-frame prediction reference index in [7] shows an example of a frame structure and the inter-frame prediction reference index in the embodiment [8] shows the data structure of a slice header in the embodiment Figure [9] a diagram showing the data structure of the macro block header in the embodiment [10] diagram showing the data structure of the predicted block header in the embodiment [11] predicted the data sub-block header in the embodiment diagram showing the structure [12] a block diagram showing the configuration of the moving image decoding apparatus according to the embodiment [Explanation of signs] 100... 101... the input image signal coding section 102... 103... 104 subtracts the orthogonal transformation. quantization unit 105 inverse-quantization section 106... 108... inverse orthogonal transformation unit 109 creates predictive picture frame memory / mode selection table... 110... 111... selected mode orthogonal transform coefficient information 112... 113... best mode information mode table 114... 115... 116 best motion vector information predictive factor of the frame / reference variable-length coding section 117...119 coding control unit 118 multiplexes the output buffer 120... 300 coded data 301 encoded data... input buffer 302. separating unit 303 multiplexes... 305 variable-length decoding unit 304 inverse-quantization inverse orthogonal transformation unit 308... to create predictive image frame memory unit 309 / 311... 310... image signal reproduction mode switching tables orthogonal transform coefficient information. mode information 312... 313... 314... 315 best motion vector information table mode... information frame prediction factor / reference

Cited by: JP2006025414 A2 ; JP2006287315 A2 ; JP2008206211 A2 ; JP2009027762 A2 ; JP2009246975 A2 ; JP2009529845 T2 ; JP2012517186 T2 ; KR100669897 B1 ; KR100829169 B1 ; US7792193 BB ; US7856053 BB ; US7876829 BB ; WO06001485 A1 ; WO06112113 A1 ; WO08123254 A1 ; WO12113197 A1 ;

(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2009-118233
(P2009-118233A)
(43)公開日　平成21年5月28日(2009.5.28)

(51)Int.Cl.		F I			テーマコード（参考）
H04N 7/32	(2006.01)	H04N	7/137	Z	5C059 5C159

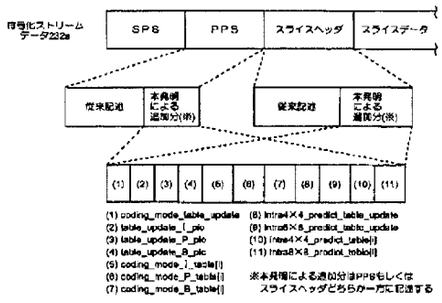
審査請求 未請求　請求項の数 10　OL　（全 18 頁）

(21)出願番号　特願2007-289577 (P2007-289577)
(22)出願日　平成19年11月7日 (2007.11.7)

(71)出願人　000208891
ＫＤＤＩ株式会社
東京都新宿区西新宿二丁目３番２号
(74)代理人　100084870
弁理士　田中　香樹
(74)代理人　100079289
弁理士　平木　道人
(74)代理人　100119688
弁理士　田邉　壽二
(72)発明者　吉野　知伸
埼玉県ふじみ野市大原二丁目１番１５号
株式会社ＫＤＤＩ研究所内
(72)発明者　内藤　整
埼玉県ふじみ野市大原二丁目１番１５号
株式会社ＫＤＤＩ研究所内
最終頁に続く

(54)【発明の名称】動画像符号化装置および復号装置

(57)【要約】
【課題】符号化サイド情報を適応的に抑制することにより、符号化シンタックスにおいて有効活用されない符号量を削減できるようにする動画像符号化装置と復号装置を提供する。
【解決手段】動画像符号化装置には、符号化モード制限情報とイントラ予測方向制限情報が入力される。拡張シンタックス記述部は、これらの制限情報にしたがって、図示のようなビットストリームデータ２３２ａを生成する。つまり、該データ２３２ａは、シーケンスパラメータセット（SPS）、ピクチャパラメータセット（PPS）、スライスヘッダおよびスライスデータで構成され、ピクチャパラメータセット（PPS）は従来のH.264シンタックス記述と本発明による追加分とから、スライスヘッダは従来のH.264シンタックス記述と本発明による追加分とから構成される。ただし、前記本発明による追加分は、前記ピクチャパラメータセット（PPS）およびスライスヘッダのどちらか一方に記述される。
【選択図】図７



【特許請求の範囲】
【請求項1】
マクロブロック単位の符号化の際に、符号化モードおよびイントラ予測方向を適応的に用いる動画像符号化装置において、
符号化モード制限判定部と、イントラ予測方向制限判定部と、符号化モード制限情報に応じて符号化に用いる符号化モードを制限するモード判定制御部と、イントラ予測方向制限情報に応じて符号化に用いるイントラ予測方向を制限するイントラ符号化予測値生成部と、符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部とを具備し、
前記シンタックス記述部は、前記符号化モードおよびイントラ予測方向に関する情報の符号化に用いる符号量を削減することを特徴とする動画像符号化装置。
【請求項2】
請求項１に記載の動画像符号化装置において、
前記符号化モード制限判定部は、外部からの制限情報、もしくは入力映像及び符号化済み符号化データに基づいて、符号化に用いる符号化モードを制限する情報を生成し出力することを特徴とする動画像符号化装置。
【請求項3】
請求項２に記載の動画像符号化装置において、
前記符号化モード制限判定部は、前記入力映像及び符号化済み符号化データに基づいて符号化に用いる符号化モードを制限する方法として、符号化済みフレームにおいて適用された符号化モードについて、発生頻度の上位以外を、処理フレームにおいて制限することを特徴とする動画像符号化装置。
【請求項4】
請求項１に記載の動画像符号化装置において、
前記イントラ予測方向制限判定部は、外部からの制限情報、もしくは入力映像及び符号化済み符号化データに基づいて、イントラ符号化に用いるイントラ予測方向を制限する情報を生成し出力することを特徴とする動画像符号化装置。
【請求項5】
請求項４に記載の動画像符号化装置において、
前記イントラ予測方向制限判定部は、前記入力映像及び符号化済みデータに基づいてイントラ符号化に用いるイントラ予測方向を制限する方法として、符号化済みフレームにおいて適用されたイントラ予測方向について、発生頻度の上位以外を、処理フレームにおいて制限することを特徴とする動画像符号化装置。
【請求項6】
請求項１に記載の動画像符号化装置において、
前記符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部は、符号化モードの制限が適用された場合、符号化データのシンタックスにおいて、符号化モードのインデックステーブルを変更することを表す情報、および変更後の符号化モードのインデックス情報を記述することを特徴とする動画像符号化装置。
【請求項7】
請求項１に記載の動画像符号化装置において、
前記符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部は、イントラ予測方向の制限が適用された場合、符号化データのシンタックスにおいて、イントラ予測方向のインデックステーブルを変更することを表す情報、および変更後のイントラ予測方向のインデックス情報を記述することを特徴とする動画像符号化装置。
【請求項8】
請求項１に記載の動画像符号化装置で符号化された符号化データを復号する動画像復号

装置において、
受信した符号化データからシンタックス記述の解析に関して符号化モードおよびイントラ予測方向の少なくとも一方が制限されているか否かを判断し、制御情報を出力する制限情報解析部と、
前記制限情報解析部から出力された制御情報に従って選択され、シンタックスに従って符号化データを解析し、符号化データから残差信号と予測情報を抽出する第１、第２のシンタックス解析部と、
前記第１、第２のシンタックス解析部から出力された残差信号と予測情報を基に復号映像を生成する手段とを具備し、
前記第１のシンタックス解析部は前記符号化モードおよびイントラ予測方向が制限されていないシンタックスに従って符号化データを解析し、前記第２のシンタックス解析部は前記符号化モードおよびイントラ予測方向の少なくとも一方が制限されているシンタックスに従って符号化データを解析することを特徴とする動画像復号装置。
【請求項9】
請求項８に記載の動画像復号装置において、
前記第２のシンタックス解析部は、前記符号化データに挿入されている識別符号から、適用する符号化モードセットを指定し、該符号化モードセット内の符号化モードもしくはイントラ予測方向を抽出することを特徴とする動画像復号装置。
【請求項10】
請求項８または９に記載の動画像復号装置であって、
請求項２ないし７のいずれかの動画像符号化装置で符号化された符号化データに対して、復号映像を生成することを特徴とする動画像復号装置。
【発明の詳細な説明】
【技術分野】
【0001】
本発明は動画像符号化装置および復号装置に関し、特にマクロブロック(MB)単位の符号化の際に、符号化モードおよびイントラ予測方向について複数の候補の中から最適なものを適用することで、高い符号化効率が得られる動画像符号化装置において、適用する符号化モードおよびイントラ予測方向を制御して、符号化サイド情報を適応的に抑制する動画像符号化装置および復号装置に関するものである。
【背景技術】
【0002】
非特許文献１では、MB単位の符号化に関して複数の符号化モードが定義されており、該非特許文献１に基づく動画像符号化装置では、映像特徴に応じて適切な符号化モードを選択して符号化することで、高い符号化効率が得られる。また、非特許文献１では、イントラ符号化におけるイントラ予測値生成に関して複数のイントラ予測方向が定義されており、非特許文献１に基づく動画像符号化装置では、映像特徴に応じて適切なイントラ予測方向を選択して符号化することで、高い符号化効率が得られる。
【0003】
非特許文献１で定義される符号化モードおよびイントラ予測方向については、様々な映像特徴に適応できるよう、あらゆる種類の符号化モードおよびイントラ予測方向が適用されている。
【非特許文献1】Joint Video Team (JVT) of ISO/IEC MPEG and ITU-VCEG, “Text of ISO/IEC 14496 10 Advanced Video Coding 3rd Edition”, July 2004.
【発明の開示】
【発明が解決しようとする課題】
【0004】
非特許文献１では、様々な映像特徴に対応できるよう、多岐にわたる符号化モードおよびイントラ予測方向が定義されている。しかしながら、ある映像に対する符号化において、実際に適用される符号化モードおよびイントラ予測方向は概ね限られており、必ずしも全ての符号化モードおよびイントラ予測方向は使われない。一方で、非特許文献１に基づ

く動画像符号化装置では、符号化シンタックスにおいて、全ての符号化モードおよびイントラ予測方向が発生しても符号化データを生成できるよう設計される。したがって、該符号化シンタックスにおける符号化データおよびイントラ予測方向に関して、ほとんどの場合において、有効活用されない符号量が発生する。該符号量は、符号化において冗長であり、削減の余地がある。

【0005】

本発明は、前記した従来技術の課題に鑑みてなされたものであり、その目的は、符号化サイド情報を適応的に抑制することにより、符号化シンタックスにおいて有効活用されない符号量を削減できるようにする動画像符号化装置および復号装置を提供することにある。

【課題を解決するための手段】

【0006】

前記した目的を達成するために、本発明は、マクロブロック単位の符号化の際に、符号化モードおよびイントラ予測方向を適応的に用いる動画像符号化装置において、符号化モード制限判定部と、イントラ予測方向制限判定部と、符号化モード制限情報に応じて符号化に用いる符号化モードを制限するモード判定制御部と、イントラ予測方向制限情報に応じて符号化に用いるイントラ予測方向を制限するイントラ符号化予測値生成部と、符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部とを具備した点に第１の特徴がある。

【0007】

また、本発明は、前記符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部は、符号化モードの制限が適用された場合、符号化データのシンタックスにおいて、符号化モードのインデックステーブルを変更することを表す情報、および変更後の符号化モードのインデックス情報を記述する点に第２の特徴がある。

【0008】

また、本発明は、前記符号化モードおよびイントラ予測方向の制限情報に応じてシンタックス記述方法を変更するシンタックス記述部は、イントラ予測方向の制限が適用された場合、符号化データのシンタックスにおいて、イントラ予測方向のインデックステーブルを変更することを表す情報、および変更後のイントラ予測方向のインデックス情報を記述する点に第３の特徴がある。

【0009】

また、本発明は、前記動画像符号化装置で符号化された符号化データを復号する動画像復号装置において、受信した符号化データからシンタックス記述の解析に関して符号化モードおよびイントラ予測方向の少なくとも一方が制限されているか否かを判断し、制御情報を出力する制限情報解析部と、前記制限情報解析部から出力された制御情報に従って選択され、シンタックスに従って符号化データを解析し、符号化データから残差信号と予測情報を抽出する第１、第２のシンタックス解析部と、前記第１、第２のシンタックス解析部から出力された残差信号と予測情報を基に復号映像を生成する手段とを具備し、前記第１のシンタックス解析部は前記符号化モードおよびイントラ予測方向が制限されていないシンタックスに従って符号化データを解析し、前記第２のシンタックス解析部は前記符号化モードおよびイントラ予測方向の少なくとも一方が制限されているシンタックスに従って符号化データを解析する点に第４の特徴がある。

【発明の効果】

【0010】

符号化データを記述するためのシンタックスでは、符号化モードおよびイントラ予測方向に関して、候補となる数に応じた符号量が必要となる。本発明の動画像符号化装置では、映像特徴に適した符号化モードおよびイントラ予測方向のみを候補として用い、符号化に影響しない（予測性能の向上に寄与しない）符号化モードおよびイントラ予測方向は候補としない。この結果、符号化モードおよびイントラ予測方向に関して、各情報を符号化

するために必要となる情報量を削減することができるようになる。また、これにより、ＳＮＲをほとんど低下することなく、符号量を削減することが可能となる。すなわち、符号化効率を改善することができるようになる。

【0011】

また、本発明の動画像復号装置では、前記動画像符号化装置で符号化された符号化データに対して、復号映像を生成することができるようになる。

【発明を実施するための最良の形態】

【0012】

以下に、図面を参照して、本発明を詳細に説明する。本発明は、本発明者等が先に特許出願した動画像符号化装置（特願２００７－８５１４９号）を前提とするものであるので、まず、該動画像符号化装置を図１のブロック図を参照して説明する。

【0013】

動画像符号化装置は、イントラ符号化予測値生成部１、インター符号化予測値生成部２、モード判定制御部３、DCT／量子化部４、逆量子化／逆DCT部５、シンタックス記述部６、エントロピー符号化部７、ローカルメモリ（１）８、ローカルメモリ（２）９とで構成される。

【0014】

前記イントラ符号化予測値生成部１では、入力映像ａの画素値、符号化済みMBにおける予測情報ｃ、符号化済みMBにおける局所復号された画素値ｂを入力とする。また、該イントラ符号化予測値生成部１の出力として、最終的に選択されたイントラ予測方向ｅ、同イントラ予測方向におけるイントラ予測値ｄ、および同イントラ予測方向におけるコスト値ｆを出力する。

【0015】

前記イントラ符号化予測値生成部１の詳細な構成を図２のブロック図を参照して説明する。図示されているように、該イントラ符号化予測値生成部１は、予測値生成部１１、歪み算出部１２、符号量算出部１３、および予測方向決定部１４で構成される。

【0016】

前記予測値生成部１１では、符号化済みブロックにおける局所復号された画素値ｂを入力とし、H.264で定義されるイントラ予測方向ｃに基づき、入力される画素値ａから、予測値１１ａを生成する。該予測値生成部１１の出力として、各予測方向に基づく予測値１１ａを出力する。なお、入力映像の画素値ａに対する、該予測値生成部１１の出力である予測値１１ａの差分を求めることにより、各予測方向に関する残差信号１２ａが得られる。

【0017】

前記歪み算出部１２は、各予測方向に関する残差信号１２ａを入力とし、各予測方向について、各画素における残差信号の二乗をブロック内で総和した値(符号化歪み量)１２ｂを求める。そして、該歪み算出部１２の出力として、各予測方向についての符号化歪み量１２ｂを出力する。

【0018】

前記符号量算出部１３では、各予測方向に関する残差信号１２ａ、処理ブロックに隣接する符号化済みブロックにおいて選択されたイントラ予測方向ｃを入力とし、各予測方向について、入力された残差信号１２ａから、残差信号の符号化に要する符号量１３ｂを求める。また、各予測方向について、入力された符号化済みブロックにおけるイントラ予測方向ｃから、予測方向の符号化に要する符号量１３ｂを求める。上記算出処理の後、各予測方向について、残差信号および予測方向の符号化に要する符号量の合計を求める。該符号量算出部１３の出力として、各予測方向について、符号化に要する符号量１３ｂを出力する。

【0019】

予測方向決定部１４では、各予測方向について、符号化歪み量１２ｂ、イントラ予測値、符号化に要する符号量１３ｂを入力とする。該予測方向決定部１４では、各イントラ予

測方向について、符号化歪み量および符号量からコスト値を算出する。更に、各予測方向のコスト値について大小関係を比較し、最もコスト値が小さい予測方向を求め、同予測方向を最適な予測方向とみなす。該予測方向決定部１４の出力として、最適な予測方向の予測値ｄ、最適な予測方向ｅ、最適な予測方向のコスト値ｆを出力する。

【0020】

図１に戻って説明を続けると、前記インター符号化予測値生成部２では、入力映像の画素値ａ、符号化済みフレームにおける局所復号された画素値ｂ、処理ブロックに隣接する符号化済みブロックにおける動きベクトルｃ’を入力とし、入力映像の画素値ａおよび符号化済みフレームにおける局所復号された画素値ｂから、適切な動きベクトルを求める。また、該動きベクトルに基づき、予測値および残差信号を求める。更に、該残差信号から、符号化歪み量およびインター符号化に要する符号量を算出するとともに、残差信号および符号量からコスト値を得る。上記の結果より、インター符号化予測値生成部２は、その出力として、適切な動きベクトルに対する予測値ｇ、適切な動きベクトルｈ、該動きベクトルｈにおけるコスト値ｉを出力する。

【0021】

前記モード判定制御部３では、イントラ符号化予測値生成部１から出力されるイントラ予測におけるコスト値ｆ、インター符号化予測値生成部２から出力されるインター予測におけるコスト値ｉを入力とし、イントラ予測およびインター予測におけるコスト値を比較し、コスト値が小さい予測を当該ブロックにおける予測とする。該モード判定制御部３の出力として、イントラ符号化およびインター符号化を切り替えるための制御情報ｊを出力する。

【0022】

前記DCT／量子化部４では、入力映像の画素値ａに対する予測値の差分である、残差信号ｋを入力とする。該DCT／量子化部４は、残差信号ｋに対してDCT処理を施す。また、DCT処理によって得られるDCT係数に対して、量子化を施す。該DCT／量子化部４の出力として、量子化されたDCT係数１を出力する。

【0023】

前記逆量子化／逆DCT部５では、量子化されたDCT係数１を入力とする。該逆量子化／逆DCT部５は、量子化されたDCT係数１に対して逆量子化を施す。また、逆量子化の結果に対して、逆DCTを施す。該逆量子化／逆DCT部５の出力として、逆DCTを施した結果を出力する。なお、該逆量子化／逆DCT部５から出力される逆DCT結果に対して、予測値を加えることで、局所復号された画素値ｂを生成する。

【0024】

前記シンタックス記述部６では、量子化されたDCT係数１および当該ブロックで用いられた予測方法(イントラ符号化であればイントラ予測方向、インター符号化であれば動きベクトル)ｃを入力とする。該シンタックス記述部６では、量子化されたDCT係数および予測方向に関する情報に対して、H.264で定められたシンタックス記述方法に従い、シンタックスに記述する内容を決定する。具体的には、各シンタックスのパラメータを決定する。該シンタックス記述部６の出力として、各シンタックスパラメータに対する設定値の情報ｍを出力する。

【0025】

前記エントロピー符号化部７では、シンタックス記述部６から出力される各シンタックスパラメータに対する設定値の情報ｍを入力とし、各シンタックスパラメータにおける設定値に従って、エントロピー符号化を施し、符号化データを生成する。そして、該エントロピー符号化部７の出力として、符号化ストリームｎを出力する。

【0026】

前記ローカルメモリ（１）８では、局所復号された画素値ｂを入力とし、該局所復号された画素値ｂを蓄積し、必要に応じて、適宜、同画素値をイントラ符号化予測値生成部１およびインター符号化予測値生成部２に供給する。

【0027】

前記ローカルメモリ（２）９では、イントラ予測（イントラ予測方向）およびインター予測（動きベクトル）に関する情報ｃ、ｃ’を入力とし、該予測に関する情報ｃ、ｃ’を蓄積し、必要に応じて、適宜、イントラ予測方向ｃをイントラ符号化予測値生成部１に、動きベクトルｃ’をインター符号化予測値生成部２に供給する。

【0028】

図３に、前記動画像符号化装置に対応する動画像復号装置のブロック図を示す。動画像復号装置は、エントロピー復号部３１、シンタックス解析部３２、予測方法判定部３３、イントラ予測値生成部３４、インター予測値生成部３５、逆ＤＣＴ／逆量子化部３６およびメモリ３７で構成される。

【0029】

前記エントロピー復号部３１では、エントロピー符号化された符号化ストリーム３１ａを入力し、入力された符号化ストリームに対してエントロピー復号を施す。そして、該エントロピー復号部３１の出力として、エントロピー復号された符号化データ３１ｂを出力する。

【0030】

前記シンタックス解析部３２では、符号化データを入力とし、Ｈ．２６４で定義されるシンタックスに従ってシンタックス解析を行う。該解析の結果、符号化データから残差信号３２ａおよび予測情報３２ｂを抽出し、該残差信号３２ａおよび予測情報３２ｂを出力する。

【0031】

予測方法判定部３３では、予測情報を入力し、該予測情報について、当該ＭＢに対する予測方法を判別する。そして、該予測方法判定部３３の出力として、イントラ符号化およびインター符号化を制御する情報を出力する。

【0032】

イントラ予測値生成部３４では、復号済み画素値および予測情報（イントラ予測方向）を入力とし、予測情報に従い、復号済み画素値を用いてイントラ予測値を生成する。その出力として、イントラ予測値を出力する。インター予測値生成部３５では、復号済み画素値および予測情報（動きベクトル）を入力とし、予測情報に従い、復号済み画素値を用いてインター予測値を生成する。その出力として、インター予測値を出力する。

【0033】

逆ＤＣＴ／逆量子化部３６では、符号化データから得られる、残差信号に関する量子化されたＤＣＴ係数を入力とし、該量子化されたＤＣＴ係数に対して、逆量子化と逆ＤＣＴを施す。逆ＤＣＴ／逆量子化部３６はその出力として、逆ＤＣＴされた値を出力する。なお、該逆ＤＣＴ／逆量子化部３６の出力と予測値（イントラ予測値もしくはインター予測値）を加算することで、復号映像３７ａが得られる。

【0034】

メモリ３７では、復号映像３７ａの画素値を入力とし、復号済み画素値を蓄積し、イントラ予測値生成部３４およびインター予測値生成部３５に対して、適宜、復号済み画素値を供給する。

【0035】

以上は、本発明の前提となる動画像符号化装置および動画像復号装置の説明であったが、以下に、本発明を詳細に説明する。

【0036】

本発明の動画像符号化装置は、MB単位の符号化の際に、符号化モードおよびイントラ予測方向を適応的に用いる動画像符号化において、映像特徴に応じて候補となる符号化モードおよびイントラ予測方向を制御することで、同情報を符号化するための符号量を削減するものである。具体的には、MB単位の符号化において、あらかじめ符号化モードの候補を取捨選択し、映像特徴に適した符号化モードのみを候補とする。また、イントラ符号化において、あらかじめイントラ予測方向を取捨選択し、映像特徴に適したイントラ予測方向のみを候補とする。

【0037】
図４は、本発明の一実施形態である動画像符号化装置の構成を示すブロック図である。図において、動画像符号化装置は、イントラ符号化予測値生成部２１、インター符号化予測値生成部２、モード判定制御部２２、DCT／量子化部４、逆量子化／逆DCT部５、シンタックス記述部２３、エントロピー符号化部７、ローカルメモリ（１）２４、ローカルメモリ(2)９、符号化モード制限判定部２５、およびイントラ予測方向制限判定部２６で構成される。なお、図４中の図１と同じ符号は、同一の機能を果たす。
【0038】
本実施形態では、後述の説明から明らかになるように、任意に、符号化モードおよびイントラ予測方向に関する制限情報を与えることが可能になる。つまり、符号化サイド情報を適応的に抑制することが可能になる。なお、該符号化モードおよびイントラ予測方向に関する制限情報は、符号化の際に用いることが可能な符号化モードおよびイントラ予測方向に関する情報が記述されているものである。
【0039】
前記符号化モード制限判定部２５では、入力映像の画素値ａ、符号化済みフレームにおける局所復号された画素値ｂを入力とする。また、動画像符号化装置への入力として符号化モードに関する制限情報２５ａが外部から与えられている場合、該情報２５ａを入力とする。符号化モード制限判定部２５では、制限情報２５ａが与えられている場合、該制限情報２５ａに従って、符号化の際に適用可能な符号化モードを決定する。また、制限情報２５ａが与えられていない場合には、入力映像の画素値ａおよび局所復号された画素値ｂから、符号化の際に適用可能な符号化モードを決定する。該符号化モード制限判定部２５の出力として、符号化の際に適用可能な符号化モードに関する情報（符号化モード制限情報）２５ｂを出力する。
【0040】
前記イントラ予測方向制限判定部２６では、入力映像の画素値ａ、符号化済みフレームにおける局所復号された画素値ｂを入力とする。また、動画像符号化装置への入力としてイントラ予測方向に関する制限情報２５ａが外部から与えられている場合、該制限情報２５ａを入力し、該制限情報２５ａに従って、符号化の際に適用可能なイントラ予測方向２６ｂを決定する。また、該制限情報２５ａが与えられていない場合、入力映像の画素値ａおよび局所復号された画素値ｂから、符号化の際に適用可能なイントラ予測方向２６ｂを決定する。該イントラ予測方向制限判定部２６の出力として、符号化の際に適用可能なイントラ予測方向に関する情報（イントラ予測方向制限情報）２６ｂを出力する。
【0041】
図５は、前記イントラ符号化予測値生成部２１の詳細な構成を示すブロック図である。該イントラ符号化予測値生成部２１は、予測値生成部１１、歪み算出部１２、符号量算出部１３、予測方向決定部２１１で構成される。ただし、予測値生成部１１、歪み算出部１２、符号量算出部１３は、図２の同部と同じ機能であるので、説明を省略する。
【0042】
前記予測方向決定部２１１では、イントラ予測方向制限情報２６ｂ、および各予測方向について、符号化歪み量１２ｂ、イントラ予測値１１ａ、および符号化に要する符号量１３ｂを入力とする。該予測方向決定部２１１では、各イントラ予測方向について、符号化歪み量１２ｂおよび符号量１３ｂからコスト値ｆを算出する。更に、各予測方向のコスト値について大小関係を比較し、最もコスト値が小さい予測方向ｅを求め、該予測方向を最適な予測方向ｅとみなす。ただし、コスト値ｆの大小関係を比較する予測方向は、イントラ予測方向制限情報２６ｂにおいて適用可能とされている予測方向に限られる。該予測方向決定部２１１の出力として、最適な予測方向の予測値ｄ、最適な予測方向ｅ、最適な予測方向のコスト値ｆを出力する。
【0043】
図４に戻って説明を続ける。モード判定制御部２２では、イントラ符号化予測値生成部２１から出力されるイントラ予測におけるコスト値ｆ、インター符号化予測値生成部２か

ら出力されるインター予測におけるコスト値ｉ、および符号化モード制限情報２５ｂを入力とする。該モード判定制御部２２では、イントラ予測およびインター予測におけるコスト値ｆ，ｉを比較し、コスト値が小さい予測を当該ブロックにおける予測とする。ただし、符号化モード制限情報２５ｂにおいて適用可能な符号化モードが制限されている場合、適用可能な符号化モードを優先して当該ブロックにおける予測とする。モード判定制御部２２の出力として、イントラ符号化およびインター符号化を切り替えるための制御情報ｊを出力する。

【0044】

ローカルメモリ（１）２４では、局所復号された画素値ｂを入力とし、局所復号された画素値を蓄積し、必要に応じて、適宜、同画素値ｂをイントラ符号化予測値生成部２１、インター符号化予測値生成部２、符号化モード制限判定部２５、およびイントラ予測方向制限判定部２６に供給する。なお、イントラ符号化予測値生成部２１には、処理ブロックに隣接する符号化済みブロックにおける局所復号された画素値ｂを供給し、インター符号化予測値生成部２、符号化モード制限判定部２５およびイントラ予測方向制限判定部２６には、符号化済みフレームにおける局所復号された画素値ｂを供給する。

【0045】

図６に、前記シンタックス記述部２３の詳細な構成のブロック図を示す。該シンタックス記述部２３は、H.264シンタックス記述部２３１、拡張シンタックス記述部２３２、シンタックス記述制御部２３３で構成される。ただし、H.264シンタックス記述部２３１は、従来手法におけるシンタックス記述部と同じ機能である。

【0046】

前記拡張シンタックス記述部２３２では、符号化における予測情報ｃ(イントラ符号化の場合はイントラ予測方向、インター符号化の場合は動きベクトル)、残差信号１、符号化モード制限情報２５ｂ、イントラ予測方向制限情報２６ｂを入力とする。該拡張シンタックス記述部２３２では、量子化されたDCT係数１に対して、H.264で定められたシンタックス記述方法に従い、シンタックスに記述する内容を決定する。また、当該ブロックにおける符号化モードに関する情報については、符号化モード制限情報２５ｂに従い、符号化モードを表現するインデックスを再構成し、再構成後のインデックスに従いシンタックスに記述する内容を決定する。同様に、当該ブロックにおけるイントラ予測方向に関する情報については、イントラ予測方向制限情報２６ｂに従い、イントラ予測方向を表現するインデックスを再構成し、再構成後のインデックスに従いシンタックスに記述する内容を決定する。該拡張シンタックス記述部２３２の出力として、各シンタックスパラメータに対する設定値の情報２３２ａを出力する。

【0047】

前記シンタックス記述制御部２３３では、符号化モード制限情報２５ｂ、イントラ予測方向制限情報２６ｂを入力とし、該符号化モード制限情報２５ｂおよびイントラ予測方向制限情報２６ｂより、符号化モードおよびイントラ予測方向に関して提案手法が適用されているかを判断する。どちらかにおいて提案手法が適用されている場合、シンタックス記述に関して拡張シンタックス記述部２３２から符号化データ２３２ａが得られるよう制御し、どちらにおいても従来手法のみが適用されている場合には、H.264シンタックス記述部２３１から符号化データ２３１ａが得られるように制御する。該符号化データ２３１ａと符号化データ２３２ａには、識別符号が付加または挿入されると共に、適用する符号化モードセット（符号化モードおよびイントラ予測方向）が指定される。

【0048】

次に、前記拡張シンタックス記述部２３２から出力される符号化データ（符号化ストリームデータ）２３２ａの概要を、図７を参照して説明する。符号化ストリームデータ２３２ａは、シーケンスパラメータセット（SPS)、ピクチャパラメータセット（PPS）、スライスヘッダおよびスライスデータで構成される。本発明により、図示されているように、ピクチャパラメータセット（PPS）は従来のH.264シンタックス記述と本発明による追加分とから構成され、スライスヘッダは従来のH.264シンタックス記述と本発明による追加分

とから構成されるようにするが、本発明による追加分は、前記ピクチャパラメータセット（PPS）およびスライスヘッダのどちらか一方に記述するようにする。

【0049】

ここに、前記本発明による追加分は、例えばパラメータ(1)coding_mode_table_update、(2)table_update_I_pic、(3)table_update_P_pic、(4)lable_update_B_pic、(5)coding_mode_I_table[i]、(6)coding_mode_P_table[i]、(7)coding_mode_B_table[i]、(8)intra4x4_predict_table_update、(9)intra8x8_predict_table_update、(10)intra4x4_predict_table[i]、(11)intra8x8_predict_table[i]からなる。なお、前記パラメータ(1)～(7)は、符号化モード制限に関するパラメータであり、前記パラメータ(8)～(11)はイントラ予測方向制限に関するパラメータである。

【0050】

ここで、前記パラメータ(１)～(１１)を説明すると、以下のようになる。

【0051】

(1)coding_mode_table_update　符号化モードを表現する情報に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。

【0052】

(2)table_update_I_pic　coding_mode_table_updateが1のとき、Iピクチャ(Intra16x16)における符号化モードを表現する情報に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。

【0053】

(3)table_update_P_pic　coding_mode_table_updateが1のとき、Pピクチャにおける符号化モードを表現する情報に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。

【0054】

(4)table_update_B_pic　coding_mode_table_updateが1のとき、Bピクチャにおける符号化モードを表現する情報に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。

【0055】

(5)coding_mode_I_table[i]　table_update_I_picが1のとき、Iピクチャ(Intra16x16)における符号化モードを表現する情報に関して、インデックステーブルの変更を表すパラメータ。iは変更後のインデックス番号であり、同パラメータに与えられる値として、同インデックス番号に割り当てられる符号化モードに該当する、従来手法における符号化モードインデックス番号を設定する。

【0056】

(6)coding_mode_P_table[i]　table_update_P_picが1のとき、Pピクチャにおける符号化モードを表現する情報に関して、インデックステーブルの変更を表すパラメータ。iは変更後のインデックス番号であり、同パラメータに与えられる値として、同インデックス番号に割り当てられる符号化モードに該当する、従来手法における符号化モードインデックス番号を設定する。

【0057】

(7)coding_mode_B_table[i]　table_update_B_picが1のとき、Bピクチャにおける符号化モードを表現する情報に関して、インデックステーブルの変更を表すパラメータ。iは変更後のインデックス番号であり、同パラメータに与えられる値として、同インデックス番号に割り当てられる符号化モードに該当する、従来手法における符号化モードインデックス番号を設定する。

【0058】
なお、前記(1)coding_mode_table_update、(2)table_update_I_pic、(3)table_update_P_pic、および(4)table_update_B_picについて、0と1を入れ替えても同じ効果が得られる。
【0059】
(8)intra4x4_predict_table_update　Intra4x4のイントラ予測方向に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。なお、0と1を入れ替えても同じ効果が得られる。
【0060】
(9)intra8x8_predict_table_update　Intra8x8のイントラ予測方向に関して、インデックステーブルの変更を行うことを表すフラグ。0であればインデックステーブルの変更を行わず、1であればインデックステーブルの変更を行うことを表す。同パラメータが存在しない場合は、0とする。なお、0と1を入れ替えても同じ効果が得られる。
【0061】
(10)intra4x4_predict_table[i]　Intra4x4のイントラ予測方向に関して、インデックステーブルの変更を表すパラメータ。iは変更後のインデックス番号であり、同パラメータに与えられる値として、同インデックス番号に割り当てるイントラ予測方向に該当する、従来手法におけるイントラ予測方向インデックス番号を設定する。
【0062】
(11)intra8x8_predict_table[i]　Intra8x8のイントラ予測方向に関して、インデックステーブルの変更を表すパラメータ。iは変更後のインデックス番号であり、同パラメータに与えられる値として、同インデックス番号に割り当てるイントラ予測方向に該当する、従来手法におけるイントラ予測方向インデックス番号を設定する。
【0063】
次に、前記した、映像特徴に応じた符号化モード制限情報を作る方法、映像特徴に応じたイントラ予測方向制限情報を作る方法、および該符号化モード制限情報とイントラ予測方向制限情報を用いて符号化した場合のシンタックス記述の具体例について、以下に説明する。
【0064】
〈１〉符号化モード制限情報を作る方法
【0065】
MB単位の符号化における符号化モードの選択に関して、映像特徴に応じた符号化モードのみを候補とする方法を説明する。符号化モードの候補を制限する方法については２つあり、(a)符号化モード制限情報２５ａまたは２５ｂを外部から入力する方法と、(b)映像特徴および符号化済み情報から符号化モード制限情報２５ａまたは２５ｂを決定する方法である。
【0066】
(a)については、入力される符号化モード制限情報に従い、MB単位の符号化において候補とする符号化モードを制御する。なお、符号化モード制限情報の生成方法として、処理フレームに対して従来手法に基づく符号化処理を施し、発生率が高い符号化モードの上位のみを候補とする方法が考えられる。
【0067】
(b)については、まず、処理フレームと直前の同一ピクチャタイプのフレームについて、画素値の平均値、画素値の分散値、フレーム間のグローバル動きから、両フレームの映像特徴が類似しているか否かを判断する。両フレームの映像特徴が類似していると判断された場合、直前の同一ピクチャタイプにおける符号化モードを踏襲する。
【0068】
また、処理フレームにおける画素値の分散値から符号化モードの分割サイズを決定し、処理フレームおよび直前フレームとのグローバル動きからイントラ符号化の適用可否を決

定することで、符号化モード制限情報を生成することが可能である。

【0069】

さらに、処理フレーム、符号化済みの局所復号された画素値、符号化済みの予測情報を用いて、処理フレームにおける符号化モードを適切かつ高速に推定する手法は、例えば下記の［2］～［4］等の文献に示されているように、多数提案されている。これらの文献に示されている手法を用いることにより、処理フレームにおける符号化モードを高速に推定し、発生頻度が高いと推測される符号化モードの上位のみを候補とする符号化モード制限情報を生成することも可能である。

【0070】

〔2〕 Xuan Jing and Lap-Pui Chau, “An efficient inter mode decision approach for H.264 video coding,” IEEE ICME2004, pp.1111-1114, 2004.

〔3〕 Hua Wang and Xiangjun Zhu, “Adaptive motion estimation and mode decision strategy for H.264 encoder,” Proceedings of the 6th World Congress on Intelligent Control and Automation, pp.10323-10327, June 2006.

〔4〕 Andy C. Yu and Graham R. Martin, "Advanced block size selection algorithm for inter frame coding in H.264/MPEG-4 AVC," ICIP2004, pp.95-98, 2004.

【0071】

上記(a)および(b)により生成される符号化モード制限情報に基づき符号化を行う場合には、符号化データのシンタックス記述に関して、従来手法からの変更が求められる。このシンタックスの変更内容は、符号化モードに関するインデックステーブル情報の追加である。

【0072】

符号化モードに関するインデックステーブルの追加に関しては、既に図７を参照して説明したように、ピクチャパラメータセット(PPS)もしくはスライス単位でインデックステーブルを付与する。すなわち、PPSおよびスライスヘッダのいずれか一方のシンタックス定義について、従来技術（従来記述）に対して、前記「本発明による追加分」を追加する。なお、従来記述におけるPPSおよびスライスヘッダのシンタックス定義については、文献〔1〕を参照。

【0073】

以下に、本発明の手法におけるシンタックス記述の具体例を説明する。下記の具体例では、Bピクチャにおける符号化モードの候補として、８×１６符号化モードが有効であると判断される場合の例を示す。

【0074】

従来手法によるＢピクチャでの８×１６符号化モード表現は、図８のようになっており、これらは５ビットで表現される。そこで、本発明による映像特徴に応じた符号化モード制限情報により、前記のパラメータ(1)coding_mode_table_updateが１、(4)table_update_B_picが１とされ、(7)coding_mode_B_table[i]において、［０］＝０，［１］＝５，［２］＝７，［３］＝９，［４］＝１３，［５］＝１５，［６］＝１７，［７］＝１９，［inferred］＝inferredとされたとすると、図９に示されているような制限された符号化モード表現になり、３ビットで表現されることになる。この場合、符号化モードに関するシンタックスを２ビット削減できたことになる。

【0075】

〈２〉イントラ予測方向制限情報を作る方法

【0076】

イントラ符号化におけるイントラ予測方向の選択に関して、映像特徴に応じたイントラ予測方向のみを候補とする方法を示す。イントラ予測方向の候補を制限する方法については２つあり、(a)イントラ予測方向制限情報２６ａまたは２６ｂを外部から入力する方法と、(b)映像特徴および符号化済み情報からイントラ予測方向制限情報２６ａまたは２６ｂを決定する方法である。

【0077】

(a)については、入力されるイントラ予測方向制限情報に従い、イントラ符号化におけるイントラ予測方向を制限する。なお、イントラ予測方向制限情報の生成方法として、処理フレームに対して従来手法に基づくイントラ符号化処理を施し、発生頻度が高いイントラ予測方向の上位のみを候補とする方法が考えられる。

(b)については、まず、処理フレームと直前のイントラピクチャのフレームについて、画素値の平均値、画素値の分散値、フレーム間のグローバル動きから、両フレームの映像特徴が類似しているかを判断する。両フレームの映像特徴が類似していると判断された場合、直前のイントラピクチャにおけるイントラ予測方向を踏襲する。また、処理フレームおよび直前のフレームとのグローバル動きからイントラ予測方向を決定することで、イントラ予測方向制限情報を生成することが可能である。

さらに、処理フレーム、符号化済みの局所復号された画素値、符号化済みの予測情報を用いて、処理フレームにおけるイントラ予測方向を適切かつ高速に推定する手法は、下記の文献［5］［6］等に示されているように、多数提案されている。これらの文献に示されている手法を用いることにより、処理フレームにおけるイントラ予測方向を高速に推定し、発生頻度が高いと推測されるイントラ予測方向の上位のみを候補とするイントラ予測方向制限情報を生成することも可能である。

【0078】

[5] Chao-Hsuing Tseng, Hung-Ming Wang, and Jar-Ferr Yang,” Enhanced intra-4x4 mode decision for H.264/AVC coders,” IEEE CSVT 2006, Vol. 16, No.8, August 2006.

[6] Zhi-Yi Mai, Chun-Ling Yang and Sheng-Li Xie, “Improved best prediction mode selection methods based on structural similarity in H.264 I-frame encoder,” Systems, Man and Cybernetics, 2005.

【0079】

上記(a)および(b)により生成されるイントラ予測方向制限情報に基づき符号化を行った場合、符号化データのシンタックス記述に関して、従来手法からの変更が求められる。このシンタックスの変更内容は、イントラ予測方向に関するインデックステーブル情報の追加、イントラ予測方向の符号化に用いるビット数の低減である。

【0080】

イントラ予測方向に関するインデックステーブルの追加に関しては、既に図７を参照して説明したように、ピクチャパラメータセット（PPS）もしくはスライス単位でインデックステーブルを付与する。すなわち、PPSおよびスライスヘッダのシンタックス定義について、従来技術（従来記述）に対して、前記「本発明による追加分」を追加する。ただし、従来記述におけるPPSおよびスライスヘッダのシンタックス定義については、文献[1]を参照。

【0081】

イントラ予測方向の符号化に用いるビット数の削減については、インデックステーブル変更後のイントラ予測方向を表現しうる最低のビット数を用いるものとする。すなわち、イントラ予測方向の数が2つ以下の場合は1ビット、３つ以上５つ以下の場合は2ビット、6つ以上の場合は3ビットで、イントラ予測方向を表現する。

【0082】

以下に、本発明の手法におけるシンタックス記述の具体例を説明する。この具体例では、Intra4×4もしくはIntra8×8におけるイントラ予測方向の候補として、水平方向予測、DC予測、水平上方向予測の3方向のみとする例を示す。

【0083】

すなわち、従来手法におけるイントラ方向予測は、図１０のようになっており、これらは3ビットで表現される。そこで、本発明による映像特徴に応じたイントラ予測方向制限により、前記のパラメータ(8)intra4x4_predict_table_update が１とされ、(10)intra4x4_predict_table[i]において、［０］＝１，［１］＝８，［２］＝２とされると、図１１に示されているような制限されたイントラ方向予測になり、２ビットで表現されること

になる。この場合、イントラ方向予測に関するシンタックスを１ビット削減できたことになる。

【0084】

次に、本発明の動画像復号装置の一実施形態を、図１２のブロック図を参照して説明する。本実施形態の動画像復号装置は、図３のシンタックス解析部３２を図１２の構成に代えたものであり、他の構成は図３と同様であるので、以下では、図１２のシンタックス記述部についてのみ説明し、他の構成の説明は省略する。

【0085】

制限情報解析部３２３では、エントロピー復号された符号化データ３１ｂを入力とし、符号化データに含まれる本発明手法適用有無に関する識別情報を抽出する。そして、符号化データ３１ｂのシンタックス解析に関して、従来手法または本発明手法のどちらを適用するかを制御する。該制限情報解析部３２３の出力として、符号化データ３１ｂに対するシンタックス解析方法に関する制御情報３２３ａを出力する。該制御情報３２３ａにより、Ｈ．２６４シンタックス解析部３２１と拡張シンタックス解析部３２２の入力側および出力側の選択（切替）が行われる。

【0086】

Ｈ．２６４シンタックス解析部３２１では、Ｈ．２６４で定義されるシンタックスに従って記述された符号化データ３１ｂを入力とし、該シンタックスに従って符号化データを解析して符号化モードセットを指定し、符号化データから残差信号３２ａと予測情報３２ｂとを抽出する。そして、該Ｈ．２６４シンタックス解析部３２１の出力として、残差信号３２ａと予測情報３２ｂとを出力する。

【0087】

拡張シンタックス解析部３２２では、本発明手法で定義されるシンタックスに従って記述された符号化データ３１ｂを入力とし、該シンタックスに従って符号化データを解析して符号化モードセットを指定し、符号化データから残差信号３２ａと予測情報３２ｂとを抽出する。そして、該拡張シンタックス解析部３２２の出力として、残差信号３２ａと予測情報３２ｂとを出力する。

【0088】

前記符号化モード制限に関して、前記拡張シンタックス解析部３２２では、前記(1)coding_mode_table_update に従い、符号化データのシンタックス解析方法を変更する。本発明手法が適用されている場合、前記(2)table_update_I_pic、(3)table_update_P_pic、および(4)table_update_B_picに従い、ピクチャタイプ毎に符号化モードのインデックス番号を再構築する。

【0089】

一方、前記イントラ予測方向制限に関して、前記拡張シンタックス解析部３２２では、前記(8)intra4x4_predict_table_updateおよび(9)intra8x8_predict_table_updateに従い、符号化データのシンタックス解析方法を変更する。本発明手法が適用されている場合、各分割ブロックサイズ毎にイントラ予測方向のインデックス番号を再構築する。

【0090】

本発明者は、本発明の手法による効果を確認するため、イントラ予測方向に関する制限についてのシミュレーション実験を行った。ＩＴＥ　ＨＤ標準動画像(1920画素×1080ライン)を対象として、ＪＭ１２．１に提案手法を実装した。なお、イントラ予測方向の制限情報については、外部から制限情報を入力するものとした。また、制限情報は、実験条件と同等の条件下で、ＪＭ１２．１を用いて符号化した際の、発生頻度が上位３位もしくは上位５位までの予測方向を用いるものとした。その結果、従来手法に対する平均符号量削減について、上位５位までの場合は０．４２％、上位３位までの場合は０．５２％のゲインが得られた。

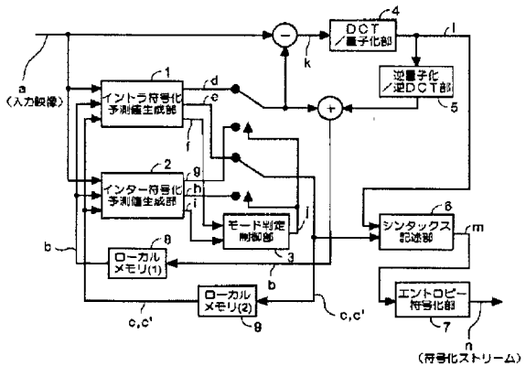
【図面の簡単な説明】

【0091】

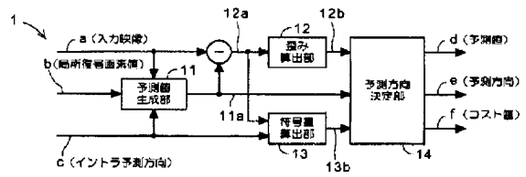
【図1】本発明の前提となる動画像符号化装置の構成を示すブロック図である。

【図2】図１のイントラ符号化予測値生成部の詳細を示すブロック図である。
【図3】本発明の前提となる動画像復号装置の構成を示すブロック図である。
【図4】本発明の一実施形態の動画像符号化装置の構成を示すブロック図である。
【図5】図４のイントラ符号化予測値生成部の詳細を示すブロック図である。
【図6】図４のシンタックス記述部の詳細を示すブロック図である。
【図7】図６の拡張シンタックス記述部の出力を示すビットストリームデータの説明図である。
【図8】Ｂピクチャでの従来の符号化モード表現の説明図である。
【図9】本発明による符号化モード制限に関するシンタックス記述例の説明図である。
【図10】従来手法によるイントラ予測方向表現の説明図である。
【図11】本発明によるイントラ予測方向制限に関するシンタックス記述例の説明図である。
【図12】本発明の動画像復号装置のシンタックス解析部の構成を示すブロック図である。
【符号の説明】
【0092】
２・・・インター符号化予測値生成部、４・・・ＤＣＴ／量子化部、５・・・逆量子化／逆ＤＣＴ部、７・・・エントロピー符号化部、９・・・ローカルメモリ(2)、１１・・・予測値生成部、１２・・・歪み算出部、１３・・・符号量算出部、２１１・・・予測方向決定部、２１・・・イントラ符号化予測値生成部、２２・・・モード判定制御部、２３・・・シンタックス記述部、２４・・・ローカルメモリ(1)、２５・・・符号化モード制限判定部、２６・・・イントラ予測方向制限判定部、２３１・・・Ｈ．２６４シンタックス記述部、２３２・・・拡張シンタックス記述部、３１・・・エントロピー復号部、３２・・・シンタックス解析部、３３・・・予測方法判定部、３４・・・イントラ予測値生成部、３５・・・インター予測値生成部、３２１・・・H.264シンタックス解析部、３２２・・・拡張シンタックス解析部、３２３・・・制限情報解析部。

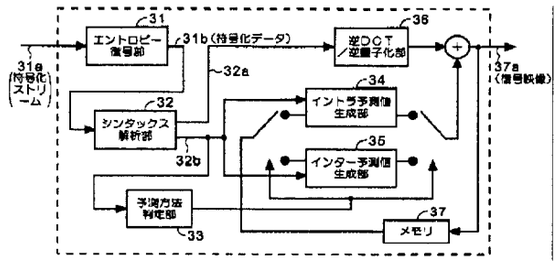
【図1】



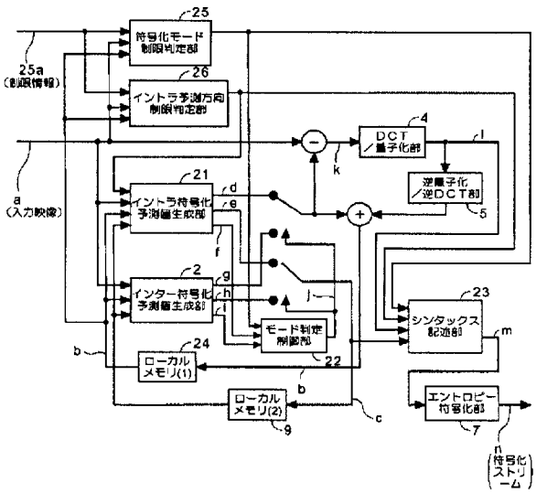
【図2】



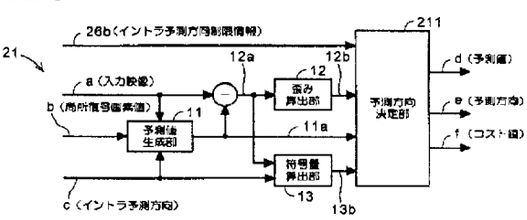
【図3】



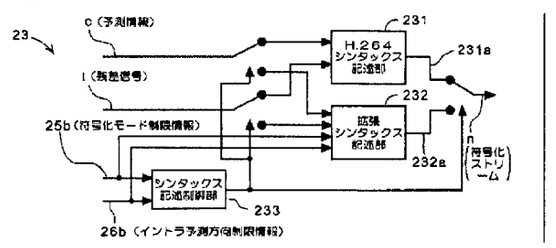
【図4】



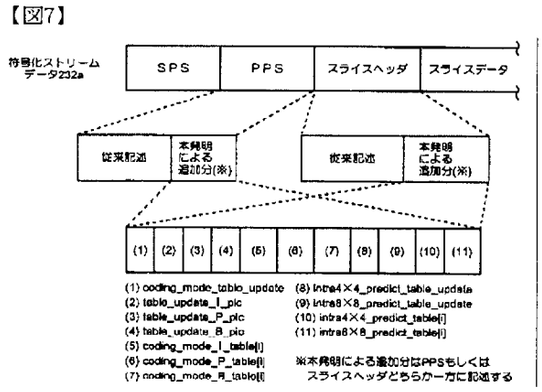
【図5】



【図6】



【図7】



【図8】

（Bピクチャでの符号化モード表現（従来技術））

インデックス番号	符号化モード
0	B_Direct_16×16
1	B_L0_16×16
2	B_L1_16×16
3	B_Bi_16×16
4	B_L0_L0_16×8
5	B_L0_L0_8×16
6	B_L1_L1_16×8
7	B_L1_L1_8×16
8	B_L0_L1_16×8
9	B_L0_L1_8×16
10	B_L1_L0_16×8
11	B_L1_L0_8×16
12	B_L0_Bi_16×8
13	B_L0_Bi_8×16
14	B_L1_Bi_16×8
15	B_L1_Bi_8×16
16	B_Bi_L0_16×8
17	B_Bi_L0_8×16
18	B_Bi_L1_16×8
19	B_Bi_L1_8×16
20	B_Bi_Bi_16×8
21	B_Bi_Bi_8×16
22	B_8×8
Inferred	B_Skip

【図9】

（符号化モード制限に関するシンタックス）

インデックス番号	符号化モード
0	B_Direct_16×16
1	B_L0_L0_8×16
2	B_L1_L1_8×16
3	B_L0_L1_8×16
4	B_L0_Bi_8×16
5	B_L1_Bi_8×16
6	B_Bi_L0_8×16
7	B_Bi_L1_8×16
Inferred	B_Skip

【図10】

（従来手法における イントラ予測方向）

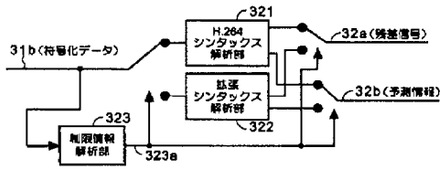
インデックス番号	予測方向
0	垂直方向予測
1	水平方向予測
2	DC予測
3	左下方向予測
4	右下方向予測
5	垂直右方向予測
6	水平下方向予測
7	垂直左方向予測
8	水平上方向予測

【図11】

（イントラ予測方向制限に関するシンタックス）

インデックス番号	予測方向
0	水平方向予測
1	水平上方向予測
2	DC予測

【図12】



(72)発明者　小池　淳
埼玉県ふじみ野市大原二丁目１番１５号　株式会社ＫＤＤＩ研究所内

Ｆターム(参考) 5C059 MA00 MA04 MA23 MC11 MC38 ME01 RC12 RC37 TA27 TC03
TD11 UA02
5C159 MA00 MA04 MA23 MC11 MC38 ME01 RC12 RC37 TA32 TC03
TD11 UA02

Patent Record Full View Thursday, August 24 2017 **Derwent Innovation**

Patent/Publication: KR20050263 18A

Key Summary Data

Patent: Dead
Publication Date: 2005-03-15
DWPI Family: Dead
Expiration Date: 2008-12-11 (estimated)
INPADOC Family: Dead
Remaining Life: 0 year(s)
Original Assignee: SAMSUNG ELECTRONICS CO. LTD., KR

Bibliography

DWPI Title

Apparatus and method for encoding and decoding video including intra skip mode, particularly regarding to increasing compression rate in case of frame having large spatial correlation

Original Title

APPARATUS AND METHOD FOR ENCODING AND DECODING VIDEO INCLUDING INTRA SKIP MODE, PARTICULARLY REGARDING TO INCREASING COMPRESSION RATE IN CASE OF FRAME HAVING LARGE SPATIAL CORRELATION

The video encoding decoding apparatus and method including the intra skip mode *(Translation from Clarivate Analytics)*

English Title

VIDEO ENCODING AND DECODING DEVICE COMPRISING INTRA SKIP MODE

Assignee/Applicant

Standardized: **SAMSUNG ELECTRONICS CO LTD**
Original: SAMSUNG ELECTRONICS CO. LTD., KR

Inventor

LEE NamSuk, KR HA HyeongSeok, CA KIM SoYoung, KR

Publication Date (Kind Code)

2005-03-15 (A)

Application Number / Date

KR200363358A / 2003-09-09

Priority Number / Date / Country

KR200363358A / 2003-09-09 / KR

Abstract

Abstract

PURPOSE: An apparatus and a method for encoding and decoding video including an intra skip mode are provided to increase a compression rate in case of a frame having large spatial correlation by using a skip mode in an intra-frame prediction mode as well as an inter-frame prediction mode.
CONSTITUTION: An intra predictor(190) executes intra prediction in a preliminarily decided intra prediction mode for generating a predicted block(P). A residual block generating unit calculates a difference between a macro block to be encoded and the predicted block for generating a residual block (Dn). A transformer(140) transforms the residual block through discrete cosine transform. A quantizer (150) quantizes the residual block for generating a transform coefficient(X). A transform coefficient judging unit(300) judges whether the transform coefficient is 0. If so, a flag setting unit(305) sets up an intra skip flag as 1. An entropy encoder(170) entropy-encodes the intra skip flag for generating bit stream.
© KIPO 2005

The invention relates to the video encoding, the decoding apparatus and method, and more specifically, to the video encoding, which additionally can reduce the compression ratio through the intra skip mode the decoding apparatus and method. According to the invention, provided are the encoding decoding apparatus and the method in which it uses the skip mode in not only the prediction mode but also the intraframe prediction mode between the frame and in that way the in-frame spatial correlation relation can increase the compression ratio in case of the large frame.
(Translation from Clarivate Analytics)

Classes/Indexing

IPC

IPC Code (1-7) **H04N 7/50**
(7)

Current IPC	Invention	Version	Additional	Version
Full	H04N 19/11 H04N 19/132	20140101 20140101	-	-
Main Group	-	-	-	-
Subclass	-	-	-	-

DWPI Manual Codes

Expand DWPI Manual Codes

Legal Status

INPADOC Legal Status

Gazette Date	Code	Description
2008-12-11	WITN -	WITHDRAWAL DUE TO NO REQUEST FOR EXAMINATION

Get Family Legal Status

Family

Family

Expand INPADOC Family (1) **INPADOC Family:** Dead

Claims

Claims

All Claims (32)

Claims (English)

1. The video encoding apparatus comprising: the intra prediction part performing the intra prediction as the video encoding apparatus to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating unit calculating the difference the predicted block and creates the residual block; the discrete cosine transform and the converting portion quantizing and creates the conversion factor the residual block; the skip mode decision part determining the decided operation of the skip mode as the int based on the conversion factor and the entropy encoding part which is the encodes entropy and generates the bitstream the macroblock generated according to the intra skip mode.

2. As for claim 1, the video encoding apparatus wherein it comprises the conversion factor discrimination section in which the conversion factor the skip mode decision part determines 0 recognition and the flag setting part setting up the intra skip flag (I skip flag) the conversion factor is 0 to 1.

3. As for claim 2, the video encoding apparatus in which the intra skip flag is positioned in data field of the macroblock as to syntax.

4. As for claim 3, the video encoding apparatus which is the flag (mb skip flag) wherein the intra skip flag, is the encoding mode of the entropy encoding part indicates one macroblock skip it is the content - base adaptive arithmetic encoding (CABAC,Context-based adaptive arithmetic coding).

5. As for claim 3, the video encoding apparatus which is the flag (mb skip run) wherein the intra skip flag, is the encoding mode of the entropy encoding part indicates the number of the macroblock which is the content - base variable length encoding (CAVLC, Context-based adaptive variable length coding); and is skipped.

6. The other macroblock of the frame containing in the intra skip flag, the target macro block the target macro block the macroblock decompression module restoring the target macro block using the information is included of the skip mode judgement portion: determining whether or not of the intra skip mode based on the intra skip flag transmitted as to the video decoding apparatus and the macroblock which is determined as the intra skip mode; and is already transmitted without the decoding of the transmitted conversion factor. And video decoding apparatus indicative of whether or not it is identical.

7. As for claim 6, the video decoding apparatus in which it includes the macroblock decompression module; and the intra skip flag is set up with the predicted block based on the conversion factor of the residual block (Dn) of the encoded macroblock wherein the intra skip flag in which the skip mode judgement portion is included within the macroblock restores the skip flag discrimination section: macroblock before being encoded according to the same mode as the above-mentioned predetermined intra prediction mode using the generated predicted block in case the intra skip flag is 1 it determines it is 1.

8. As for claim 7, the video decoding apparatus in which the intra skip flag the conversion factor of the residual block (Dn) of the encoded macroblock is set up with the predicted block in 0 in the encoding unit as 1.

9. As for claim 7, the video decoding apparatus in which the above-mentioned predetermined prediction mode is everlastingly fixedly determined as the spec of the encoder and decoder.

10. As for claim 7, the video decoding apparatus in which the above-mentioned predetermined prediction mode is differently set up according to the sequence, the picture, and the slice.

11. As for claim 7, the above-mentioned predetermined prediction mode is the video decoding apparatus which is DC prediction mode.

12. The video encoding apparatus comprising: the intra prediction part performing the intra prediction as the video encoding apparatus to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating unit calculating the difference the predicted block and creates the residual block; the discrete cosine transform and the converting portion quantizing and creates the conversion factor the residual block; the peripheral macro block of the predetermined position and the intra skip mode judgement portion: flag setting part: which sets up the intra skip flag (I skip flag) as 1 it determines the intra skip mode judgement portion uses the intra skip mode and the entropy encoding part which is within data field of the macroblock the encodes entropy and generates the bitstream the intra skip flag determines the availability of the intra skip mode it determines the paper in which the macroblock is identical.

13. As for claim 12, the video encoding apparatus wherein the conversion factor after doing the production, the conversion factor of the residual block (C_R) consisting of difference of the sample of the peripheral macro block (B_N) of the macroblock (B) in which the intra skip mode judgement portion is encoded and the predetermined position determines the availability of the intra skip mode 0 recognition is determined.

14. As for claim 12, the video encoding apparatus which determines the conversion factor of the peripheral macro block (B_N) of the conversion factor of the macroblock (B) in which the intra skip mode judgement portion is encoded and the predetermined position after doing the production whether it is identical with this conversion factor; and determines the availability of the intra skip mode.

15. As for claim 13 or 14, the video encoding apparatus which is the flag (mb skip flag) indicated one macroblock skip the encoding mode of the entropy encoding part the intra skip flag is the content - base adaptive arithmetic encoding (CABAC,Context-based adaptive arithmetic coding).

16. As for claim 13 or 14, the video encoding apparatus which is the flag (mb skip run) instructing the number of the macroblock which the encoding mode of the entropy encoding part the intra skip flag is the content - base variable length encoding (CAVLC, Context-based adaptive variable length coding); and is skipped.

17. The video decoding apparatus comprises the decodes entropy is the entropy decoding part the bitstream in which the macroblock encoded by the predetermined intra prediction mode is included; the skip flag discrimination section which determines whether the intra skip flag included within the macroblock is 1 or not; and the macroblock decompression module which the intra skip flag is 1; and restores the macroblock decoded using the information of the peripheral macro block of the predetermined position. And the video decoding apparatus which the peripheral macro block of the predetermined position is identical with the macroblock in which the intra skip flag is encoded; and is set up in the encoding unit as 1.

18. As for claim 17, the video decoding apparatus wherein the peripheral macro block of the position predetermined with the encoded macroblock determines; and is determined the conversion factor after doing the production, is 0 recognition the conversion factor of the residual block (C_R) which is identical gets accomplished of the above-mentioned encoded macroblock (B) and difference of the sample of the peripheral macro block (B_N) of the predetermined position.

19. As for claim 17, the peripheral macro block information is the sampled-data of the peripheral macro block. And the video decoding apparatus which restores the macroblock since the macroblock converting portion copies the peripheral macro block each sampled-data.

20. As for claim 17, the video decoding apparatus wherein the peripheral macro block of the position predetermined with the encoded macroblock determines after doing the production whether is identical with this conversion factor; and is determined whether or not, is the conversion factor of the peripheral macro block (B_N) of the conversion factor of the above-mentioned encoded macroblock (B) and the predetermined position it is identical.

21. As for claim 18, the peripheral macro block information is the conversion factor of the peripheral macro block. And the video decoding apparatus which restores the macroblock since the macroblock converting portion copies the peripheral macro block conversion factor.

22. As for claim 21, the peripheral macro block information is a part among the conversion factor of the peripheral macro block. And the video decoding apparatus which restores the macroblock since the macroblock converting portion copies the partial conversion factor of the peripheral macro block.

23. As for claim 21, the video decoding apparatus which is the peripheral macro block information is the DC value of each sampled-data.

24. The encoding method comprising: the intra prediction step performing the intra prediction to the predetermined intra prediction mode and creates the predicted block as the video encoding method using the intra skip mode; the encoded macroblock and the residual block generating step calculating the difference the predicted block and creates the residual block; the step of generating the discrete cosine transform and conversion factor it quantizes; the skip mode decision step determining the decided operation of the intra skip mode based on the conversion factor and the encodes entropy is the step the macroblock generated according to the intra skip mode.

25. As for claim 24, the video encoding method wherein it comprises the conversion factor decision step that the conversion factor the skip mode decision step determines 0 recognition and the flag setting step setting up the intra skip flag (I skip flag) the conversion factor is 0 to 1.

26. As for claim 24, the video encoding method wherein it comprises the macroblock (B) in which the skip mode decision step is encoded and the difference of the sample of the peripheral macro block (B_N) of the predetermined position the step of determining the availability of the intra skip mode the conversion factor determines 0 recognition after doing the production.

27. As for claim 24, the video encoding method wherein it comprises the conversion factor of the macroblock (B) in which the skip mode decision step is encoded and the step of determining the availability of the intra skip mode the it is identical with this conversion factor paper is determined after doing the production.

28. A video decoding method comprising the steps of: as the video decoding method using the intra skip mode, it is determined as the skip mode decision step: intra skip mode determining whether or not of the intra skip mode based on the transmitted intra skip flag; including the macroblock reconstitution step which restores the target macro block using the information of the macroblock which is already transmitted without the decoding of the conversion factor which is transmitted; and it shows whether it is identical with the other macroblock of the frame containing in the intra skip flag, the target macro block the target macro block or not.

29. As for claim 28, the video decoding method wherein it comprises the skip flag decision step: the above-mentioned predetermined intra prediction mode which it determines whether the intra skip flag in which the skip mode decision step is included within the macroblock is 1 or not the intra skip flag is 1 and the macroblock reconstitution step restoring the macroblock before being encoded according to the same mode using the generated predicted block.

30. As for claim 28, the video decoding method for comprising the macroblock reconstitution step wherein the intra skip flag included within the macroblock restores the macroblock which the skip flag decision step: intra skip flag which whether is 1 or not determines is 1; and is decoded using the information of the peripheral macro block of the predetermined position.

31. The video decoding method of claim 29, wherein the peripheral macro block information comprises the step that is the sampled-data of the peripheral macro block; and restores the macroblock since the macroblock reconstitution step copies the peripheral macro block each sampled-data.

32. The video decoding method of claim 29, wherein the peripheral macro block information comprises the step that is the conversion factor of the peripheral macro block; and restores the macroblock since the macroblock reconstitution step copies the peripheral macro block conversion factor.

(Translation from Clarivate Analytics)

Description

Background/ Summary

Expand Background/Summary

Drawing Description

Drawing Description

Figures 1 and 2 are the block diagram of the decoding apparatus and encoding performing the operation of the skip mode between the frame being used in the existing video compression technique of the H.264 standard.

Figures 3 and 4 are drawing indicative of the video encoding and decoding apparatus according to the first embodiment of the invention.

Figures 5 and 6 are drawing indicative of the video encoding and decoding apparatus according to the second embodiment of the invention.

Figures 7 and 8 are drawing indicative of the video encoding and decoding apparatus according to the third embodiment of the invention.

Figure 9 is a time flow diagram indicative of the encoding and decoding method according to a preferred embodiment of the present invention.

(Translation from Clarivate Analytics)

Description

Description

The video encoding decoding apparatus and method [Video Encoding and Decoding device comprising intra skip mode] including the intra skip mode**Background Art**

The invention relates to the video encoding, the decoding apparatus and method, and more specifically, to the video encoding, which additionally can reduce the compression ratio through the intra skip mode the decoding apparatus and method.

In video compression technique such as the MPEG4 and H.264 standard, the projection technology of the various is used in order to improve the compressible effect. There can be the macroblock skip mode belonging to the motion prediction technology as such projection technology. This skip mode is the method it does not transmit the information of the macroblock which is given when it has the redundancy in macro block data between the frame and for transmitting only the skip flag of the single bit as the predetermined appointment between the decoder and the encoder.

In the MPEG 4, the result motion vector of the motion prediction of the macroblock which is given among the squeezing operation of the predictive coded picture is 0 and if the transform coefficient value of the quantization this after is 0 the skip flag (not coded) is set up as 1 and it transmits and the other other information does not transmit. In the decoder, the macroblock existing in the position which is identical in the reference picture is copied and the target macro block is reproduced.

In the H.264, the result of the motion prediction of the macroblock, and the motion vector are identical with the motion vector predictor within P slice and 0, the quantization coefficient of the reference index altogether transmits 0 rear side, and the skip flag. In case the encoding mode is CABAC the skip flag is the skip flag (mb skip flag) set up as 1 and in case the encoding mode is the CAVLC) the encoder transmits the number of the macroblock skipped using the skip flag (mb skip run). At this time, in the decoder, the macroblock positioned in the place where the motion vector indicates within the reference picture is copied and the target macro block is reproduced. In case it is B slice the transform coefficient value after the quantization altogether sets up 0 rear side skip flag in the Direct 16*16 mode as 1 and by copying 4*4 block 16 with the method designated in the Direct 16*16 mode in the decoder using the motion vector and reference index the target macro block is reproduced.

Figures 1 and 2 are the block diagram of the decoding apparatus and encoding indicative of the operation of the skip mode (inter skip mode) between the frame being used in the existing video compression technique of the H.264 standard.

By the motion predict (110) receiving the current frame (Fn) and comparing with the reference frame (F'_{n-1}) the motion vector (111) and reference index (112) are created. The motion compensator (120) receives the reference frame (F'_{n-1}), and the motion vector (111) and reference index (112) and the predicted block is created. The residual block generator (130) creates the residual block (Dn) by taking the differential of the reference frame (F'_{n-1}) and current frame (Fn). The residual block (Dn) is quantized with the discrete cosine transform and the discrete cosine transformer (140) and quantizer (150) create the conversion factor (X).

The flag setting part (160) receives the motion vector (111) and reference index (112) from the motion predict and the motion vector and reference index examine 0 recognition. Moreover, the flag setting part (160) the conversion factor (X) of the residual block examines 0 recognition. The motion vector (111), and the reference index (112) and conversion factor (X) set up the skip flag (mb skip flag) in which 0 rear side , and the flag setting part (160) altogether exist within data field of the macroblock of the single bit as 1. The entropy encoding part (170) encodes only this skip flag (mb skip flag) and the bitstream is created but any information about the target macro block does not encode. The generated bitstream is transmitted to the decoder through the transmission media (NAL).

In the decoder, the transmitted bitstream determines whether the skip flag (mb skip flag) existing in data field of the target macro block is with the skip flag discrimination section (165) 1 after it is decoded in the entropy decoding part (175) or not. The play frame (F'n) is created since the skip flag copies the target macro block using 1 rear side, and the information of the previously transmitted reference frame (F'_{n-1}). That is, the prediction is not performed between the frame using the reference frame (F'_{n-1}) which the decoder is already transmitted in order to reproduce the frame (F'n) and becomes with the regeneration (reconstruct).

But according to such prior art, the skip mode is performed about the macroblock of P or B type and the skip mode is not performed about the macroblock of the type I. That is, the opportunity which can obtain the higher compression efficiency since the peripheral macro block within the same picture does not utilize such information even in case of being identical with the target macro block is missed.

Abstract Problem

Therefore, the present invention is to provide apparatus for encoding and decoding and the method which to the utmost increases the compression efficiency by applying the skip mode to the picture of not only P or B type but also the type I.

Invention Configuration

It comprises the intra prediction part which is the invention relates to the video encoding apparatus for achieving the above-described purpose and it performs the intra prediction to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating unit calculating the difference the predicted block and creates the residual block; the discrete cosine transform and the converting portion quantizing and creates the conversion factor the residual block; and the flag setting part setting up the intra skip flag (I skip flag) the conversion factor discrimination section: conversion factor determined is 0 to 1 and the entropy encoding part which is the encodes entropy and generates the bitstream the intra skip flag.

Moreover, the decodes entropy includes entropy decoding part, the skip flag discrimination section, and the macroblock decompression module the bitstream in which the macroblock encoded by the intra prediction mode which as for the video decoding apparatus of the invention, is predetermined for achieving the above-described purpose is included and the intra skip flag the conversion factor of the residual block (Dn) of the encoded macroblock is set up with the predicted block in 0 in the encoding unit as 1. The skip flag discrimination section determines whether the intra skip flag included within the macroblock is 1 or not. The macroblock decompression module restores the macroblock before being encoded according to the same mode as the above-mentioned predetermined intra prediction mode using the generated predicted block in case the intra skip flag is 1.

Moreover, the invention relates to the video encoding apparatus. And it comprises the intra prediction part performing the intra prediction to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating unit calculating the difference the predicted block and creates the residual block; the discrete cosine transform and the converting portion quantizing and creates the conversion factor the residual block; and the intra skip mode judgement portion: flag setting part: which sets up the intra skip flag (I skip flag) as 1 it determines the intra skip mode judgement portion uses the intra skip mode and the entropy encoding part which is within data field of the macroblock the encodes entropy and generates the bitstream the intra skip flag determines the availability of the intra skip mode it determines the peripheral macro block and the paper in which the macroblock is identical of the predetermined position.

In the preferred embodiment, the conversion factor of the residual block (C_R) consisting of difference of the sample of the peripheral macro block (B_N) of the macroblock (B) in which the intra skip mode judgement portion is encoded and the predetermined position the availability of the intra skip mode is determined after doing the production since the conversion factor determines 0 recognition.

In another preferred embodiment, the conversion factor of the peripheral macro block (B_N) of the conversion factor of the macroblock (B) in which the intra skip mode judgement portion is encoded and the predetermined position are determined after doing the production whether it is identical with this conversion factor. In that way the availability of the intra skip mode is determined.

Moreover, the invention relates to the video decoding apparatus. And the decodes entropy includes the entropy decoding part, the skip flag discrimination section, and the macroblock decompression module the bitstream in which the macroblock encoded by the predetermined intra prediction mode is included and when the peripheral macro block of the predetermined position is identical with the macroblock in which the intra skip flag is encoded the position is set up in the encoding unit as 1. The skip flag discrimination section determines whether the intra skip flag included within the macroblock is 1 or not. The macroblock decompression module in case the intra skip flag is 1 restores the macroblock decoded using the information of the peripheral macro block of the predetermined position.

In the preferred embodiment, the peripheral macro block of the predetermined position is identical with the encoded macroblock the conversion factor determines 0 recognition the conversion factor of the residual block (C_R) consisting of the above-mentioned encoded macroblock (B) and difference of the sample of the peripheral macro block (B_N) of the predetermined position after doing the production. In that way it is determined.

Moreover, at this time, the peripheral macro block information is the sampled-data of the peripheral macro block. And the macroblock is restored since the macroblock converting portion copies the peripheral macro block each sampled-data.

In another preferred embodiment, the peripheral macro block of the predetermined position is identical with the encoded macroblock the conversion factor of the peripheral macro block (B_N) of the conversion factor of the above-mentioned encoded macroblock (B) and the predetermined position whether it is identical with this conversion factor. In that way paper are determined after doing the production it is determined.

Moreover, at this time, the peripheral macro block information is the conversion factor of the peripheral macro block. And the macroblock is restored since the macroblock converting portion copies the peripheral macro block conversion factor.

Moreover, it comprises the intra prediction step which is the invention relates to the video encoding decoding method using the intra skip mode for achieving the above-described purpose and it performs the intra prediction to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating step calculating the difference the predicted block and creates the residual block; the step of generating the discrete cosine transform and conversion factor it quantizes; and the skip flag decision step: macroblock reconstitution step which restores the macroblock before being encoded according to the same mode as the above-mentioned predetermined intra prediction mode using the generated predicted block in case the intra skip flag is 1 it determines the intra skip flag included within the flag setting step setting up the intra skip flag (I skip flag) the conversion factor decision step: conversion factor determined is 0 to 1: macroblock is 1.

Moreover, it comprises the intra prediction step which is the invention relates to the video encoding decoding method using the intra skip mode for achieving the above-described purpose and it performs the intra prediction to the predetermined intra prediction mode and creates the predicted block; the encoded macroblock and the residual block generating step calculating the difference the predicted block and creates the residual block; the discrete cosine transform and the conversion step quantizing and creates the conversion factor the residual block; and the macroblock reconstitution step which in case determines that the intra skip mode decision step: intra skip mode judgement portion which determines the availability of the intra skip mode by determining whether the macroblock is identical or not predetermined uses the intra skip mode in case the flag setting step: skip flag decision step: intra skip flag which determines whether the intra skip flag included within the macroblock is 1 or not is the intra skip flag (I skip flag) 1 sets up as 1 restores the macroblock decoded using the information of the peripheral macro block of the predetermined position.

Hereinafter, the principles of the invention is made to be equal to next, it schematically explains.

In the present invention, the intra skip mode used in the prediction process of the video codec is suggested. The intra skip mode names the obtained definite value or the prediction mode copying the specific block in the corresponding I picture and when transmitting data of the given macroblock the prediction mode which the other any information except the skip flag information of the single bit does not transmit is referred to.

In the encoder, the result intra skip flag (I skip flag) of the prediction operation performance of I picture (slice) 0 rear side , and the encoder creates the predicted block (P) like the other time through the intra prediction mode and the residual block (D) is created and the residual block (D) the conventional process of coding including the discrete cosine transform etc. is passed through. Quantizes and creates the conversion factor (X).

But in the encoder, 1 rear side , and the other any information (the prediction mode, the residual block etc) about the target macro block does not code and the result intra skip flag (I skip flag) of the prediction operation performance of I picture (slice) sets up within the macroblock field only the skip flag as 1 and it transmits.

The case where the intra skip flag (I skip flag) is set up as 1 is the force kinds case is made possible. And in case of each, the method for decoding the target macro block in the decoder is changed.

The attached preferred embodiment according to the present invention is hereinafter particularly explained.

Figures 3 and 4 are drawing indicative of the video encoding and decoding apparatus according to the first embodiment of the invention.

The video encoding apparatus according to this embodiment comprises the predicted block generated in the intra prediction mode, the conversion factor discrimination section (300) in which the conversion factor (X) of the residual block showing the difference of the target macro block determines 0 recognition, and the flag setting part (310) which sets up the intra skip flag indicating the intra skip mode as 1 in case the conversion factor is 0.

The frame in which the current frame (Fn) is encoded by the intramode is shown. And this frame is processed by unit of the macroblock. The frame in which the reproduced frame (F'n) is reproduced in the encoding mode according to the salvage pathway is shown. And this is reproduced through the prediction mode which is identical in the decoder. Whenever it is necessary while being stored in the variable delay memory (the not illustrated) in which the current frame (Fn) and play frame (F'n) are altogether called because of being the prediction memory it is recorded with the read.

After the residual block (Dn) showing difference of the predicted block (P) and target macro blocks (Fn, 101) is generated the residual block is quantized with the discrete cosine transform with the discrete cosine transformer (140) and quantizer (150) and the conversion factor (X) is obtained. Here, the predicted block (P) is generated between the encoder and decoder according to the predetermined specific prediction mode. Therefore, the decoder already knows at the encoder whether the predicted block was created to any kind of mode or not.

All conversion factors of the residual block the conversion factor discrimination section (300) determine 0 recognition. If it is altogether not conversion factor (X) of the residual block with 0 this the flag setting part (305) sets up the intra skip flag (I skip flag) as 0. If the intra skip flag is set up as 0 the bitstream which is the encodes entropy and in which the entropy encoder (170) is the conversion factor (X) encoded according to the conventional encoding method is inserted within data field of the macroblock.

The conversion factor (X) of the residual block 0 rear side , and the flag setting part (305) altogether sets up the intra skip flag (I skip flag) as 1. The entropy encoder (170) the intra skip flag is set up as 1 transmits the bitstream it is only the intra skip flag (I skip flag) the encodes entropy to the decoder through the transmission media (NAL) this it generates. The information of the intra skip flag extrinsic is not transmitted about the target macro block.

In the decoder, it is the decodes entropy and the bitstream in which the entropy decoder (175) is received the intra skip flag (I skip flag) is extracted. 1 recognition is determined whether the intra skip flag in which the skip flag discrimination section (305) is extracted is 0 or not. It is the inverse quantization and cosine-transform:v reversely and the intra skip flag generates the residual block (Dn) the conversion factor in which 0 rear side , and the inverse quantizer (155) and inverter (145) are received. The adder (135) adds the predicted block (P) generated with the predetermined prediction mode to this residual block and macroblocks (F'n,103) which are reproduced by filtering low frequency the block (uF'n) in which the filter (180) is added are created.

According to the prediction mode in which the intra skip flag 1 rear side , and the macroblock decompression module (320) is predetermined, the target macro block is reproduced. According to the prediction mode in which the macroblock decompression module (320) is used in the encoder, the predicted block (P) of the target macro block is created and it is considered as the macroblock (321) in which this predicted block (P) is reproduced and it is stored in the prediction memory (103).

In the present preferred embodiment, the redundancy information in the intra prediction mode uses the conversion factor of the residual block. That is, when the discrete cosine transform and the conversion factor after quantizing of the residual block are 0 the target macro block is reproduced in the intra prediction mode through the specific predicted block used. When the conversion factor after quantizing is 0 the intra skip mode is used. Therefore although the sampled-data of the predicted block and sampled-data of the target macro block completely does not coincide with the intra skip mode is used it notes.

Generally, the predetermined above-described prediction mode may be the DC prediction mode. And it more specifically can be the intra 16*16 luma DC prediction mode, and **4*4** luma DC prediction mode or the chroma DC prediction mode. Moreover, the crystallization and prearranged mutual agreement of the prediction mode sequence and it can be selected by unit of the picture or the slice or it can be forever fixed.

Figures 5 and 6 are drawing indicative of the video encoding and decoding apparatus according to the second embodiment of the invention.

As shown in figure 5, the encoder according to this embodiment includes the target macro block and the intra skip mode judgement portion (530) in which the conversion factor (C_R) of the residual block (R) of the peripheral macro block determines 0 recognition. The residual block (R) formed with difference of the intra skip mode judgement portion (530) is the target macro block (B) and peripheral macro block (B_N) is created and the conversion factor (C_R) of the residual block altogether determines 0 recognition. It is distinguished from the conversion factor (X) of the residual block formed with difference of the conversion factor (C_R) of the residual block is the target block and predicted block in the intra prediction mode.

The conversion factor (C_R) of the residual block 0 rear side , and the intra skip mode judgement portion (530) transmits the intra skip set signal (531) with the flag setting part (500) and the flag setting part (500) receiving the intra skip set signal sets up the intra skip flag (I skip flag) as 1. If it is not conversion factor (C_R) of the residual block with 0 this it confronts with the target macro block (B) and the final conversion factor is inserted into data field of the macroblock according to the conventional prediction operation after the production of the residual block (Dn), and the conversion and quantization. At this time, the intra skip flag is set up as 0.

The position of the peripheral macro block is the position agreed between the encoder and decoder in advance. This can be forever fixed and it can be selected by unit of the sequence, and the picture or the slice.

As shown in figure 6, the decoder according to this embodiment includes the skip flag discrimination section (510) in which the intra skip flag included in the target macro block data field determines 1 recognition and the macroblock decompression module (520) restoring the target macro block according to the intra skip flag. In case the intra skip flag is 1 the macroblock decompression module (520) copies the sampled-data (521) of the peripheral macro block and it stores in the position of the prediction memory (103) corresponding to the target macro block. Since the position of the peripheral macro block is the position agreed on the spec of the decoder and encoder the macroblock decompression module (520) has the position about this information to the SPEC value.

Figures 7 and 8 are drawing indicative of the video encoding and decoding apparatus according to the third embodiment of the invention.

As shown in figure 7, similarly the encoder by this embodiment includes the intra skip discrimination section (730) and skip flag setting part (700) with the encoder by the second embodiment. But the intra skip discrimination section (730) by this embodiment determines whether the conversion factor (C) of the target macro block and conversion factor (C') of the peripheral macro block are identical or not. The conversion factor of the residual block of the amount block determines 0 recognition as to the this, the second embodiment is distinguished. The discrete cosine transformer (140) and quantizer (150) are used so that the intra skip discrimination section (730) save the conversion factor (C) of the target macro block and conversion factor (C') of the peripheral macro block.

The intra skip discrimination section (730) the conversion factor (C) of the target macro block and conversion factor (C') of the peripheral macro block are identical transmit the intra skip set signal (731) to the skip flag setting part (700) it generates. The skip flag setting part (700) receiving the intra skip set signal (731) sets up the intra skip flag (I skip flag) as 1 and this is inserted into data field of the macroblock of the corresponding sequence.

If the conversion factor (C) of the target macro block and conversion factor (C') of the peripheral macro block are not identical the intra skip discrimination section (730) does not create the intra skip set signal (731) and it confronts with the target macro block (B) and the final conversion factor

generated according to the conventional prediction operation is inserted into data field of the macroblock. At this time, the intra skip flag is set up as 0.

As shown in figure 8, the decoder according to this embodiment includes the skip flag discrimination section (710) in which the intra skip flag included in the target macro block data field determines 1 recognition and the macroblock decompression module (720) restoring the target macro block according to the intra skip flag. It is different with the decoder of the second embodiment. The target macro block is restored since the macroblock decompression module (720) of the decoder by this embodiment copies the conversion factor of the peripheral macro block. That is, the conversion factor of the peripheral macro block is transmitted from the prediction memory within the decoder and this is regarded as the conversion factor (721) of the target macro block. Therefore, it is restored to the macroblock having the sampled-data the reverse conversion and intra prediction operation are gone through the output of the macroblock decompression module (720) dequantizes.

Since the position of the peripheral macro block is the position agreed on the spec of the decoder and encoder the macroblock decompression module (720) has the position about this information like the second embodiment to the SPEC value.

Figure 9 is a time flow diagram indicative of the encoding and decoding method according to a preferred embodiment of the present invention.

Firstly, in the encoder, the residual block generating unit creates the residual block of the predicted block and target macro block (900 step). Here, the predicted block is generated between the encoder and decoder using the predetermined intra prediction mode. In the preferred embodiment, the intra prediction mode is the DC prediction mode. The discrete cosine transform and quantization unit quantize the residual block with the discrete cosine transform and the conversion factor (X) is created (905 step). The conversion factor discrimination section the conversion factor determines 0 recognition. 0 rear side intra skip flag (I skip flag) is set up as 1 and the conversion factor is inserted within data field of the target macro block (920 step). If it is not conversion factor with 0 this it is the encodes entropy and the conversion factor of the residual block generated about the target macro block is altogether inserted within data field of the target macro block (925 step). The sequence containing the target macro block is transmitted to the decoder (930 step). In the decoder, it is the decodes entropy and the entropy decoder extracts the intra skip flag the target macro block and it determines whether the intra skip flag is 1 (940 step). According to the prediction mode in which the intra skip flag 1 rear side , and the macroblock decompression module is predetermined between the encoder and decoder, the predicted block is created (950 step) and by storing the above-described predicted block in the position of the prediction memory corresponding to the target macro block the target macro block is restored (960 step). The existing decoding process is altogether performed the intra skip flag inverses transform all conversion factors within data field of 0 rear side target macro block and etc the macroblock is reproduced.

The present invention has been reviewed centering on preferred embodiments. In the technical field in which the invention belongs, it will be able to understand being implemented as the form in which a person skilled in the art is transfomed in the range that the invention does not deviate from the intrinsic property of the invention. It is not point of view that the disclosed embodiments are the restrictive but point of view has to be considered in the point of view of being descriptive. It should be interpreted that the scope of the present invention all differences within that and the equivalent range above-described description shows up in the patent claim it is not above-described description are included in the invention.

Advantageous Effects

As described above, provided are the encoding decoding apparatus and the method in which it uses the skip mode in not only the prediction mode but also the intraframe prediction mode between the frame and in that way the in-frame spatial correlation relation can increase the compression ratio in case of the large frame.

Moreover, the encoding decoding system of the existing H.264 standard can not be changed and the above-described encoding decoding apparatus and the method like that can apply and the implementation is facilitated.

Figures 1 and 2 are the block diagram of the decoding apparatus and encoding performing the operation of the skip mode between the frame being used in the existing video compression technique of the H.264 standard.

Figures 3 and 4 are drawing indicative of the video encoding and decoding apparatus according to the first embodiment of the invention.

Figures 5 and 6 are drawing indicative of the video encoding and decoding apparatus according to the second embodiment of the invention.

Figures 7 and 8 are drawing indicative of the video encoding and decoding apparatus according to the third embodiment of the invention.

Figure 9 is a time flow diagram indicative of the encoding and decoding method according to a preferred embodiment of the present invention.

(Translation from Clarivate Analytics)

Citations

Citation *Record level*

Expand Citing Patents (29)

Cited Patents (0)

Cited Non-patents (0)

DPCI Citation *Patent family level*

DPCI Citing Patents (0)

DPCI Cited Patents (0)

DPCI Cited Non-patents (0)

DPCI Citation Counts

DPCI Field of Search

-

DPCI Citing Patents Country

-

DPCI Cited Patents Country

-

Other

Attorney / Agent

Y.P.LEE MOCK&PARTNERS,

LEE HAE YOUNG

Custom Fields

PMS Active Portfolio

-

Type

-

PMS Family Number

-

PMS Case Reference

-

PMS Case Key

-

Cluster RPPR

-

Copyright 2007-2017 CLARIVATE ANALYTICS

KR20090065401 A - METHOD OF INTER-FRAME/INTRA BLOCK PREDICTIVE CODING IN VIDEO CODING

Bibliographic data

Publication
KR20090065401 A [2009-06-22]
Application
KR20080012337 [2008-02-11]

Classifications
IPC
H04N19/102; H04N19/51
CPC
H04N19/119; H04N19/124; H04N19/13; H04N19/176; H04N19/51

Priorities
KR20070132435 [2007-12-17]

Inventors
김문철; 박창섭; 함상진; 박근수; KIM MUN CHURL; PARK CHANG SEOB; HAHM SANG JIN; PARK KEUN SOO

Assignees
한국정보통신대학교 산학협력단; 한국방송공사; ICU RES & IND COOP GROUP; KOREAN BROADCASTING SYSTEM; Korean Broadcasting System Corp

Titles
[EN]
METHOD OF INTER-FRAME/INTRA BLOCK PREDICTIVE CODING IN VIDEO CODING
[KO]
비디오 부호화에서의 화면간/화면내 예측 부호화 방법

Abstract

KO	EN
본 발명은 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 관한 것으로, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면간 예측을 수행한 후, 그 중 어느 하나를 화면간 예측 부호화 결과로 선택하여 부호화함으로써, 부호화 효율을 향상 향상시킬 수 있으며, 기존의 블록 매칭을 이용한 부호화 방식뿐만 아니라 움직임 모델 기반의 픽셀 와핑(warping)을 이용하여 예측 정확도를 크게 향상시킴으로써 발생 잔차 신호의 양을 줄여 부호화 효율을 더욱 개선할 수 있는 효과가 있다. H.264, AVC, 수퍼 매크로 블록, 블록 분할, 움직임 모델, 화면간 예측, 율-왜곡 비용, 부호화기	An inter/intra-frame predictive encoding method for video coding is provided to efficiency perform predictive encoding by decreasing the number of motion vectors to be encoded and load for predicting motion. An encoded block unit is expanded to more than one super macroblock comprising a bigger size than a 16x16 macroblock(S101). Each super macroblock is divided into sub brocks having various sizes(S102). Internet-frame prediction for the divided sub blocks is performed(S106). One of the predictions is selected and encoded as a predictive encoding(S107).

Claims

[KO]	[MT]
1. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면간 예측을 수행한 후, 그 중 어느 하나를 화면간 예측 부호화 결과로 선택하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	1. In the inter prediction encoding method in video encoding, Extends a coding block unit to at least one super macro block having a size larger than 16 × 16 macro blocks, divides each super macro block into sub blocks of various sizes, and divides sub blocks including each super macro block. The inter prediction prediction method in video encoding according to claim 1, wherein after performing the inter prediction with respect to the encoding, one of them is selected and encoded as the inter prediction prediction result.
2. 제1 항에 있어서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들을 통한 잔차 신호에 대하여 변환 및 양자화를 수행한 후, 그 결과를 엔트로피 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	2. According to claim 1, And transforming and quantizing the residual signal through the divided lower blocks including each super macro block, and then entropy-encoding the result.
3. 제1 항에 있어서, 상기 수퍼 매크로 블록의 크기는 16×16 매크로 블록의 크기보다 큰 가로 및 세로 길이가 2의 거듭제곱으로 정의되는 정사각형이며, 이를 분할한 하위 블록은 정사각형 및 직사각형을 포함하는 2의 거듭제곱으로 정의되는 블록으로 이루어진 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	3. According to claim 1, The size of the super macro block is a square in which the horizontal and vertical lengths larger than the size of the 16 × 16 macro block are defined as powers of two, and the sub-blocks divided therefrom are defined as powers of two including squares and rectangles. An inter prediction encoding method in video encoding, characterized by consisting of blocks.
4. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 입력 화면을 부호화 단위 블록으로 분할하고, 상기 분할된 각 부호화 단위 블록에 대해 움직임 예측을 위한 적어도 하나의 움직임 모델을 선택적으로 적용하여 예측 부호화를 수행한 후, 그 중 어느 하나를 화면간 예측 결과로 선택하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	4. In the inter prediction encoding method in video encoding, After splitting the input screen into coding unit blocks, performing prediction encoding by selectively applying at least one motion model for motion prediction to each of the divided coding unit blocks, and converting any one of them into an inter prediction result. An inter-prediction encoding method in video encoding, which is selected and encoded.
5. 제4 항에 있어서, 상기 부호화 단위 블록은 매크로 블록 또는 16×16 매크로 블록보다 큰 크기의 수퍼 매크로 블록인 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	5. The method of claim 4, wherein The coding unit block is a macroblock or a super macroblock having a size larger than that of a 16 × 16 macroblock.

6. 제4 항에 있어서, 상기 움직임 모델은 파라미터로 모형화 된 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	6. The method of claim 4, wherein And the motion model is modeled as a parameter.
7. 제4 항에 있어서, 상기 움직임 모델은 병진 움직임 모델, 의사(擬似)(Affine) 움직임 모델, 이중선형(Bilinear) 움직임 모델 또는 투시(Perspective) 움직임 모델 중 적어도 하나를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	7. The method of claim 4, wherein The motion model includes at least one of a translational motion model, an affine motion model, a bilinear motion model, or a perspective motion model. Way.
8. 제4 항에 있어서, 상기 움직임 모델을 적용한 예측 부호화 과정은, 움직임 모델 파라미터를 추정하고, 상기 추정된 움직임 모델 파라미터를 이용하여 참조 화면으로부터 움직임 예측을 수행하고, 그 잔차 신호를 변환하여 변환 계수를 양자화한 후, 상기 움직임 모델 파라미터와 양자화된 변환 계수에 대해 엔트로피 부호화를 수행하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	8. The method of claim 4, wherein The prediction encoding process to which the motion model is applied, After estimating a motion model parameter, performing motion prediction from a reference picture using the estimated motion model parameter, transforming the residual signal, quantizing transform coefficients, and entropy for the motion model parameter and quantized transform coefficient. An inter prediction encoding method in video encoding, characterized in that encoding is performed.
9. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a) 입력 화면을 미리 정의된 수퍼 매크로 블록 단위로 확장하는 단계; (b) 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하는 단계; 및 (c) 각 수퍼 매크로 블록 단위로 상기 분할된 하위 블록들에 대해 참조 영상 으로부터 움직임 예측을 수행한 후, 최적의 블록 모드를 결정하고 이를 통해 구해진 잔차 신호와 움직임 벡터를 부호화하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	9. In the inter prediction encoding method in video encoding, (a) expanding the input screen by a predefined super macro block unit; (b) dividing each super macro block into sub-blocks of various sizes; And (c) performing motion prediction from the reference image on the divided lower blocks in units of each super macro block, and then determining an optimal block mode and encoding the residual signal and the motion vector obtained through the prediction. An inter prediction encoding method in video encoding.

10. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a') 부호화 대상 블록에 대해 움직임 모델을 선택하여 움직임 모델 파라미터를 획득하는 단계; (b') 상기 획득된 움직임 모델 파라미터를 이용하여 참조 화면으로부터 움직임 보상 블록 픽셀을 생성하는 단계; (c') 상기 생성된 움직임 보상 블록 픽셀값과 현재 부호화하고자 하는 블록의 픽셀값의 잔차 신호를 구하는 단계; (d') 상기 단계(b') 및 단계(c')를 선택 가능한 움직임 모델에 대해 수행하고, 현재 부호화하고자 하는 블록에 대한 움직임 모델을 최종 결정하는 단계; 및 (e') 상기 최종 선택된 움직임 모델을 이용하여 화면간 움직임 예측 부호화를 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	10. In the inter prediction encoding method in video encoding, (a ') obtaining a motion model parameter by selecting a motion model with respect to an encoding target block; (b ') generating a motion compensation block pixel from a reference picture using the obtained motion model parameter; obtaining a residual signal of the generated motion compensation block pixel value and the pixel value of the block to be currently encoded; (d ') performing steps (b') and (c ') on a selectable motion model, and finally determining a motion model for a block to be currently encoded; And (e ') performing inter-screen motion prediction encoding using the finally selected motion model.
11. 제10 항에 있어서, 상기 단계(d')에서, 상기 움직임 모델 최종 결정은 율-왜곡 비용을 고려하여 결정하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	11. The method of claim 10, In the step (d '), the final determination of the motion model is determined in consideration of the rate-distortion cost.
12. 제9 항 또는 제10 항에 있어서, 상기 단계(a) 또는 단계(a') 이전에 부호화를 위한 화면 순서 및 블록 순서를 초기화하는 단계를 더 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	12. The method of claim 9 or 10, And initiating a picture order and a block order for encoding before step (a) or step (a ').
13. 비디오 부호화에서의 화면내 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 비교하여 더 유리한 부호화 결과를 선택하여 최종 부호화하는 것을 특징으로 하는 비디오 부호화에 있어서 화면내 예측 부호화 방법.	13. In the intra prediction encoding method in video encoding, Extends a coding block unit to at least one super macro block having a size larger than 16 × 16 macro blocks, divides each super macro block into sub blocks of various sizes, and divides sub blocks including each super macro block. The intra prediction prediction method according to the video encoding, characterized in that after performing the intra prediction for the video encoding, the results are compared and the more favorable encoding result is selected and finally encoded.

14. 제13 항에 있어서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 율-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면내 예측 부호화 방법.	14. The method of claim 13, Intra prediction encoding is performed on the divided subblocks including each super macro block by performing directional prediction and encoding the directional prediction mode that generates the least rate-distortion cost as the optimal final intra prediction. An intra prediction encoding method in video encoding.
15. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 부호화 할 화면내 예측 결과와 부호화 할 화면간 예측 결과를 비교하여 더 유리한 예측을 선택하여 최종 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	15. In the inter prediction encoding method in video encoding, Extends a coding block unit to at least one super macro block having a size larger than 16 × 16 macro blocks, divides each super macro block into sub blocks of various sizes, and divides sub blocks including each super macro block. After the intra prediction is performed on the inter prediction, the inter prediction prediction encoding in the video encoding is performed by comparing the intra prediction result to be encoded with the inter prediction result to be encoded, and finally selecting a more advantageous prediction. Way.

16. 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a") 부호화하고자 하는 화면을 입력하는 단계; (b") 입력된 화면을 부호화 단위 블록으로 분할하는 단계; (c") 현재의 입력 화면이 화면간 예측 부호화를 수행할 것인지를 판별하는 단계; (d") 현재의 입력 화면이 화면간 예측이 아닌 경우 화면내 예측 부호화를 수행하는 단계; (e") 현재의 입력 화면이 화면간 예측인 경우에는 입력 화면에서 부호화되는 단위 블록의 순서를 초기화하는 단계; (f") 부호화하고자 하는 단위 블록에 움직임 예측을 수행하기 위해 적용할 움직임 모델을 하나 선택한 후 움직임 예측을 수행하는 단계; (g") 상기 선택된 움직임 예측 결과를 이용하여 예측 부호화를 수행하는 단계; (h") 상기 단계(g")에서의 예측 부호화 결과 및 율-왜곡 비용값을 저장하는 단계; (i") 상기 적용 움직임 모델이 최종 모델인지 판단하여, 최종 움직임 모델이면 최적의 움직임 모델을 결정하고, 해당 부호화 결과를 최종 선택하는 단계; (j") 상기 적용 움직임 모델이 최종 모델이 아니면, 상기 단계(f")로 리턴하는 단계; (k") 현재 단위 블록이 현재의 입력 화면에서 최종 블록인지를 판별하는 단계; 및 (l") 현재 단위 블록이 현재의 입력 화면에서 최종 블록이면, 현재의 입력 화면이 최종 화면인지 판별하여, 최종 화면이 될 때까지 상기 단계(a") 내지 단계(k")를 반복 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	16. In the inter prediction encoding method in video encoding, (a ") inputting a screen to be encoded; (b ") dividing the input screen into coding unit blocks; (c ") determining whether the current input screen is to perform inter prediction encoding; (d ") performing intra prediction encoding if the current input screen is not inter prediction; (e ") initializing the order of the unit blocks encoded in the input screen when the current input screen is inter prediction; (f ") selecting one motion model to be applied to perform motion prediction on the unit block to be encoded and performing motion prediction; (g ") performing predictive encoding using the selected motion prediction result; (h ") storing the predictive coding result and the rate-distortion cost value in step g"; (i ") determining whether the applied motion model is a final model, determining an optimal motion model if the final motion model, and finally selecting a corresponding encoding result; (j ") if the applied motion model is not a final model, returning to step f "; (k ") determining whether the current unit block is the last block in the current input screen; and (l ") If the current unit block is the last block in the current input screen, it is determined whether the current input screen is the last screen, and the above steps (a") to (k ") are repeatedly performed until the final screen. An inter prediction prediction method in video encoding, comprising the steps of:
17. 제16 항에 있어서, 상기 단계(f")에서, 상기 움직임 모델을 적용하여 움직임 예측 수행 시 움직임 모델 파라미터를 추정하는 단계를 더 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.	17. The method of claim 16, And estimating a motion model parameter when performing motion prediction by applying the motion model in step (f ").
18. 제1 항 내지 제17 항 중 어느 한 항의 방법을 컴퓨터로 실행시킬 수 있는 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.	18. A computer-readable recording medium having recorded thereon a program capable of executing the method of any one of claims 1 to 17.

Description

[KO]	[MT]
본 발명은 비디오 부호화에서 수퍼 매크로 블록 또는 움직임 모델을 이용한 예측 부호화 방법에 관한 것으로, 기존의 16×16 픽셀 단위의 매크로 블록 이상으로 확장하고 하위 블록의 크기를 다양한 크기로 분할한 후, 예측 부호화함으로써 부호화 효율을 더욱 향상시킬 수 있으며, 이를 기존의 블록 매칭을 이용한 부호화 방식뿐만 아니라 움직임 모델 기반의 픽셀 와핑을 이용하여 예측 정확도를 크게 향상시킴으로써 발생 잔차 신호의 양을 줄여 부호화 효율을 더욱 개선할 수 있도록 한 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 관한 것이다.	The present invention relates to a predictive encoding method using a super macro block or a motion model in video encoding. The present invention relates to a method of expanding a macroblock of 16 × 16 pixel units or more by dividing a size of a lower block into various sizes and then predicting encoding. Encoding efficiency can be further improved, and this can be further improved by reducing the amount of residual signal generated by greatly improving the prediction accuracy by using pixel warping based on motion model as well as the conventional coding method using block matching. The present invention relates to an inter-screen / intra-prediction coding method in a video encoding.
일반적으로, ISO/IEC와 ITU-T에서 공동으로 표준화한 비디오 압축 부호화에 기술인 ISO/IEC 14496-10(MPEG-4 Advanced Video Coding) 또는 H.264 및 ISO/IEC 14496-10 Amendment 3(MPEG-4 Scalable Video Coding), SMPTE 표준인 VC-1, 이밖에 Audio Video coding Standard(AVS) 규격 등은 비디오 데이터 압축 효율에 있어 많 은 진전을 이루었다.	Generally, ISO / IEC 14496-10 (MPEG-4 Advanced Video Coding) or H.264 and ISO / IEC 14496-10 Amendment 3 (MPEG-), a technique for video compression encoding jointly standardized by ISO / IEC and ITU-T. 4 Scalable Video Coding), the SMPTE standard VC-1, and the Audio Video coding Standard (AVS) standard have made significant progress in video data compression efficiency.
이러한 비디오 압축 효율의 향상 요인은 여러 가지가 있을 수 있으나 특히, 기존의 비디오 부호화 표준들(MPEG-1 Video, MPEG-2 Video, MPEG-4 Part 2 Visual, H.261, H.263 등)이 부호화하고자 하는 화면의 크기를 매크로 블록(16×16) 단위로 분할한 후 이를 예측 부호화하는 것과는 달리, 매크로 블록을 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 및 4×4 단위로 더욱 세분화하여 분할하고 이들 하위 블록에 대해 예측 부호화를 수행하고 율-왜곡 비용의 관점에서 비용 최소화를 발생시키는 블록을 최적의 블록모드로 하여 부호화한다.	There are many factors that can improve the video compression efficiency, but in particular, existing video coding standards (MPEG-1 Video, MPEG-2 Video, MPEG-4 Part 2 Visual, H.261, H.263, etc.) Unlike dividing the size of the screen to be encoded into units of macroblocks (16 × 16) and then predictively encoding the macroblocks, the macroblocks are 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 Further subdividing into x8 and 4x4 units, performing prediction coding on these sub-blocks, and encoding a block that generates cost minimization in terms of rate-distortion cost is an optimal block mode.
이렇게 함으로써 미세한 움직임이나 복잡한 영상의 움직임 예측을 보다 효과적으로 수행하여 발생되는 잔차 신호를 크게 줄임으로써 압축 효율을 크게 향상할 수 있었다.	In this way, the compression efficiency can be greatly improved by greatly reducing the residual signal generated by performing the motion prediction of the fine motion or the complex image more effectively.
도 1은 종래 기술의 H.264|AVC 부호화기에서 화면간 예측 부호화를 위한 16×16 매크로 블록 단위 블록의 분할 블록 타입을 나타낸 도면으로서, H.264에서 사용되는 7가지의 움직임 예측 블록 분할 타입을 나타낸다.	FIG. 1 is a diagram illustrating a partition block type of a 16 × 16 macroblock unit block for inter prediction encoding in an H.264	AVC encoder according to the prior art. FIG. 1 illustrates seven motion prediction block partition types used in H.264. Indicates.

도 1을 참조하면, 주로 작은 크기의 블록들(8×8, 8×4, 4×8 및 4×4)은 영상 내의 움직임 영역이 작거나 미세한 움직임의 경우 예측 성능의 큰 향상을 제공한다.	Referring to FIG. 1, mainly small blocks 8x8, 8x4, 4x8 and 4x4 provide a large improvement in prediction performance in the case of small or fine motion regions of an image.
반면에 크기가 큰 블록들(16×16, 16×8 및 8×16)은 영상 픽셀 값이 균질한 영역이나 움직임이 적은 영역 또는 병진 운동이 존재하는 움직임 영역 등에서 움직임 예측 부호화 시에 주로 선택되는 경향을 띤다. 이와 같이 움직임 예측에 큰 블록을 사용하는 것은 그만큼 부호화 시에 움직임 벡터 발생량이 줄어 부호화 효율을 증가시키는 요인이 된다.	On the other hand, large blocks 16 × 16, 16 × 8, and 8 × 16 are mainly selected for motion prediction coding in a homogeneous region, a region with little movement, or a motion region in which translational motion exists. Tends to As such, the use of a large block for motion prediction reduces the amount of motion vector generated at the time of encoding, thereby increasing the coding efficiency.
결론적으로, 움직임 예측 부호화 시에 다양한 블록의 크기를 사용하여 움직임 예측 부호화를 수행하고, 이 중 하기의 수학식 1과 같이 율-왜곡 비용을 가장 적게 발생시키는 블록이 최적의 부호화 최종 블록모드로 결정된다.	In conclusion, motion prediction coding is performed by using various block sizes in motion prediction coding, and among the blocks that generate the least rate-distortion cost as the following Equation 1, the optimal coding final block mode is determined. do.
$MODE = \arg\min_{Mode}\{D(c,r,Mode \mid QP) + \lambda_{MODE} \cdot R(c,r,Mode \mid QP)\}$	$MODE = \arg\min_{Mode}\{D(c,r,Mode \mid QP) + \lambda_{MODE} \cdot R(c,r,Mode \mid QP)\}$
여기서, c와 r은 현재 블록과 참조 블록을 의미하고, D는 두 블록간의 왜곡도를 나타낸다. λ_{MODE}는 라그랑지안 승수(Lagrangian Multiplier)이다. 또한, MODE는 예측 부호화에 사용되는 율-왜곡 최적화를 수행하는 모든 블록모드(또는 크기)를 의미한다.	Here, c and r represent the current block and the reference block, and D represents the distortion degree between the two blocks. λ_{MODE} is the Lagrangian Multiplier. In addition, MODE means all block modes (or sizes) for performing rate-distortion optimization used for predictive encoding.
예컨대, 영상의 균질 영역이 많은 부분을 차지할 경우나 움직임이 일어나지 않는 영역의 부분이 많은 경우 매크로 블록(16×16)이 최종 움직임 예측 블록으로 결정되는 빈도가 크게 증가한다.	For example, when the homogeneous region of the image occupies a large portion or when the portion of the region where no motion occurs is large, the frequency at which the macro block 16x16 is determined as the final motion prediction block increases greatly.
특히, 부호화하고자 하는 입력 영상의 해상도가 매우 큰 경우 이러한 영상의 균질한 영역이 많은 부분을 차지하나 16×16 크기의 매크로 블록이 최대 블록이므로, 즉 최대 처리블록 단위의 크기가 16×16으로 제한되어, 움직임 예측 부호화 효율을 증대시키는데 한계가 있다. 이로 인해 초고해상도 영상을 부호화하고자 하는 경우 기존의 최대 크기인 16×16 매크로 블록을 사용하면 부호화 효율 증대에 한계 를 가지는 문제점이 있다.	In particular, when the resolution of the input image to be encoded is very large, the homogeneous area of the image occupies a large portion, but since the macro block of 16 × 16 size is the largest block, that is, the size of the maximum processing block unit is limited to 16 × 16. There is a limit to increasing the motion prediction coding efficiency. For this reason, when a super high resolution image is to be encoded, using a 16 × 16 macroblock, which is the existing maximum size, has a problem in that the encoding efficiency is limited.

한편, 움직임 예측 부호화 시에 기존의 비디오 부호화 방법들은 부호화하고자 하는 블록을 참조 화면 내의 탐색 영역에서 블록 매칭을 통하여 가장 유사한 픽셀 블록의 위치를 찾고 이점을 움직임 벡터로 전송한다.	Meanwhile, in motion prediction encoding, existing video encoding methods locate the most similar pixel block through block matching in a search region in a reference picture and transmit the benefit as a motion vector.
이때, 사용되는 블록 매칭 기법은 기본적으로 병진 운동을 가정하고 움직임 예측을 수행하는 경우이다. 그러나, 카메라의 움직임이나 영상 내의 객체의 움직임은 병진(Tracking) 운동뿐만 아니라, 줌(Zoom-in: 카메라의 위치를 고정시킨 채 줌 렌즈의 초점거리를 변화시켜 피사체에 접근하여 가는 것처럼 촬영, Zoom-out: 카메라의 위치를 고정시킨 채 줌 렌즈의 초점거리를 변화 시켜 피사체로부터 멀어져 가는 것처럼 촬영), 회전(Roll: 카메라를 회전시켜 피사체를 촬영), 패닝(Panning: 카메라의 수직축이 고정되어 있는 상태에서의 카메라의 수평적 움직임), 붐(Boom-up: 카메라의 수직적 상승 이동, Boom-down: 카메라의 수직적 하강 이동), 틸팅(Tilting: 카메라의 수평축이 고정되어 있는 상태에서의 카메라의 수직적 움직임), 달리(Dolly-in: 카메라가 피사체에 접근, Dolly-out: 카메라가 피사체에서 물러남)등이 있다. 도 2는 비디오 영상에서의 카메라의 움직임에 의한 움직임 생성 타입을 설명하기 위한 도면으로서, 카메라의 움직임 형태에 대한 도식을 나타낸다.	In this case, the block matching technique used is basically a case of performing a motion prediction assuming a translational motion. However, the movement of the camera or the movement of an object in the image is not only a tracking movement, but also a zoom-in. The zoom lens is fixed by changing the focal length of the zoom lens as if approaching a subject. -out: Changes the focal length of the zoom lens while fixing the position of the camera and shoots it away from the subject), Roll (Roll: Shoots the subject by rotating the camera), and Panning: The vertical axis of the camera is fixed. Horizontal movement of the camera in the state), Boom-up: vertical upward movement of the camera, Boom-down: vertical downward movement of the camera, and tilting (vertical movement of the camera while the horizontal axis of the camera is fixed). Movement), otherwise (Dolly-in: the camera approaches the subject, Dolly-out: the camera withdraws from the subject). FIG. 2 is a diagram for describing a motion generation type by camera movement in a video image, and illustrates a diagram of a camera's movement pattern.
도 3은 2차원 영상 공간에 투영된 움직임 모델을 설명하기 위한 개념도로서, 실제 3차원 공간에서의 피사체의 움직임이 2차원 공간에 투영되었을 때 투영된 2차원 공간에서의 움직임을 나타내며, 하기의 수학식 2와 수학식 3은 투영된 공간에서의 x축과 y축으로의 움직임을 수식으로 표현한다.	3 is a conceptual diagram illustrating a motion model projected in a two-dimensional image space, and illustrates a motion in a two-dimensional space projected when a motion of a subject in a real three-dimensional space is projected in the two-dimensional space. Equation 2 and Equation 3 express the movement in the projected space in the x-axis and y-axis by the equation.
$u=-\frac{f}{Z}(T_x-xT_z)+\frac{xy}{f}\cdot R_x-f\left(1+\frac{x^2}{f^2}\right)\cdot R_y+y\cdot R_z+f.\tan^{-1}\left(\frac{x}{f}\right)\left(1+\frac{x^2}{f^2}\right)\cdot R_{zoom}$	$u=-\frac{f}{Z}(T_x-xT_z)+\frac{xy}{f}\cdot R_x-f\left(1+\frac{x^2}{f^2}\right)\cdot R_y+y\cdot R_z+f.\tan^{-1}\left(\frac{x}{f}\right)\left(1+\frac{x^2}{f^2}\right)\cdot R_{zoom}$
$v=-\frac{f}{Z}(T_y-yT_z)-\frac{xy}{f}\cdot R_y+f\left(1+\frac{y^2}{f^2}\right)\cdot R_x-x\cdot R_z+f.\tan^{-1}\left(\frac{y}{f}\right)\left(1+\frac{y^2}{f^2}\right)\cdot R_{zoom}$	$v=-\frac{f}{Z}(T_y-yT_z)-\frac{xy}{f}\cdot R_y+f\left(1+\frac{y^2}{f^2}\right)\cdot R_x-x\cdot R_z+f.\tan^{-1}\left(\frac{y}{f}\right)\left(1+\frac{y^2}{f^2}\right)\cdot R_{zoom}$
여기서, u와 v는 움직임 벡터를 의미하고, f는 카메라의 초점 거리, T_x, T_y, T_z는 OXYZ 공간에서 각각 X축, Y축, Z축에서의 병진 속도(Translational Velocity)를 나타낸다. 한편, R_x, R_y, R_z는 각각 각속도(Angular Velocity)를 나타낸다. x, y는 투영 이미지 공간에서의 좌표 값을 의미한다.	Here, u and v denote motion vectors, and f denotes a focal length of the camera, and T_x, T_y, and T_z denote translational velocities in the X, Y, and Z axes in OXYZ space, respectively. . On the other hand, R_x, R_y, and R_z represent Angular Velocity, respectively. x and y mean coordinate values in the projection image space.

이와 같이 기존의 움직임 보상을 통한 예측 부호화는 병진 운동만을 고려한 움직임 예측이므로 실제 2차원 공간에서의 복잡한 움직임을 제대로 예측하지 못하며 이로 인한 예측 부호화 성능 향상의 한계점을 가지고 있다.	As the conventional predictive coding using motion compensation is a motion prediction considering only the translational motion, it does not properly predict the complex motion in the actual two-dimensional space, and thus has a limitation of improving the predictive coding performance.
본 발명은 전술한 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 목적은 비디오 부호화 시에 최대 16×16 매크로 블록을 통한 예측 부호화 방식을 확장하여 16×16 보다 큰 크기의 블록(이하, "수퍼 매크로 블록"이라 칭함)을 사용하여 움직임 예측 및 보상을 수행하여 보다 효과적인 예측 부호화 방법을 제공하는데 있다.	SUMMARY OF THE INVENTION The present invention has been made to solve the above-described problem, and an object of the present invention is to extend a prediction coding scheme through a maximum of 16 × 16 macroblocks in video encoding, thereby to block blocks having a size larger than 16 × 16 (hereinafter, referred to as "super Macroblock ") to perform motion prediction and compensation to provide a more efficient predictive encoding method.
본 발명의 다른 목적은 블록 기반의 예측 부호화에 있어 기존의 병진 운동에 의한 참조 화면에서의 탐색영역에서 최대 유사 블록을 찾아 예측 부호화함으로써 카메라 움직임 및 객체 움직임 등의 보다 복잡한 복합적인 움직임을 예측하는데 한계가 있는 것을 극복하기 위해 움직임 모델을 이용하여 움직임을 보다 정확히 예측하고 이를 기반으로 예측 부호화를 수행함으로서 기대되는 잔차 신호(Residual Signals)의 발생량을 줄여 부호화 효율을 향상시키는 방법을 제공하는데 있다.	Another object of the present invention is to predict a more complex complex motion such as a camera motion and an object motion by finding and predicting a maximum likelihood block in a search region in a reference screen by a conventional translational motion in block-based predictive coding. The present invention provides a method of improving encoding efficiency by reducing the amount of residual signals expected to be generated by more accurately predicting a motion using a motion model and performing predictive coding based on the motion model.
전술한 목적을 달성하기 위하여 본 발명의 제1 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면간 예측을 수행한 후, 그 중 어느 하나를 화면간 예측 부호화 결과로 선택하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	In order to achieve the above object, a first aspect of the present invention provides a method for inter-prediction encoding in video encoding, wherein a coding block unit is extended to at least one super macro block having a size larger than 16 × 16 macroblocks. After splitting each super macro block into sub-blocks of various sizes, performing inter-prediction on the divided sub-blocks including each super macro block, selecting one of them as an inter-prediction encoding result. The present invention provides an inter prediction encoding method in video encoding.
여기서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들을 통한 잔차 신호에 대하여 변환 및 양자화를 수행한 후, 그 결과를 엔트로피 부호화하는 것이 바람직하다.	Here, it is preferable to perform transform and quantization on the residual signal through the divided lower blocks including each super macro block, and then entropy-encode the result.

바람직하게, 상기 수퍼 매크로 블록의 크기는 16×16 매크로 블록의 크기보다 큰 가로 및 세로 길이가 2의 거듭제곱으로 정의되는 정사각형이며, 이를 분할한 하위 블록은 정사각형 및 직사각형을 포함하는 2의 거듭제곱으로 정의되는 블록으로 이루어질 수 있다.	Preferably, the size of the super macro block is a square whose horizontal and vertical lengths greater than the size of the 16 × 16 macro block are defined as powers of two, and the sub-blocks that divide the power are two powers including squares and rectangles. It may be made of a block defined as.
본 발명의 제2 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 입력 화면을 부호화 단위 블록으로 분할하고, 상기 분할된 각 부호화 단위 블록에 대해 움직임 예측을 위한 적어도 하나의 움직임 모델을 선택적으로 적용하여 예측 부호화를 수행한 후, 그 중 어느 하나를 화면간 예측 결과로 선택하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	According to a second aspect of the present invention, in an inter prediction encoding method in video encoding, the input screen is divided into coding unit blocks, and at least one motion model for motion prediction is selectively selected for each of the divided coding unit blocks. The present invention provides an inter-prediction encoding method in video encoding, wherein the prediction encoding is performed by applying the L-P encoding, and then any one of the encoding is selected as an inter-screen prediction result.
여기서, 상기 부호화 단위 블록은 매크로 블록 또는 16×16 매크로 블록보다 큰 크기의 수퍼 매크로 블록인 것이 바람직하다.	Herein, the coding unit block is preferably a macroblock or a super macroblock having a size larger than that of a 16 × 16 macroblock.
바람직하게, 상기 움직임 모델은 파라미터로 모형화 된 것이다.	Preferably, the motion model is modeled with parameters.
바람직하게, 상기 움직임 모델은 병진 움직임 모델, 의사(擬似)(Affine) 움 직임 모델, 이중선형(Bilinear) 움직임 모델 또는 투시(Perspective) 움직임 모델 중 적어도 하나를 포함할 수 있다.	Preferably, the motion model may include at least one of a translational motion model, an affine motion model, a bilinear motion model, or a perspective motion model.
바람직하게, 상기 움직임 모델을 적용한 예측 부호화 과정은, 움직임 모델 파라미터를 추정하고, 상기 추정된 움직임 모델 파라미터를 이용하여 참조 화면으로부터 움직임 예측을 수행하고, 그 잔차 신호를 변환하여 변환 계수를 양자화한 후, 상기 움직임 모델 파라미터와 양자화된 변환계수에 대해 엔트로피 부호화를 수행할 수 있다.	Preferably, the prediction encoding process applying the motion model estimates a motion model parameter, performs motion prediction from a reference picture using the estimated motion model parameter, transforms the residual signal, and quantizes the transform coefficients. In addition, entropy encoding may be performed on the motion model parameter and the quantized transform coefficient.

본 발명의 제3 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a) 입력 화면을 미리 정의된 수퍼 매크로 블록 단위로 확장하는 단계; (b) 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하는 단계; 및 (c) 각 수퍼 매크로 블록 단위로 상기 분할된 하위 블록들에 대해 참조 영상으로부터 움직임 예측을 수행한 후, 최적의 블록 모드를 결정하고 이를 통해 구해진 잔차 신호와 움직임 벡터를 부호화하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	According to a third aspect of the present invention, there is provided a method of inter-prediction encoding in video encoding, comprising: (a) extending an input screen in a unit of a predefined super macro block; (b) dividing each super macro block into sub-blocks of various sizes; And (c) after performing motion prediction on the divided lower blocks in units of each super macro block, determining an optimal block mode and encoding a residual signal and a motion vector obtained through the prediction. An inter prediction encoding method in video encoding is provided.
본 발명의 제4 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a') 부호화 대상 블록에 대해 움직임 모델을 선택하여 움직임 모델 파라미터를 획득하는 단계; (b') 상기 획득된 움직임 모델 파라미터를 이용하여 참조 화면으로부터 움직임 보상 블록 픽셀을 생성하는 단계; (c') 상기 생성된 움직임 보 상 블록 픽셀값과 현재 부호화하고자 하는 블록의 픽셀값의 잔차 신호를 구하는 단계; (d') 상기 단계(b') 및 단계(c')를 선택 가능한 움직임 모델에 대해 수행하고, 현재 부호화하고자 하는 블록에 대한 움직임 모델을 최종 결정하는 단계; 및 (e') 상기 최종 선택된 움직임 모델을 이용하여 화면간 움직임 예측 부호화를 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	A fourth aspect of the present invention provides a method for inter prediction encoding in video encoding, comprising: (a ') obtaining a motion model parameter by selecting a motion model for a block to be encoded; (b ') generating a motion compensation block pixel from a reference picture using the obtained motion model parameter; obtaining a residual signal of the generated motion compensation block pixel value and the pixel value of the block to be currently encoded; (d ') performing steps (b') and (c ') on a selectable motion model, and finally determining a motion model for a block to be currently encoded; And (e ') performing inter-screen motion prediction encoding by using the finally selected motion model.
상기 단계(d')에서, 상기 움직임 모델 최종 결정은 율-왜곡 비용을 고려하여 결정함이 바람직하다.	In the step (d '), the final determination of the motion model is preferably made in consideration of the rate-distortion cost.
바람직하게, 상기 단계(a) 또는 단계(a') 이전에 부호화를 위한 화면 순서 및 블록 순서를 초기화하는 단계를 더 포함할 수 있다.	Preferably, the method may further include initializing a screen order and a block order for encoding before step (a) or step (a ').
바람직하게, 상기 단계(c)에서 움직임 예측을 수행한 후, 또는 상기 단계(a') 이후에 부호화를 위해 순서에 맞게 화면을 입력하고, 상기 입력된 화면을 블록 단위로 분할한 후, 한 화면간 예측 부호화에서 부호화되는 블록의 순서를 초기화하는 단계를 더 포함할 수 있다.	Preferably, after performing the motion prediction in step (c) or after the step (a '), the screen is input in order for encoding, the input screen is divided in block units, and then one screen The method may further include initializing an order of blocks to be encoded in the inter prediction encoding.
바람직하게, 상기 단계(c) 또는 단계(d')에서, 상기 화면간 예측을 통한 율-왜곡 비용을 계산하고 부호화하고자 하는 매크로 블록 또는 수퍼 매크로 블록으로 합할 수 있다.	Preferably, in step (c) or step (d '), the rate-distortion cost through the inter prediction may be calculated and summed into a macro block or a super macro block to be encoded.

본 발명의 제5 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화하고자 하는 화면을 입력하는 단계; 상기 입력된 화면을 미리 정의된 수퍼 매크로 블록 단위로 분할하는 단계; 각 수퍼 매크로 블록에 대해 하위 크기의 블록으로 분할하고, 각 수퍼 매크로 블록 및 그 하위 블록에 대해 화면간 예측을 수행한 후, 그 중 어느 하나의 블록 타입을 선정하는 단계; 및 상기 선정된 블록 타입의 예측 결과를 이용하여 화면간 예측 부호화를 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	According to a fifth aspect of the present invention, there is provided an inter prediction encoding method in video encoding, comprising: inputting a screen to be encoded; Dividing the input screen into units of a predefined super macro block; Dividing each super macro block into blocks having a lower size, performing inter-screen prediction on each super macro block and the lower block, and selecting any one of the block types; And performing inter prediction encoding by using the selected prediction result of the block type.
본 발명의 제6 측면은 비디오 부호화에서의 화면내 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 비교하여 더 유리한 부호화 결과를 선택하여 최종 부호화하는 것을 특징으로 하는 비디오 부호화에 있어서 화면내 예측 부호화 방법을 제공하는 것이다.	In a sixth aspect of the present invention, in an intra prediction encoding method in video encoding, a coding block unit is extended to at least one super macro block having a size larger than 16 × 16 macro blocks, and each super macro block is variously sized. The video encoding, characterized by dividing into sub-blocks of, performing intra-prediction on the divided sub-blocks including each super macro block, comparing the results, selecting a more advantageous encoding result, and performing final encoding. In the present invention, an intra prediction encoding method is provided.
여기서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 율-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화함이 바람직하다.	In this case, the intra prediction encoding is performed on the sub-blocks including each super macro block by performing directional prediction, and encoding the directional prediction mode that generates the least rate-distortion cost as the optimal final intra prediction. desirable.
본 발명의 제7 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하 나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 부호화 할 화면내 예측 결과와 부호화 할 화면간 예측 결과를 비교하여 더 유리한 예측을 선택하여 최종 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	According to a seventh aspect of the present invention, in an inter prediction encoding method in video encoding, a coding block unit is extended to at least one super macro block having a size larger than 16 × 16 macro blocks, and each super macro block is expanded in various ways. After sub-blocks are divided into sub-blocks of size, intra-prediction is performed on the divided sub-blocks including each super macro block, and the intra-prediction result to be encoded is compared with the inter-prediction result to be encoded. It is an object of the present invention to provide an inter-prediction encoding method in video encoding, characterized by selecting and encoding an advantageous prediction.

본 발명의 제8 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a") 부호화하고자 하는 화면을 입력하는 단계; (b") 입력된 화면을 부호화 단위 블록으로 분할하는 단계; (c") 현재의 입력 화면이 화면간 예측부호화를 수행할 것인지를 판별하는 단계; (d") 현재의 입력 화면이 화면간 예측이 아닌 경우 화면내 예측 부호화를 수행하는 단계; (e") 현재의 입력 화면이 화면간 예측인 경우에는 입력 화면에서 부호화되는 단위 블록의 순서를 초기화하는 단계; (f") 부호화하고자 하는 단위 블록에 움직임 예측을 수행하기 위해 적용할 움직임 모델을 하나 선택한 후 움직임 예측을 수행하는 단계; (g") 상기 선택된 움직임 예측 결과를 이용하여 예측 부호화를 수행하는 단계; (h") 상기 단계(g")에서의 예측 부호화 결과 및 율-왜곡 비용값을 저장하는 단계; (i") 상기 적용 움직임 모델이 최종 모델인지 판단하여, 최종 움직임 모델이면 최적의 움직임 모델을 결정하고, 해당 부호화 결과를 최종 선택하는 단계; (j") 상기 적용 움직임 모델이 최종 모델이 아니면, 상기 단계(f")로 리턴하는 단계; (k") 현재 단위 블록이 현재의 입력 화면에서 최종 블록인지를 판별하는 단계; 및 (l") 현재 단위 블록이 현재의 입력 화면에서 최종 블록이면, 현재의 입력 화면이 최종 화면인지 판별하여, 최종 화면이 될 때까지 상기 단계(a") 내지 단계(k")를 반복 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법을 제공하는 것이다.	An eighth aspect of the present invention provides a method of inter-prediction encoding in video encoding, comprising: (a ") inputting a screen to be encoded, (b") dividing an input screen into coding unit blocks; (c ") determining whether the current input screen is to perform inter-prediction encoding; (d") performing intra prediction encoding if the current input screen is not inter prediction; (e ") initializing the order of unit blocks encoded in the input screen when the current input screen is inter-screen prediction; (f") selecting a motion model to be applied to perform motion prediction on the unit block to be encoded. Selecting one and performing motion prediction; (g ") performing predictive encoding using the selected motion prediction result; (h") storing the predictive encoding result and the rate-distortion cost value in step (g "); (i") the Determining whether the applied motion model is a final model, determining an optimal motion model if the final motion model is selected, and finally selecting a corresponding encoding result; (j ") if the applied motion model is not a final model, returning to step f "; (k ") determining whether the current unit block is the last block in the current input screen; and (l") if the current unit block is the last block in the current input screen, determining whether the current input screen is the last screen, It provides a method of inter-prediction encoding in video encoding, comprising the step of repeating the steps (a ") to (k") until the final picture.
상기 단계(f")에서, 상기 움직임 모델을 적용하여 움직임 예측 수행 시 움직임 모델 파라미터를 추정하는 단계를 더 포함함이 바람직하다.	In the step (f "), it is preferable to further include the step of estimating the motion model parameters when performing the motion prediction by applying the motion model.
본 발명의 제9 측면은, 상술한 비디오 부호화에서의 화면간 예측 부호화 방법을 실행시키기 위한 프로그램을 기록한 기록매체를 제공한다.	A ninth aspect of the present invention provides a recording medium on which a program for executing the above-described inter prediction encoding method in video encoding is recorded.
이상에서 설명한 바와 같은 본 발명의 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 따르면, 부호화하고자 하는 블록의 크기를 기존의 16×16 매크로 블록 크기 이상으로 확장하고, 이를 다시 하위 블록으로 분할하여 선택적으로 사용하며, 영상의 균질한 영역이 많은 영상에 대해 효과적으로 사용함으로써, 부호화되어야 하는 움직임 벡터 수의 감소와 움직임 예측의 부하를 줄여 효과적으로 예측 부호화를 수행할 수 있는 이점이 있다.	According to the inter-screen / intra-prediction coding method of the video encoding of the present invention as described above, the size of the block to be encoded is extended beyond the size of the existing 16x16 macroblock, and this is further divided into lower blocks. By selectively using and effectively using an image having a large number of homogeneous regions, there is an advantage in that predictive encoding can be effectively performed by reducing the number of motion vectors to be coded and reducing the load of motion prediction.

또한, 본 발명에 따르면, 다양한 움직임 모델을 사용하여 보다 정확한 움직임 예측을 수행함으로써 발생되는 잔차 신호의 양을 줄여 부호화 효율을 증대시킬 수 있을 뿐만 아니라 움직임 모델에 의한 참조 영상의 와핑(Warping)을 통해 정화소로부터 1/2-화소, 1/4-화소, 1/8-화소의 서브-펠(Sub-Pel) 단계로 움직임 보상을 수행함으로서 소요되는 움직임 예측의 많은 연산량의 감소를 얻을 수 있는 이점이 있다.	In addition, according to the present invention, the coding efficiency can be increased by reducing the amount of residual signals generated by performing more accurate motion prediction using various motion models, and by warping the reference image by the motion model. By performing motion compensation from the refinery to sub-pel steps of 1 / 2-pixel, 1 / 4-pixel, and 1 / 8-pixel, it is possible to obtain a large amount of computational reduction of the motion prediction required. have.
이하, 첨부 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다. 그러나, 다음에 예시하는 본 발명의 실시예는 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 다음에 상술하는 실시예에 한정되는 것은 아니다. 본 발명의 실시예는 당업계에서 통상의 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위하여 제공되어지는 것이다.	Hereinafter, embodiments of the present invention will be described in detail with reference to the accompanying drawings. However, embodiments of the present invention illustrated below may be modified in many different forms, and the scope of the present invention is not limited to the embodiments described below. The embodiments of the present invention are provided to more completely explain the present invention to those skilled in the art.
먼저, 통상적인 화면간 예측은 부호화하고자 블록을 최대 16×16 크기인 매크로 블록 단위로 분할하여 부호화한다. 특히, ISO/IEC 14496-10(MPEG-4 Advanced Video Coding) 또는 H.264 규격에서는 도 1과 같이 매크로 블록을 7가지의 하위 블록으로 분할하여 부호화 시에 율-왜곡 비용을 최소화하는 블록을 최종 선택하여 예측 부호화를 수행한다.	First, in conventional inter prediction, a block is encoded by dividing a block to be encoded into macroblock units having a maximum size of 16 × 16. In particular, in ISO / IEC 14496-10 (MPEG-4 Advanced Video Coding) or H.264 standard, as shown in FIG. 1, the macro block is divided into seven sub-blocks to finalize a block that minimizes rate-distortion cost in encoding. Selectively perform prediction encoding.
전술한 바와 같이 부호화하고자 하는 블록 단위를 16×16 매크로 블록 단위로 제한하는 것은 영상 내의 균질 영역(Homogeneous Regions) 또는 움직임이 없는 영역에서 화면간 예측을 수행할 때에 매크로 블록의 수가 많이 증가한다. 특히, 초 고해상도 영상의 부호화에서는 매크로 블록의 크기가 상대적으로 너무 작아 부호화 효율을 떨어뜨리는 결과를 초래한다.	As described above, limiting the block unit to be encoded to 16 × 16 macroblock units greatly increases the number of macroblocks when performing inter-prediction in homogeneous regions or regions without motion. In particular, in the encoding of an ultra high resolution image, the size of the macro block is relatively small, resulting in a decrease in the coding efficiency.
도 4는 본 발명의 일 실시예에 적용된 16×16 매크로 블록 크기 이상의 수퍼 매트로 블록 및 블록 분할 타입의 일 예를 나타낸 도면으로서, 매크로 블록의 크기를 확장하고 하위 블록을 다양한 크기로 분할하여 사용함으로써 상술한 균질 영역 부호화나 움직임이 없는 영상의 부호화에 매우 효과적으로 적용될 수 있다.	FIG. 4 is a diagram illustrating an example of a block and block division type with a super mat of 16 × 16 macroblock size or larger applied to an embodiment of the present invention. The macroblock is expanded in size and the lower block is divided into various sizes. Thus, the present invention can be effectively applied to the above-mentioned homogeneous region encoding or encoding of a motion-free image.
즉, 32×32 수퍼 매크로 블록 및 그 하위 분할 블록, 64×64 수퍼 매크로 블록 및 그 하위 분할 블록, 128×128 수퍼 매크로 블록 및 그 하위 분할 블록 등의 형태로 블록 크기를 확장할 수 있다.	That is, the block size can be extended in the form of a 32x32 super macro block and its subdivided blocks, a 64x64 super macro block and its subdivided blocks, a 128x128 super macro block and its subdivided blocks, and the like.

또한, 부호화기의 복잡도를 고려하여 블록 크기의 확장을 제한하거나 그 하위 블록의 블록 분할 타입의 수를 제한할 수 있다. 실제 부호화 시에 전술한 수학식 1에서와 같이 율-왜곡 비용을 최소화하는 블록 모드(Mode)를 선택하여 부호화할 수 있다.	In addition, in consideration of the complexity of the encoder, it is possible to limit the expansion of the block size or limit the number of block division types of the lower block. In actual encoding, as in Equation 1 described above, a block mode that minimizes the rate-distortion cost may be selected and encoded.
뿐만 아니라 크기가 증가된 수퍼 매크로 블록의 잔차 신호에 대해 변환 부호화(Transform Coding)를 적용할 경우 기존의 4×4, 8×8 또는 16×16 보다 큰 변환 커널(Kernel)을 사용함으로써 부호화 효율을 증가시킬 수 있다.	In addition, when transform coding is applied to a residual signal of an increased sized super macro block, encoding efficiency is increased by using a transform kernel larger than 4 × 4, 8 × 8, or 16 × 16. Can be increased.
또한, 화면간 예측 부호화에 부호화하고자 하는 블록에 대해 움직임 예측 모델을 선택적으로 적용하여 부호화 효율을 더욱 증가시킬 수 있다.	In addition, coding efficiency may be further increased by selectively applying a motion prediction model to a block to be encoded in inter prediction encoding.
이하에는 움직임 예측 모델에 대하여 구체적으로 설명하기로 한다.	Hereinafter, the motion prediction model will be described in detail.
움직임 예측 모델을 보다 상세히 설명하기 위해 먼저, f_c와 f_r을 각각 부호화하고자 하는 현재 영상과 참조 영상으로 한다. (x,y)를 f_c의 부호화하고자 하는 현재 블록의 한 픽셀 좌표값을 나타낸다. 또한, 부호화하고자 하는 현재 프레임의 예측을 $\hat{f}_c$ 라고 하면 참조 영상으로부터 하기의 수학식 4와 같이 정의할 수 있다.	In order to describe the motion prediction model in more detail, first, f_c and f_r are used as current and reference pictures to be encoded, respectively. (x, y) represents one pixel coordinate value of the current block to be encoded by f_c. In addition, the prediction of the current frame to be encoded $\hat{f}_c$ In this case, it can be defined as shown in Equation 4 below from the reference image.
$\hat{f}_c(x,y) = f_r(u,v)$	$\hat{f}_c(x,y) = f_r(u,v)$
여기서, (u,v)는 참조 영상의 좌표값을 나타낸다. 상기의 수학식 4는 움직임 보상이 참조 영상의 좌표값을 현재 영상의 좌표값에 대응시키는 기하학적 변환(Geometric Transformation)을 의미하며 하기의 수학식 5와 같이 일반화하여 표현할 수 있다.	Here, (u, v) represents a coordinate value of the reference image. Equation 4 denotes a geometric transformation in which motion compensation corresponds to a coordinate value of the current image, and may be generalized as shown in Equation 5 below.
$u = g_x(u,v), \quad v = g_y(u,v)$	$u = g_x(u,v), \quad v = g_y(u,v)$
여기서, g_x와 g_y는 공간 변환 함수를 나타낸다. 이러한 공간 변환을 텍스쳐 매핑(Texture Mapping) 또는 이미지 매핑(Image Mapping)으로 불린다.	Where g_x and g_y represent spatial transform functions. This spatial transformation is called texture mapping or image mapping.
움직임 예측에 일반적으로 사용되는 블록 정합 방법인 2차원 병진 움직임 모형은 하기의 수학식 6과 같이 표현된다.	A 2D translational motion model, which is a block matching method generally used for motion prediction, is expressed by Equation 6 below.

$u=g_x(u,v)=x+a_1=x+d_x$ $v=g_y(u,v)=y+a_2=y+d_y$	$u=g_x(u,v)=x+a_1=x+d_x$ $v=g_y(u,v)=y+a_2=y+d_y$
그러나, 전술한 바와 같이 실제 2차원 투영 영상에는 병진 운동뿐만 아니라 줌(Zoom), 회전(Roll), 패닝(Panning), 붐(Boom), 틸팅(Tilting), 달리(Dolly) 등의 움직임이 복합적으로 혼재되어 있다.	However, as described above, not only the translational movement but also the movement of zoom, rotation, panning, boom, tilting, and dolly are combined in the actual two-dimensional projection image. Are mixed.
이러한 복잡한 움직임을 제대로 예측하기 위해서는 보다 정교한 움직임 모델이 필요하다. 하기의 수학식 7은 의사(擬似)(Affine) 움직임 모델, 수학식 8은 이중선형(Bilinear) 움직임 모델, 그리고 수학식 9는 투시(Perspective) 움직임 모델을 각각 나타낸다.	More accurate motion models are needed to properly predict these complex movements. Equation 7 below represents an Affine motion model, Equation 8 represents a bilinear motion model, and Equation 9 represents a perspective motion model.
$u = g_x(u,v) = a_1x + a_2y + a_3$ $v = g_y(u,v) = a_4x + a_5y + a_6$	$u = g_x(u,v) = a_1x + a_2y + a_3$ $v = g_y(u,v) = a_4x + a_5y + a_6$
$u=g_x(u,v)=a_1xy+a_2x+a_3y+a_4$ $v=g_y(u,v)=a_5xy+a_6x+a_7y+a_8$	$u=g_x(u,v)=a_1xy+a_2x+a_3y+a_4$ $v=g_y(u,v)=a_5xy+a_6x+a_7y+a_8$
$u = g_x(u,v) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1}$ $v = g_y(u,v) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1}$	$u = g_x(u,v) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1}$ $v = g_y(u,v) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1}$
여기서, a_i는 움직임 모델 파라미터를 나타낸다. 움직임 모델 기반의 예측 부호화를 수행할 경우, 상기 모델들 중 가장 율-왜곡 비용을 최소화하는 모델을 적용하여 부호화함으로써 부호화 효율을 증가시킬 수 있다.	Here, a $_i$ represents a motion model parameter. When performing prediction coding based on motion model, encoding efficiency may be increased by applying a model that minimizes the rate-distortion cost among the models.
본 발명에서는 움직임 모델을 적용하는 단위 블록의 크기는 수퍼 매크로 블록과 그 하위 블록이 될 수 있으며 부호화 효율에 따라 최적의 움직임 모델을 선택할 수 있다. 한편, 본 발명의 일 실시예에 적용된 움직임 모델은 파라미터로 모형화됨이 바람직하다.	In the present invention, the size of a unit block to which a motion model is applied may be a super macro block and a lower block thereof, and an optimal motion model may be selected according to coding efficiency. Meanwhile, the motion model applied to the embodiment of the present invention is preferably modeled by a parameter.
한편, 본 발명의 일 실시예에서는 비디오 부호화에서의 화면간 예측 부호화 방법 위주로 설명하였지만, 이에 국한하지 않으며, 비디오 부호화에서의 화면내 예측 부호화에 적용시킬 수도 있다.	Meanwhile, although an embodiment of the present invention has been described mainly for the inter prediction encoding method in video encoding, the present invention is not limited thereto and may be applied to the intra prediction encoding in video encoding.

즉, 화면내 예측 부호화를 수행할 경우, 먼저, 부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 비교하여 더 유리한 부호화 결과를 선택하여 최종 부호화한다.	That is, when performing intra prediction encoding, first, a coding block unit is extended to at least one super macro block having a size larger than 16 × 16 macro blocks, and each super macro block is divided into sub-blocks having various sizes. Intra-picture prediction is performed on the divided lower blocks including each super macro block, and the results are compared to select a more advantageous encoding result and to perform final encoding.
이때, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 율-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화함이 바람직하다.	In this case, intra prediction encoding is performed on the sub-blocks including each super macro block by performing directional prediction, and encoding the directional prediction mode that generates the least rate-distortion cost as the optimal final intra prediction. desirable.
도 5는 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간 예측 부호화 방법을 설명하기 위한 전체적인 흐름도로서, 비디오 부호화에서 선택적 움직임 모델 기반의 화면간 예측 부호화 방법을 나타낸 것이다.	FIG. 5 is a flowchart illustrating an inter prediction encoding method in video encoding according to an embodiment of the present invention, and illustrates an inter motion prediction encoding method based on a selective motion model in video encoding.
도 5를 참조하면, 먼저, 부호화를 위한 화면(i) 순서를 초기화(i=0)한 후(S100), 부호화를 위해 순서에 맞게 화면(i)을 입력한다(S101).	Referring to FIG. 5, first, the order of screen i for encoding is initialized (i = 0) (S100), and then the screen i is input in order for encoding (S101).
그런 다음, 상기 단계S101에서 입력된 화면(i)을 부호화 단위 블록(매크로 블록 또는 수퍼 매크로 블록)으로 분할한 후(S102), 현재의 화면(i)이 화면간 예측부호화를 수행할 것인지를 판단한다(S103).	Then, after dividing the screen i input in step S101 into coding unit blocks (macro blocks or super macro blocks) (S102), it is determined whether the current screen i is to perform inter-prediction encoding. (S103).
만약, 상기 단계S103에서의 판단 결과, 현재의 화면(i)이 화면간 예측이 아닌 경우 화면내 예측 부호화를 수행하고(S104), 그렇지 않고 현재의 화면(i)이 화면간 예측인 경우에는 한 화면(i)내에서 부호화되는 단위 블록(j)의 순서를 초기화(j=0)한다(S105).	If it is determined in step S103 that the current screen i is not inter-screen prediction, intra prediction coding is performed (S104). Otherwise, if the current screen i is inter-screen prediction, The order of the unit blocks j encoded in the screen i is initialized (j = 0) (S105).
이후에, 부호화하고자 하는 단위 블록(j)에 움직임 예측을 수행하기 위해 적용할 움직임 모델을 하나 선택한 후 이에 대해 움직임 예측을 수행하고(S106), 그 결과를 이용하여 예측 부호화를 수행한 후(S107), 그 결과인 예측 부호화 결과 및 율-왜곡 비용값을 저장한다(S108).	Subsequently, one motion model to be applied to perform the motion prediction on the unit block j to be encoded is selected, motion prediction is performed on the motion block (S106), and prediction prediction is performed using the result (S107). The predictive encoding result and the rate-distortion cost value are stored (S108).

그런 다음, 사용 움직임 모델이 최종 움직임 모델인지를 판단하여(S109), 최종 움직임 모델이 아닐 경우 상기 단계S106으로 리턴되어 다음 적용할 움직임 모델을 선택하고 움직임 예측을 수행하며 이 과정을 반복한다.	Then, it is determined whether the used motion model is the final motion model (S109), and if it is not the final motion model, the method returns to step S106 to select a motion model to be applied next, performs motion prediction, and repeats this process.
그렇지 않고, 상기 단계S109에서의 판단 결과, 사용 움직임 모델이 최종 움직임 모델이면 최적의 움직임 모델을 결정하고 해당 부호화 결과를 최종 선택한다(S110).	Otherwise, if the use motion model is the final motion model as a result of the determination in step S109, an optimal motion model is determined and the corresponding encoding result is finally selected (S110).
이후에, 현재 단위 블록(j)이 화면(i)에서 최종 블록인지를 판단하여(S111), 최종 블록이 아니면 다음 부호화 단위 블록(j)을 입력하고 상기 단계S106으로 리턴한다.	Subsequently, it is determined whether the current unit block j is the last block in the screen i (S111). If not, the next coding unit block j is input and the process returns to step S106.
그렇지 않고, 상기 단계S111에서의 판단 결과, 현재 단위 블록(j)이 화면(i)에서 최종 블록이면, 현재 화면(i)이 최종 화면인지 판단하여(S112), 최종 화면이면 종료하고 그렇지 않으면, 상기 단계S101로 리턴되어 다음 화면을 입력하고 전술한 과정들을 수행한다.	Otherwise, as a result of the determination in step S111, if the current unit block j is the last block in the screen i, it is determined whether the current screen i is the last screen (S112). Returning to step S101, the next screen is input and the above-described processes are performed.
한편, 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간/화면내 예측 부호화 방법은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현되는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다.	Meanwhile, the inter-screen / in-screen predictive encoding method in video encoding according to an embodiment of the present invention may also be implemented as computer readable codes on a computer readable recording medium. The computer-readable recording medium includes all kinds of recording devices in which data that can be read by a computer system is stored.
예컨대, 컴퓨터가 읽을 수 있는 기록매체로는 롬(ROM), 램(RAM), 시디-롬(CD-ROM), 자기 테이프, 하드디스크, 플로피디스크, 이동식 저장장치, 비휘발성 메모리(Flash Memory), 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들면, 인터넷을 통한 전송)의 형태로 구현되는 것도 포함된다.	For example, the computer-readable recording medium may be a ROM, a RAM, a CD-ROM, a magnetic tape, a hard disk, a floppy disk, a removable storage device, a nonvolatile memory (Flash memory). Optical data storage, and the like, and also implemented in the form of a carrier wave (eg, transmission over the Internet).
또한, 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터 통신망으로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 읽을 수 있는 코드로서 저장되고 실행될 수 있다.	The computer readable recording medium can also be distributed over computer systems connected over a computer network so that the computer readable code is stored and executed in a distributed fashion.

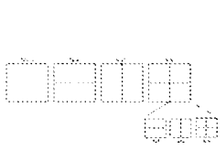
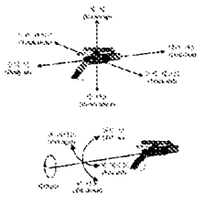
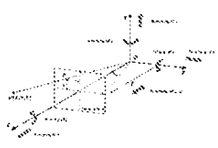
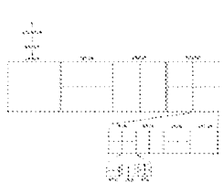
전술한 본 발명에 따른 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 대한 바람직한 실시예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 특허청구범위와 발명의 상세한 설명 및 첨부한 도면의 범위 안에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 본 발명에 속한다.	Although preferred embodiments of the inter-screen / in-screen prediction coding method in video encoding according to the present invention have been described above, the present invention is not limited thereto, but the claims and the detailed description of the invention and the scope of the accompanying drawings. Various modifications can be made therein and this also belongs to the present invention.
도 1은 종래 기술의 H.264|AVC 부호화기에서 화면간 예측 부호화를 위한 16×16 매크로 블록 단위 블록의 분할 블록 타입을 나타낸 도면.	BRIEF DESCRIPTION OF THE DRAWINGS Fig. 1 is a diagram showing a divided block type of a 16 × 16 macroblock unit block for inter prediction encoding in a conventional H.264	AVC encoder.
도 2는 비디오 영상에서의 카메라의 움직임에 의한 움직임 생성 타입을 설명하기 위한 도면.	FIG. 2 is a diagram for describing a motion generation type due to camera movement in a video image. FIG.
도 3은 2차원 영상 공간에 투영된 움직임 모델을 설명하기 위한 개념도.	3 is a conceptual diagram illustrating a motion model projected on a two-dimensional image space.
도 4는 본 발명의 일 실시예에 적용된 16×16 매크로 블록 크기 이상의 수퍼 매트로 블록 및 블록 분할 타입의 일 예를 나타낸 도면.	4 is a diagram illustrating an example of a block and a block division type of a super mat having a size of 16 × 16 macroblocks or more applied to an embodiment of the present invention.
도 5는 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간 예측 부호화 방법을 설명하기 위한 전체적인 흐름도.	5 is an overall flowchart illustrating a method of inter prediction encoding in video encoding according to an embodiment of the present invention.

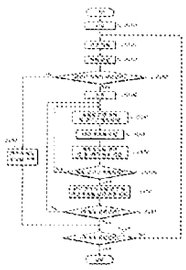
Legal events

Code	Title	Date	Country	Impact	Details
A201	REQUEST FOR EXAMINATION	2008-02-11	KR		
N231	NOTIFICATION OF CHANGE OF APPLICANT	2009-07-07	KR		
E902	NOTIFICATION OF REASON FOR REFUSAL	2009-09-24	KR	-	
E701	DECISION TO GRANT OR REGISTRATION OF PATENT RIGHT	2010-03-22	KR	+	
GRNT	WRITTEN DECISION TO GRANT	2010-03-29	KR	+	
FPAY	ANNUAL FEE PAYMENT	2013-03-04	KR		payment date 20130304 year of fee payment 4
FPAY	ANNUAL FEE PAYMENT	2014-03-03	KR		payment date 20140303 year of fee payment 5
FPAY	ANNUAL FEE PAYMENT	2015-07-16	KR		payment date 20150716 year of fee payment 6

Code	Title	Date	Country	Impact	Details
FPAY	ANNUAL FEE PAYMENT	2016-02-25	KR		payment date 20160225 year of fee payment 7
FPAY	ANNUAL FEE PAYMENT	2017-02-24	KR		payment date 20170224 year of fee payment 8
FPAY	ANNUAL FEE PAYMENT	2018-02-26	KR		payment date 20180226 year of fee payment 9
FPAY	ANNUAL FEE PAYMENT	2019-03-04	KR		payment date 20190304 year of fee payment 10
FPAY	ANNUAL FEE PAYMENT	2020-03-02	KR		payment date 20200302 year of fee payment 11

Images



ITU - Telecommunications Standardization Sector
STUDY GROUP 16 Question 6
Video Coding Experts Group (VCEG)
36th Meeting: San Diego, USA, 8-10 October, 2008

Document VCEG-AJ23
Filename: VCEG-AJ23.doc

Question: Q.6/SG16 (VCEG)

Source: P. Chen, Y. Ye and M. Karczewicz
5775 Morehouse Dr
San Diego, CA 92121

Tel: +1-858-8455080
Fax: +1-848-6585673
Email: {peisongc,yye,martak}@qualcomm.com

Title: **Video Coding Using Extended Block Sizes**

Purpose: Proposal

Abstract

In this contribution, video coding using motion partitions larger than 16x16 is implemented. In particular, partition sizes of 32x32, 32x16, and 16x32 are used in addition to the partition sizes already allowed in H.264. Larger block transforms are also used. Under VCEG common testing conditions, the proposed scheme achieves 12.4% bit rate reduction on average for the HD test sequences for hierarchical B coding.

The Proposed Solution

It has been demonstrated that video coding using blocks bigger than 16x16 can provide substantial coding gain compared to H.264, especially when applied on high definition video sequences [2]. In this contribution, we propose a coding system that encodes the input video using bigger than 16x16 blocks as follows:

1. New motion partitions (32x32, 32x16, and 16x32) are used.

In addition to the existing motion partition sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4) in H.264, inter coding using 32x32, 32x16, and 16x32 partitions is also enabled. Figure 1 shows all the motion block partitions allowed in the proposed scheme. Consider each 32x32 block as a "super macroblock" or SMB. Input video is coded in units of SMBs. For each SMB, smb_skip_flag signals the SKIP or DIRECT mode in the same way as in H.264. In addition, four values of smb_type are signaled with the following meanings:

0: 32x32 partition
1: 32x16 partition
2: 16x32 partition
3: 16x16 partition

The smb_type of an SMB is signaled in the same way as the mb_type of a macroblock; in other words, all the CABAC contexts and context initialization remain the same. If smb_type is between 0 and 2 (i.e., bigger than 16x16 partition is used), then one motion vector per prediction list is signaled for each partition. If smb_type is between 0 and 2, or if the SMB has DIRECT mode, residual coefficients are signaled for each 16x16 MB in the SMB in raster scan order, re-using the same syntax elements as in residual_block() in H.264 (except when bigger transforms are used as described in 2).

If the mb_type of an SMB is 3 (i.e., 16x16 partition is used), then the four 16x16 macroblocks in the SMB are signaled in raster scan order using the same syntax elements as macroblock_layer() in H.264. In other words, each 16x16 macroblock may be partitioned further in the quadtree manner, from size 16x16 down to size 4x4. Also the signaling of cbp and transform coefficients remain the same as in H.264 (except for bigger block transforms as described in 2).

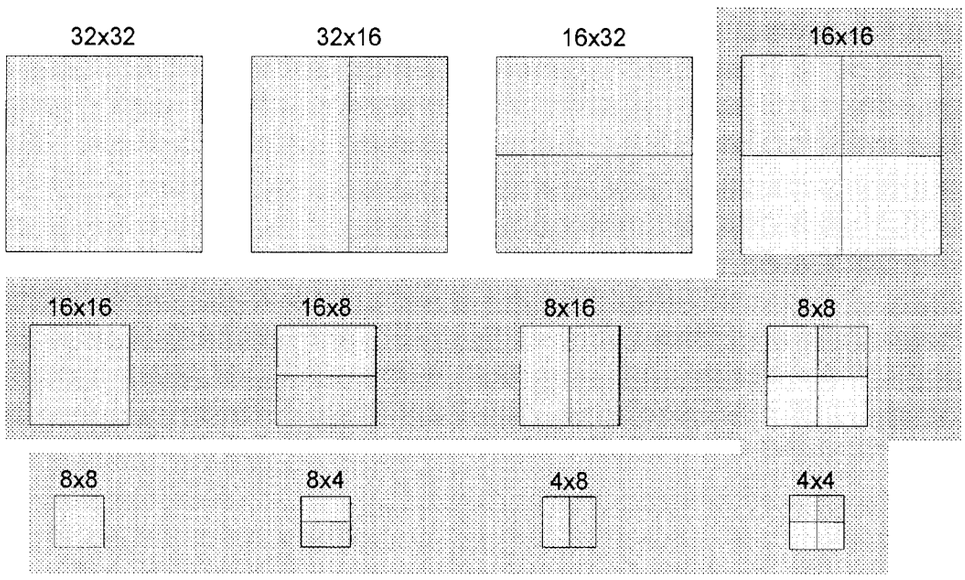


Figure 1: Motion partitions used in the proposed scheme

2. New block transforms with larger basis functions are used.

Bigger transforms allow to better capture the usually smoother content in high definition video. For inter coded macroblocks, 16x16, 16x8, and 8x16 transforms are used in addition to 4x4 and 8x8 integer transforms. For each motion partition of size MxN, where M, N <= 16, MxN transform is used. For bigger motion partitions, 16x16 transform is used. For intra coded macroblocks, MDDT [3] is used. The bigger block transform coefficients are sent with minimal changes to the significance, last significance, and magnitude coding in CABAC.

Simulation Results

The proposed scheme is implemented in KTA2.0 and tested under the VCEG common testing conditions [1]. The configuration UseRDO_Q is set to 1 but macroblock level delta QP is disabled due to software constraint. Tables 1 shows the percentages of BD rate reduction for all HD test sequences. For IPPP coding, average rate reduction of 6.5% is achieved; for IBBP coding, average rate reduction of 10.4% is achieved; and for hierarchical B coding, average rate reduction of 12.4% is achieved. The proposed scheme becomes more effective as the video resolution increases. For the 1080p sequence "rolling_tomatoes", the proposed scheme achieves up to 21% rate reduction.

Table 2: Simulation results using the proposed scheme

		IPPP	IBBP	Hier B
HD	bigships	-9.1	-6.72	-7.08
	city	-6.1	-10.8	-14.88
	crew	-7.37	-10.99	-12.59
	night	-1.11	-4.54	-5.48
	shuttlestart	-6.26	-11.42	-13.42
	rolling tomatoes	-9.2	-18.19	-20.71
Average		-6.52	-10.44	-12.36



References

[1] TK Tan, G. Sullivan, and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 1," ITU-T Q.6/SG16 VCEG, VCEG-AE10, Marrakech, Morocco, January 2007.

[2] K. Lee et. al., "Technical considerations for Ad Hoc Group on New Challenges in Video Coding Standardization." ISO/IEC JTC1/SC29/WG11 doc. M1558, July 2008, Hannover, Germany.

[3] Y. Ye and M. Karczewicz, "Improved Intra Coding," ITU-T Q.6/SG16 VCEG, VCEG-AG11, Shenzhen, China, October 2007.

IPR Notification: Qualcomm may have IPR relating to the technology described this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

Home Mail Search News Sports Finance Weather Games Answers Get the Yahoo Toolbar Get it for Firefox Mobile
Search Conversations
Tiobias Mail

Groups Home
vceg-experts 1
All My Groups
Manage My Groups
Create a Group
Browse Groups
Terms Privacy
Guidelines Feedback
Help Blog

ITU-T Video Coding Experts Group (VCEG)

Restricted Group, 721 members

511 **Contributions VCEG-AL28 and AL29 have been...**

정세윤 Jul 1, 2009

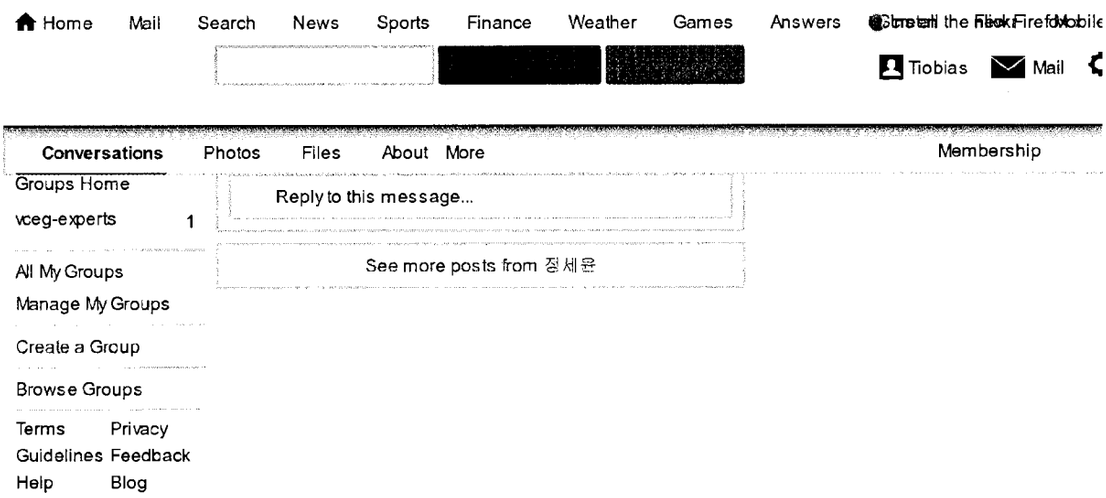
Dear experts,

VCEG-AL28 and VCEG-AL29 have been uploaded to the dropbox.

VCEG-AL28: "Intra coding using extended block size"
Abstract
Inter coding using extended block size was already adopted in KTA. In this contribution, we focus on intra coding using extended block size, which allows intra_32x32 prediction as well as the existing intra predictions. Larger block transforms are also used and more directional prediction modes for intra_16x16 are added. Under VCEG common testing conditions except RDO-Q, the proposed method achieved 1.83% BD-RATE reduction on average over all test sequences for only intra coding.

VCEG-AL29: "Intra Prediction with Offset"
Abstract
The methods using the offset in order to improve coding efficiency have been proposed and some of those are adopted into KTA S/W. Qualcomm proposed to use DC offset on the frame-level or on the sub-pixel position level with switched interpolation filters (SIFO). Tandberg and Nokia used LDCO (Local DC Offset), local fixed offsets such as +1 and -1, on the fractional position in their directional interpolation filter scheme. In intra prediction mode, neighboring pixels in reconstructed blocks can be used as intra reference block for the current block to be coded. In order to reduce further the prediction error of the intra reference block, the proposed method introduces an intra prediction offset which is determined in a sense of the rate-distortion optimization and is added to the conventional intra prediction block. The proposed method was implemented on the KTA2.1 and its experimental results showed average of 0.99% and up to 2.08% of bitrate saving over all test sequences under the VCEG common condition VCEG-AJ10r1. Furthermore, when the proposed method is used together with MDDT and RDO-Q, average 11.70% of bitrate saving was achieved in comparison with JM11.0.

Best Regards,


Home Mail Search News Sports Finance Weather Games Answers
Tiobias Mail
Conversations Photos Files About More Membership
Groups Home
vceg-experts 1
All My Groups
Manage My Groups
Create a Group
Browse Groups
Terms Privacy
Guidelines Feedback
Help Blog
Reply to this message...
See more posts from 정세윤

Home Mail Search News Sports Finance Weather Games Answers Get it all the new Firefox Mobile
Tiobias Mail
Conversations Photos Files About More Membership
Groups Home
vceg-experts 1
All My Groups
Manage My Groups
Create a Group
Browse Groups
Terms Privacy
Guidelines Feedback
Help Blog

INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2009-2012

COM 16 – C 123 – E

January 2009

English only

Original: English

Question(s): 6/16

STUDY GROUP 16 – CONTRIBUTION 123

Source: Qualcomm Inc.

Title: Video Coding Using Extended Block Sizes

Abstract

In this contribution, video coding using motion partitions larger than 16x16 pixels is implemented. In particular, macroblocks of size 64x64 is used. Larger block transforms are also used. Under VCEG common testing conditions, the use of bigger motion partitions and larger block transforms achieves 19.9% BD-Rate reduction on average for the HD test sequences for hierarchical B coding.

1 The proposed solution

It has been demonstrated that video coding using blocks bigger than 16x16 pixels can provide substantial coding gain compared to H.264, especially when applied to high definition video sequences [2]. In this contribution, we propose a coding system that encodes the input video by using blocks bigger than 16x16 pixels as follows:

1.1 New motion partitions (32x32, 32x16, and 16x32) for 32x32 blocks

In addition to the existing motion partition sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4) in H.264, inter coding using 32x32, 32x16, and 16x32 partitions is also enabled. Figure 1 shows the motion block partitions allowed in the proposed scheme. For each 32x32 block, mb32_skip_flag signals the SKIP or DIRECT mode in the way similar to that of H.264. In addition, the original mb_type for MxN (M =8, or 16 and N =8, or 16) partition in H.264 is also used to signal 2Mx2N partition in a 32x32 block. For example mb_type 3 for 8x16 macroblock partition in a P slice in H.264 is also used for 16x32 partition in a P slice in this proposal. The mb32_type of a 32x32 block is encoded in a similar way as the mb_type of a macroblock; in other words, mb32_type is binarized in a similar way as mb_type of H.264 and shares the same context index as mb_type of H.264.

If partition bigger than 16x16 is used, one motion vector per prediction list is signalled for each partition (for modes other than skip mode and direct mode). Residual coefficients are signalled in a

Contact:	Peisong Chenn Qualcomm Inc USA	Tel: +1-858-845-1347 Fax: +1-858-651-2236 Email: peisongc@qualcomm.com
Contact:	Yan Ye Qualcomm Inc USA	Tel: +1-858-845-5080 Fax: +1-858-651-2236 Email: yye@qualcomm.com
Contact:	Marta Karczewicz Qualcomm Inc USA	Tel:+1-858-658-5673 Fax:+1-858-651-2236 Email: martak@qualcomm.com

Attention: This is not a publication made available to the public, but **an internal ITU-T Document** intended only for use by the Member States of ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of ITU-T.

hierarchical way. A 1 bit syntax cbp32 has been introduced for the whole 32x32 block with 1 meaning that the whole block has at least 1 nonzero coefficient and 0 meaning there is no coefficient with a magnitude greater than zero. If cbp32 is equal to 1, for each 16x16 block in the 32x32 block in raster scan order, cbp and coefficient information for the 16x16 block are encoded in the same way as H.264 (except when bigger transforms are used, case described in section 3).

If the mb32_type of a 32x32 indicates that a 16x16 partition is used, then the four 16x16 blocks are signalled in raster scan order by using the same syntax elements as macroblock_layer() in H.264 except that the delta QP information is sent once for the whole 32x32 block. In other words, the whole 32x32 block shares the same QP. Each 16x16 block may be partitioned further in the quadtree manner, from size 16x16 down to size 4x4. Also the signalling of cbp and residual coefficients remains the same as in H.264 (except for the special case described in section 3).

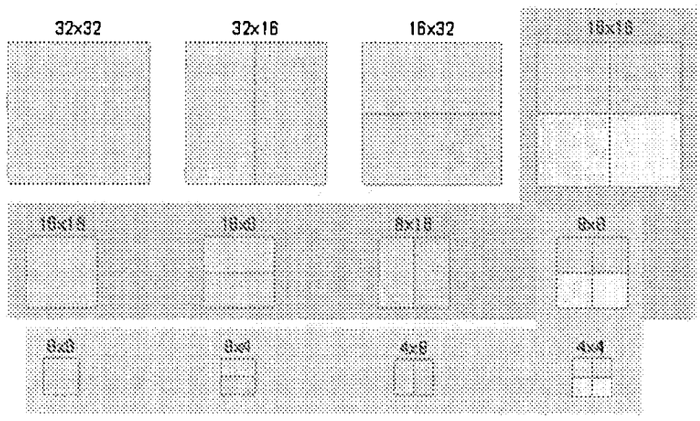


Figure 1: Motion partitions used in the proposed 32x32 blocks

1.2 New motion partitions (64x64, 64x32, and 32x64) for macroblock size 64x64.

For macroblock size 64x64, we also introduce 64x64, 64x32, and 32x64 above the partitions used in 32x32 blocks. Therefore we have one more hierarchy layer in the macroblock partition above block size 32x32. The original mb_type for MxN (M =8, or 16 and N =8, or 16) macroblock partition in H.264 is used also to signal a 4Mx4N macroblock partition in 64x64 macroblocks. If 32x32 macroblock partition is used for a 64x64 block, each 32x32 block will be handled in the same way described in the previous section.

If macroblock partition bigger than 32x32 is used, one motion vector per prediction list is signalled for each partition (for modes other than skip mode and direct mode). Residual coefficients are signalled in a hierarchical way. A 1 bit syntax cbp64 has been introduced for the whole 64x64 block and with 1 meaning that the whole block has at least 1 nonzero coefficient and 0 meaning that there is no coefficient with magnitude greater than zero. If cbp64 is 1, for each 32x32 block in the 64x64 block in raster scan order, cbp32 and coefficient information for the 32x32 block are handled in the same way as described in the previous section.

1.3 New block transforms with larger basis functions are used.

Bigger transforms allow to better capture the usually smoother content in high definition video. For inter coded macroblocks, 16x16, 16x8, and 8x16 transforms are used in addition to 4x4 and 8x8 integer transforms. Specifically, for each motion partition of size 16x16, 16x8, and 8x16, 16x16, 16x8 and 8x16 transforms may be used in addition to the existing 4x4 and 8x8 transforms. The

additional transform choice is signalled by transform_size_8x8_flag = 2. For these motion partitions, an additional bit cbp16 is signalled to indicate whether there is non-zero luma coefficient in the macroblock. If cbp16 is 1, then transform_size_8x8_flag is signalled. If transform_size_8x8_flag is not equal to 2, or if partition size is not 16x16, then the existing syntax element cbp is signalled. If cbp16 is 0, no more luma residual information is sent. When cbp16 = 1 and transform_size_8x8_flag = 2, the bigger block transform coefficients for luma component are sent with minimal changes to the significance, last significance, and magnitude coding in CABAC. For chroma residual, if cbp16=0 or if transform_size_8x8_flag=2 and partition size is 16x16, then chroma residual information is signalled by sending chroma_cbp and chroma residual coefficients using the existing H.264 syntax elements.

For motion partitions bigger than 16x16, 16x16 transform is used in addition to 4x4 and 8x8 transforms. For these motion partitions, an additional bit cbp16 is signalled for each 16x16 block to indicate whether there is non-zero luma coefficient in the macroblock. If cbp16 is 1, then transform_size_8x8_flag is signalled. If transform_size_8x8_flag is not equal to 2, then the existing syntax element cbp is signalled. If cbp16 is 0, no more luma residual information is sent. When cbp16 = 1 and transform_size_8x8_flag = 2, the bigger block transform coefficients are sent with minimal changes to the significance, last significance, and magnitude coding in CABAC. For chroma residual, if cbp16=0 or transform_size_8x8_flag=2, then chroma residual information is signalled by sending chroma_cbp and chroma residual coefficients using the existing H.264 syntax elements.

2 Simulation Results

The proposed scheme is implemented in KTA2.0 and tested under the VCEG common testing conditions [1]. The configuration parameters UseRDO_Q, UseHPFilter and UseNewOffset are set to 1 in both the reference and our proposed algorithm. In our proposed algorithm we enable bigger motion partition, MDDT [3] (for intra coded macroblocks) and also bigger transform for inter coded macroblocks. Tables 1 shows the percentages of BD-Rate reduction for all HD test sequences. On average, our proposed algorithm provides 15.1% and 19.9% BD-Rate reduction for IPPP and hierB coding.

Table 1: Simulation results using bigger partition and bigger transform

	IPPP	hierB
bigships	-16.9	-13.2
city	-12.9	-23.7
crew	-19.2	-23.0
night	-6.3	-8.5
jets	-26.5	-36.2
raven	-25.0	-33.5
Janine	-5.4	-3.9
1080 CrowdRun	-2.2	-2.8
1080 ParkJoy	-2.3	-3.9
1080 Toysandcalendar	-21.3	-34.1
1080 Sunflower	-27.9	-36.4
Average	**-15.1**	**-19.9**

References

[1] TK Tan, G. Sullivan, and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 3," VCEG-AI10. July 2008

[2] K. Lee et. al., "Technical considerations for Ad Hoc Group on New Challenges in Video Coding Standardization." ISO/IEC JTC1/SC29/WG11 doc. M1558, July 2008, Hannover, Germany.

[3] Y. Ye and M. Karczewicz, "Improved Intra Coding," ITU-T Q.6/SG16 VCEG, VCEG-AG11, Shenzhen, China, October 2007.

Electronic attachment

- Associated code
  http://itu.int/md/dologin_md.asp?id=T09-SG16-C-0123!A1!ZIP-E

IPR Notification: Qualcomm may have IPR relating to the technology described this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.264

(03/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video

Advanced video coding for generic audiovisual services

ITU-T Recommendation H.264

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	**H.260–H.279**
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.264

Advanced video coding for generic audiovisual services

Summary

This Recommendation | International Standard represents an evolution of the existing video coding standards (H.261, H.262, and H.263) and it was developed in response to the growing need for higher compression of moving pictures for various applications such as videoconferencing, digital storage media, television broadcasting, Internet streaming, and communication. It is also designed to enable the use of the coded video representation in a flexible manner for a wide variety of network environments. The use of this Recommendation | International Standard allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels.

The revision approved 2005-03 contains modifications of the video coding standard to add four new profiles, referred to as the High, High 10, High 4:2:2, and High 4:4:4 profiles, to improve video quality capability and to extend the range of applications addressed by the standard (for example, by including support for a greater range of picture sample precision and higher-resolution chroma formats). Additionally, a definition of new types of supplemental data has been specified to further broaden the applicability of the video coding standard. Finally, a number of corrections to errors in the published text have been included. This revision, in addition to enhancing video coding capability, will serve to maintain technical alignment with the corresponding jointly-developed ISO/IEC 14496-10 standard.

Corrigendum 1 to ITU-T Rec. H.264 corrected and updated various minor aspects to bring the ITU-T version of the text up to date relative to the April 2005 output status approved as a new edition of the corresponding jointly-developed and technically-aligned text ISO/IEC 14496-10. It additionally fixes a number of minor errors and needs for clarification and defines three previously-reserved sample aspect ratio indicators.

This edition includes the text approved 2005-03 and its Corrigendum 1 approved 2005-09.

Source

ITU-T Recommendation H.264 was approved on 1 March 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure. It includes modifications introduced by H.264 (2005) Cor.1 approved on 13 September 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2005

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

Foreword ..... **xiv**

0 Introduction **1**

0.1 Prologue *1*

0.2 Purpose *1*

0.3 Applications *1*

0.4 Publication and versions of this specification *1*

0.5 Profiles and levels *2*

0.6 Overview of the design characteristics *2*

0.6.1 Predictive coding 3

0.6.2 Coding of progressive and interlaced video 3

0.6.3 Picture partitioning into macroblocks and smaller partitions 3

0.6.4 Spatial redundancy reduction 3

0.7 How to read this specification *3*

1 Scope **4**

2 Normative references **4**

3 Definitions **4**

4 Abbreviations **12**

5 Conventions **12**

5.1 Arithmetic operators *13*

5.2 Logical operators *13*

5.3 Relational operators *13*

5.4 Bit-wise operators *13*

5.5 Assignment operators *14*

5.6 Range notation *14*

5.7 Mathematical functions *14*

5.8 Variables, syntax elements, and tables *15*

5.9 Text description of logical operations *16*

5.10 Processes *17*

6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships **17**

6.1 Bitstream formats *17*

6.2 Source, decoded, and output picture formats *17*

6.3 Spatial subdivision of pictures and slices *22*

6.4 Inverse scanning processes and derivation processes for neighbours *23*

6.4.1 Inverse macroblock scanning process 23

6.4.2 Inverse macroblock partition and sub-macroblock partition scanning process 24

6.4.2.1 Inverse macroblock partition scanning process 25

6.4.2.2 Inverse sub-macroblock partition scanning process 25

6.4.3 Inverse 4x4 luma block scanning process 26

6.4.4 Inverse 8x8 luma block scanning process 26

6.4.5 Derivation process of the availability for macroblock addresses 26

6.4.6 Derivation process for neighbouring macroblock addresses and their availability 27

6.4.7 Derivation process for neighbouring macroblock addresses and their availability in MBAFF frames 27

6.4.8 Derivation processes for neighbouring macroblocks, blocks, and partitions 28

6.4.8.1 Derivation process for neighbouring macroblocks 29

6.4.8.2 Derivation process for neighbouring 8x8 luma block 29

6.4.8.3 Derivation process for neighbouring 4x4 luma blocks 30

6.4.8.4 Derivation process for neighbouring 4x4 chroma blocks 30

6.4.8.5 Derivation process for neighbouring partitions 31

6.4.9 Derivation process for neighbouring locations 33

6.4.9.1 Specification for neighbouring locations in fields and non-MBAFF frames 33

6.4.9.2 Specification for neighbouring locations in MBAFF frames 34

7 Syntax and semantics **36**

7.1 Method of describing syntax in tabular form *36*

7.2 Specification of syntax functions, categories, and descriptors *37*

7.3 Syntax in tabular form *38*

7.3.1 NAL unit syntax 38

7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax 39

7.3.2.1 Sequence parameter set RBSP syntax 39

7.3.2.1.1 Scaling list syntax 40
7.3.2.1.2 Sequence parameter set extension RBSP syntax 40
7.3.2.2 Picture parameter set RBSP syntax 41
7.3.2.3 Supplemental enhancement information RBSP syntax 42
7.3.2.3.1 Supplemental enhancement information message syntax 42
7.3.2.4 Access unit delimiter RBSP syntax 43
7.3.2.5 End of sequence RBSP syntax 43
7.3.2.6 End of stream RBSP syntax 43
7.3.2.7 Filler data RBSP syntax 43
7.3.2.8 Slice layer without partitioning RBSP syntax 43
7.3.2.9 Slice data partition RBSP syntax 43
7.3.2.9.1 Slice data partition A RBSP syntax 43
7.3.2.9.2 Slice data partition B RBSP syntax 44
7.3.2.9.3 Slice data partition C RBSP syntax 44
7.3.2.10 RBSP slice trailing bits syntax 44
7.3.2.11 RBSP trailing bits syntax 44
7.3.3 Slice header syntax 45
7.3.3.1 Reference picture list reordering syntax 46
7.3.3.2 Prediction weight table syntax 47
7.3.3.3 Decoded reference picture marking syntax 48
7.3.4 Slice data syntax 49
7.3.5 Macroblock layer syntax 50
7.3.5.1 Macroblock prediction syntax 51
7.3.5.2 Sub-macroblock prediction syntax 52
7.3.5.3 Residual data syntax 53
7.3.5.3.1 Residual block CAVLC syntax 54
7.3.5.3.2 Residual block CABAC syntax 55
7.4 Semantics 56
7.4.1 NAL unit semantics 56
7.4.1.1 Encapsulation of an SODB within an RBSP (informative) 58
7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences 59
7.4.1.2.1 Order of sequence and picture parameter set RBSPs and their activation 59
7.4.1.2.2 Order of access units and association to coded video sequences 60
7.4.1.2.3 Order of NAL units and coded pictures and association to access units 60
7.4.1.2.4 Detection of the first VCL NAL unit of a primary coded picture 62
7.4.1.2.5 Order of VCL NAL units and association to coded pictures 63
7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics 63
7.4.2.1 Sequence parameter set RBSP semantics 63
7.4.2.1.1 Scaling list semantics 68
7.4.2.1.2 Sequence parameter set extension RBSP semantics 69
7.4.2.2 Picture parameter set RBSP semantics 70
7.4.2.3 Supplemental enhancement information RBSP semantics 73
7.4.2.3.1 Supplemental enhancement information message semantics 73
7.4.2.4 Access unit delimiter RBSP semantics 73
7.4.2.5 End of sequence RBSP semantics 73
7.4.2.6 End of stream RBSP semantics 73
7.4.2.7 Filler data RBSP semantics 73
7.4.2.8 Slice layer without partitioning RBSP semantics 73
7.4.2.9 Slice data partition RBSP semantics 74
7.4.2.9.1 Slice data partition A RBSP semantics 74
7.4.2.9.2 Slice data partition B RBSP semantics 74
7.4.2.9.3 Slice data partition C RBSP semantics 74
7.4.2.10 RBSP slice trailing bits semantics 74
7.4.2.11 RBSP trailing bits semantics 75
7.4.3 Slice header semantics 75
7.4.3.1 Reference picture list reordering semantics 80
7.4.3.2 Prediction weight table semantics 81
7.4.3.3 Decoded reference picture marking semantics 82
7.4.4 Slice data semantics 85
7.4.5 Macroblock layer semantics 85
7.4.5.1 Macroblock prediction semantics 92
7.4.5.2 Sub-macroblock prediction semantics 93
7.4.5.3 Residual data semantics 95

7.4.5.3.1 Residual block CAVLC semantics 96
7.4.5.3.2 Residual block CABAC semantics 96
8 Decoding process 96
8.1 NAL unit decoding process 97
8.2 Slice decoding process 98
8.2.1 Decoding process for picture order count 98
8.2.1.1 Decoding process for picture order count type 0 99
8.2.1.2 Decoding process for picture order count type 1 100
8.2.1.3 Decoding process for picture order count type 2 101
8.2.2 Decoding process for macroblock to slice group map 102
8.2.2.1 Specification for interleaved slice group map type 103
8.2.2.2 Specification for dispersed slice group map type 103
8.2.2.3 Specification for foreground with left-over slice group map type 104
8.2.2.4 Specification for box-out slice group map types 104
8.2.2.5 Specification for raster scan slice group map types 105
8.2.2.6 Specification for wipe slice group map types 105
8.2.2.7 Specification for explicit slice group map type 105
8.2.2.8 Specification for conversion of map unit to slice group map to macroblock to slice group map 105
8.2.3 Decoding process for slice data partitioning 105
8.2.4 Decoding process for reference picture lists construction 106
8.2.4.1 Decoding process for picture numbers 107
8.2.4.2 Initialisation process for reference picture lists 107
8.2.4.2.1 Initialisation process for the reference picture list for P and SP slices in frames 108
8.2.4.2.2 Initialisation process for the reference picture list for P and SP slices in fields 108
8.2.4.2.3 Initialisation process for reference picture lists for B slices in frames 109
8.2.4.2.4 Initialisation process for reference picture lists for B slices in fields 109
8.2.4.2.5 Initialisation process for reference picture lists in fields 110
8.2.4.3 Reordering process for reference picture lists 110
8.2.4.3.1 Reordering process of reference picture lists for short-term reference pictures 111
8.2.4.3.2 Reordering process of reference picture lists for long-term reference pictures 112
8.2.5 Decoded reference picture marking process 112
8.2.5.1 Sequence of operations for decoded reference picture marking process 113
8.2.5.2 Decoding process for gaps in frame_num 113
8.2.5.3 Sliding window decoded reference picture marking process 114
8.2.5.4 Adaptive memory control decoded reference picture marking process 114
8.2.5.4.1 Marking process of a short-term reference picture as “unused for reference” 114
8.2.5.4.2 Marking process of a long-term reference picture as “unused for reference” 115
8.2.5.4.3 Assignment process of a LongTermFrameIdx to a short-term reference picture 115
8.2.5.4.4 Decoding process for MaxLongTermFrameIdx 115
8.2.5.4.5 Marking process of all reference pictures as “unused for reference” and setting MaxLongTermFrameIdx to “no long-term frame indices” 115
8.2.5.4.6 Process for assigning a long-term frame index to the current picture 116
8.3 Intra prediction process 116
8.3.1 Intra_4x4 prediction process for luma samples 117
8.3.1.1 Derivation process for the Intra4x4PredMode 117
8.3.1.2 Intra_4x4 sample prediction 119
8.3.1.2.1 Specification of Intra_4x4_Vertical prediction mode 119
8.3.1.2.2 Specification of Intra_4x4_Horizontal prediction mode 119
8.3.1.2.3 Specification of Intra_4x4_DC prediction mode 120
8.3.1.2.4 Specification of Intra_4x4_Diagonal_Down_Left prediction mode 120
8.3.1.2.5 Specification of Intra_4x4_Diagonal_Down_Right prediction mode 120
8.3.1.2.6 Specification of Intra_4x4_Vertical_Right prediction mode 121
8.3.1.2.7 Specification of Intra_4x4_Horizontal_Down prediction mode 121
8.3.1.2.8 Specification of Intra_4x4_Vertical_Left prediction mode 122
8.3.1.2.9 Specification of Intra_4x4_Horizontal_Up prediction mode 122
8.3.2 Intra_8x8 prediction process for luma samples 122
8.3.2.1 Derivation process for Intra8x8PredMode 123
8.3.2.2 Intra_8x8 sample prediction 124
8.3.2.2.1 Reference sample filtering process for Intra_8x8 sample prediction 125
8.3.2.2.2 Specification of Intra_8x8_Vertical prediction mode 126
8.3.2.2.3 Specification of Intra_8x8_Horizontal prediction mode 126
8.3.2.2.4 Specification of Intra_8x8_DC prediction mode 127
8.3.2.2.5 Specification of Intra_8x8_Diagonal_Down_Left prediction mode 127

8.3.2.2.6 Specification of Intra_8x8_Diagonal_Down_Right prediction mode 127
8.3.2.2.7 Specification of Intra_8x8_Vertical_Right prediction mode 128
8.3.2.2.8 Specification of Intra_8x8_Horizontal_Down prediction mode 128
8.3.2.2.9 Specification of Intra_8x8_Vertical_Left prediction mode 129
8.3.2.2.10 Specification of Intra_8x8_Horizontal_Up prediction mode 129
8.3.3 Intra_16x16 prediction process for luma samples 129
8.3.3.1 Specification of Intra_16x16_Vertical prediction mode 130
8.3.3.2 Specification of Intra_16x16_Horizontal prediction mode 130
8.3.3.3 Specification of Intra_16x16_DC prediction mode 130
8.3.3.4 Specification of Intra_16x16_Plane prediction mode 131
8.3.4 Intra prediction process for chroma samples 131
8.3.4.1 Specification of Intra_Chroma_DC prediction mode 132
8.3.4.2 Specification of Intra_Chroma_Horizontal prediction mode 134
8.3.4.3 Specification of Intra_Chroma_Vertical prediction mode 134
8.3.4.4 Specification of Intra_Chroma_Plane prediction mode 134
8.3.5 Sample construction process for I_PCM macroblocks 134
8.4 Inter prediction process 135
8.4.1 Derivation process for motion vector components and reference indices 137
8.4.1.1 Derivation process for luma motion vectors for skipped macroblocks in P and SP slices 138
8.4.1.2 Derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 139
8.4.1.2.1 Derivation process for the co-located 4x4 sub-macroblock partitions 139
8.4.1.2.2 Derivation process for spatial direct luma motion vector and reference index prediction mode ... 142
8.4.1.2.3 Derivation process for temporal direct luma motion vector and reference index prediction mode 144
8.4.1.3 Derivation process for luma motion vector prediction 146
8.4.1.3.1 Derivation process for median luma motion vector prediction 147
8.4.1.3.2 Derivation process for motion data of neighbouring partitions 148
8.4.1.4 Derivation process for chroma motion vectors 149
8.4.2 Decoding process for Inter prediction samples 149
8.4.2.1 Reference picture selection process 150
8.4.2.2 Fractional sample interpolation process 151
8.4.2.2.1 Luma sample interpolation process 152
8.4.2.2.2 Chroma sample interpolation process 155
8.4.2.3 Weighted sample prediction process 156
8.4.2.3.1 Default weighted sample prediction process 156
8.4.2.3.2 Weighted sample prediction process 157
8.5 Transform coefficient decoding process and picture construction process prior to deblocking filter process. 159
8.5.1 Specification of transform decoding process for 4x4 luma residual blocks 160
8.5.2 Specification of transform decoding process for luma samples of Intra_16x16 macroblock prediction mode 160
8.5.3 Specification of transform decoding process for 8x8 luma residual blocks 161
8.5.4 Specification of transform decoding process for chroma samples 162
8.5.5 Inverse scanning process for transform coefficients 164
8.5.6 Inverse scanning process for 8x8 luma transform coefficients 164
8.5.7 Derivation process for the chroma quantisation parameters and scaling function 166
8.5.8 Scaling and transformation process for luma DC transform coefficients for Intra_16x16 macroblock type168
8.5.9 Scaling and transformation process for chroma DC transform coefficients 169
8.5.10 Scaling and transformation process for residual 4x4 blocks 171
8.5.11 Scaling and transformation process for residual 8x8 luma blocks 173
8.5.12 Picture construction process prior to deblocking filter process 176
8.5.13 Residual colour transform process 177
8.6 Decoding process for P macroblocks in SP slices or SI macroblocks 178
8.6.1 SP decoding process for non-switching pictures 178
8.6.1.1 Luma transform coefficient decoding process 178
8.6.1.2 Chroma transform coefficient decoding process 179
8.6.2 SP and SI slice decoding process for switching pictures 181
8.6.2.1 Luma transform coefficient decoding process 181
8.6.2.2 Chroma transform coefficient decoding process 181
8.7 Deblocking filter process 182
8.7.1 Filtering process for block edges 186
8.7.2 Filtering process for a set of samples across a horizontal or vertical block edge 188
8.7.2.1 Derivation process for the luma content dependent boundary filtering strength 188
8.7.2.2 Derivation process for the thresholds for each block edge 190
8.7.2.3 Filtering process for edges with bS less than 4 191

8.7.2.4 Filtering process for edges for bS equal to 4 ... 193
9 Parsing process ... 194
9.1 Parsing process for Exp-Golomb codes ... 194
9.1.1 Mapping process for signed Exp-Golomb codes ... 195
9.1.2 Mapping process for coded block pattern ... 196
9.2 CAVLC parsing process for transform coefficient levels ... 198
9.2.1 Parsing process for total number of transform coefficient levels and trailing ones ... 199
9.2.2 Parsing process for level information ... 201
9.2.2.1 Parsing process for level_prefix ... 202
9.2.3 Parsing process for run information ... 203
9.2.4 Combining level and run information ... 206
9.3 CABAC parsing process for slice data ... 206
9.3.1 Initialisation process ... 207
9.3.1.1 Initialisation process for context variables ... 208
9.3.1.2 Initialisation process for the arithmetic decoding engine ... 218
9.3.2 Binarization process ... 219
9.3.2.1 Unary (U) binarization process ... 221
9.3.2.2 Truncated unary (TU) binarization process ... 221
9.3.2.3 Concatenated unary/ k-th order Exp-Golomb (UEGk) binarization process ... 222
9.3.2.4 Fixed-length (FL) binarization process ... 222
9.3.2.5 Binarization process for macroblock type and sub-macroblock type ... 222
9.3.2.6 Binarization process for coded block pattern ... 225
9.3.2.7 Binarization process for mb_qp_delta ... 225
9.3.3 Decoding process flow ... 225
9.3.3.1 Derivation process for ctxIdx ... 226
9.3.3.1.1 Assignment process of ctxIdxInc using neighbouring syntax elements ... 228
9.3.3.1.1.1 Derivation process of ctxIdxInc for the syntax element mb_skip_flag ... 228
9.3.3.1.1.2 Derivation process of ctxIdxInc for the syntax element mb_field_decoding_flag ... 228
9.3.3.1.1.3 Derivation process of ctxIdxInc for the syntax element mb_type ... 229
9.3.3.1.1.4 Derivation process of ctxIdxInc for the syntax element coded_block_pattern ... 229
9.3.3.1.1.5 Derivation process of ctxIdxInc for the syntax element mb_qp_delta ... 230
9.3.3.1.1.6 Derivation process of ctxIdxInc for the syntax elements ref_idx_l0 and ref_idx_l1 ... 230
9.3.3.1.1.7 Derivation process of ctxIdxInc for the syntax elements mvd_l0 and mvd_l1 ... 231
9.3.3.1.1.8 Derivation process of ctxIdxInc for the syntax element intra_chroma_pred_mode ... 232
9.3.3.1.1.9 Derivation process of ctxIdxInc for the syntax element coded_block_flag ... 233
9.3.3.1.1.10 Derivation process of ctxIdxInc for the syntax element transform_size_8x8_flag ... 234
9.3.3.1.2 Assignment process of ctxIdxInc using prior decoded bin values ... 234
9.3.3.1.3 Assignment process of ctxIdxInc for syntax elements significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 ... 235
9.3.3.2 Arithmetic decoding process ... 237
9.3.3.2.1 Arithmetic decoding process for a binary decision ... 238
9.3.3.2.1.1 State transition process ... 239
9.3.3.2.2 Renormalization process in the arithmetic decoding engine ... 241
9.3.3.2.3 Bypass decoding process for binary decisions ... 242
9.3.3.2.4 Decoding process for binary decisions before termination ... 242
9.3.4 Arithmetic encoding process (informative) ... 243
9.3.4.1 Initialisation process for the arithmetic encoding engine (informative) ... 243
9.3.4.2 Encoding process for a binary decision (informative) ... 243
9.3.4.3 Renormalization process in the arithmetic encoding engine (informative) ... 244
9.3.4.4 Bypass encoding process for binary decisions (informative) ... 246
9.3.4.5 Encoding process for a binary decision before termination (informative) ... 246
9.3.4.6 Byte stuffing process (informative) ... 247
Annex A Profiles and levels ... 249
A.1 Requirements on video decoder capability ... 249
A.2 Profiles ... 249
A.2.1 Baseline profile ... 249
A.2.2 Main profile ... 250
A.2.3 Extended profile ... 250
A.2.4 High profile ... 250
A.2.5 High 10 profile ... 251
A.2.6 High 4:2:2 profile ... 251
A.2.7 High 4:4:4 profile ... 252

A.3 Levels *252*
A.3.1 Level limits common to the Baseline, Main, and Extended profiles 252
A.3.2 Level limits common to the High, High 10, High 4:2:2, and High 4:4:4 profiles 254
A.3.3 Profile-specific level limits 255
A.3.3.1 Baseline profile limits 256
A.3.3.2 Main, High, High 10, High 4:2:2, or High 4:4:4 profile limits 257
A.3.3.3 Extended Profile Limits 258
A.3.4 Effect of level limits on frame rate (informative) 259
Annex B Byte stream format 262
B.1 Byte stream NAL unit syntax and semantics *262*
B.1.1 Byte stream NAL unit syntax 262
B.1.2 Byte stream NAL unit semantics 262
B.2 Byte stream NAL unit decoding process *262*
B.3 Decoder byte-alignment recovery (informative) *263*
Annex C Hypothetical reference decoder 264
C.1 Operation of coded picture buffer (CPB) *266*
C.1.1 Timing of bitstream arrival 266
C.1.2 Timing of coded picture removal 267
C.2 Operation of the decoded picture buffer (DPB) *268*
C.2.1 Decoding of gaps in frame_num and storage of "non-existing" frames 268
C.2.2 Picture decoding and output 268
C.2.3 Removal of pictures from the DPB before possible insertion of the current picture 268
C.2.4 Current decoded picture marking and storage 269
C.2.4.1 Marking and storage of a reference decoded picture into the DPB 269
C.2.4.2 Storage of a non-reference picture into the DPB 269
C.3 Bitstream conformance *269*
C.4 Decoder conformance *271*
C.4.1 Operation of the output order DPB 272
C.4.2 Decoding of gaps in frame_num and storage of "non-existing" pictures 272
C.4.3 Picture decoding 272
C.4.4 Removal of pictures from the DPB before possible insertion of the current picture 272
C.4.5 Current decoded picture marking and storage 272
C.4.5.1 Storage and marking of a reference decoded picture into the DPB 272
C.4.5.2 Storage and marking of a non-reference decoded picture into the DPB 273
C.4.5.3 "Bumping" process 273
Annex D Supplemental enhancement information 275
D.1 SEI payload syntax *276*
D.1.1 Buffering period SEI message syntax 277
D.1.2 Picture timing SEI message syntax 277
D.1.3 Pan-scan rectangle SEI message syntax 278
D.1.4 Filler payload SEI message syntax 278
D.1.5 User data registered by ITU-T Recommendation T.35 SEI message syntax 279
D.1.6 User data unregistered SEI message syntax 279
D.1.7 Recovery point SEI message syntax 279
D.1.8 Decoded reference picture marking repetition SEI message syntax 279
D.1.9 Spare picture SEI message syntax 280
D.1.10 Scene information SEI message syntax 280
D.1.11 Sub-sequence information SEI message syntax 281
D.1.12 Sub-sequence layer characteristics SEI message syntax 281
D.1.13 Sub-sequence characteristics SEI message syntax 281
D.1.14 Full-frame freeze SEI message syntax 282
D.1.15 Full-frame freeze release SEI message syntax 282
D.1.16 Full-frame snapshot SEI message syntax 282
D.1.17 Progressive refinement segment start SEI message syntax 282
D.1.18 Progressive refinement segment end SEI message syntax 282
D.1.19 Motion-constrained slice group set SEI message syntax 282
D.1.20 Film grain characteristics SEI message syntax 283
D.1.21 Deblocking filter display preference SEI message syntax 283
D.1.22 Stereo video information SEI message syntax 284
D.1.23 Reserved SEI message syntax 284
D.2 SEI payload semantics *284*
D.2.1 Buffering period SEI message semantics 284

D.2.2 Picture timing SEI message semantics 285
D.2.3 Pan-scan rectangle SEI message semantics 288
D.2.4 Filler payload SEI message semantics 290
D.2.5 User data registered by ITU-T Recommendation T.35 SEI message semantics 290
D.2.6 User data unregistered SEI message semantics 290
D.2.7 Recovery point SEI message semantics 290
D.2.8 Decoded reference picture marking repetition SEI message semantics 291
D.2.9 Spare picture SEI message semantics 292
D.2.10 Scene information SEI message semantics 293
D.2.11 Sub-sequence information SEI message semantics 295
D.2.12 Sub-sequence layer characteristics SEI message semantics 296
D.2.13 Sub-sequence characteristics SEI message semantics 297
D.2.14 Full-frame freeze SEI message semantics 299
D.2.15 Full-frame freeze release SEI message semantics 299
D.2.16 Full-frame snapshot SEI message semantics 299
D.2.17 Progressive refinement segment start SEI message semantics 299
D.2.18 Progressive refinement segment end SEI message semantics 300
D.2.19 Motion-constrained slice group set SEI message semantics 300
D.2.20 Film grain characteristics SEI message semantics 301
D.2.21 Deblocking filter display preference SEI message semantics 306
D.2.22 Stereo video information SEI message semantics 308
D.2.23 Reserved SEI message semantics 309
Annex E Video usability information 310
E.1 VUI syntax 311
E.1.1 VUI parameters syntax 311
E.1.2 HRD parameters syntax 312
E.2 VUI semantics 312
E.2.1 VUI parameters semantics 312
E.2.2 HRD parameters semantics 323

LIST OF FIGURES

Figure 6-1 – Nominal vertical and horizontal locations of 4:2:0 luma and chroma samples in a frame 19
Figure 6-2 – Nominal vertical and horizontal sampling locations of 4:2:0 samples in top and bottom fields. 20
Figure 6-3 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a frame 20
Figure 6-4 – Nominal vertical and horizontal sampling locations of 4:2:2 samples top and bottom fields 21
Figure 6-5 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a frame 21
Figure 6-6 – Nominal vertical and horizontal sampling locations of 4:4:4 samples top and bottom fields 22
Figure 6-7 – A picture with 11 by 9 macroblocks that is partitioned into two slices 23
Figure 6-8 – Partitioning of the decoded frame into macroblock pairs 23
Figure 6-9 – Macroblock partitions, sub-macroblock partitions, macroblock partition scans, and sub-macroblock partition scans 25
Figure 6-10 – Scan for 4x4 luma blocks 26
Figure 6-11 – Scan for 8x8 luma blocks 26
Figure 6-12 – Neighbouring macroblocks for a given macroblock 27
Figure 6-13 – Neighbouring macroblocks for a given macroblock in MBAFF frames 28
Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative) 29
Figure 7-1 – Structure of an access unit not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive. 62
Figure 8-1 – Intra_4x4 prediction mode directions (informative) 118
Figure 8-2 –Example for temporal direct-mode motion vector inference (informative) 146
Figure 8-3 – Directional segmentation prediction (informative) 147

Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation 153
Figure 8-5 – Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D 155
Figure 8-6 – Assignment of the indices of dcY to luma4x4BlkIdx 161
Figure 8-7 – Assignment of the indices of dcC to chroma4x4BlkIdx: (a) chroma_format_idc equal to 1, (b) chroma_format_idc equal to 2, (c) chroma_format_idc equal to 3 163
Figure 8-8 – 4x4 block scans. (a) Zig-zag scan. (b) Field scan (informative) 164
Figure 8-9 – 8x8 block scans. (a) 8x8 zig-zag scan. (b) 8x8 field scan (informative) 165
Figure 8-10 – Boundaries in a macroblock to be filtered 183
Figure 8-11 – Convention for describing samples across a 4x4 block horizontal or vertical boundary 187
Figure 9-1 – Illustration of CABAC parsing process for a syntax element SE (informative) 207
Figure 9-2 – Overview of the arithmetic decoding process for a single bin (informative) 238
Figure 9-3 – Flowchart for decoding a decision 239
Figure 9-4 – Flowchart of renormalization 241
Figure 9-5 – Flowchart of bypass decoding process 242
Figure 9-6 – Flowchart of decoding a decision before termination 243
Figure 9-7 – Flowchart for encoding a decision 244
Figure 9-8 – Flowchart of renormalization in the encoder 245
Figure 9-9 – Flowchart of PutBit(B) 245
Figure 9-10 – Flowchart of encoding bypass 246
Figure 9-11 – Flowchart of encoding a decision before termination 247
Figure 9-12 – Flowchart of flushing at termination 247
Figure C-1 – Structure of byte streams and NAL unit streams for HRD conformance checks 264
Figure C-2 – HRD buffer model 265
Figure E-1 – Location of chroma samples for top and bottom fields as a function of chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field 320

LIST OF TABLES

Table 6-1 –SubWidthC, and SubHeightC values derived from chroma_format_idc 18
Table 6-2 – Specification of input and output assignments for subclauses 6.4.8.1 to 6.4.8.5 29
Table 6-3 – Specification of mbAddrN 33
Table 6-4 - Specification of mbAddrN and yM 35
Table 7-1 – NAL unit type codes 57
Table 7-2 – Assignment of mnemonic names to scaling list indices and specification of fall-back rule 65
Table 7-3 – Specification of default scaling lists Default_4x4_Intra and Default_4x4_Inter 66
Table 7-4 – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter 66
Table 7-5 – Meaning of primary_pic_type 73
Table 7-6 – Name association to slice_type 75
Table 7-7 – reordering_of_pic_nums_idc operations for reordering of reference picture lists 81
Table 7-8 – Interpretation of adaptive_ref_pic_marking_mode_flag 83

Table 7-9 – Memory management control operation (memory_management_control_operation) values 84
Table 7-10 – Allowed collective macroblock types for slice_type 86
Table 7-11 – Macroblock types for I slices 87
Table 7-12 – Macroblock type with value 0 for SI slices 88
Table 7-13 – Macroblock type values 0 to 4 for P and SP slices 89
Table 7-14 – Macroblock type values 0 to 22 for B slices 90
Table 7-15 – Specification of CodedBlockPatternChroma values 92
Table 7-16 – Relationship between intra_chroma_pred_mode and spatial prediction modes 92
Table 7-17 – Sub-macroblock types in P macroblocks 93
Table 7-18 – Sub-macroblock types in B macroblocks 94
Table 8-1 – Refined slice group map type 102
Table 8-2 – Specification of Intra4x4PredMode[luma4x4BlkIdx] and associated names 117
Table 8-3 – Specification of Intra8x8PredMode[luma8x8BlkIdx] and associated names 123
Table 8-4 – Specification of Intra16x16PredMode and associated names 130
Table 8-5 – Specification of Intra chroma prediction modes and associated names 132
Table 8-6 – Specification of the variable colPic 140
Table 8-7 – Specification of PicCodingStruct(X) 140
Table 8-8 – Specification of mbAddrCol, yM, and vertMvScale 141
Table 8-9 – Assignment of prediction utilization flags 143
Table 8-10 – Derivation of the vertical component of the chroma vector in field coding mode 149
Table 8-11 – Differential full-sample luma locations 153
Table 8-12 – Assignment of the luma prediction sample predPartLX$_L$[x_L, y_L] 155
Table 8-13 – Specification of mapping of idx to c_{ij} for zig-zag and field scan 164
Table 8-14 – Specification of mapping of idx to c_{ij} for 8x8 zig-zag and 8x8 field scan 166
Table 8-15 – Specification of QP_C as a function of qP_I 167
Table 8-16 – Derivation of offset dependent threshold variables α' and β' from indexA and indexB 191
Table 8-17 – Value of variable t'_{C0} as a function of indexA and bS 192
Table 9-1 – Bit strings with “prefix” and “suffix” bits and assignment to codeNum ranges (informative) 194
Table 9-2 – Exp-Golomb bit strings and codeNum in explicit form and used as ue(v) (informative) 195
Table 9-3 – Assignment of syntax element to codeNum for signed Exp-Golomb coded syntax elements se(v) 196
Table 9-4 – Assignment of codeNum to values of coded_block_pattern for macroblock prediction modes 196
Table 9-5 – coeff_token mapping to TotalCoeff(coeff_token) and TrailingOnes(coeff_token) 200
Table 9-6 – Codeword table for level_prefix (informative) 203
Table 9-7 – total_zeros tables for 4x4 blocks with TotalCoeff(coeff_token) 1 to 7 204
Table 9-8 – total_zeros tables for 4x4 blocks with TotalCoeff(coeff_token) 8 to 15 205
Table 9-9 – total_zeros tables for chroma DC 2x2 and 2x4 blocks 205
Table 9-10 – Tables for run_before 206
Table 9-11 – Association of ctxIdx and syntax elements for each slice type in the initialisation process 208
Table 9-12 – Values of variables m and n for ctxIdx from 0 to 10 209
Table 9-13 – Values of variables m and n for ctxIdx from 11 to 23 210

Table 9-14 – Values of variables m and n for ctxIdx from 24 to 39 210
Table 9-15 – Values of variables m and n for ctxIdx from 40 to 53 210
Table 9-16 – Values of variables m and n for ctxIdx from 54 to 59, and 399 to 401 211
Table 9-17 – Values of variables m and n for ctxIdx from 60 to 69 211
Table 9-18 – Values of variables m and n for ctxIdx from 70 to 104 212
Table 9-19 – Values of variables m and n for ctxIdx from 105 to 165 213
Table 9-20 – Values of variables m and n for ctxIdx from 166 to 226 214
Table 9-21 – Values of variables m and n for ctxIdx from 227 to 275 215
Table 9-22 – Values of variables m and n for ctxIdx from 277 to 337 216
Table 9-23 – Values of variables m and n for ctxIdx from 338 to 398 217
Table 9-24 – Values of variables m and n for ctxIdx from 402 to 459 218
Table 9-25 – Syntax elements and associated types of binarization, maxBinIdxCtx, and ctxIdxOffset 220
Table 9-26 – Bin string of the unary binarization (informative) 221
Table 9-27 – Binarization for macroblock types in I slices 223
Table 9-28 – Binarization for macroblock types in P, SP, and B slices 224
Table 9-29 – Binarization for sub-macroblock types in P, SP, and B slices 225
Table 9-30 – Assignment of ctxIdxInc to binIdx for all ctxIdxOffset values except those related to the syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 227
Table 9-31 – Assignment of ctxIdxBlockCatOffset to ctxBlockCat for syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 228
Table 9-32 – Specification of ctxIdxInc for specific values of ctxIdxOffset and binIdx 235
Table 9-33 – Specification of ctxBlockCat for the different blocks 235
Table 9-34 – Mapping of scanning position to ctxIdxInc for ctxBlockCat == 5 236
Table 9-35 – Specification of rangeTabLPS depending on pStateIdx and qCodIRangeIdx 240
Table 9-36 – State transition table 241
Table A-1 – Level limits 254
Table A-2 – Specification of cpbBrVclFactor and cpbBrNalFactor 256
Table A-3 – Baseline profile level limits 257
Table A-4 – Main, High, High 10, High 4:2:2, or High 4:4:4 profile level limits 257
Table A-5 – Extended profile level limits 258
Table A-6 – Maximum frame rates (frames per second) for some example frame sizes 259
Table D-1 – Interpretation of pic_struct 286
Table D-2 – Mapping of ct_type to source picture scan 287
Table D-3 – Definition of counting_type values 287
Table D-4 – scene_transition_type values. 294
Table D-5 – model_id values 301
Table D-6 – blending_mode_id values 302
Table E-1 – Meaning of sample aspect ratio indicator 313
Table E-2 – Meaning of video_format 314
Table E-3 – Colour primaries 315

Table E-4 – Transfer characteristics 316
Table E-5 – Matrix coefficients 319
Table E-6 – Divisor for computation of $\Delta t_{fi,dpb}(n)$ 321

Foreword

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardising telecommunications on a world-wide basis. The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups that, in turn, produce Recommendations on these topics. The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1. In some areas of information technology that fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialised system for world-wide standardisation. National Bodies that are members of ISO and IEC participate in the development of International Standards through technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

This Recommendation | International Standard was prepared jointly by ITU-T SG 16 Q.6, also known as VCEG (Video Coding Experts Group), and by ISO/IEC JTC 1/SC 29/WG 11, also known as MPEG (Moving Picture Experts Group). VCEG was formed in 1997 to maintain prior ITU-T video coding standards and develop new video coding standard(s) appropriate for a wide range of conversational and non-conversational services. MPEG was formed in 1988 to establish standards for coding of moving pictures and associated audio for various applications such as digital storage media, distribution, and communication.

In this Recommendation | International Standard Annexes A through E contain normative requirements and are an integral part of this Recommendation | International Standard.

ITU-T Recommendation H.264

Advanced video coding for generic audiovisual services

0 Introduction

This clause does not form an integral part of this Recommendation | International Standard.

0.1 Prologue

This subclause does not form an integral part of this Recommendation | International Standard.

As the costs for both processing power and memory have reduced, network support for coded video data has diversified, and advances in video coding technology have progressed, the need has arisen for an industry standard for compressed video representation with substantially increased coding efficiency and enhanced robustness to network environments. Toward these ends the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) formed a Joint Video Team (JVT) in 2001 for development of a new Recommendation | International Standard.

0.2 Purpose

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard was developed in response to the growing need for higher compression of moving pictures for various applications such as videoconferencing, digital storage media, television broadcasting, internet streaming, and communication. It is also designed to enable the use of the coded video representation in a flexible manner for a wide variety of network environments. The use of this Recommendation | International Standard allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels.

0.3 Applications

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard is designed to cover a broad range of applications for video content including but not limited to the following:

CATV	Cable TV on optical networks, copper, etc.
DBS	Direct broadcast satellite video services
DSL	Digital subscriber line video services
DTTB	Digital terrestrial television broadcasting
ISM	Interactive storage media (optical disks, etc.)
MMM	Multimedia mailing
MSPN	Multimedia services over packet networks
RTC	Real-time conversational services (videoconferencing, videophone, etc.)
RVS	Remote video surveillance
SSM	Serial storage media (digital VTR, etc.)

0.4 Publication and versions of this specification

This subclause does not form an integral part of this Recommendation | International Standard.

This specification has been jointly developed by ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group. It is published as technically-aligned twin text in both organizations ITU-T and ISO/IEC.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 1 refers to the first (2003) approved version of this Recommendation | International Standard.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 2 refers to the integrated text containing the corrections specified in the first technical corrigendum.

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 3 refers to the integrated text containing both the first technical corrigendum (2004) and the first amendment, which is referred to as the "Fidelity range extensions".

ITU-T Rec. H.264 | ISO/IEC 14496-10 version 4 (the current specification) refers to the integrated text containing the first technical corrigendum (2004), the first amendment (the "Fidelity range extensions"), and an additional technical corrigendum (2005). In the ITU-T, the next published version after version 2 was version 4 (due to the completion of the drafting work for version 4 prior to the approval opportunity for a final version 3 text).

0.5 Profiles and levels

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard is designed to be generic in the sense that it serves a wide range of applications, bit rates, resolutions, qualities, and services. Applications should cover, among other things, digital storage media, television broadcasting and real-time communications. In the course of creating this Specification, various requirements from typical applications have been considered, necessary algorithmic elements have been developed, and these have been integrated into a single syntax. Hence, this Specification will facilitate video data interchange among different applications.

Considering the practicality of implementing the full syntax of this Specification, however, a limited number of subsets of the syntax are also stipulated by means of "profiles" and "levels". These and other related terms are formally defined in clause 3.

A "profile" is a subset of the entire bitstream syntax that is specified by this Recommendation | International Standard. Within the bounds imposed by the syntax of a given profile it is still possible to require a very large variation in the performance of encoders and decoders depending upon the values taken by syntax elements in the bitstream such as the specified size of the decoded pictures. In many applications, it is currently neither practical nor economic to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile.

In order to deal with this problem, "levels" are specified within each profile. A level is a specified set of constraints imposed on values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively they may take the form of constraints on arithmetic combinations of values (e.g. picture width multiplied by picture height multiplied by number of pictures decoded per second).

Coded video content conforming to this Recommendation | International Standard uses a common syntax. In order to achieve a subset of the complete syntax, flags, parameters, and other syntax elements are included in the bitstream that signal the presence or absence of syntactic elements that occur later in the bitstream.

0.6 Overview of the design characteristics

This subclause does not form an integral part of this Recommendation | International Standard.

The coded representation specified in the syntax is designed to enable a high compression capability for a desired image quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 profile and the I_PCM mode of operation in all profiles, the algorithm is typically not lossless, as the exact source sample values are typically not preserved through the encoding and decoding processes. A number of techniques may be used to achieve highly efficient compression. Encoding algorithms (not specified in this Recommendation | International Standard) may select between inter and intra coding for block-shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial statistical dependencies in the source signal for a single picture. Motion vectors and intra prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation inside the transform block before it is quantised, producing an irreversible process that typically discards less important visual information while forming a close approximation to the source samples. Finally, the motion vectors or intra prediction modes are combined with the quantised transform coefficient information and encoded using either variable length codes or arithmetic coding.

0.6.1 Predictive coding

This subclause does not form an integral part of this Recommendation | International Standard.

Because of the conflicting requirements of random access and highly efficient compression, two main coding types are specified. Intra coding is done without reference to other pictures. Intra coding may provide access points to the coded sequence where decoding can begin and continue correctly, but typically also shows only moderate compression efficiency. Inter coding (predictive or bi-predictive) is more efficient using inter prediction of each block of sample values from some previously decoded picture selected by the encoder. In contrast to some other video coding standards, pictures coded using bi-predictive inter prediction may also be used as references for inter coding of other pictures.

The application of the three coding types to pictures in a sequence is flexible, and the order of the decoding process is generally not the same as the order of the source picture capture process in the encoder or the output order from the decoder for display. The choice is left to the encoder and will depend on the requirements of the application. The decoding order is specified such that the decoding of pictures that use inter-picture prediction follows later in decoding order than other pictures that are referenced in the decoding process.

0.6.2 Coding of progressive and interlaced video

This subclause does not form an integral part of this Recommendation | International Standard.

This Recommendation | International Standard specifies a syntax and decoding process for video that originated in either progressive-scan or interlaced-scan form, which may be mixed together in the same sequence. The two fields of an interlaced frame are separated in capture time while the two fields of a progressive frame share the same capture time. Each field may be coded separately or the two fields may be coded together as a frame. Progressive frames are typically coded as a frame. For interlaced video, the encoder can choose between frame coding and field coding. Frame coding or field coding can be adaptively selected on a picture-by-picture basis and also on a more localized basis within a coded frame. Frame coding is typically preferred when the video scene contains significant detail with limited motion. Field coding typically works better when there is fast picture-to-picture motion.

0.6.3 Picture partitioning into macroblocks and smaller partitions

This subclause does not form an integral part of this Recommendation | International Standard.

As in previous video coding Recommendations and International Standards, a macroblock, consisting of a 16x16 block of luma samples and two corresponding blocks of chroma samples, is used as the basic processing unit of the video decoding process.

A macroblock can be further partitioned for inter prediction. The selection of the size of inter prediction partitions is a result of a trade-off between the coding gain provided by using motion compensation with smaller blocks and the quantity of data needed to represent the data for motion compensation. In this Recommendation | International Standard the inter prediction process can form segmentations for motion representation as small as 4x4 luma samples in size, using motion vector accuracy of one-quarter of the luma sample grid spacing displacement. The process for inter prediction of a sample block can also involve the selection of the picture to be used as the reference picture from a number of stored previously-decoded pictures. Motion vectors are encoded differentially with respect to predicted values formed from nearby encoded motion vectors.

Typically, the encoder calculates appropriate motion vectors and other data elements represented in the video data stream. This motion estimation process in the encoder and the selection of whether to use inter prediction for the representation of each region of the video content is not specified in this Recommendation | International Standard.

0.6.4 Spatial redundancy reduction

This subclause does not form an integral part of this Recommendation | International Standard.

Both source pictures and prediction residuals have high spatial redundancy. This Recommendation | International Standard is based on the use of a block-based transform method for spatial redundancy removal. After inter prediction from previously-decoded samples in other pictures or spatial-based prediction from previously-decoded samples within the current picture, the resulting prediction residual is split into 4x4 blocks. These are converted into the transform domain where they are quantised. After quantisation many of the transform coefficients are zero or have low amplitude and can thus be represented with a small amount of encoded data. The processes of transformation and quantisation in the encoder are not specified in this Recommendation | International Standard.

0.7 How to read this specification

This subclause does not form an integral part of this Recommendation | International Standard.

It is suggested that the reader starts with clause 1 (Scope) and moves on to clause 3 (Definitions). Clause 6 should be read for the geometrical relationship of the source, input, and output of the decoder. Clause 7 (Syntax and semantics) specifies the order to parse syntax elements from the bitstream. See subclauses 7.1-7.3 for syntactical order and see subclause 7.4 for semantics; i.e., the scope, restrictions, and conditions that are imposed on the syntax elements. The actual parsing for most syntax elements is specified in clause 9 (Parsing process). Finally, clause 8 (Decoding process) specifies how the syntax elements are mapped into decoded samples. Throughout reading this specification, the reader should refer to clauses 2 (Normative references), 4 (Abbreviations), and 5 (Conventions) as needed. Annexes A through E also form an integral part of this Recommendation | International Standard.

Annex A specifies seven profiles (Baseline, Main, Extended, High, High 10, High 4:2:2 and High 4:4:4), each being tailored to certain application domains, and defines the so-called levels of the profiles. Annex B specifies syntax and semantics of a byte stream format for delivery of coded video as an ordered stream of bytes. Annex C specifies the hypothetical reference decoder and its use to check bitstream and decoder conformance. Annex D specifies syntax and semantics for supplemental enhancement information message payloads. Finally, Annex E specifies syntax and semantics of the video usability information parameters of the sequence parameter set.

Throughout this specification, statements appearing with the preamble "NOTE -" are informative and are not an integral part of this Recommendation | International Standard.

1 Scope

This document specifies ITU-T Recommendation H.264 | ISO/IEC International Standard ISO/IEC 14496-10 video coding.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

- ITU-T Recommendation T.35 (2000), *Procedure for the allocation of ITU-T defined codes for non-standard facilities*.
- ISO/IEC 11578:1996, Annex A, *Universal Unique Identifier*.
- ISO/CIE 10527:1991, *Colorimetric Observers*.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 **access unit**: A set of *NAL units* always containing exactly one *primary coded picture*. In addition to the *primary coded picture*, an access unit may also contain one or more *redundant coded pictures* or other *NAL units* not containing *slices* or *slice data partitions* of a *coded picture*. The decoding of an access unit always results in a *decoded picture*.

3.2 **AC transform coefficient**: Any *transform coefficient* for which the *frequency index* in one or both dimensions is non-zero.

3.3 **adaptive binary arithmetic decoding process**: An entropy *decoding process* that derives the values of *bins* from a *bitstream* produced by an *adaptive binary arithmetic encoding process*.

3.4 **adaptive binary arithmetic encoding process**: An entropy *encoding process*, not normatively specified in this Recommendation | International Standard, that codes a sequence of *bins* and produces a *bitstream* that can be decoded using the *adaptive binary arithmetic decoding process*.

3.5 **alpha blending**: A process not specified by this Recommendation | International Standard, in which an *auxiliary coded picture* is used in combination with a *primary coded picture* and with other data not specified by this Recommendation | International Standard in the *display process*. In an alpha blending process, the samples of an *auxiliary coded picture* are interpreted as indications of the degree of opacity (or, equivalently, the degrees of transparency) associated with the corresponding *luma* samples of the *primary coded picture*.

3.6 **arbitrary slice order**: A *decoding order* of *slices* in which the *macroblock address* of the first *macroblock* of some *slice* of a *picture* may be less than the *macroblock address* of the first *macroblock* of some other preceding *slice* of the same *coded picture*.

3.7 **auxiliary coded picture**: A *picture* that supplements the *primary coded picture* that may be used in combination with other data not specified by this Recommendation | International Standard in the *display process*. An auxiliary coded picture has the same syntactic and semantic restrictions as a monochrome *redundant coded picture*. An auxiliary coded picture must contain the same number of *macroblocks* as the *primary coded picture*. Auxiliary coded pictures have no normative effect on the *decoding process*. See also *primary coded picture* and *redundant coded picture*.

3.8 **B slice**: A *slice* that may be decoded using *intra prediction* from decoded samples within the same *slice* or *inter prediction* from previously-decoded *reference pictures*, using at most two *motion vectors* and *reference indices* to *predict* the sample values of each *block*.

3.9 **bin**: One bit of a *bin string*.

3.10 **binarization**: A set of *bin strings* for all possible values of a *syntax element*.

3.11 **binarization process**: A unique mapping process of all possible values of a *syntax element* onto a set of *bin strings*.

3.12 **bin string**: A string of *bins*. A bin string is an intermediate binary representation of values of *syntax elements* from the *binarization* of the *syntax element*.

3.13 **bi-predictive slice:** See *B slice*.

3.14 **bitstream**: A sequence of bits that forms the representation of *coded pictures* and associated data forming one or more *coded video sequences*. Bitstream is a collective term used to refer either to a *NAL unit stream* or a *byte stream*.

3.15 **block**: An MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients*.

3.16 **bottom field**: One of two *fields* that comprise a *frame*. Each row of a *bottom field* is spatially located immediately below a corresponding row of a *top field*.

3.17 **bottom macroblock (of a macroblock pair)**: The *macroblock* within a *macroblock pair* that contains the samples in the bottom row of samples for the *macroblock pair*. For a *field macroblock pair*, the bottom macroblock represents the samples from the region of the *bottom field* of the *frame* that lie within the spatial region of the *macroblock pair*. For a *frame macroblock pair*, the bottom macroblock represents the samples of the *frame* that lie within the bottom half of the spatial region of the *macroblock pair*.

3.18 **broken link**: A location in a *bitstream* at which it is indicated that some subsequent *pictures* in *decoding order* may contain serious visual artefacts due to unspecified operations performed in the generation of the *bitstream*.

3.19 **byte**: A sequence of 8 bits, written and read with the most significant bit on the left and the least significant bit on the right. When represented in a sequence of data bits, the most significant bit of a byte is first.

3.20 **byte-aligned**: A position in a *bitstream* is byte-aligned when the position is an integer multiple of 8 bits from the position of the first bit in the *bitstream*. A bit or *byte* or *syntax element* is said to be byte-aligned when the position at which it appears in a *bitstream* is byte-aligned.

3.21 **byte stream**: An encapsulation of a *NAL unit stream* containing *start code prefixes* and *NAL units* as specified in Annex B.

3.22 **can**: A term used to refer to behaviour that is allowed, but not necessarily required.

3.23 **category**: A number associated with each *syntax element*. The category is used to specify the allocation of *syntax elements* to *NAL units* for *slice data partitioning*. It may also be used in a manner determined by the application to refer to classes of *syntax elements* in a manner not specified in this Recommendation | International Standard.

3.24 **chroma**: An adjective specifying that a sample array or single sample is representing one of the two colour difference signals related to the primary colours. The symbols used for a chroma array or sample are Cb and Cr.

NOTE – The term chroma is used rather than the term chrominance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term chrominance.

3.25 **coded field**: A *coded representation* of a *field*.

3.26 **coded frame**: A *coded representation* of a *frame*.

3.27 **coded picture**: A *coded representation* of a *picture*. A coded picture may be either a *coded field* or a *coded frame*. Coded picture is a collective term referring to a *primary coded picture* or a *redundant coded picture*, but not to both together.

3.28 **coded picture buffer (CPB)**: A first-in first-out buffer containing *access units* in *decoding order* specified in the *hypothetical reference decoder* in Annex C.

3.29 **coded representation**: A data element as represented in its coded form.

3.30 **coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*.

3.31 **component**: An array or single sample from one of the three arrays (*luma* and two *chroma*) that make up a *field* or *frame*.

3.32 **complementary field pair:** A collective term for a *complementary reference field pair* or a *complementary non-reference field pair*.

3.33 **complementary non-reference field pair**: Two *non-reference fields* that are in consecutive *access units* in *decoding order* as two *coded fields* of opposite parity where the first *field* is not already a paired *field*.

3.34 **complementary reference field pair**: Two *reference fields* that are in consecutive *access units* in *decoding order* as two *coded fields* and share the same value of the frame_num *syntax element*, where the second *field* in *decoding order* is not an *IDR picture* and does not include a memory_management_control_operation *syntax element* equal to 5.

3.35 **context variable**: A variable specified for the *adaptive binary arithmetic decoding process* of a *bin* by an equation containing recently decoded *bins*.

3.36 **DC transform coefficient**: A *transform coefficient* for which the *frequency index* is zero in all dimensions.

3.37 **decoded picture**: A *decoded picture* is derived by decoding a *coded picture*. A *decoded picture* is either a decoded *frame*, or a decoded *field*. A decoded *field* is either a decoded *top field* or a decoded *bottom field*.

3.38 **decoded picture buffer (DPB)**: A buffer holding *decoded pictures* for reference, output reordering, or output delay specified for the *hypothetical reference decoder* in Annex C.

3.39 **decoder**: An embodiment of a *decoding process*.

3.40 **decoding order**: The order in which *syntax elements* are processed by the *decoding process*.

3.41 **decoding process**: The process specified in this Recommendation | International Standard that reads a *bitstream* and derives *decoded pictures* from it.

3.42 **direct prediction**: An *inter prediction* for a *block* for which no *motion vector* is decoded. Two direct *prediction* modes are specified that are referred to as spatial direct *prediction* and temporal *prediction* mode.

3.43 **display process**: A process not specified in this Recommendation | International Standard having, as its input, the cropped decoded *pictures* that are the output of the *decoding process*.

3.44 **decoder under test (DUT)**: A *decoder* that is tested for conformance to this Recommendation | International Standard by operating the *hypothetical stream scheduler* to deliver a conforming *bitstream* to the *decoder* and to the *hypothetical reference decoder* and comparing the values and timing of the output of the two *decoders*.

3.45 **emulation prevention byte**: A *byte* equal to 0x03 that may be present within a *NAL unit*. The presence of emulation prevention bytes ensures that no sequence of consecutive *byte-aligned bytes* in the *NAL unit* contains a *start code prefix*.

3.46 **encoder**: An embodiment of an *encoding process*.

3.47 **encoding process**: A process, not specified in this Recommendation | International Standard, that produces a *bitstream* conforming to this Recommendation | International Standard.

3.48 **field**: An assembly of alternate rows of a *frame*. A *frame* is composed of two *fields*, a *top field* and a *bottom field*.

3.49 **field macroblock**: A macroblock containing samples from a single *field*. All *macroblocks* of a *coded field* are field macroblocks. When *macroblock-adaptive frame/field decoding* is in use, some *macroblocks* of a *coded frame* may be field macroblocks.

3.50 **field macroblock pair**: A *macroblock pair* decoded as two *field macroblocks*.

3.51 **field scan**: A specific sequential ordering of *transform coefficients* that differs from the *zig-zag scan* by scanning columns more rapidly than rows. Field scan is used for *transform coefficients* in *field macroblocks*.

3.52 **flag**: A variable that can take one of the two possible values 0 and 1.

3.53 **frame**: A *frame* contains an array of *luma* samples and two corresponding arrays of *chroma* samples. A *frame* consists of two *fields*, a *top field* and a *bottom field*.

3.54 **frame macroblock**: A *macroblock* representing samples from the two *fields* of a *coded frame*. When *macroblock-adaptive frame/field decoding* is not in use, all macroblocks of a *coded frame* are frame macroblocks. When *macroblock-adaptive frame/field decoding* is in use, some macroblocks of a *coded frame* may be frame macroblocks.

3.55 **frame macroblock pair**: A *macroblock pair* decoded as two *frame macroblocks*.

3.56 **frequency index**: A one-dimensional or two-dimensional index associated with a *transform coefficient* prior to an *inverse transform* part of the *decoding process*.

3.57 **hypothetical reference decoder (HRD)**: A hypothetical *decoder* model that specifies constraints on the variability of conforming *NAL unit streams* or conforming *byte streams* that an encoding process may produce.

3.58 **hypothetical stream scheduler (HSS)**: A hypothetical delivery mechanism for the timing and data flow of the input of a *bitstream* into the *hypothetical reference decoder*. The HSS is used for checking the conformance of a *bitstream* or a *decoder*.

3.59 **I slice**: A *slice* that is not an *SI slice* that is decoded using *prediction* only from decoded samples within the same *slice*.

3.60 **informative**: A term used to refer to content provided in this Recommendation | International Standard that is not an integral part of this Recommendation | International Standard. Informative content does not establish any mandatory requirements for conformance to this Recommendation | International Standard.

3.61 **instantaneous decoding refresh (IDR) access unit**: An *access unit* in which the *primary coded picture* is an *IDR picture*.

3.62 **instantaneous decoding refresh (IDR) picture**: A *coded picture* in which all *slices* are *I* or *SI slices* that causes the *decoding process* to mark all *reference pictures* as "unused for reference" immediately after decoding the IDR picture. After the decoding of an IDR picture all following *coded pictures* in *decoding order* can be decoded without *inter prediction* from any *picture* decoded prior to the IDR picture. The first *picture* of each *coded video sequence* is an IDR picture.

3.63 **inter coding**: Coding of a *block*, *macroblock*, *slice*, or *picture* that uses *inter prediction*.

3.64 **inter prediction**: A *prediction* derived from decoded samples of *reference pictures* other than the current *decoded picture*.

3.65 **interpretation sample value**: A possibly-altered value corresponding to a decoded sample value of an *auxiliary coded picture* that may be generated for use in the *display process*. Interpretation sample values are not used in the *decoding process* and have no normative effect on the *decoding process*.

3.66 **intra coding**: Coding of a *block, macroblock*, *slice*, or *picture* that uses *intra prediction*.

3.67 **intra prediction**: A *prediction* derived from the decoded samples of the same decoded *slice*.

3.68 **intra slice**: See *I slice*.

3.69 **inverse transform**: A part of the *decoding process* by which a set of *transform coefficients* are converted into spatial-domain values, or by which a set of *transform coefficients* are converted into *DC transform coefficients*.

3.70 **layer**: One of a set of syntactical structures in a non-branching hierarchical relationship. Higher layers contain lower layers. The coding layers are the *coded video sequence*, *picture*, *slice*, and *macroblock* layers.

3.71 **level**: A defined set of constraints on the values that may be taken by the *syntax elements* and variables of this Recommendation | International Standard. The same set of levels is defined for all *profiles*, with most aspects of the definition of each level being in common across different *profiles*. Individual implementations may, within specified constraints, support a different level for each supported *profile*. In a different context, level is the value of a *transform coefficient* prior to *scaling*.

3.72 **list 0 (list 1) motion vector**: A *motion vector* associated with a *reference index* pointing into *reference picture list 0* (*list 1*).

3.73 **list 0 (list 1) prediction**: *Inter prediction* of the content of a *slice* using a *reference index* pointing into *reference picture list 0* (*list 1*).

3.74 **luma**: An adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours. The symbol or subscript used for luma is Y or L.

NOTE – The term luma is used rather than the term luminance in order to avoid the implication of the use of linear light transfer characteristics that is often associated with the term luminance. The symbol L is sometimes used instead of the symbol Y to avoid confusion with the symbol y as used for vertical location.

3.75 **macroblock**: A 16x16 *block* of *luma* samples and two corresponding *blocks* of *chroma* samples. The division of a *slice* or a *macroblock pair* into macroblocks is a *partitioning*.

3.76 **macroblock-adaptive frame/field decoding**: A *decoding process* for *coded frames* in which some *macroblocks* may be decoded as *frame macroblocks* and others may be decoded as *field macroblocks*.

3.77 **macroblock address**: When *macroblock-adaptive frame/field decoding* is not in use, a macroblock address is the index of a macroblock in a *macroblock raster scan* of the *picture* starting with zero for the top-left *macroblock* in a *picture*. When *macroblock-adaptive frame/field decoding* is in use, the macroblock address of the *top macroblock* of a *macroblock pair* is two times the index of the *macroblock pair* in a *macroblock pair raster scan* of the *picture*, and the macroblock address of the *bottom macroblock* of a *macroblock pair* is the macroblock address of the corresponding *top macroblock* plus 1. The macroblock address of the *top macroblock* of each *macroblock pair* is an even number and the macroblock address of the *bottom macroblock* of each *macroblock pair* is an odd number.

3.78 **macroblock location**: The two-dimensional coordinates of a *macroblock* in a *picture* denoted by (x, y). For the top left *macroblock* of the *picture* (x, y) is equal to (0, 0). x is incremented by 1 for each *macroblock* column from left to right. When *macroblock-adaptive frame/field decoding* is not in use, y is incremented by 1 for each *macroblock* row from top to bottom. When *macroblock-adaptive frame/field decoding* is in use, y is incremented by 2 for each *macroblock pair* row from top to bottom, and is incremented by an additional 1 when a macroblock is a *bottom macroblock*.

3.79 **macroblock pair**: A pair of vertically contiguous *macroblocks* in a *frame* that is coupled for use in *macroblock-adaptive frame/field decoding*. The division of a *slice* into macroblock pairs is a *partitioning*.

3.80 **macroblock partition**: A *block* of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *macroblock* for *inter prediction*.

3.81 **macroblock to slice group map**: A means of mapping *macroblocks* of a *picture* into *slice groups*. The macroblock to slice group map consists of a list of numbers, one for each coded *macroblock*, specifying the *slice group* to which each coded *macroblock* belongs.

3.82 **map unit to slice group map**: A means of mapping *slice group map units* of a *picture* into *slice groups*. The map unit to slice group map consists of a list of numbers, one for each *slice group map unit*, specifying the *slice group* to which each coded *slice group map unit* belongs.

3.83 **may**: A term used to refer to behaviour that is allowed, but not necessarily required. In some places where the optional nature of the described behaviour is intended to be emphasized, the phrase "may or may not" is used to provide emphasis.

3.84 **memory management control operation**: Seven operations that control *reference picture marking*.

3.85 **motion vector**: A two-dimensional vector used for *inter prediction* that provides an offset from the coordinates in the *decoded picture* to the coordinates in a *reference picture*.

3.86 **must**: A term used in expressing an observation about a requirement or an implication of a requirement that is specified elsewhere in this Recommendation | International Standard. This term is used exclusively in an *informative* context.

3.87 **NAL unit**: A syntax structure containing an indication of the type of data to follow and *bytes* containing that data in the form of an *RBSP* interspersed as necessary with *emulation prevention bytes*.

3.88 **NAL unit stream**: A sequence of *NAL units*.

3.89 **non-paired field**: A collective term for a *non-paired reference field* or a *non-paired non-reference field*.

3.90 **non-paired non-reference field:** A decoded *non-reference field* that is not part of a *complementary non-reference field pair*.

3.91 **non-paired reference field**: A decoded *reference field* that is not part of a *complementary reference field pair*.

3.92 **non-reference field**: A *field* coded with nal_ref_idc equal to 0.

3.93 **non-reference frame**: A *frame* coded with nal_ref_idc equal to 0.

3.94 **non-reference picture**: A *picture* coded with nal_ref_idc equal to 0. A *non-reference picture* is not used for *inter prediction* of any other *pictures*.

3.95 **note**: A term used to prefix *informative* remarks. This term is used exclusively in an *informative* context.

3.96 **opposite parity**: The *opposite parity* of *top* is *bottom*, and vice versa.

3.97 **output order**: The order in which the *decoded pictures* are output from the *decoded picture buffer*.

3.98 **P slice**: A *slice* that may be decoded using *intra prediction* from decoded samples within the same *slice* or *inter prediction* from previously-decoded *reference pictures*, using at most one *motion vector* and *reference index* to *predict* the sample values of each *block*.

3.99 **parameter**: A *syntax element* of a *sequence parameter set* or a *picture parameter set*. Parameter is also used as part of the defined term *quantisation parameter*.

3.100 **parity**: The parity of a *field* can be *top* or *bottom*.

3.101 **partitioning**: The division of a set into subsets such that each element of the set is in exactly one of the subsets.

3.102 **picture**: A collective term for a *field* or a *frame*.

3.103 **picture parameter set**: A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded pictures* as determined by the pic_parameter_set_id *syntax element* found in each *slice header*.

3.104 **picture order count**: A variable having a value that is non-decreasing with increasing *picture* position in output order relative to the previous *IDR picture* in *decoding order* or relative to the previous *picture* containing the *memory management control operation* that marks all *reference pictures* as "unused for reference".

3.105 **prediction**: An embodiment of the *prediction process*.

3.106 **prediction process**: The use of a *predictor* to provide an estimate of the sample value or data element currently being decoded.

3.107 **predictive slice**: See *P slice*.

3.108 **predictor**: A combination of specified values or previously decoded sample values or data elements used in the *decoding process* of subsequent sample values or data elements.

3.109 **primary coded picture**: The coded representation of a *picture* to be used by the *decoding process* for a bitstream conforming to this Recommendation | International Standard. The primary coded picture contains all *macroblocks* of the *picture*. The only *pictures* that have a normative effect on the *decoding process* are primary coded pictures. See also *redundant coded picture*.

3.110 **profile**: A specified subset of the syntax of this Recommendation | International Standard.

3.111 **quantisation parameter**: A variable used by the *decoding process* for *scaling* of *transform coefficient levels*.

3.112 **random access**: The act of starting the decoding process for a *bitstream* at a point other than the beginning of the stream.

3.113 **raster scan**: A mapping of a rectangular two-dimensional pattern to a one-dimensional pattern such that the first entries in the one-dimensional pattern are from the first top row of the two-dimensional pattern scanned from left to right, followed similarly by the second, third, etc. rows of the pattern (going down) each scanned from left to right.

3.114 **raw byte sequence payload (RBSP):** A syntax structure containing an integer number of *bytes* that is encapsulated in a *NAL unit*. An RBSP is either empty or has the form of a *string of data bits* containing *syntax elements* followed by an *RBSP stop bit* and followed by zero or more subsequent bits equal to 0.

3.115 **raw byte sequence payload (RBSP) stop bit:** A bit equal to 1 present within a *raw byte sequence payload (RBSP)* after a *string of data bits*. The location of the end of the *string of data bits* within an *RBSP* can be identified by searching from the end of the *RBSP* for the *RBSP stop bit*, which is the last non-zero bit in the *RBSP*.

3.116 **recovery point**: A point in the *bitstream* at which the recovery of an exact or an approximate representation of the *decoded pictures* represented by the *bitstream* is achieved after a *random access* or *broken link*.

3.117 **redundant coded picture**: A coded representation of a *picture* or a part of a *picture*. The content of a redundant coded picture shall not be used by the *decoding process* for a *bitstream* conforming to this Recommendation | International Standard. A *redundant coded picture* is not required to contain all *macroblocks* in the *primary coded picture*. Redundant coded pictures have no normative effect on the *decoding process*. See also *primary coded picture*.

3.118 **reference field**: A *reference field* may be used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded field* or *field macroblocks* of a *coded frame* are decoded. See also *reference picture*.

3.119 **reference frame**: A *reference frame* may be used for *inter prediction* when *P*, *SP*, and *B slices* of a *coded frame* are decoded. See also *reference picture*.

3.120 **reference index**: An index into a *reference picture list*.

3.121 **reference picture**: A *picture* with nal_ref_idc not equal to 0. A *reference picture* contains samples that may be used for *inter prediction* in the *decoding process* of subsequent *pictures* in *decoding order*.

3.122 **reference picture list**: A list of *reference pictures* that is used for *inter prediction* of a *P, B,* or *SP slice*. For the *decoding process* of a *P* or *SP slice,* there is one reference picture list. For the *decoding process* of a *B slice*, there are two reference picture lists.

3.123 **reference picture list 0**: A *reference picture list* used for *inter prediction* of a *P*, *B*, or *SP slice*. All *inter prediction* used for *P* and *SP slices* uses reference picture list 0. Reference picture list 0 is one of two *reference picture lists* used for *inter prediction* for a *B slice*, with the other being *reference picture list 1*.

3.124 **reference picture list 1**: A *reference picture list* used for *inter prediction* of a *B slice*. Reference picture list 1 is one of two lists of *reference picture lists* used for *inter prediction* for a *B slice*, with the other being *reference picture list 0*.

3.125 **reference picture marking**: Specifies, in the bitstream, how the *decoded pictures* are marked for *inter prediction*.

3.126 **reserved**: The term reserved, when used in the clauses specifying some values of a particular *syntax element*, are for future use by ITU-T | ISO/IEC. These values shall not be used in *bitstreams* conforming to this Recommendation | International Standard, but may be used in future extensions of this Recommendation | International Standard by ITU-T | ISO/IEC.

3.127 **residual**: The decoded difference between a *prediction* of a sample or data element and its decoded value.

3.128 **run**: A number of consecutive data elements represented in the decoding process. In one context, the number of zero-valued *transform coefficient levels* preceding a non-zero *transform coefficient level* in the list of *transform coefficient levels* generated by a *zig-zag scan* or a *field scan*. In other contexts, run refers to a number of *macroblocks*.

3.129 **sample aspect ratio**: Specifies, for assisting the display process, which is not specified in this Recommendation | International Standard, the ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the *luma* sample array in a *frame*. Sample aspect ratio is expressed as *h*:*v*, where *h* is horizontal width and *v* is vertical height (in arbitrary units of spatial distance).

3.130 **scaling**: The process of multiplying *transform coefficient levels* by a factor, resulting in *transform coefficients*.

3.131 **sequence parameter set**: A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded video sequences* as determined by the content of a seq_parameter_set_id *syntax element* found in the *picture parameter set* referred to by the pic_parameter_set_id *syntax element* found in each *slice header*.

3.132 **shall**: A term used to express mandatory requirements for conformance to this Recommendation | International Standard. When used to express a mandatory constraint on the values of *syntax elements* or on the results obtained by operation of the specified *decoding process*, it is the responsibility of the *encoder* to ensure that the constraint is fulfilled. When used in reference to operations performed by the *decoding process*, any *decoding process* that produces identical results to the *decoding process* described herein conforms to the *decoding process* requirements of this Recommendation | International Standard.

3.133 **should**: A term used to refer to behaviour of an implementation that is encouraged to be followed under anticipated ordinary circumstances, but is not a mandatory requirement for conformance to this Recommendation | International Standard.

3.134 **SI slice**: A *slice* that is coded using *prediction* only from decoded samples within the same *slice* and using quantisation of the *prediction* samples. An SI slice can be coded such that its decoded samples can be constructed identically to an *SP slice*.

3.135 **skipped macroblock**: A *macroblock* for which no data is coded other than an indication that the *macroblock* is to be decoded as "skipped". This indication may be common to several *macroblocks*.

3.136 **slice**: An integer number of *macroblocks* or *macroblock pairs* ordered consecutively in the *raster scan* within a particular *slice group*. For the *primary coded picture*, the division of each *slice group* into slices is a *partitioning*. Although a slice contains *macroblocks* or *macroblock pairs* that are consecutive in the *raster scan* within a *slice group*, these *macroblocks* or *macroblock pairs* are not necessarily consecutive in the *raster scan* within the *picture*. The addresses of the *macroblocks* are derived from the address of the first *macroblock* in a slice (as represented in the *slice header*) and the *macroblock to slice group map*.

3.137 **slice data partitioning**: A method of *partitioning* selected *syntax elements* into *syntax structures* based on a *category* associated with each *syntax element*.

3.138 **slice group**: A subset of the *macroblocks* or *macroblock pairs* of a *picture*. The division of the *picture* into slice groups is a *partitioning* of the *picture*. The partitioning is specified by the *macroblock to slice group map*.

3.139 **slice group map units**: The units of the *map unit to slice group map*.

3.140 **slice header**: A part of a coded *slice* containing the data elements pertaining to the first or all *macroblocks* represented in the *slice*.

3.141 **source**: Term used to describe the video material or some of its attributes before encoding.

3.142 **SP slice**: A *slice* that is coded using *inter prediction* from previously-decoded *reference pictures*, using at most one *motion vector* and *reference index* to *predict* the sample values of each *block*. An SP slice can be coded such that its decoded samples can be constructed identically to another SP slice or an *SI slice*.

3.143 **start code prefix**: A unique sequence of three *bytes* equal to 0x000001 embedded in the *byte stream* as a prefix to each *NAL unit*. The location of a start code prefix can be used by a *decoder* to identify the beginning of a new *NAL unit* and the end of a previous *NAL unit*. Emulation of start code prefixes is prevented within *NAL units* by the inclusion of *emulation prevention bytes*.

3.144 **string of data bits (SODB)**: A sequence of some number of bits representing *syntax elements* present within a *raw byte sequence payload* prior to the *raw byte sequence payload stop bit*. Within an SODB, the left-most bit is considered to be the first and most significant bit, and the right-most bit is considered to be the last and least significant bit.

3.145 **sub-macroblock**: One quarter of the samples of a *macroblock*, i.e., an 8x8 *luma block* and two corresponding *chroma blocks* of which one corner is located at a corner of the *macroblock*.

3.146 **sub-macroblock partition**: A *block* of *luma* samples and two corresponding *blocks* of *chroma* samples resulting from a *partitioning* of a *sub-macroblock* for *inter prediction*.

3.147 **switching I slice**: See *SI slice*.

3.148 **switching P slice**: See *SP slice*.

3.149 **syntax element**: An element of data represented in the *bitstream*.

3.150 **syntax structure**: Zero or more *syntax elements* present together in the *bitstream* in a specified order.

3.151 **top field**: One of two *fields* that comprise a *frame*. Each row of a *top field* is spatially located immediately above the corresponding row of the *bottom field*.

3.152 **top macroblock (of a macroblock pair)**: The *macroblock* within a *macroblock pair* that contains the samples in the top row of samples for the *macroblock pair*. For a *field macroblock pair*, the top macroblock represents the samples from the region of the *top field* of the *frame* that lie within the spatial region of the *macroblock pair*. For a *frame macroblock pair*, the top macroblock represents the samples of the *frame* that lie within the top half of the spatial region of the *macroblock pair*.

3.153 **transform coefficient**: A scalar quantity, considered to be in a frequency domain, that is associated with a particular one-dimensional or two-dimensional *frequency index* in an *inverse transform* part of the *decoding process*.

3.154 **transform coefficient level**: An integer quantity representing the value associated with a particular two-dimensional frequency index in the *decoding process* prior to *scaling* for computation of a *transform coefficient* value.

3.155 **universal unique identifier (UUID)**: An identifier that is unique with respect to the space of all universal unique identifiers.

3.156 **unspecified:** The term unspecified, when used in the clauses specifying some values of a particular *syntax element*, indicates that the values have no specified meaning in this Recommendation | International Standard and will not have a specified meaning in the future as an integral part of this Recommendation | International Standard.

3.157 **variable length coding (VLC)**: A reversible procedure for entropy coding that assigns shorter bit strings to *symbols* expected to be more frequent and longer bit strings to *symbols* expected to be less frequent.

3.158 **zig-zag scan**: A specific sequential ordering of *transform coefficient levels* from (approximately) the lowest spatial frequency to the highest. Zig-zag scan is used for *transform coefficient levels* in *frame macroblocks.*

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CBR	Constant Bit Rate
CPB	Coded Picture Buffer
DPB	Decoded Picture Buffer
DUT	Decoder under test
FIFO	First-In, First-Out
HRD	Hypothetical Reference Decoder
HSS	Hypothetical Stream Scheduler
IDR	Instantaneous Decoding Refresh
LSB	Least Significant Bit
MB	Macroblock
MBAFF	Macroblock-Adaptive Frame-Field Coding
MSB	Most Significant Bit
NAL	Network Abstraction Layer
RBSP	Raw Byte Sequence Payload
SEI	Supplemental Enhancement Information
SODB	String Of Data Bits
UUID	Universal Unique Identifier
VBR	Variable Bit Rate
VCL	Video Coding Layer
VLC	Variable Length Coding
VUI	Video Usability Information

5 Conventions

NOTE – The mathematical operators used in this Specification are similar to those used in the C programming language. However, integer division and arithmetic shift operations are specifically defined. Numbering and counting conventions generally begin from 0.

5.1 Arithmetic operators

The following arithmetic operators are defined as follows.

+	Addition
–	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	Multiplication
x^y	Exponentiation. Specifies x to the power of y. In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	Integer division with truncation of the result toward zero. For example, 7/4 and –7/–4 are truncated to 1 and –7/4 and 7/–4 are truncated to –1.
÷	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\frac{x}{y}$	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\sum_{i=x}^{y} f(i)$	The summation of f(i) with i taking all integer values from x up to and including y.
x % y	Modulus. Remainder of x divided by y, defined only for integers x and y with x >= 0 and y > 0.

When order of precedence is not indicated explicitly by use of parenthesis, the following rules apply:

- multiplication and division operations are considered to take place before addition and subtraction;
- multiplication and division operations in sequence are evaluated sequentially from left to right;
- addition and subtraction operations in sequence are evaluated sequentially from left to right.

5.2 Logical operators

The following logical operators are defined as follows:

x && y	Boolean logical "and" of x and y
x || y	Boolean logical "or" of x and y
!	Boolean logical "not"
x ? y : z	If x is TRUE or not equal to 0, evaluates to the value of y; otherwise, evaluates to the value of z

5.3 Relational operators

The following relational operators are defined as follows:

>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
= =	Equal to
!=	Not equal to

When a relational operator is applied to a syntax element or variable that has been assigned the value "na" (not applicable), the value "na" is treated as a distinct value for the syntax element or variable. The value "na" is considered not to be equal to any other value.

5.4 Bit-wise operators

The following bit-wise operators are defined as follows:

&	Bit-wise "and". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

| Bit-wise "or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

x >> y Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for positive integer values of y. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of x prior to the shift operation.

x << y Arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for positive integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.5 Assignment operators

The following arithmetic operators are defined as follows:

= Assignment operator.

++ Increment, i.e., *x*++ is equivalent to *x* = *x* + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation.

–– Decrement, i.e., *x*–– is equivalent to *x* = *x* – 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation.

+= Increment by amount specified, i.e., x += 3 is equivalent to x = x + 3, and x += (-3) is equivalent to x = x + (-3).

–= Decrement by amount specified, i.e., x –= 3 is equivalent to x = x – 3, and x –= (-3) is equivalent to x = x – (-3).

5.6 Range notation

The following notation is used to specify a range of values

x = y .. z x takes on integer values starting from y to z inclusive, with x, y, and z being integer numbers.

5.7 Mathematical functions

The following mathematical functions are defined as follows:

$$\text{Abs}(x) = \begin{cases} x & ; \quad x >= 0 \\ -x & ; \quad x < 0 \end{cases} \tag{5-1}$$

Ceil(x) the smallest integer greater than or equal to x. (5-2)

$$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 << \text{BitDepth}_Y) - 1, x) \tag{5-3}$$

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 << \text{BitDepth}_C) - 1, x) \tag{5-4}$$

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} \tag{5-5}$$

Floor(x) the greatest integer less than or equal to x. (5-6)

$$\text{InverseRasterScan}(a, b, c, d, e) = \begin{cases} (a\%(d / b)) * b; & e == 0 \\ (a / (d / b)) * c; & e == 1 \end{cases} \tag{5-7}$$

Log2(x) returns the base-2 logarithm of x. (5-8)

Log10(x) returns the base-10 logarithm of x. (5-9)

$$Median(x, y, z) = x + y + z - Min(x, Min(y, z)) - Max(x, Max(y, z)) \quad (5\text{-}10)$$

$$Min(x, y) = \begin{cases} x & ; \quad x <= y \\ y & ; \quad x > y \end{cases} \quad (5\text{-}11)$$

$$Max(x, y) = \begin{cases} x & ; \quad x >= y \\ y & ; \quad x < y \end{cases} \quad (5\text{-}12)$$

$$Round(x) = Sign(x) * Floor(Abs(x) + 0.5) \quad (5\text{-}13)$$

$$Sign(x) = \begin{cases} 1 & ; \quad x >= 0 \\ -1 & ; \quad x < 0 \end{cases} \quad (5\text{-}14)$$

$$Sqrt(x) = \sqrt{x} \quad (5\text{-}15)$$

5.8 Variables, syntax elements, and tables

Syntax elements in the bitstream are represented in **bold** type. Each syntax element is described by its name (all lower case letters with underscore characters), its one or two syntax categories, and one or two descriptors for its method of coded representation. The decoding process behaves according to the value of the syntax element and to the values of previously decoded syntax elements. When a value of a syntax element is used in the syntax tables or the text, it appears in regular (i.e., not bold) type.

In some cases the syntax tables may use the values of other variables derived from syntax elements values. Such variables appear in the syntax tables, or text, named by a mixture of lower case and upper case letter and without any underscore characters. Variables starting with an upper case letter are derived for the decoding of the current syntax structure and all depending syntax structures. Variables starting with an upper case letter may be used in the decoding process for later syntax structures mentioning the originating syntax structure of the variable. Variables starting with a lower case letter are only used within the subclause in which they are derived.

In some cases, "mnemonic" names for syntax element values or variable values are used interchangeably with their numerical values. Sometimes "mnemonic" names are used without any associated numerical values. The association of values and names is specified in the text. The names are constructed from one or more groups of letters separated by an underscore character. Each group starts with an upper case letter and may contain more upper case letters.

NOTE – The syntax is described in a manner that closely follows the C-language syntactic constructs.

Functions are described by their names, which are constructed as syntax element names, with left and right round parentheses including zero or more variable names (for definition) or values (for usage), separated by commas (if more than one variable).

A one-dimensional array is referred to as a list. A two-dimensional array is referred to as a matrix. Arrays can either be syntax elements or variables. Subscripts or square parentheses are used for the indexing of arrays. In reference to a visual depiction of a matrix, the first subscript is used as a row (vertical) index and the second subscript is used as a column (horizontal) index. The indexing order is reversed when using square parentheses rather than subscripts for indexing. Thus, an element of a matrix s at horizontal position x and vertical position y may be denoted either as s[x, y] or as s_{yx}.

Binary notation is indicated by enclosing the string of bit values by single quote marks. For example, '01000001' represents an eight-bit string having only its second and its last bits equal to 1.

Hexadecimal notation, indicated by prefixing the hexadecimal number by "0x", may be used instead of binary notation when the number of bits is an integer multiple of 4. For example, 0x41 represents an eight-bit string having only its second and its last bits equal to 1.

Numerical values not enclosed in single quotes and not prefixed by "0x" are decimal values.

A value equal to 0 represents a FALSE condition in a test statement. The value TRUE is represented by any other value different than zero.

5.9 Text description of logical operations

In the text, a statement of logical operations as would be described in pseudo-code as

```
if( condition 0 )
  statement 0
else if ( condition 1 )
  statement 1
…
else /* informative remark on remaining condition */
  statement n
```

may be described in the following manner:

... as follows / ... the following applies.

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- …
- Otherwise (informative remark on remaining condition), statement n

Each "If...Otherwise, if...Otherwise, ..." statement in the text is introduced with "... as follows" or "... the following applies" immediately followed by "If ... ". The last condition of the "If...Otherwise, if...Otherwise, ..." is always an "Otherwise, ...". Interleaved "If...Otherwise, if...Otherwise, ..." statements can be identified by matching "... as follows" or "... the following applies" with the ending "Otherwise, ...".

In the text, a statement of logical operations as would be described in pseudo-code as

```
if( condition 0a  &&  condition 0b )
  statement 0
else if ( condition 1a  ||  condition 1b )
  statement 1
…
else
  statement n
```

may be described in the following manner:

... as follows / ... the following applies.

- If all of the following conditions are true, statement 0
  - condition 0a
  - condition 0b
- Otherwise, if any of the following conditions are true, statement 1
  - condition 1a
  - condition 1b
- …
- Otherwise, statement n

In the text, a statement of logical operations as would be described in pseudo-code as

```
if( condition 0 )
  statement 0
if ( condition 1 )
  statement 1
```

may be described in the following manner:

When condition 0, statement 0

When condition 1, statement 1

5.10 Processes

Processes are used to describe the decoding of syntax elements. A process has a separate specification and invoking. All syntax elements and upper case variables that pertain to the current syntax structure and depending syntax structures are available in the process specification and invoking. A process specification may also have a lower case variable explicitly specified as the input. Each process specification has explicitly specified an output. The output is a variable that can either be an upper case variable or a lower case variable.

The assignment of variables is specified as follows.

- If invoking a process, variables are explicitly assigned to lower case input or output variables of the process specification in case these do not have the same name.
- Otherwise (when the variables at the invoking and specification have the same name), assignment is implied.

In the specification of a process, a specific macroblock may be referred to by the variable name having a value equal to the address of the specific macroblock.

6 Source, coded, decoded and output data formats, scanning processes, and neighbouring relationships

6.1 Bitstream formats

This subclause specifies the relationship between the NAL unit stream and byte stream, either of which are referred to as the bitstream.

The bitstream can be in one of two formats: the NAL unit stream format or the byte stream format. The NAL unit stream format is conceptually the more "basic" type. It consists of a sequence of syntax structures called NAL units. This sequence is ordered in decoding order. There are constraints imposed on the decoding order (and contents) of the NAL units in the NAL unit stream.

The byte stream format can be constructed from the NAL unit stream format by ordering the NAL units in decoding order and prefixing each NAL unit with a start code prefix and zero or more zero-valued bytes to form a stream of bytes. The NAL unit stream format can be extracted from the byte stream format by searching for the location of the unique start code prefix pattern within this stream of bytes. Methods of framing the NAL units in a manner other than use of the byte stream format are outside the scope of this Recommendation | International Standard. The byte stream format is specified in Annex B.

6.2 Source, decoded, and output picture formats

This subclause specifies the relationship between source and decoded frames and fields that is given via the bitstream.

The video source that is represented by the bitstream is a sequence of either or both frames or fields (called collectively pictures) in decoding order.

The source and decoded pictures (frames or fields) are each comprised of one or more sample arrays:

- Luma (Y) only (monochrome), with or without an auxiliary array.
- Luma and two Chroma (YCbCr or YCgCo), with or without an auxiliary array.
- Green, Blue and Red (GBR, also known as RGB), with or without an auxiliary array.

– Arrays representing other unspecified monochrome or tri-stimulus colour samplings (for example, YZX, also known as XYZ), with or without an auxiliary array.

For convenience of notation and terminology in this specification, the variables and terms associated with these arrays are referred to as luma (or L or Y) and chroma, where the two chroma arrays are referred to as Cb and Cr; regardless of the actual colour representation method in use. The actual colour representation method in use can be indicated in syntax that is specified in Annex E. The (monochrome) auxiliary arrays, which may or may not be present as auxiliary pictures in a coded video sequence, are optional for decoding and can be used for such purposes as alpha blending.

The variables SubWidthC, and SubHeightC are specified in Table 6-1, depending on the chroma format sampling structure, which is specified through chroma_format_idc. An entry marked as "-" in Table 6-1 denotes an undefined value for SubWidthC or SubHeightC. Other values of chroma_format_idc, SubWidthC, and SubHeightC may be specified in the future by ITU-T | ISO/IEC.

Table 6-1 – SubWidthC, and SubHeightC values derived from chroma_format_idc

chroma_format_idc	Chroma Format	SubWidthC	SubHeightC
0	monochrome	–	–
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

In monochrome sampling there is only one sample array, which is nominally considered the luma array.

In 4:2:0 sampling, each of the two chroma arrays has half the height and half the width of the luma array.

In 4:2:2 sampling, each of the two chroma arrays has the same height and half the width of the luma array.

In 4:4:4 sampling, each of the two chroma arrays has the same height and width as the luma array.

The width and height of the luma sample arrays are each an integer multiple of 16. In bitstreams using 4:2:0 chroma sampling, the width and height of chroma sample arrays are each an integer multiple of 8. In bitstreams using 4:2:2 sampling, the width of the chroma sample arrays is an integer multiple of 8 and the height is an integer multiple of 16. The height of a luma array that is coded as two separate fields or in macroblock-adaptive frame-field coding (see below) is an integer multiple of 32. In bitstreams using 4:2:0 chroma sampling, the height of each chroma array that is coded as two separate fields or in macroblock-adaptive frame-field coding (see below) is an integer multiple of 16. The width or height of pictures output from the decoding process need not be an integer multiple of 16 and can be specified using a cropping rectangle.

The syntax for the luma and (when present) chroma arrays are ordered such when data for all three colour components is present, the data for the luma array is first, followed by any data for the Cb array, followed by any data for the Cr array, unless otherwise specified.

The width of fields coded referring to a specific sequence parameter set is the same as that of frames coded referring to the same sequence parameter set (see below). The height of fields coded referring to a specific sequence parameter set is half that of frames coded referring to the same sequence parameter set (see below).

The number of bits necessary for the representation of each of the samples in the luma and chroma arrays in a video sequence is in the range of 8 to 12, and the number of bits used in the luma array may differ from the number of bits used in the chroma arrays.

When the value of chroma_format_idc is equal to 1, the nominal vertical and horizontal relative locations of luma and chroma samples in frames are shown in Figure 6-1. Alternative chroma sample relative locations may be indicated in video usability information (see Annex E).

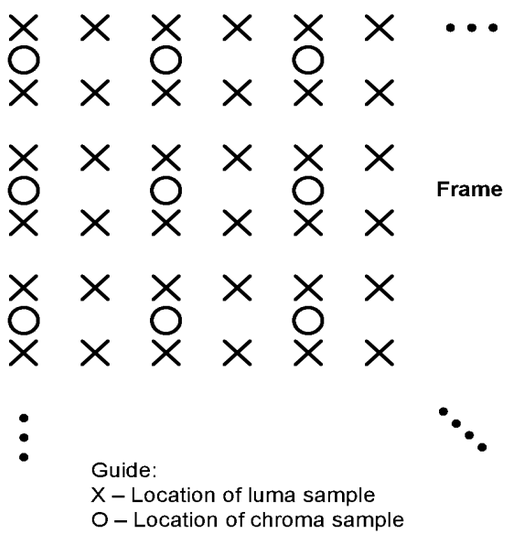


Figure 6-1 – Nominal vertical and horizontal locations of 4:2:0 luma and chroma samples in a frame

A frame consists of two fields as described below. A coded picture may represent a coded frame or an individual coded field. A coded video sequence conforming to this Recommendation | International Standard may contain arbitrary combinations of coded frames and coded fields. The decoding process is also specified in a manner that allows smaller regions of a coded frame to be coded either as a frame or field region, by use of macroblock-adaptive frame-field coding.

Source and decoded fields are one of two types: top field or bottom field. When two fields are output at the same time, or are combined to be used as a reference frame (see below), the two fields (which shall be of opposite parity) are interleaved. The first (i.e., top), third, fifth, etc. rows of a decoded frame are the top field rows. The second, fourth, sixth, etc. rows of a decoded frame are the bottom field rows. A top field consists of only the top field rows of a decoded frame. When the top field or bottom field of a decoded frame is used as a reference field (see below) only the even rows (for a top field) or the odd rows (for a bottom field) of the decoded frame are used.

When the value of chroma_format_idc is equal to 1, the nominal vertical and horizontal relative locations of luma and chroma samples in top and bottom fields are shown in Figure 6-2. The nominal vertical sampling relative locations of the chroma samples in a top field are specified as shifted up by one-quarter luma sample height relative to the field-sampling grid. The vertical sampling locations of the chroma samples in a bottom field are specified as shifted down by one-quarter luma sample height relative to the field-sampling grid. Alternative chroma sample relative locations may be indicated in the video usability information (see Annex E).

NOTE – The shifting of the chroma samples is in order for these samples to align vertically to the usual location relative to the full-frame sampling grid as shown in Figure 6-1.

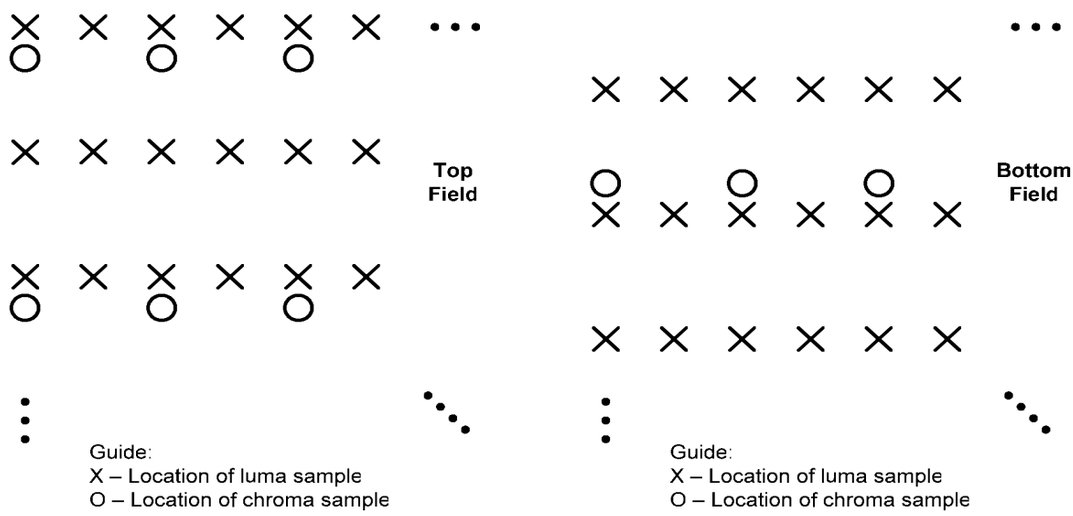


Figure 6-2 – Nominal vertical and horizontal sampling locations of 4:2:0 samples in top and bottom fields

When the value of chroma_format_idc is equal to 2, the chroma samples are co-sited with the corresponding luma samples and the nominal locations in a frame and in fields are as shown in Figure 6-3 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a frameandFigure 6-4, respectively.

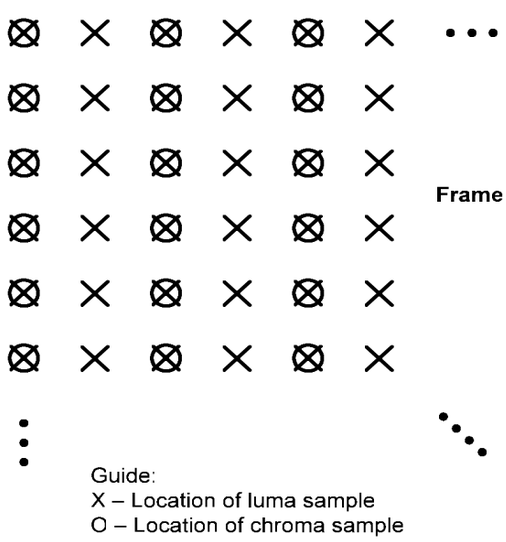


Figure 6-3 – Nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a frame

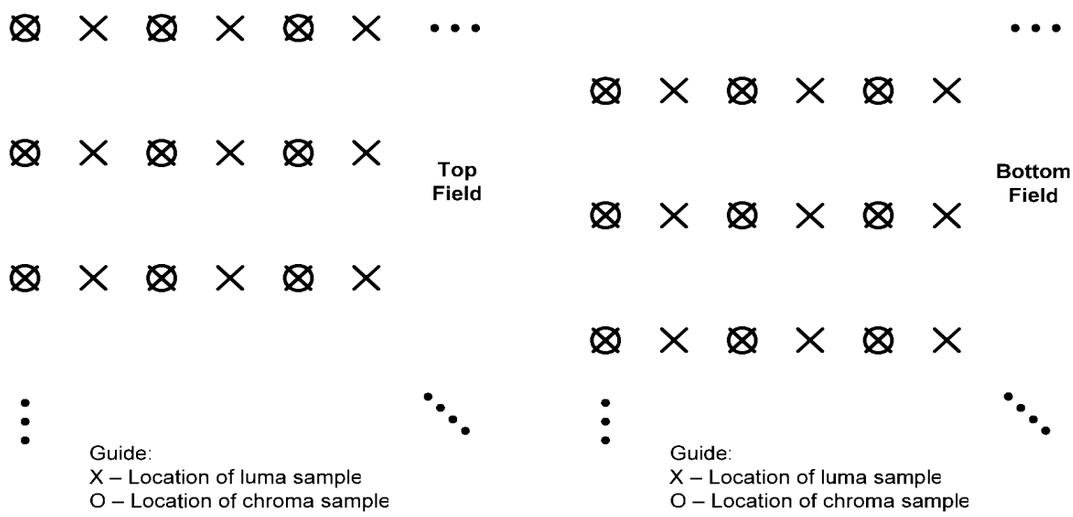


Figure 6-4 – Nominal vertical and horizontal sampling locations of 4:2:2 samples top and bottom fields

When the value of chroma_format_idc is equal to 3, all array samples are co-sited for all cases of frames and fields and the nominal locations in a frame and in fields are as shown in Figures 6-5 and 6-6, respectively.

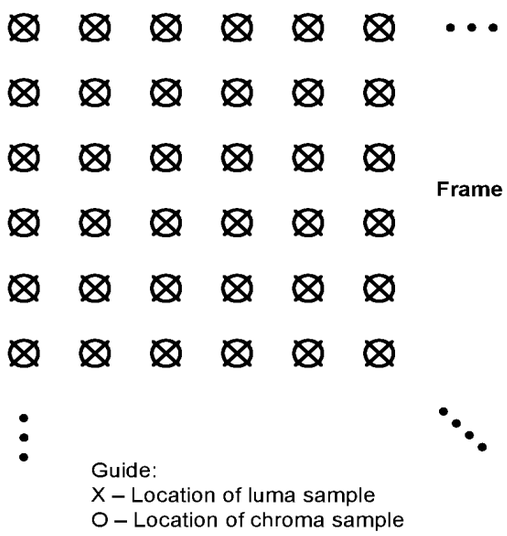


Figure 6-5 – Nominal vertical and horizontal locations of 4:4:4 luma and chroma samples in a frame

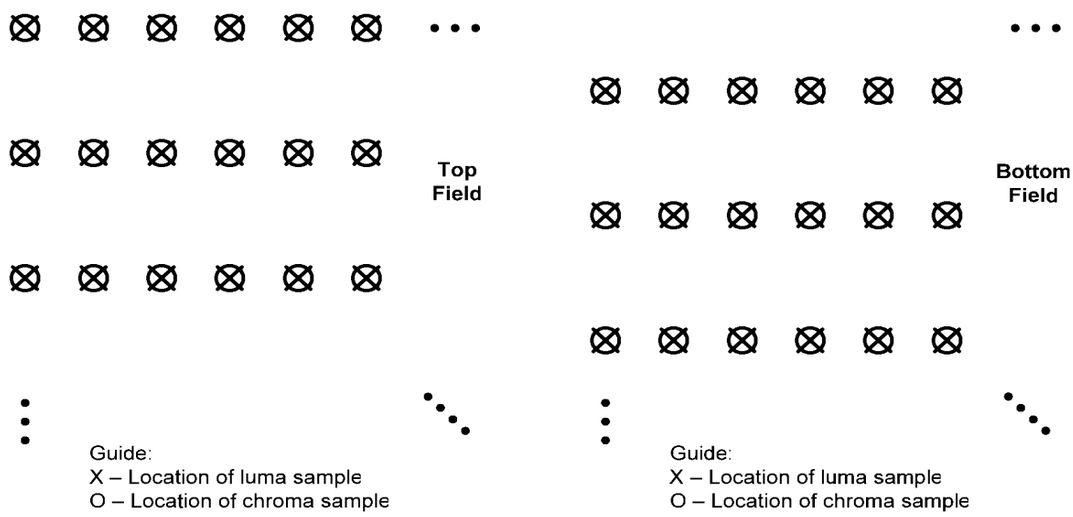


Figure 6-6 – Nominal vertical and horizontal sampling locations of 4:4:4 samples top and bottom fields

The samples are processed in units of macroblocks. The luma array for each macroblock is 16 samples in both width and height. The variables MbWidthC and MbHeightC, which specify the width and height, respectively, of the chroma arrays for each macroblock, are derived as follows.

– If chroma_format_idc is equal to 0 (monochrome), MbWidthC and MbHeightC are both equal to 0 (as no chroma arrays are specified for monochrome video).

– Otherwise, MbWidthC and MbHeightC are derived as

$$MbWidthC = 16 / SubWidthC \quad (6\text{-}1)$$
$$MbHeightC = 16 / SubHeightC \quad (6\text{-}2)$$

6.3 Spatial subdivision of pictures and slices

This subclause specifies how a picture is partitioned into slices and macroblocks. Pictures are divided into slices. A slice is a sequence of macroblocks, or, when macroblock-adaptive frame/field decoding is in use, a sequence of macroblock pairs.

Each macroblock is comprised of one 16x16 luma array and, when the video format is not monochrome, two corresponding chroma sample arrays. When macroblock-adaptive frame/field decoding is not in use, each macroblock represents a spatial rectangular region of the picture. For example, a picture may be divided into two slices as shown in Figure 6-7.

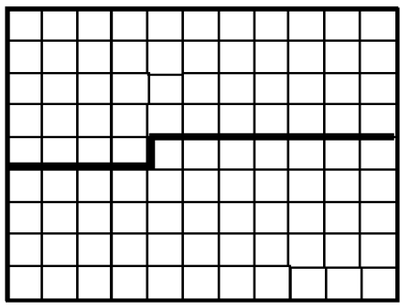


Figure 6-7 – A picture with 11 by 9 macroblocks that is partitioned into two slices

When macroblock-adaptive frame/field decoding is in use, the picture is partitioned into slices containing an integer number of macroblock pairs as shown in Figure 6-8. Each macroblock pair consists of two macroblocks.

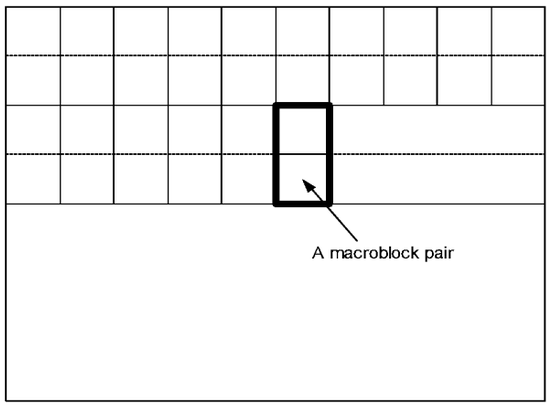


Figure 6-8 – Partitioning of the decoded frame into macroblock pairs

6.4 Inverse scanning processes and derivation processes for neighbours

This subclause specifies inverse scanning processes; i.e., the mapping of indices to locations, and derivation processes for neighbours.

6.4.1 Inverse macroblock scanning process

Input to this process is a macroblock address mbAddr.

Output of this process is the location (x, y) of the upper-left luma sample for the macroblock with address mbAddr relative to the upper-left sample of the picture.

The inverse macroblock scanning process is specified as follows.

– If MbaffFrameFlag is equal to 0,

$$x = InverseRasterScan(mbAddr, 16, 16, PicWidthInSamples_L, 0) \quad (6\text{-}3)$$

$$y = InverseRasterScan(mbAddr, 16, 16, PicWidthInSamples_L, 1) \quad (6\text{-}4)$$

– Otherwise (MbaffFrameFlag is equal to 1), the following applies.

$$xO = InverseRasterScan(mbAddr / 2, 16, 32, PicWidthInSamples_L, 0) \quad (6\text{-}5)$$

$$yO = InverseRasterScan(mbAddr / 2, 16, 32, PicWidthInSamples_L, 1) \quad (6\text{-}6)$$

Depending on the current macroblock the following applies.

– If the current macroblock is a frame macroblock

$$x = xO \quad (6\text{-}7)$$

$$y = yO + (mbAddr \% 2) * 16 \quad (6\text{-}8)$$

– Otherwise (the current macroblock is a field macroblock),

$$x = xO \quad (6\text{-}9)$$

$$y = yO + (mbAddr \% 2) \quad (6\text{-}10)$$

6.4.2 Inverse macroblock partition and sub-macroblock partition scanning process

Macroblocks or sub-macroblocks may be partitioned, and the partitions are scanned for inter prediction as shown in Figure 6-9. The outer rectangles refer to the samples in a macroblock or sub-macroblock, respectively. The rectangles refer to the partitions. The number in each rectangle specifies the index of the inverse macroblock partition scan or inverse sub-macroblock partition scan.

The functions MbPartWidth( ), MbPartHeight( ), SubMbPartWidth( ), and SubMbPartHeight( ) describing the width and height of macroblock partitions and sub-macroblock partitions are specified in Tables 7-13, 7-14, 7-17, and 7-18. MbPartWidth( ) and MbPartHeight( ) are set to appropriate values for each macroblock, depending on the macroblock type. SubMbPartWidth( ) and SubMbPartHeight( ) are set to appropriate values for each sub-macroblock of a macroblock with mb_type equal to P_8x8, P_8x8ref0, or B_8x8, depending on the sub-macroblock type.

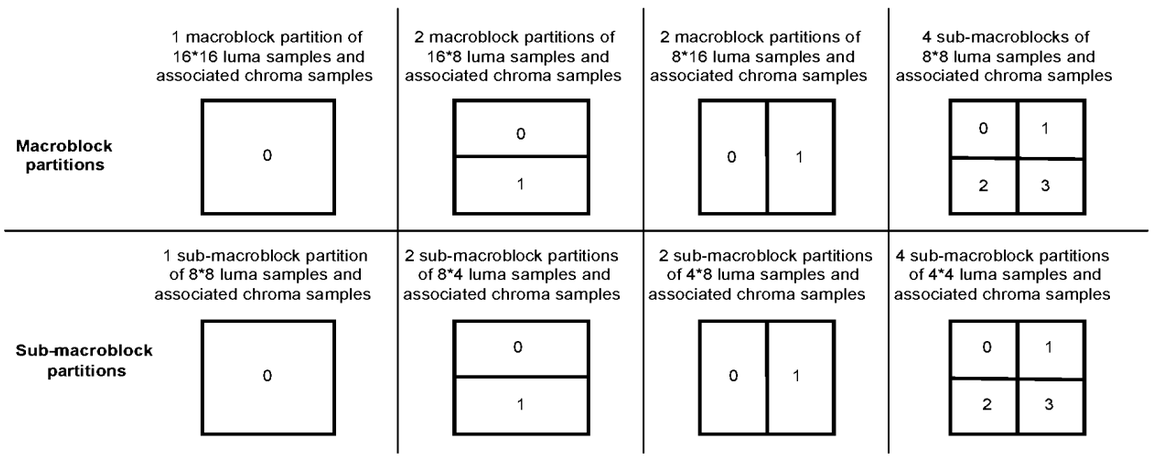


Figure 6-9 – Macroblock partitions, sub-macroblock partitions, macroblock partition scans, and sub-macroblock partition scans

6.4.2.1 Inverse macroblock partition scanning process

Input to this process is the index of a macroblock partition mbPartIdx.

Output of this process is the location (x, y) of the upper-left luma sample for the macroblock partition mbPartIdx relative to the upper-left sample of the macroblock.

The inverse macroblock partition scanning process is specified by

$$x = \text{InverseRasterScan}(\text{mbPartIdx}, \text{MbPartWidth}(\text{mb_type}), \text{MbPartHeight}(\text{mb_type}), 16, 0) \quad (6\text{-}11)$$

$$y = \text{InverseRasterScan}(\text{mbPartIdx}, \text{MbPartWidth}(\text{mb_type}), \text{MbPartHeight}(\text{mb_type}), 16, 1) \quad (6\text{-}12)$$

6.4.2.2 Inverse sub-macroblock partition scanning process

Inputs to this process are the index of a macroblock partition mbPartIdx and the index of a sub-macroblock partition subMbPartIdx.

Output of this process is the location (x, y) of the upper-left luma sample for the sub-macroblock partition subMbPartIdx relative to the upper-left sample of the sub-macroblock.

The inverse sub-macroblock partition scanning process is specified as follows.

– If mb_type is equal to P_8x8, P_8x8ref0, or B_8x8,

$$x = \text{InverseRasterScan}(\text{subMbPartIdx}, \text{SubMbPartWidth}(\text{sub_mb_type}[\text{mbPartIdx}]), \text{SubMbPartHeight}(\text{sub_mb_type}[\text{mbPartIdx}]), 8, 0) \quad (6\text{-}13)$$

$$y = \text{InverseRasterScan}(\text{subMbPartIdx}, \text{SubMbPartWidth}(\text{sub_mb_type}[\text{mbPartIdx}]), \text{SubMbPartHeight}(\text{sub_mb_type}[\text{mbPartIdx}]), 8, 1) \quad (6\text{-}14)$$

– Otherwise,

$$x = \text{InverseRasterScan}(\text{subMbPartIdx}, 4, 4, 8, 0) \quad (6\text{-}15)$$

$$y = \text{InverseRasterScan}(\text{subMbPartIdx}, 4, 4, 8, 1) \quad (6\text{-}16)$$

6.4.3 Inverse 4x4 luma block scanning process

Input to this process is the index of a 4x4 luma block luma4x4BlkIdx.

Output of this process is the location (x, y) of the upper-left luma sample for the 4x4 luma block with index luma4x4BlkIdx relative to the upper-left luma sample of the macroblock.

Figure 6-10 shows the scan for the 4x4 luma blocks.

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Figure 6-10 – Scan for 4x4 luma blocks

The inverse 4x4 luma block scanning process is specified by

x = InverseRasterScan(luma4x4BlkIdx / 4, 8, 8, 16, 0) + InverseRasterScan(luma4x4BlkIdx % 4, 4, 4, 8, 0) (6-17)

y = InverseRasterScan(luma4x4BlkIdx / 4, 8, 8, 16, 1) + InverseRasterScan(luma4x4BlkIdx % 4, 4, 4, 8, 1) (6-18)

6.4.4 Inverse 8x8 luma block scanning process

Input to this process is the index of an 8x8 luma block luma8x8BlkIdx.

Output of this process is the location (x, y) of the upper-left luma sample for the 8x8 luma block with index luma8x8BlkIdx relative to the upper-left luma sample of the macroblock.

Figure 6-11 shows the scan for the 8x8 luma blocks.

0	1
2	3

Figure 6-11 – Scan for 8x8 luma blocks

The inverse 8x8 luma block scanning process is specified by

x = InverseRasterScan(luma8x8BlkIdx, 8, 8, 16, 0) (6-19)

y = InverseRasterScan(luma8x8BlkIdx, 8, 8, 16, 1) (6-20)

6.4.5 Derivation process of the availability for macroblock addresses

Input to this process is a macroblock address mbAddr.

Output of this process is the availability of the macroblock mbAddr.

NOTE – The meaning of availability is determined when this process is invoked.

The macroblock is marked as available, unless one of the following conditions is true in which case the macroblock is marked as not available:

– mbAddr < 0

– mbAddr > CurrMbAddr

– the macroblock with address mbAddr belongs to a different slice than the macroblock with address CurrMbAddr

6.4.6 Derivation process for neighbouring macroblock addresses and their availability

This process can only be invoked when MbaffFrameFlag is equal to 0.

The outputs of this process are

– mbAddrA: the address and availability status of the macroblock to the left of the current macroblock.

– mbAddrB: the address and availability status of the macroblock above the current macroblock.

– mbAddrC: the address and availability status of the macroblock above-right of the current macroblock.

– mbAddrD: the address and availability status of the macroblock above-left of the current macroblock.

Figure 6-12 shows the relative spatial locations of the macroblocks with mbAddrA, mbAddrB, mbAddrC, and mbAddrD relative to the current macroblock with CurrMbAddr.

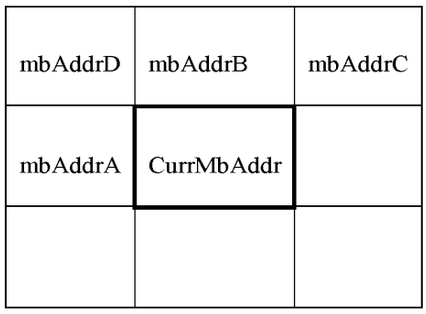


Figure 6-12 – Neighbouring macroblocks for a given macroblock

Input to the process in subclause 6.4.5 is mbAddrA = CurrMbAddr – 1 and the output is whether the macroblock mbAddrA is available. In addition, mbAddrA is marked as not available when CurrMbAddr % PicWidthInMbs is equal to 0.

Input to the process in subclause 6.4.5 is mbAddrB = CurrMbAddr – PicWidthInMbs and the output is whether the macroblock mbAddrB is available.

Input to the process in subclause 6.4.5 is mbAddrC = CurrMbAddr – PicWidthInMbs + 1 and the output is whether the macroblock mbAddrC is available. In addition, mbAddrC is marked as not available when (CurrMbAddr + 1) % PicWidthInMbs is equal to 0.

Input to the process in subclause 6.4.5 is mbAddrD = CurrMbAddr – PicWidthInMbs - 1 and the output is whether the macroblock mbAddrD is available. In addition, mbAddrD is marked as not available when CurrMbAddr % PicWidthInMbs is equal to 0.

6.4.7 Derivation process for neighbouring macroblock addresses and their availability in MBAFF frames

This process can only be invoked when MbaffFrameFlag is equal to 1.

The outputs of this process are

– mbAddrA: the address and availability status of the top macroblock of the macroblock pair to the left of the current macroblock pair.

– mbAddrB: the address and availability status of the top macroblock of the macroblock pair above the current macroblock pair.

– mbAddrC: the address and availability status of the top macroblock of the macroblock pair above-right of the current macroblock pair.
– mbAddrD: the address and availability status of the top macroblock of the macroblock pair above-left of the current macroblock pair.

Figure 6-13 shows the relative spatial locations of the macroblocks with mbAddrA, mbAddrB, mbAddrC, and mbAddrD relative to the current macroblock with CurrMbAddr.

mbAddrA, mbAddrB, mbAddrC, and mbAddrD have identical values regardless whether the current macroblock is the top or the bottom macroblock of a macroblock pair.

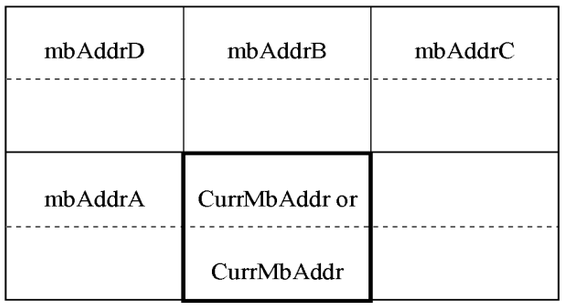


Figure 6-13 – Neighbouring macroblocks for a given macroblock in MBAFF frames

Input to the process in subclause 6.4.5 is mbAddrA =2 * (CurrMbAddr / 2 – 1) and the output is whether the macroblock mbAddrA is available. In addition, mbAddrA is marked as not available when (CurrMbAddr / 2) % PicWidthInMbs is equal to 0.

Input to the process in subclause 6.4.5 is mbAddrB =2 * (CurrMbAddr / 2 – PicWidthInMbs) and the output is whether the macroblock mbAddrB is available.

Input to the process in subclause 6.4.5 is mbAddrC = 2 * (CurrMbAddr / 2 – PicWidthInMbs + 1) and the output is whether the macroblock mbAddrC is available. In addition, mbAddrC is marked as not available when (CurrMbAddr / 2 + 1) % PicWidthInMbs is equal to 0.

Input to the process in subclause 6.4.5 is mbAddrD = 2 * (CurrMbAddr / 2 – PicWidthInMbs - 1) and the output is whether the macroblock mbAddrD is available. In addition, mbAddrD is marked as not available when (CurrMbAddr / 2) % PicWidthInMbs is equal to 0.

6.4.8 Derivation processes for neighbouring macroblocks, blocks, and partitions

Subclause 6.4.8.1 specifies the derivation process for neighbouring macroblocks.

Subclause 6.4.8.2 specifies the derivation process for neighbouring 8x8 luma blocks.

Subclause 6.4.8.3 specifies the derivation process for neighbouring 4x4 luma blocks.

Subclause 6.4.8.4 specifies the derivation process for neighbouring 4x4 chroma blocks.

Subclause 6.4.8.5 specifies the derivation process for neighbouring partitions.

Table 6-2 specifies the values for the difference of luma location (xD, yD) for the input and the replacement for N in mbAddrN, mbPartIdxN, subMbPartIdxN, luma8x8BlkIdxN, luma4x4BlkIdxN, and chroma4x4BlkIdxN for the output. These input and output assignments are used in subclauses 6.4.8.1 to 6.4.8.5. The variable predPartWidth is specified when Table 6-2 is referred to.

Table 6-2 – Specification of input and output assignments for subclauses 6.4.8.1 to 6.4.8.5

N	xD	yD
A	-1	0
B	0	-1
C	predPartWidth	-1
D	-1	-1

Figure 6-14 illustrates the relative location of the neighbouring macroblocks, blocks, or partitions A, B, C, and D to the current macroblock, partition, or block, when the current macroblock, partition, or block is in frame coding mode.

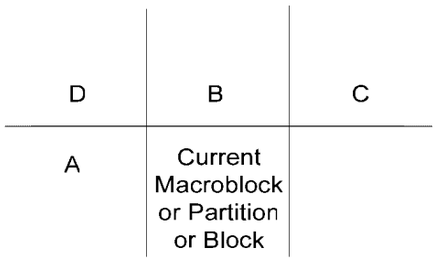


Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative)

6.4.8.1 Derivation process for neighbouring macroblocks

Outputs of this process are

– mbAddrA: the address of the macroblock to the left of the current macroblock and its availability status and

– mbAddrB: the address of the macroblock above the current macroblock and its availability status.

mbAddrN (with N being A or B) is derived as follows.

– The difference of luma location (xD, yD) is set according to Table 6-2.

– The derivation process for neighbouring locations as specified in subclause 6.4.9 is invoked for luma locations with (xN, yN) equal to (xD, yD), and the output is assigned to mbAddrN.

6.4.8.2 Derivation process for neighbouring 8x8 luma block

Input to this process is an 8x8 luma block index luma8x8BlkIdx.

The luma8x8BlkIdx specifies the 8x8 luma blocks of a macroblock in a raster scan.

Outputs of this process are

– mbAddrA: either equal to CurrMbAddr or the address of the macroblock to the left of the current macroblock and its availability status,

– luma8x8BlkIdxA: the index of the 8x8 luma block to the left of the 8x8 block with index luma8x8BlkIdx and its availability status,

– mbAddrB: either equal to CurrMbAddr or the address of the macroblock above the current macroblock and its availability status,

– luma8x8BlkIdxB: the index of the 8x8 luma block above the 8x8 block with index luma8x8BlkIdx and its availability status.

mbAddrN and luma8x8BlkIdxN (with N being A or B) are derived as follows.

– The difference of luma location (xD, yD) is set according to Table 6-2.

– The luma location (xN, yN) is specified by

$$xN = (luma8x8BlkIdx \% 2) * 8 + xD \quad (6\text{-}21)$$

$$yN = (luma8x8BlkIdx / 2) * 8 + yD \quad (6\text{-}22)$$

– The derivation process for neighbouring locations as specified in subclause 6.4.9 is invoked for luma locations with (xN, yN) as the input and the output is assigned to mbAddrN and (xW, yW).

– The variable luma8x8BlkIdxN is derived as follows.

  – If mbAddrN is not available, luma8x8BlkIdxN is marked as not available.

  – Otherwise (mbAddrN is available), the 8x8 luma block in the macroblock mbAddrN covering the luma location (xW, yW) is assigned to luma8x8BlkIdxN.

6.4.8.3 Derivation process for neighbouring 4x4 luma blocks

Input to this process is a 4x4 luma block index luma4x4BlkIdx.

Outputs of this process are

– mbAddrA: either equal to CurrMbAddr or the address of the macroblock to the left of the current macroblock and its availability status,

– luma4x4BlkIdxA: the index of the 4x4 luma block to the left of the 4x4 block with index luma4x4BlkIdx and its availability status,

– mbAddrB: either equal to CurrMbAddr or the address of the macroblock above the current macroblock and its availability status,

– luma4x4BlkIdxB: the index of the 4x4 luma block above the 4x4 block with index luma4x4BlkIdx and its availability status.

mbAddrN and luma4x4BlkIdxN (with N being A or B) are derived as follows.

– The difference of luma location (xD, yD) is set according to Table 6-2.

– The inverse 4x4 luma block scanning process as specified in subclause 6.4.3 is invoked with luma4x4BlkIdx as the input and (x, y) as the output.

– The luma location (xN, yN) is specified by

$$xN = x + xD \quad (6\text{-}23)$$

$$yN = y + yD \quad (6\text{-}24)$$

– The derivation process for neighbouring locations as specified in subclause 6.4.9 is invoked for luma locations with (xN, yN) as the input and the output is assigned to mbAddrN and (xW, yW).

– The variable luma4x4BlkIdxN is derived as follows.

  – If mbAddrN is not available, luma4x4BlkIdxN is marked as not available.

  – Otherwise (mbAddrN is available), the 4x4 luma block in the macroblock mbAddrN covering the luma location (xW, yW) is assigned to luma4x4BlkIdxN.

6.4.8.4 Derivation process for neighbouring 4x4 chroma blocks

Input to this process is a 4x4 chroma block index chroma4x4BlkIdx.

Outputs of this process are

– mbAddrA (either equal to CurrMbAddr or the address of the macroblock to the left of the current macroblock) and its availability status,

– chroma4x4BlkIdxA (the index of the 4x4 chroma block to the left of the 4x4 chroma block with index chroma4x4BlkIdx) and its availability status,

– mbAddrB (either equal to CurrMbAddr or the address of the macroblock above the current macroblock) and its availability status,
– chroma4x4BlkIdxB (the index of the 4x4 chroma block above the 4x4 chroma block with index chroma4x4BlkIdx) and its availability status.

mbAddrN and chroma4x4BlkIdxN (with N being A or B) are derived as follows.

– The difference of chroma location (xD, yD) is set according to Table 6-2.
– Depending on chroma_format_idc, the position (x, y) of the upper-left sample of the 4x4 chroma block with index chroma4x4BlkIdx is derived as follows
  – If chroma_format_idc is equal to 1 or 2, the following applies

$$x = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0) \quad (6\text{-}25)$$

$$y = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1) \quad (6\text{-}26)$$

  – Otherwise (chroma_format_idc is equal to 3), the following applies

$$x = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0) \quad (6\text{-}27)$$

$$y = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1) \quad (6\text{-}28)$$

– The chroma location (xN, yN) is specified by

$$xN = x + xD \quad (6\text{-}29)$$

$$yN = y + yD \quad (6\text{-}30)$$

– The derivation process for neighbouring locations as specified in subclause 6.4.9 is invoked for chroma locations with (xN, yN) as the input and the output is assigned to mbAddrN and (xW, yW).
– The variable chroma4x4BlkIdxN is derived as follows.
  – If mbAddrN is not available, chroma4x4BlkIdxN is marked as not available.
  – Otherwise (mbAddrN is available), the 4x4 chroma block in the macroblock mbAddrN covering the chroma location (xW, yW) is assigned to chroma4x4BlkIdxN.

6.4.8.5 Derivation process for neighbouring partitions

Inputs to this process are

– a macroblock partition index mbPartIdx
– a current sub-macroblock type currSubMbType
– a sub-macroblock partition index subMbPartIdx

Outputs of this process are

– mbAddrA\mbPartIdxA\subMbPartIdxA: specifying the macroblock or sub-macroblock partition to the left of the current macroblock and its availability status, or the sub-macroblock partition CurrMbAddr\mbPartIdx\subMbPartIdx and its availability status,

– mbAddrB\mbPartIdxB\subMbPartIdxB: specifying the macroblock or sub-macroblock partition above the current macroblock and its availability status, or the sub-macroblock partition CurrMbAddr\mbPartIdx\subMbPartIdx and its availability status,

– mbAddrC\mbPartIdxC\subMbPartIdxC: specifying the macroblock or sub-macroblock partition to the right-above of the current macroblock and its availability status, or the sub-macroblock partition CurrMbAddr\mbPartIdx\subMbPartIdx and its availability status,

– mbAddrD\mbPartIdxD\subMbPartIdxD: specifying the macroblock or sub-macroblock partition to the left-above of the current macroblock and its availability status, or the sub-macroblock partition CurrMbAddr\mbPartIdx\subMbPartIdx and its availability status.

mbAddrN, mbPartIdxN, and subMbPartIdx (with N being A, B, C, or D) are derived as follows.

– The inverse macroblock partition scanning process as described in subclause 6.4.2.1 is invoked with mbPartIdx as the input and (x, y) as the output.

– The location of the upper-left luma sample inside a macroblock partition (xS, yS) is derived as follows.

  — If mb_type is equal to P_8x8, P_8x8ref0 or B_8x8, the inverse sub-macroblock partition scanning process as described in subclause 6.4.2.2 is invoked with subMbPartIdx as the input and (xS, yS) as the output.

  — Otherwise, (xS, yS) are set to (0, 0).

– The variable predPartWidth in Table 6-2 is specified as follows.

  – If mb_type is equal to P_Skip, B_Skip, or B_Direct_16x16, predPartWidth = 16.

  – Otherwise, if mb_type is equal to B_8x8, the following applies.

    – If currSubMbType is equal to B_Direct_8x8, predPartWidth = 16.

      NOTE 1 – When currSubMbType is equal to B_Direct_8x8 and direct_spatial_mv_pred_flag is equal to 1, the predicted motion vector is the predicted motion vector for the complete macroblock.

    – Otherwise, predPartWidth = SubMbPartWidth(sub_mb_type[mbPartIdx]).

  – Otherwise, if mb_type is equal to P_8x8 or P_8x8ref0, predPartWidth = SubMbPartWidth(sub_mb_type[mbPartIdx]).

  – Otherwise, predPartWidth = MbPartWidth(mb_type).

– The difference of luma location (xD, yD) is set according to Table 6-2.

– The neighbouring luma location (xN, yN) is specified by

$$xN = x + xS + xD \tag{6-31}$$

$$yN = y + yS + yD \tag{6-32}$$

– The derivation process for neighbouring locations as specified in subclause 6.4.9 is invoked for luma locations with (xN, yN) as the input and the output is assigned to mbAddrN and (xW, yW).

– Depending on mbAddrN, the following applies.

  – If mbAddrN is not available, the macroblock or sub-macroblock partition – mbAddrN\mbPartIdxN\subMbPartIdxN is marked as not available.

  – Otherwise (mbAddrN is available), the following applies.

    – The macroblock partition in the macroblock mbAddrN covering the luma location (xW, yW) is assigned to mbPartIdxN and the sub-macroblock partition inside the macroblock partition mbPartIdxN covering the sample (xW, yW) in the macroblock mbAddrN is assigned to subMbPartIdxN.

    – When the partition given by mbPartIdxN and subMbPartIdxN is not yet decoded, the macroblock partition mbPartIdxN and the sub-macroblock partition subMbPartIdxN are marked as not available.

    NOTE 2 – The latter condition is, for example, the case when mbPartIdx = 2, subMbPartIdx = 3, xD = 4, yD = -1, i.e., when neighbour C of the last 4x4 luma block of the third sub-macroblock is requested.

6.4.9 Derivation process for neighbouring locations

Input to this process is a luma or chroma location (xN, yN) expressed relative to the upper left corner of the current macroblock.

Outputs of this process are

– mbAddrN: either equal to CurrMbAddr or to the address of neighbouring macroblock that contains (xN, yN) and its availability status,

– (xW, yW): the location (xN, yN) expressed relative to the upper-left corner of the macroblock mbAddrN (rather than relative to the upper-left corner of the current macroblock).

Let maxW and maxH be variables specifying maximum values of the location components xN, xW, and yN, yW, respectively. maxW and maxH are derived as follows.

– If this process is invoked for neighbouring luma locations,

$$maxW = maxH = 16 \tag{6-33}$$

– Otherwise (this process is invoked for neighbouring chroma locations),

$$maxW = MbWidthC \tag{6-34}$$

$$maxH = MbHeightC \tag{6-35}$$

Depending on the variable MbaffFrameFlag, the neighbouring locations are derived as follows.

– If MbaffFrameFlag is equal to 0, the specification for neighbouring locations in fields and non-MBAFF frames as described in subclause 6.4.9.1 is applied.

– Otherwise (MbaffFrameFlag is equal to 1), the specification for neighbouring locations in MBAFF frames as described in subclause 6.4.9.2 is applied.

6.4.9.1 Specification for neighbouring locations in fields and non-MBAFF frames

The specifications in this subclause are applied when MbaffFrameFlag is equal to 0.

The derivation process for neighbouring macroblock addresses and their availability in subclause 6.4.6 is invoked with mbAddrA, mbAddrB, mbAddrC, and mbAddrD as well as their availability status as the output.

Table 6-3 specifies mbAddrN depending on (xN, yN).

Table 6-3 – Specification of mbAddrN

xN	yN	mbAddrN
< 0	< 0	mbAddrD
< 0	0 .. maxH - 1	mbAddrA
0 .. maxW - 1	< 0	mbAddrB
0 .. maxW - 1	0 .. maxH - 1	CurrMbAddr
> maxW - 1	< 0	mbAddrC
> maxW - 1	0 .. maxH - 1	not available
	> maxH - 1	not available

The neighbouring location (xW, yW) relative to the upper-left corner of the macroblock mbAddrN is derived as

$$xW = (xN + maxW) \% maxW \tag{6-36}$$

$$yW = (yN + maxH) \% maxH \tag{6-37}$$

6.4.9.2 Specification for neighbouring locations in MBAFF frames

The specifications in this subclause are applied when MbaffFrameFlag is equal to 1.

The derivation process for neighbouring macroblock addresses and their availability in subclause 6.4.7 is invoked with mbAddrA, mbAddrB, mbAddrC, and mbAddrD as well as their availability status as the output.

Table 6-4 specifies the macroblock addresses mbAddrN and yM in two ordered steps:

1. Specification of a macroblock address mbAddrX depending on (xN, yN) and the following variables:
   - The variable currMbFrameFlag is derived as follows.
     - If the macroblock with address CurrMbAddr is a frame macroblock, currMbFrameFlag is set equal to 1,
     - Otherwise (the macroblock with address CurrMbAddr is a field macroblock), currMbFrameFlag is set equal to 0.
   - The variable mbIsTopMbFlag is derived as follows.
     - If the macroblock with address CurrMbAddr is a top macroblock (CurrMbAddr % 2 is equal to 0), mbIsTopMbFlag is set equal to 1;
     - Otherwise (the macroblock with address CurrMbAddr is a bottom macroblock, CurrMbAddr % 2 is equal to 1), mbIsTopMbFlag is set equal to 0.
2. Depending on the availability of mbAddrX, the following applies.
   - If mbAddrX is not available, mbAddrN is marked as not available.
   - Otherwise (mbAddrX is available), mbAddrN is marked as available and Table 6-4 specifies mbAddrN and yM depending on (xN, yN), currMbFrameFlag, mbIsTopMbFlag, and the variable mbAddrXFrameFlag, which is derived as follows.
     - If the macroblock mbAddrX is a frame macroblock, mbAddrXFrameFlag is set equal to 1,
     - Otherwise (the macroblock mbAddrX is a field macroblock), mbAddrXFrameFlag is set equal to 0.

Unspecified values (na) of the above flags in Table 6-4 indicate that the value of the corresponding flag is not relevant for the current table rows.

Table 6-4 – Specification of mbAddrN and yM

xN	yN	currMbFrameFlag	mbIsTopMbFlag	mbAddrX	mbAddrXFrameFlag	additional condition	mbAddrN	yM
< 0	< 0	1	1	mbAddrD			mbAddrD + 1	yN
			0	mbAddrA	1		mbAddrA	yN
					0		mbAddrA + 1	(yN + maxH) >> 1
		0	1	mbAddrD	1		mbAddrD + 1	2*yN
					0		mbAddrD	yN
			0	mbAddrD			mbAddrD + 1	yN
< 0	0 .. maxH - 1	1	1	mbAddrA	1		mbAddrA	yN
					0	yN % 2 = = 0	mbAddrA	yN >> 1
						yN % 2 != 0	mbAddrA + 1	yN >> 1
			0	mbAddrA	1		mbAddrA + 1	yN
					0	yN % 2 = = 0	mbAddrA	(yN + maxH) >> 1
						yN % 2 != 0	mbAddrA + 1	(yN + maxH) >> 1
		0	1	mbAddrA	1	yN < (maxH / 2)	mbAddrA	yN <<1
						yN >= (maxH / 2)	mbAddrA + 1	(yN <<1) - maxH
					0		mbAddrA	yN
			0	mbAddrA	1	yN < (maxH / 2)	mbAddrA	(yN <<1) + 1
						yN >= (maxH / 2)	mbAddrA + 1	(yN <<1) + 1 – maxH
					0		mbAddrA + 1	yN
0 .. maxW – 1	< 0	1	1	mbAddrB			mbAddrB + 1	yN
			0	CurrMbAddr			CurrMbAddr - 1	yN
		0	1	mbAddrB	1		mbAddrB + 1	2 * yN
					0		mbAddrB	yN
			0	mbAddrB			mbAddrB + 1	yN
0 .. maxW – 1	0 .. maxH - 1			CurrMbAddr			CurrMbAddr	yN
> maxW – 1	<0	1	1	mbAddrC			mbAddrC + 1	yN
			0	not available			not available	na
		0	1	mbAddrC	1		mbAddrC + 1	2 * yN
					0		mbAddrC	yN
			0	mbAddrC			mbAddrC + 1	yN
> maxW – 1	0 .. maxH - 1			not available			not available	na
	> maxH - 1			not available			not available	na

The neighbouring luma location (xW, yW) relative to the upper-left corner of the macroblock mbAddrN is derived as

$$xW = (xN + maxW) \% maxW \quad (6\text{-}38)$$

$$yW = (yM + maxH) \% maxH \quad (6\text{-}39)$$

7 Syntax and semantics

7.1 Method of specifying syntax in tabular form

The syntax tables specify a superset of the syntax of all allowed bitstreams. Additional constraints on the syntax may be specified, either directly or indirectly, in other clauses.

NOTE – An actual decoder should implement means for identifying entry points into the bitstream and means to identify and handle non-conforming bitstreams. The methods for identifying and handling errors and other such situations are not specified here.

The following table lists examples of pseudo code used to describe the syntax. When **syntax_element** appears, it specifies that a syntax element is parsed from the bitstream and the bitstream pointer is advanced to the next position beyond the syntax element in the bitstream parsing process.

	C	Descriptor
/* A statement can be a syntax element with an associated syntax category and descriptor or can be an expression used to specify conditions for the existence, type, and quantity of syntax elements, as in the following two examples */		
syntax_element	3	ue(v)
conditioning statement		
/* A group of statements enclosed in curly brackets is a compound statement and is treated functionally as a single statement. */		
{		
statement		
statement		
…		
}		
/* A "while" structure specifies a test of whether a condition is true, and if true, specifies evaluation of a statement (or compound statement) repeatedly until the condition is no longer true */		
while(condition)		
statement		
/* A "do … while" structure specifies evaluation of a statement once, followed by a test of whether a condition is true, and if true, specifies repeated evaluation of the statement until the condition is no longer true */		
do		
statement		
while(condition)		
/* An "if … else" structure specifies a test of whether a condition is true, and if the condition is true, specifies evaluation of a primary statement, otherwise, specifies evaluation of an alternative statement. The "else" part of the structure and the associated alternative statement is omitted if no alternative statement evaluation is needed */		
if(condition)		
primary statement		
else		
alternative statement		
/* A "for" structure specifies evaluation of an initial statement, followed by a test of a condition, and if the condition is true, specifies repeated evaluation of a primary statement followed by a subsequent statement until the condition is no longer true. */		
for(initial statement; condition; subsequent statement)		
primary statement		

7.2 Specification of syntax functions, categories, and descriptors

The functions presented here are used in the syntactical description. These functions assume the existence of a bitstream pointer with an indication of the position of the next bit to be read by the decoding process from the bitstream.

byte_aligned( ) is specified as follows.

- If the current position in the bitstream is on a byte boundary, i.e., the next bit in the bitstream is the first bit in a byte, the return value of byte_aligned( ) is equal to TRUE.
- Otherwise, the return value of byte_aligned( ) is equal to FALSE.

more_data_in_byte_stream( ), which is used only in the byte stream NAL unit syntax structure specified in Annex B, is specified as follows.

- If more data follow in the byte stream, the return value of more_data_in_byte_stream( ) is equal to TRUE.
- Otherwise, the return value of more_data_in_byte_stream( ) is equal to FALSE.

more_rbsp_data( ) is specified as follows.

- If there is more data in an RBSP before rbsp_trailing_bits( ), the return value of more_rbsp_data( ) is equal to TRUE.
- Otherwise, the return value of more_rbsp_data( ) is equal to FALSE.

The method for enabling determination of whether there is more data in the RBSP is specified by the application (or in Annex B for applications that use the byte stream format).

more_rbsp_trailing_data( ) is specified as follows.

- If there is more data in an RBSP, the return value of more_rbsp_trailing_data( ) is equal to TRUE.
- Otherwise, the return value of more_rbsp_trailing_data( ) is equal to FALSE.

next_bits(n) provides the next bits in the bitstream for comparison purposes, without advancing the bitstream pointer. Provides a look at the next n bits in the bitstream with n being its argument. When used within the byte stream as specified in Annex B, next_bits(n) returns a value of 0 if fewer than n bits remain within the byte stream.

read_bits(n) reads the next n bits from the bitstream and advances the bitstream pointer by n bit positions. When n is equal to 0, read_bits(n) is specified to return a value equal to 0 and to not advance the bitstream pointer.

Categories (labelled in the table as **C**) specify the partitioning of slice data into at most three slice data partitions. Slice data partition A contains all syntax elements of category 2. Slice data partition B contains all syntax elements of category 3. Slice data partition C contains all syntax elements of category 4. The meaning of other category values is not specified. For some syntax elements, two category values, separated by a vertical bar, are used. In these cases, the category value to be applied is further specified in the text. For syntax structures used within other syntax structures, the categories of all syntax elements found within the included syntax structure are listed, separated by a vertical bar. A syntax element or syntax structure with category marked as "All" is present within all syntax structures that include that syntax element or syntax structure. For syntax structures used within other syntax structures, a numeric category value provided in a syntax table at the location of the inclusion of a syntax structure containing a syntax element with category marked as "All" is considered to apply to the syntax elements with category "All".

The following descriptors specify the parsing process of each syntax element. For some syntax elements, two descriptors, separated by a vertical bar, are used. In these cases, the left descriptors apply when entropy_coding_mode_flag is equal to 0 and the right descriptor applies when entropy_coding_mode_flag is equal to 1.

- ae(v): context-adaptive arithmetic entropy-coded syntax element. The parsing process for this descriptor is specified in subclause 9.3.
- b(8): byte having any pattern of bit string (8 bits). The parsing process for this descriptor is specified by the return value of the function read_bits(8).
- ce(v): context-adaptive variable-length entropy-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.2.
- f(n): fixed-pattern bit string using n bits written (from left to right) with the left bit first. The parsing process for this descriptor is specified by the return value of the function read_bits(n).
- i(n): signed integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the

return value of the function read_bits(n) interpreted as a two's complement integer representation with most significant bit written first.

– me(v): mapped Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1.

– se(v): signed integer Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1.

– te(v): truncated Exp-Golomb-coded syntax element with left bit first. The parsing process for this descriptor is specified in subclause 9.1.

– u(n): unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies in a manner dependent on the value of other syntax elements. The parsing process for this descriptor is specified by the return value of the function read_bits(n) interpreted as a binary representation of an unsigned integer with most significant bit written first.

– ue(v): unsigned integer Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in subclause 9.1.

7.3 Syntax in tabular form

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
for(i = 1; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */	All	f(8)
} else		
rbsp_byte[NumBytesInRBSP++ **]**	All	b(8)
}		
}		

7.3.2 Raw byte sequence payloads and RBSP trailing bits syntax

7.3.2.1 Sequence parameter set RBSP syntax

seq_parameter_set_rbsp( ) {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 || profile_idc == 110 || profile_idc == 122 || profile_idc == 144) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc == 3)		
residual_colour_transform_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < 8; i++) {		
seq_scaling_list_present_flag[i]	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		
scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	0	
else		
scaling_list(ScalingList8x8[i – 6], 64, UseDefaultScalingMatrix8x8Flag[i – 6])	0	
}		
}		
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type == 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)

frame_mbs_only_flag	0	u(1)
if(!frame_mbs_only_flag)		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if(frame_cropping_flag) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if(vui_parameters_present_flag)		
vui_parameters( )	0	
rbsp_trailing_bits( )	0	
}		

7.3.2.1.1 Scaling list syntax

scaling_list(scalingList, sizeOfScalingList, useDefaultScalingMatrixFlag) {	**C**	**Descriptor**
lastScale = 8		
nextScale = 8		
for(j = 0; j < sizeOfScalingList; j++) {		
if(nextScale != 0) {		
delta_scale	0 | 1	se(v)
nextScale = (lastScale + delta_scale + 256) % 256		
useDefaultScalingMatrixFlag = (j == 0 && nextScale == 0)		
}		
scalingList[j] = (nextScale == 0) ? lastScale : nextScale		
lastScale = scalingList[j]		
}		
}		

7.3.2.1.2 Sequence parameter set extension RBSP syntax

seq_parameter_set_extension_rbsp( ) {	**C**	**Descriptor**
seq_parameter_set_id	10	ue(v)
aux_format_idc	10	ue(v)
if(aux_format_idc != 0) {		
bit_depth_aux_minus8	10	ue(v)
alpha_incr_flag	10	u(1)
alpha_opaque_value	10	u(v)
alpha_transparent_value	10	u(v)
}		
additional_extension_flag	10	u(1)
rbsp_trailing_bits()	10	
}		

7.3.2.2 Picture parameter set RBSP syntax

pic_parameter_set_rbsp( ) {	C	Descriptor
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode_flag	1	u(1)
pic_order_present_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
slice_group_map_type	1	ue(v)
if(slice_group_map_type = = 0)		
for(iGroup = 0; iGroup <= num_slice_groups_minus1; iGroup++)		
run_length_minus1[iGroup **]**	1	ue(v)
else if(slice_group_map_type = = 2)		
for(iGroup = 0; iGroup < num_slice_groups_minus1; iGroup++) {		
top_left[iGroup **]**	1	ue(v)
bottom_right[iGroup **]**	1	ue(v)
}		
else if(slice_group_map_type = = 3 || slice_group_map_type = = 4 || slice_group_map_type = = 5) {		
slice_group_change_direction_flag	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
} else if(slice_group_map_type = = 6) {		
pic_size_in_map_units_minus1	1	ue(v)
for(i = 0; i <= pic_size_in_map_units_minus1; i++)		
slice_group_id[i **]**	1	u(v)
}		
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_idc	1	u(2)
pic_init_qp_minus26 /* relative to 26 */	1	se(v)
pic_init_qs_minus26 /* relative to 26 */	1	se(v)
chroma_qp_index_offset	1	se(v)
deblocking_filter_control_present_flag	1	u(1)
constrained_intra_pred_flag	1	u(1)
redundant_pic_cnt_present_flag	1	u(1)
if(more_rbsp_data( )) {		
transform_8x8_mode_flag	1	u(1)
pic_scaling_matrix_present_flag	1	u(1)
if(pic_scaling_matrix_present_flag)		
for(i = 0; i < 6 + 2* transform_8x8_mode_flag; i++) {		
pic_scaling_list_present_flag[i **]**	1	u(1)
if(pic_scaling_list_present_flag[i])		
if(i < 6)		

scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	1	
else		
scaling_list(ScalingList8x8[i – 6], 64, UseDefaultScalingMatrix8x8Flag[i – 6])	1	
}		
second_chroma_qp_index_offset	1	se(v)
}		
rbsp_trailing_bits( )	1	
}		

7.3.2.3 Supplemental enhancement information RBSP syntax

sei_rbsp( ) {	C	Descriptor
do		
sei_message( )	5	
while(more_rbsp_data( ))		
rbsp_trailing_bits( )	5	
}		

7.3.2.3.1 Supplemental enhancement information message syntax

sei_message( ) {	C	Descriptor
payloadType = 0		
while(next_bits(8) == 0xFF) {		
ff_byte /* equal to 0xFF */	5	f(8)
payloadType += 255		
}		
last_payload_type_byte	5	u(8)
payloadType += last_payload_type_byte		
payloadSize = 0		
while(next_bits(8) == 0xFF) {		
ff_byte /* equal to 0xFF */	5	f(8)
payloadSize += 255		
}		
last_payload_size_byte	5	u(8)
payloadSize += last_payload_size_byte		
sei_payload(payloadType, payloadSize)	5	
}		

7.3.2.4 Access unit delimiter RBSP syntax

access_unit_delimiter_rbsp( ) {	C	Descriptor
primary_pic_type	6	u(3)
rbsp_trailing_bits( )	6	
}		

7.3.2.5 End of sequence RBSP syntax

end_of_seq_rbsp( ) {	C	Descriptor
}		

7.3.2.6 End of stream RBSP syntax

end_of_stream_rbsp( ) {	C	Descriptor
}		

7.3.2.7 Filler data RBSP syntax

filler_data_rbsp( ) {	C	Descriptor
while(next_bits(8) == 0xFF)		
ff_byte /* equal to 0xFF */	9	f(8)
rbsp_trailing_bits( )	9	
}		

7.3.2.8 Slice layer without partitioning RBSP syntax

slice_layer_without_partitioning_rbsp( ) {	C	Descriptor
slice_header( )	2	
slice_data( ) /* all categories of slice_data( ) syntax */	2 | 3 | 4	
rbsp_slice_trailing_bits( )	2	
}		

7.3.2.9 Slice data partition RBSP syntax

7.3.2.9.1 Slice data partition A RBSP syntax

slice_data_partition_a_layer_rbsp( ) {	C	Descriptor
slice_header( )	2	
slice_id	All	ue(v)
slice_data( ) /* only category 2 parts of slice_data( ) syntax */	2	
rbsp_slice_trailing_bits( )	2	
}		

7.3.2.9.2 Slice data partition B RBSP syntax

slice_data_partition_b_layer_rbsp( ) {	C	Descriptor
slice_id	All	ue(v)
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	All	ue(v)
slice_data( ) /* only category 3 parts of slice_data( ) syntax */	3	
rbsp_slice_trailing_bits( )	3	
}		

7.3.2.9.3 Slice data partition C RBSP syntax

slice_data_partition_c_layer_rbsp( ) {	C	Descriptor
slice_id	All	ue(v)
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	All	ue(v)
slice_data( ) /* only category 4 parts of slice_data( ) syntax */	4	
rbsp_slice_trailing_bits( )	4	
}		

7.3.2.10 RBSP slice trailing bits syntax

rbsp_slice_trailing_bits( ) {	C	Descriptor
rbsp_trailing_bits( )	All	
if(entropy_coding_mode_flag)		
while(more_rbsp_trailing_data( ))		
cabac_zero_word /* equal to 0x0000 */	All	f(16)
}		

7.3.2.11 RBSP trailing bits syntax

rbsp_trailing_bits( ) {	C	Descriptor
rbsp_stop_one_bit /* equal to 1 */	All	f(1)
while(!byte_aligned( ))		
rbsp_alignment_zero_bit /* equal to 0 */	All	f(1)
}		

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type = = 5)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type = = 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type = = 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0 **]**	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1 **]**	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type = = B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type = = P | | slice_type = = SP | | slice_type = = B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type = = B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
ref_pic_list_reordering( )	2	
if((weighted_pred_flag && (slice_type = = P | | slice_type = = SP)) | | (weighted_bipred_idc = = 1 && slice_type = = B))		
pred_weight_table( )	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking( )	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type = = SP | | slice_type = = SI) {		
if(slice_type = = SP)		
sp_for_switch_flag	2	u(1)

slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

7.3.3.1 Reference picture list reordering syntax

ref_pic_list_reordering( ) {	C	Descriptor
if(slice_type != I && slice_type != SI) {		
ref_pic_list_reordering_flag_l0	2	u(1)
if(ref_pic_list_reordering_flag_l0)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 || reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
if(slice_type == B) {		
ref_pic_list_reordering_flag_l1	2	u(1)
if(ref_pic_list_reordering_flag_l1)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 || reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
}		

7.3.3.2 Prediction weight table syntax

pred_weight_table( ) {	C	Descriptor
luma_log2_weight_denom	2	ue(v)
if(chroma_format_idc != 0)		
chroma_log2_weight_denom	2	ue(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {		
luma_weight_l0_flag	2	u(1)
if(luma_weight_l0_flag) {		
luma_weight_l0[i]	2	se(v)
luma_offset_l0[i]	2	se(v)
}		
if (chroma_format_idc != 0) {		
chroma_weight_l0_flag	2	u(1)
if(chroma_weight_l0_flag)		
for(j =0; j < 2; j++) {		
chroma_weight_l0[i][j]	2	se(v)
chroma_offset_l0[i][j]	2	se(v)
}		
}		
}		
if(slice_type = = B)		
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {		
luma_weight_l1_flag	2	u(1)
if(luma_weight_l1_flag) {		
luma_weight_l1[i]	2	se(v)
luma_offset_l1[i]	2	se(v)
}		
if(chroma_format_idc != 0) {		
chroma_weight_l1_flag	2	u(1)
if(chroma_weight_l1_flag)		
for(j = 0; j < 2; j++) {		
chroma_weight_l1[i][j]	2	se(v)
chroma_offset_l1[i][j]	2	se(v)
}		
}		
}		
}		

7.3.3.3 Decoded reference picture marking syntax

dec_ref_pic_marking( ) {	C	Descriptor
if(nal_unit_type == 5) {		
no_output_of_prior_pics_flag	2 | 5	u(1)
long_term_reference_flag	2 | 5	u(1)
} else {		
adaptive_ref_pic_marking_mode_flag	2 | 5	u(1)
if(adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	2 | 5	ue(v)
if(memory_management_control_operation == 1 || memory_management_control_operation == 3)		
difference_of_pic_nums_minus1	2 | 5	ue(v)
if(memory_management_control_operation == 2)		
long_term_pic_num	2 | 5	ue(v)
if(memory_management_control_operation == 3 || memory_management_control_operation == 6)		
long_term_frame_idx	2 | 5	ue(v)
if(memory_management_control_operation == 4)		
max_long_term_frame_idx_plus1	2 | 5	ue(v)
} while(memory_management_control_operation != 0)		
}		
}		

7.3.4 Slice data syntax

slice_data( ) {	C	Descriptor
if(entropy_coding_mode_flag)		
while(!byte_aligned( ))		
cabac_alignment_one_bit	2	f(1)
CurrMbAddr = first_mb_in_slice * (1 + MbaffFrameFlag)		
moreDataFlag = 1		
prevMbSkipped = 0		
do {		
if(slice_type != I && slice_type != SI)		
if(!entropy_coding_mode_flag) {		
mb_skip_run	2	ue(v)
prevMbSkipped = (mb_skip_run > 0)		
for(i=0; i<mb_skip_run; i++)		
CurrMbAddr = NextMbAddress(CurrMbAddr)		
moreDataFlag = more_rbsp_data( )		
} else {		
mb_skip_flag	2	ae(v)
moreDataFlag = !mb_skip_flag		
}		
if(moreDataFlag) {		
if(MbaffFrameFlag && (CurrMbAddr % 2 == 0 || (CurrMbAddr % 2 == 1 && prevMbSkipped)))		
mb_field_decoding_flag	2	u(1) | ae(v)
macroblock_layer( )	2 | 3 | 4	
}		
if(!entropy_coding_mode_flag)		
moreDataFlag = more_rbsp_data( )		
else {		
if(slice_type != I && slice_type != SI)		
prevMbSkipped = mb_skip_flag		
if(MbaffFrameFlag && CurrMbAddr % 2 == 0)		
moreDataFlag = 1		
else {		
end_of_slice_flag	2	ae(v)
moreDataFlag = !end_of_slice_flag		
}		
}		
CurrMbAddr = NextMbAddress(CurrMbAddr)		
} while(moreDataFlag)		
}		

7.3.5 Macroblock layer syntax

macroblock_layer( ) {	C	Descriptor
mb_type	2	ue(v) | ae(v)
if(mb_type == I_PCM) {		
while(!byte_aligned( ))		
pcm_alignment_zero_bit	2	f(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i **]**	2	u(v)
for(i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i **]**	2	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		
if(mb_type != I_NxN && MbPartPredMode(mb_type, 0) != Intra_16x16 && NumMbPart(mb_type) == 4) {		
sub_mb_pred(mb_type)	2	
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8) {		
if(NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else if(!direct_8x8_inference_flag)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else {		
if(transform_8x8_mode_flag && mb_type == I_NxN)		
transform_size_8x8_flag	2	u(1) | ae(v)
mb_pred(mb_type)	2	
}		
if(MbPartPredMode(mb_type, 0) != Intra_16x16) {		
coded_block_pattern	2	me(v) | ae(v)
if(CodedBlockPatternLuma > 0 && transform_8x8_mode_flag && mb_type != I_NxN && noSubMbPartSizeLessThan8x8Flag && (mb_type != B_Direct_16x16 || direct_8x8_inference_flag))		
transform_size_8x8_flag	2	u(1) | ae(v)
}		
if(CodedBlockPatternLuma > 0 || CodedBlockPatternChroma > 0 || MbPartPredMode(mb_type, 0) == Intra_16x16) {		
mb_qp_delta	2	se(v) | ae(v)
residual( )	3 | 4	
}		
}		
}		

7.3.5.1 Macroblock prediction syntax

mb_pred(mb_type) {	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4 || MbPartPredMode(mb_type, 0) == Intra_8x8 || MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx **]**	2	u(1) | ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx **]**	2	u(3) | ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag[luma8x8BlkIdx **]**	2	u(1) | ae(v)
if(!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode[luma8x8BlkIdx **]**	2	u(3) | ae(v)
}		
if(chroma_format_idc != 0)		
intra_chroma_pred_mode	2	ue(v) | ae(v)
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
ref_idx_l0[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
ref_idx_l1[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode (mb_type, mbPartIdx) != Pred_L1)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx **][** 0 **][** compIdx **]**	2	se(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx **][** 0 **][** compIdx **]**	2	se(v) | ae(v)
}		
}		

7.3.5.2 Sub-macroblock prediction syntax

sub_mb_pred(mb_type) {	C	Descriptor
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
sub_mb_type[mbPartIdx **]**	2	ue(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag) && mb_type != P_8x8ref0 && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
ref_idx_l0[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag) && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
ref_idx_l1[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx **][** subMbPartIdx **][** compIdx **]**	2	se(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx **][** subMbPartIdx **][** compIdx **]**	2	se(v) | ae(v)
}		

7.3.5.3 Residual data syntax

residual( ) {	C	Descriptor
if(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
else		
residual_block = residual_block_cabac		
if(MbPartPredMode(mb_type, 0) = = Intra_16x16)		
residual_block(Intra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
if(!transform_size_8x8_flag || !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) = = Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
else		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 | 4	
else if(MbPartPredMode(mb_type, 0) = = Intra_16x16)		
for(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] = LumaLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(LumaLevel8x8[i8x8], 64)	3 | 4	
else		
for(i = 0; i < 64; i++)		
LumaLevel8x8[i8x8][i] = 0		
if(chroma_format_idc != 0) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
if(CodedBlockPatternChroma & 3) /* chroma DC residual present */		
residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8)	3 | 4	
else		
for(i = 0; i < 4 * NumC8x8; i++)		
ChromaDCLevel[iCbCr][i] = 0		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
for(i8x8 = 0; i8x8 < NumC8x8; i8x8++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
if(CodedBlockPatternChroma & 2) /* chroma AC residual present */		
residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4], 15)	3 | 4	
else		
for(i = 0; i < 15; i++)		
ChromaACLevel[iCbCr][i8x8*4+i4x4][i] = 0		
}		

7.3.5.3.1 Residual block CAVLC syntax

residual_block_cavlc(coeffLevel, maxNumCoeff) {	C	Descriptor
for(i = 0; i < maxNumCoeff; i++)		
coeffLevel[i] = 0		
coeff_token	3 | 4	ce(v)
if(TotalCoeff(coeff_token) > 0) {		
if(TotalCoeff(coeff_token) > 10 && TrailingOnes(coeff_token) < 3)		
suffixLength = 1		
else		
suffixLength = 0		
for(i = 0; i < TotalCoeff(coeff_token); i++)		
if(i < TrailingOnes(coeff_token)) {		
trailing_ones_sign_flag	3 | 4	u(1)
level[i] = 1 – 2 * trailing_ones_sign_flag		
} else {		
level_prefix	3 | 4	ce(v)
levelCode = (Min(15, level_prefix) << suffixLength)		
if(suffixLength > 0 || level_prefix >= 14) {		
level_suffix	3 | 4	u(v)
levelCode += level_suffix		
}		
if(level_prefix >= 15 && suffixLength == 0)		
levelCode += 15		
if(level_prefix >= 16)		
levelCode += (1 << (level_prefix – 3)) – 4096		
if(i == TrailingOnes(coeff_token) && TrailingOnes(coeff_token) < 3)		
levelCode += 2		
if(levelCode % 2 == 0)		
level[i] = (levelCode + 2) >> 1		
else		
level[i] = (–levelCode – 1) >> 1		
if(suffixLength == 0)		
suffixLength = 1		
if(Abs(level[i]) > (3 << (suffixLength – 1)) && suffixLength < 6)		
suffixLength++		
}		
if(TotalCoeff(coeff_token) < maxNumCoeff) {		
total_zeros	3 | 4	ce(v)
zerosLeft = total_zeros		
} else		
zerosLeft = 0		
for(i = 0; i < TotalCoeff(coeff_token) – 1; i++) {		
if(zerosLeft > 0) {		
run_before	3 | 4	ce(v)
run[i] = run_before		
} else		
run[i] = 0		

zerosLeft = zerosLeft – run[i]		
}		
run[TotalCoeff(coeff_token) – 1] = zerosLeft		
coeffNum = -1		
for(i = TotalCoeff(coeff_token) – 1; i >= 0; i--) {		
coeffNum += run[i] + 1		
coeffLevel[coeffNum] = level[i]		
}		
}		
}		

7.3.5.3.2 Residual block CABAC syntax

residual_block_cabac(coeffLevel, maxNumCoeff) {	C	Descriptor
if(maxNumCoeff = = 64)		
coded_block_flag = 1		
else		
coded_block_flag	3 | 4	ae(v)
if(coded_block_flag) {		
numCoeff = maxNumCoeff		
i = 0		
do {		
significant_coeff_flag[i **]**	3 | 4	ae(v)
if(significant_coeff_flag[i]) {		
last_significant_coeff_flag[i **]**	3 | 4	ae(v)
if(last_significant_coeff_flag[i]) {		
numCoeff = i + 1		
for(j = numCoeff; j < maxNumCoeff; j++)		
coeffLevel[j] = 0		
}		
}		
i++		
} while(i < numCoeff - 1)		
coeff_abs_level_minus1[numCoeff - 1 **]**	3 | 4	ae(v)
coeff_sign_flag[numCoeff - 1 **]**	3 | 4	ae(v)
coeffLevel[numCoeff - 1] = (coeff_abs_level_minus1[numCoeff – 1] + 1) * (1 – 2 * coeff_sign_flag[numCoeff – 1])		
for(i = numCoeff - 2; i >= 0; i--)		
if(significant_coeff_flag[i]) {		
coeff_abs_level_minus1[i **]**	3 | 4	ae(v)
coeff_sign_flag[i **]**	3 | 4	ae(v)
coeffLevel[i] = (coeff_abs_level_minus1[i] + 1) * (1 – 2 * coeff_sign_flag[i])		
} else		
coeffLevel[i] = 0		
} else		
for(i = 0; i < maxNumCoeff; i++)		
coeffLevel[i] = 0		
}		

7.4 Semantics

Semantics associated with the syntax structures and with the syntax elements within these structures are specified in this subclause. When the semantics of a syntax element are specified using a table or a set of tables, any values that are not specified in the table(s) shall not be present in the bitstream unless otherwise specified in this Recommendation | International Standard.

7.4.1 NAL unit semantics

NOTE 1 – The VCL is specified to efficiently represent the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance on a variety of communication channels or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and byte stream is identical except that each NAL unit can be preceded by a start code prefix and extra padding bytes in the byte stream format.

NumBytesInNALunit specifies the size of the NAL unit in bytes. This value is required for decoding of the NAL unit. Some form of demarcation of NAL unit boundaries is necessary to enable inference of NumBytesInNALunit. One such demarcation method is specified in Annex B for the byte stream format. Other methods of demarcation may be specified outside of this Recommendation | International Standard.

forbidden_zero_bit shall be equal to 0.

nal_ref_idc not equal to 0 specifies that the content of the NAL unit contains a sequence parameter set or a picture parameter set or a slice of a reference picture or a slice data partition of a reference picture.

nal_ref_idc equal to 0 for a NAL unit containing a slice or slice data partition indicates that the slice or slice data partition is part of a non-reference picture.

nal_ref_idc shall not be equal to 0 for sequence parameter set or sequence parameter set extension or picture parameter set NAL units. When nal_ref_idc is equal to 0 for one slice or slice data partition NAL unit of a particular picture, it shall be equal to 0 for all slice and slice data partition NAL units of the picture.

nal_ref_idc shall not be equal to 0 for IDR NAL units, i.e., NAL units with nal_unit_type equal to 5.

nal_ref_idc shall be equal to 0 for all NAL units having nal_unit_type equal to 6, 9, 10, 11, or 12.

nal_unit_type specifies the type of RBSP data structure contained in the NAL unit as specified in Table 7-1. VCL NAL units are specified as those NAL units having nal_unit_type equal to 1 to 5, inclusive. All remaining NAL units are called non-VCL NAL units.

The column marked "C" in Table 7-1 lists the categories of the syntax elements that may be present in the NAL unit. In addition, syntax elements with syntax category "All" may be present, as determined by the syntax and semantics of the RBSP data structure. The presence or absence of any syntax elements of a particular listed category is determined from the syntax and semantics of the associated RBSP data structure. nal_unit_type shall not be equal to 3 or 4 unless at least one syntax element is present in the RBSP data structure having a syntax element category value equal to the value of nal_unit_type and not categorized as "All".

Table 7-1 – NAL unit type codes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C
0	Unspecified	
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3
6	Supplemental enhancement information (SEI) sei_rbsp( )	5
7	Sequence parameter set seq_parameter_set_rbsp( )	0
8	Picture parameter set pic_parameter_set_rbsp( )	1
9	Access unit delimiter access_unit_delimiter_rbsp( )	6
10	End of sequence end_of_seq_rbsp( )	7
11	End of stream end_of_stream_rbsp( )	8
12	Filler data filler_data_rbsp( )	9
13	Sequence parameter set extension seq_parameter_set_extension_rbsp( )	10
14..18	Reserved	
19	Coded slice of an auxiliary coded picture without partitioning slice_layer_without_partitioning_rbsp( )	2, 3, 4
20..23	Reserved	
24..31	Unspecified	

NAL units having nal_unit_type equal to 13 and 19 may be discarded by decoders without affecting the decoding process for NAL units having nal_unit_type not equal to 13 or 19 and without affecting conformance to this Recommendation | International Standard.

NAL units that use nal_unit_type equal to 0 or in the range of 24..31, inclusive, shall not affect the decoding process specified in this Recommendation | International Standard.

NOTE 2 – NAL unit types 0 and 24..31 may be used as determined by the application. No decoding process for these values of nal_unit_type is specified in this Recommendation | International Standard.

Decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units that use reserved values of nal_unit_type.

NOTE 3 – This requirement allows future definition of compatible extensions to this Recommendation | International Standard.

In the text, coded slice NAL unit collectively refers to a coded slice of a non-IDR picture NAL unit or to a coded slice of an IDR picture NAL unit.

When the value of nal_unit_type is equal to 5 for a NAL unit containing a slice of a coded picture, the value of nal_unit_type shall be 5 in all other VCL NAL units of the same coded picture. Such a picture is referred to as an IDR picture.

NOTE 4 – Slice data partitioning cannot be used for IDR pictures.

rbsp_byte[i **]** is the i-th byte of an RBSP. An RBSP is specified as an ordered sequence of bytes as follows.

The RBSP contains an SODB as follows.

– If the SODB is empty (i.e., zero bits in length), the RBSP is also empty.

– Otherwise, the RBSP contains the SODB as follows.

1) The first byte of the RBSP contains the (most significant, left-most) eight bits of the SODB; the next byte of the RBSP shall contain the next eight bits of the SODB, etc., until fewer than eight bits of the SODB remain.

2) rbsp_trailing_bits( ) are present after the SODB as follows:

i) The first (most significant, left-most) bits of the final RBSP byte contains the remaining bits of the SODB, (if any)

ii) The next bit consists of a single rbsp_stop_one_bit equal to 1, and

iii) When the rbsp_stop_one_bit is not the last bit of a byte-aligned byte, one or more rbsp_alignment_zero_bit is present to result in byte alignment.

3) One or more cabac_zero_word 16-bit syntax elements equal to 0x0000 may be present in some RBSPs after the rbsp_trailing_bits( ) at the end of the RBSP.

Syntax structures having these RBSP properties are denoted in the syntax tables using an "_rbsp" suffix. These structures shall be carried within NAL units as the content of the rbsp_byte[i] data bytes. The association of the RBSP syntax structures to the NAL units shall be as specified in Table 7-1.

NOTE 5 – When the boundaries of the RBSP are known, the decoder can extract the SODB from the RBSP by concatenating the bits of the bytes of the RBSP and discarding the rbsp_stop_one_bit, which is the last (least significant, right-most) bit equal to 1, and discarding any following (less significant, farther to the right) bits that follow it, which are equal to 0. The data necessary for the decoding process is contained in the SODB part of the RBSP.

emulation_prevention_three_byte is a byte equal to 0x03. When an emulation_prevention_three_byte is present in the NAL unit, it shall be discarded by the decoding process.

The last byte of the NAL unit shall not be equal to 0x00.

Within the NAL unit, the following three-byte sequences shall not occur at any byte-aligned position:

– 0x000000

– 0x000001

– 0x000002

Within the NAL unit, any four-byte sequence that starts with 0x000003 other than the following sequences shall not occur at any byte-aligned position:

– 0x00000300

– 0x00000301

– 0x00000302

– 0x00000303

NOTE 6 – When nal_unit_type is equal to 0, particular care must be exercised in the design of encoders to avoid the presence of the above-listed three-byte and four-byte patterns at the beginning of the NAL unit syntax structure, as the syntax element emulation_prevention_three_byte cannot be the third byte of a NAL unit.

7.4.1.1 Encapsulation of an SODB within an RBSP (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

The form of encapsulation of an SODB within an RBSP and the use of the emulation_prevention_three_byte for encapsulation of an RBSP within a NAL unit is specified for the following purposes:

– to prevent the emulation of start codes within NAL units while allowing any arbitrary SODB to be represented within a NAL unit,

– to enable identification of the end of the SODB within the NAL unit by searching the RBSP for the rbsp_stop_one_bit starting at the end of the RBSP, and

– to enable a NAL unit to have a size larger than that of the SODB under some circumstances (using one or more cabac_zero_word).

The encoder can produce a NAL unit from an RBSP by the following procedure:

The RBSP data is searched for byte-aligned bits of the following binary patterns:

'00000000 00000000 000000xx' (where xx represents any 2 bit pattern: 00, 01, 10, or 11),

and a byte equal to 0x03 is inserted to replace these bit patterns with the patterns

'00000000 00000000 00000011 000000xx',

and finally, when the last byte of the RBSP data is equal to 0x00 (which can only occur when the RBSP ends in a cabac_zero_word), a final byte equal to 0x03 is appended to the end of the data.

The resulting sequence of bytes is then prefixed with the first byte of the NAL unit containing the indication of the type of RBSP data structure it contains. This results in the construction of the entire NAL unit.

This process can allow any SODB to be represented in a NAL unit while ensuring that

– no byte-aligned start code prefix is emulated within the NAL unit, and

– no sequence of 8 zero-valued bits followed by a start code prefix, regardless of byte-alignment, is emulated within the NAL unit.

7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences

This subclause specifies constraints on the order of NAL units in the bitstream. Any order of NAL units in the bitstream obeying these constraints is referred to in the text as the decoding order of NAL units. Within a NAL unit, the syntax in subclauses 7.3, D.1, and E.1 specifies the decoding order of syntax elements. Decoders conforming to this Recommendation | International Standard shall be capable of receiving NAL units and their syntax elements in decoding order.

7.4.1.2.1 Order of sequence and picture parameter set RBSPs and their activation

NOTE 1 – The sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. Sequence and picture parameter sets may, in some applications, be conveyed "out-of-band" using a reliable transport mechanism.

A picture parameter set RBSP includes parameters that can be referred to by the coded slice NAL units or coded slice data partition A NAL units of one or more coded pictures. Each picture parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one picture parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular picture parameter set RBSP results in the deactivation of the previously-active picture parameter set RBSP (if any).

When a picture parameter set RBSP (with a particular value of pic_parameter_set_id) is not active and it is referred to by a coded slice NAL unit or coded slice data partition A NAL unit (using that value of pic_parameter_set_id), it is activated. This picture parameter set RBSP is called the active picture parameter set RBSP until it is deactivated by the activation of another picture parameter set RBSP. A picture parameter set RBSP, with that particular value of pic_parameter_set_id, shall be available to the decoding process prior to its activation.

Any picture parameter set NAL unit containing the value of pic_parameter_set_id for the active picture parameter set RBSP shall have the same content as that of the active picture parameter set RBSP unless it follows the last VCL NAL unit of a coded picture and precedes the first VCL NAL unit of another coded picture.

A sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. Each sequence parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one sequence parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular sequence parameter set RBSP results in the deactivation of the previously-active sequence parameter set RBSP (if any).

When a sequence parameter set RBSP (with a particular value of seq_parameter_set_id) is not already active and it is referred to by activation of a picture parameter set RBSP (using that value of seq_parameter_set_id) or is referred to by an SEI NAL unit containing a buffering period SEI message (using that value of seq_parameter_set_id), it is activated. This sequence parameter set RBSP is called the active sequence parameter set RBSP until it is deactivated by the activation of another sequence parameter set RBSP. A sequence parameter set RBSP, with that particular value of seq_parameter_set_id, shall be available to the decoding process prior to its activation. An activated sequence parameter set RBSP shall remain active for the entire coded video sequence.

NOTE 2 – Because an IDR access unit begins a new coded video sequence and an activated sequence parameter set RBSP must remain active for the entire coded video sequence, a sequence parameter set RBSP can only be activated by a buffering period SEI message when the buffering period SEI message is part of an IDR access unit.

Any sequence parameter set NAL unit containing the value of seq_parameter_set_id for the active sequence parameter set RBSP shall have the same content as that of the active sequence parameter set RBSP unless it follows the last access unit of a coded video sequence and precedes the first VCL NAL unit and the first SEI NAL unit containing a buffering period SEI message (when present) of another coded video sequence.

NOTE 3 – If picture parameter set RBSP or sequence parameter set RBSP are conveyed within the bitstream, these constraints impose an order constraint on the NAL units that contain the picture parameter set RBSP or sequence parameter set RBSP, respectively. Otherwise (picture parameter set RBSP or sequence parameter set RBSP are conveyed by other means not specified in this Recommendation | International Standard), they must be available to the decoding process in a timely fashion such that these constraints are obeyed.

When present, a sequence parameter set extension RBSP includes parameters having a similar function to those of a sequence parameter set RBSP. For purposes of establishing constraints on the syntax elements of the sequence parameter set extension RBSP and for purposes of determining activation of a sequence parameter set extension RBSP, the sequence parameter set extension RBSP shall be considered part of the preceding sequence parameter set RBSP with the same value of seq_parameter_set_id. When a sequence parameter set RBSP is present that is not followed by a sequence parameter set extension RBSP with the same value of seq_parameter_set_id prior to the activation of the sequence parameter set RBSP, the sequence parameter set extension RBSP and its syntax elements shall be considered not present for the active sequence parameter set RBSP.

All constraints that are expressed on the relationship between the values of the syntax elements (and the values of variables derived from those syntax elements) in sequence parameter sets and picture parameter sets and other syntax elements are expressions of constraints that apply only to the active sequence parameter set and the active picture parameter set. If any sequence parameter set RBSP is present that is not activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream. If any picture parameter set RBSP is present that is not ever activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream.

During operation of the decoding process (see clause 8), the values of parameters of the active picture parameter set and the active sequence parameter set shall be considered in effect. For interpretation of SEI messages, the values of the parameters of the picture parameter set and sequence parameter set that are active for the operation of the decoding process for the VCL NAL units of the primary coded picture in the same access unit shall be considered in effect unless otherwise specified in the SEI message semantics.

7.4.1.2.2 Order of access units and association to coded video sequences

A bitstream conforming to this Recommendation | International Standard consists of one or more coded video sequences.

A coded video sequence consists of one or more access units. The order of NAL units and coded pictures and their association to access units is described in subclause 7.4.1.2.3.

The first access unit of each coded video sequence is an IDR access unit. All subsequent access units in the coded video sequence are non-IDR access units.

The values of picture order count for the coded pictures in consecutive access units in decoding order containing non-reference pictures shall be non-decreasing.

When present, an access unit following an access unit that contains an end of sequence NAL unit shall be an IDR access unit.

When an SEI NAL unit contains data that pertain to more than one access unit (for example, when the SEI NAL unit has a coded video sequence as its scope), it shall be contained in the first access unit to which it applies.

When an end of stream NAL unit is present in an access unit, this access unit shall be the last access unit in the bitstream and the end of stream NAL unit shall be the last NAL unit in that access unit.

7.4.1.2.3 Order of NAL units and coded pictures and association to access units

An access unit consists of one primary coded picture, zero or more corresponding redundant coded pictures, and zero or more non-VCL NAL units. The association of VCL NAL units to primary or redundant coded pictures is described in subclause 7.4.1.2.5.

The first access unit in the bitstream starts with the first NAL unit of the bitstream.

The first of any of the following NAL units after the last VCL NAL unit of a primary coded picture specifies the start of a new access unit.

– access unit delimiter NAL unit (when present)

– sequence parameter set NAL unit (when present)

– picture parameter set NAL unit (when present)

– SEI NAL unit (when present)

– NAL units with nal_unit_type in the range of 14 to 18, inclusive

– first VCL NAL unit of a primary coded picture (always present)

The constraints for the detection of the first VCL NAL unit of a primary coded picture are specified in subclause 7.4.1.2.4.

The following constraints shall be obeyed by the order of the coded pictures and non-VCL NAL units within an access unit.

– When an access unit delimiter NAL unit is present, it shall be the first NAL unit. There shall be at most one access unit delimiter NAL unit in any access unit.

– When any SEI NAL units are present, they shall precede the primary coded picture.

– When an SEI NAL unit containing a buffering period SEI message is present, the buffering period SEI message shall be the first SEI message payload of the first SEI NAL unit in the access unit

– The primary coded picture shall precede the corresponding redundant coded pictures.

– When redundant coded pictures are present, they shall be ordered in ascending order of the value of redundant_pic_cnt.

– When a sequence parameter set extension NAL unit is present, it shall be the next NAL unit after a sequence parameter set NAL unit having the same value of seq_parameter_set_id as in the sequence parameter set extension NAL unit.

– When one or more coded slice of an auxiliary coded picture without partitioning NAL units is present, they shall follow the primary coded picture and all redundant coded pictures (if any).

– When an end of sequence NAL unit is present, it shall follow the primary coded picture and all redundant coded pictures (if any) and all coded slice of an auxiliary coded picture without partitioning NAL units (if any).

– When an end of stream NAL unit is present, it shall be the last NAL unit.

– NAL units having nal_unit_type equal to 0, 12, or in the range of 20 to 31, inclusive, shall not precede the first VCL NAL unit of the primary coded picture.

NOTE 1 – Sequence parameter set NAL units or picture parameter set NAL units may be present in an access unit, but cannot follow the last VCL NAL unit of the primary coded picture within the access unit, as this condition would specify the start of a new access unit.

NOTE 2 – When a NAL unit having nal_unit_type equal to 7 or 8 is present in an access unit, it may or may not be referred to in the coded pictures of the access unit in which it is present, and may be referred to in coded pictures of subsequent access units.

The structure of access units not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive, is shown in Figure 7-1.

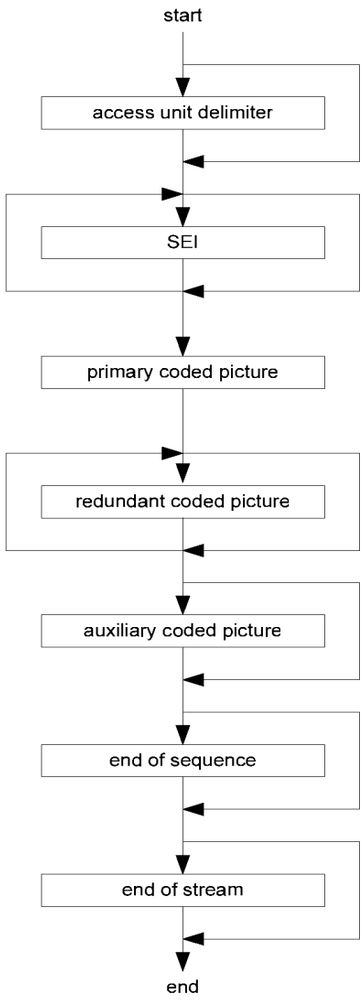


Figure 7-1 – Structure of an access unit not containing any NAL units with nal_unit_type equal to 0, 7, 8, or in the range of 12 to 18, inclusive, or in the range of 20 to 31, inclusive

7.4.1.2.4 Detection of the first VCL NAL unit of a primary coded picture

This subclause specifies constraints on VCL NAL unit syntax that are sufficient to enable the detection of the first VCL NAL unit of each primary coded picture.

Any coded slice NAL unit or coded slice data partition A NAL unit of the primary coded picture of the current access unit shall be different from any coded slice NAL unit or coded slice data partition A NAL unit of the primary coded picture of the previous access unit in one or more of the following ways.

– frame_num differs in value. The value of frame_num used to test this condition is the value of frame_num that appears in the syntax of the slice header, regardless of whether that value is inferred to have been equal to 0 for subsequent use in the decoding process due to the presence of memory_management_control_operation equal to 5.

> NOTE 1 – A consequence of the above statement is that a primary coded picture having frame_num equal to 1 cannot contain a memory_management_control_operation equal to 5 unless some other condition listed below is fulfilled for the next primary coded picture that follows after it (if any).

– pic_parameter_set_id differs in value.

– field_pic_flag differs in value.

– bottom_field_flag is present in both and differs in value.

– nal_ref_idc differs in value with one of the nal_ref_idc values being equal to 0.

– pic_order_cnt_type is equal to 0 for both and either pic_order_cnt_lsb differs in value, or delta_pic_order_cnt_bottom differs in value.

– pic_order_cnt_type is equal to 1 for both and either delta_pic_order_cnt[0] differs in value, or delta_pic_order_cnt[1] differs in value.

– nal_unit_type differs in value with one of the nal_unit_type values being equal to 5.

– nal_unit_type is equal to 5 for both and idr_pic_id differs in value.

NOTE 2 – Some of the VCL NAL units in redundant coded pictures or some non-VCL NAL units (e.g. an access unit delimiter NAL unit) may also be used for the detection of the boundary between access units, and may therefore aid in the detection of the start of a new primary coded picture.

7.4.1.2.5 Order of VCL NAL units and association to coded pictures

Each VCL NAL unit is part of a coded picture.

The order of the VCL NAL units within a coded IDR picture is constrained as follows.

– If arbitrary slice order is allowed as specified in Annex A, coded slice of an IDR picture NAL units may have any order relative to each other.

– Otherwise (arbitrary slice order is not allowed), the order of coded slice of an IDR picture NAL units shall be in the order of increasing macroblock address for the first macroblock of each coded slice of an IDR picture NAL unit.

The order of the VCL NAL units within a coded non-IDR picture is constrained as follows.

– If arbitrary slice order is allowed as specified in Annex A, coded slice of a non-IDR picture NAL units or coded slice data partition A NAL units may have any order relative to each other. A coded slice data partition A NAL unit with a particular value of slice_id shall precede any present coded slice data partition B NAL unit with the same value of slice_id. A coded slice data partition A NAL unit with a particular value of slice_id shall precede any present coded slice data partition C NAL unit with the same value of slice_id. When a coded slice data partition B NAL unit with a particular value of slice_id is present, it shall precede any present coded slice data partition C NAL unit with the same value of slice_id.

– Otherwise (arbitrary slice order is not allowed), the order of coded slice of a non-IDR picture NAL units or coded slice data partition A NAL units shall be in the order of increasing macroblock address for the first macroblock of each coded slice of a non-IDR picture NAL unit or coded slice data partition A NAL unit. A coded slice data partition A NAL unit with a particular value of slice_id shall immediately precede any present coded slice data partition B NAL unit with the same value of slice_id. A coded slice data partition A NAL unit with a particular value of slice_id shall immediately precede any present coded slice data partition C NAL unit with the same value of slice_id, when a coded slice data partition B NAL unit with the same value of slice_id is not present. When a coded slice data partition B NAL unit with a particular value of slice_id is present, it shall immediately precede any present coded slice data partition C NAL unit with the same value of slice_id.

NAL units having nal_unit_type equal to 12 may be present in the access unit but shall not precede the first VCL NAL unit of the primary coded picture within the access unit.

NAL units having nal_unit_type equal to 0 or in the range of 24 to 31, inclusive, which are unspecified, may be present in the access unit but shall not precede the first VCL NAL unit of the primary coded picture within the access unit.

NAL units having nal_unit_type in the range of 20 to 23, inclusive, which are reserved, shall not precede the first VCL NAL unit of the primary coded picture within the access unit (when specified in the future by ITU-T | ISO/IEC).

7.4.2 Raw byte sequence payloads and RBSP trailing bits semantics

7.4.2.1 Sequence parameter set RBSP semantics

profile_idc and **level_idc** indicate the profile and level to which the bitstream conforms, as specified in Annex A.

constraint_set0_flag equal to 1 indicates that the bitstream obeys all constraints specified in subclause A.2.1. constraint_set0_flag equal to 0 indicates that the bitstream may or may not obey all constraints specified in subclause A.2.1.

constraint_set1_flag equal to 1 indicates that the bitstream obeys all constraints specified in subclause A.2.2. constraint_set1_flag equal to 0 indicates that the bitstream may or may not obey all constraints specified in subclause A.2.2.

constraint_set2_flag equal to 1 indicates that the bitstream obeys all constraints specified in subclause A.2.3. constraint_set2_flag equal to 0 indicates that the bitstream may or may not obey all constraints specified in subclause A.2.3.

NOTE 1 – When one or more than one of constraint_set0_flag, constraint_set1_flag, or constraint_set2_flag are equal to 1, the bitstream must obey the constraints of all of the indicated subclauses of subclause A.2. When profile_idc is equal to 100, 110, 122, or 144, the values of constraint_set0_flag, constraint_set1_flag, and constraint_set2_flag must all be equal to 0.

constraint_set3_flag indicates the following.

– If profile_idc is equal to 66, 77, or 88 and level_idc is equal to 11, constraint_set3_flag equal to 1 indicates that the bitstream obeys all constraints specified in Annex A for level 1b and constraint_set3_flag equal to 0 indicates that the bitstream may or may not obey all constraints specified in Annex A for level 1b.

– Otherwise (profile_idc is equal to 100, 110, 122, or 144 or level_idc is not equal to 11), the value of 1 for constraint_set3_flag is reserved for future use by ITU-T | ISO/IEC. constraint_set3_flag shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard when profile_idc is equal to 100, 110, 122, or 144 or level_idc is not equal to 11. Decoders conforming to this Recommendation | International Standard shall ignore the value of constraint_set3_flag when profile_idc is equal to 100, 110, 122, or 144 or level_idc is not equal to 11.

reserved_zero_4bits shall be equal to 0. Other values of reserved_zero_4bits may be specified in the future by ITU-T | ISO/IEC. Decoders shall ignore the value of reserved_zero_4bits.

seq_parameter_set_id identifies the sequence parameter set that is referred to by the picture parameter set. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

NOTE 2 – When feasible, encoders should use distinct values of seq_parameter_set_id when the values of other sequence parameter set syntax elements differ rather than changing the values of the syntax elements associated with a specific value of seq_parameter_set_id.

chroma_format_idc specifies the chroma sampling relative to the luma sampling as specified in subclause 6.2. The value of chroma_format_idc shall be in the range of 0 to 3, inclusive. When chroma_format_idc is not present, it shall be inferred to be equal to 1 (4:2:0 chroma format).

residual_colour_transform_flag equal to 1 specifies that the residual colour transform is applied as specified in subclause 8.5. residual_colour_transform_flag equal to 0 specifies that the residual colour transform is not applied. When residual_colour_transform_flag is not present, it shall be inferred to be equal to 0.

bit_depth_luma_minus8 specifies the bit depth of the samples of the luma array and the value of the luma quantisation parameter range offset $QpBdOffset_Y$, as specified by

$$BitDepth_Y = 8 + bit_depth_luma_minus8 \tag{7-1}$$

$$QpBdOffset_Y = 6 * bit_depth_luma_minus8 \tag{7-2}$$

When bit_depth_luma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_luma_minus8 shall be in the range of 0 to 4, inclusive.

bit_depth_chroma_minus8 specifies the bit depth of the samples of the chroma arrays and the value of the chroma quantisation parameter range offset $QpBdOffset_C$, as specified by

$$BitDepth_C = 8 + bit_depth_chroma_minus8 \tag{7-3}$$

$$QpBdOffset_C = 6 * (bit_depth_chroma_minus8 + residual_colour_transform_flag) \tag{7-4}$$

When bit_depth_chroma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_chroma_minus8 shall be in the range of 0 to 4, inclusive.

The variable RawMbBits is derived as

$$RawMbBits = 256 * BitDepth_Y + 2 * MbWidthC * MbHeightC * BitDepth_C \tag{7-5}$$

qpprime_y_zero_transform_bypass_flag equal to 1 specifies that, when QP'_Y is equal to 0, a transform bypass operation for the transform coefficient decoding process and picture construction process prior to deblocking filter process as specified in subclause 8.5 shall be applied. qpprime_y_zero_transform_bypass_flag equal to 0 specifies that the transform coefficient decoding process and picture construction process prior to deblocking filter process shall not use the transform bypass operation. When qpprime_y_zero_transform_bypass_flag is not present, it shall be inferred to be equal to 0.

seq_scaling_matrix_present_flag equal to 1 specifies that the flags seq_scaling_list_present_flag[i] for i = 0..7 are present. seq_scaling_matrix_present_flag equal to 0 specifies that these flags are not present and the sequence-level scaling list specified by Flat_4x4_16 shall be inferred for i = 0..5 and the sequence-level scaling list specified by Flat_8x8_16 shall be inferred for i = 6..7. When seq_scaling_matrix_present_flag is not present, it shall be inferred to be equal to 0.

The scaling lists Flat_4x4_16 and Flat_8x8_16 are specified as follows:

Flat_4x4_16[i] = 16, with i = 0..15, (7-6)

Flat_8x8_16[i] = 16, with i = 0..63. (7-7)

seq_scaling_list_present_flag[i **]** equal to 1 specifies that the syntax structure for scaling list i is present in the sequence parameter set. seq_scaling_list_present_flag[i] equal to 0 specifies that the syntax structure for scaling list i is not present in the sequence parameter set and the scaling list fall-back rule set A specified in Table 7-2 shall be used to infer the sequence-level scaling list for index i.

Table 7-2 – Assignment of mnemonic names to scaling list indices and specification of fall-back rule

Value of scaling list index	Mnemonic name	Block size	MB prediction type	Component	Scaling list fall-back rule set A	Scaling list fall-back rule set B	Default scaling list
0	Sl_4x4_Intra_Y	4x4	Intra	Y	default scaling list	sequence-level scaling list	Default_4x4_Intra
1	Sl_4x4_Intra_Cb	4x4	Intra	Cb	scaling list for i = 0	scaling list for i = 0	Default_4x4_Intra
2	Sl_4x4_Intra_Cr	4x4	Intra	Cr	scaling list for i = 1	scaling list for i = 1	Default_4x4_Intra
3	Sl_4x4_Inter_Y	4x4	Inter	Y	default scaling list	sequence-level scaling list	Default_4x4_Inter
4	Sl_4x4_Inter_Cb	4x4	Inter	Cb	scaling list for i = 3	scaling list for i = 3	Default_4x4_Inter
5	Sl_4x4_Inter_Cr	4x4	Inter	Cr	scaling list for i = 4	scaling list for i = 4	Default_4x4_Inter
6	Sl_8x8_Intra_Y	8x8	Intra	Y	default scaling list	sequence-level scaling list	Default_8x8_Intra
7	Sl_8x8_Inter_Y	8x8	Inter	Y	default scaling list	sequence-level scaling list	Default_8x8_Inter

Table 7-3 specifies the default scaling lists Default_4x4_Intra and Default_4x4_Inter. Table 7-4 specifies the default scaling lists Default_8x8_Intra and Default_8x8_Inter.

Table 7-3 – Specification of default scaling lists Default_4x4_Intra and Default_4x4_Inter

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Default_4x4_Intra[idx]	6	13	13	20	20	20	28	28	28	28	32	32	32	37	37	42
Default_4x4_Inter[idx]	10	14	14	20	20	20	24	24	24	24	27	27	27	30	30	34

Table 7-4 – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Default_8x8_Intra[idx]	6	10	10	13	11	13	16	16	16	16	18	18	18	18	18	23
Default_8x8_Inter[idx]	9	13	13	15	13	15	17	17	17	17	19	19	19	19	19	21

Table 7-4 (continued) – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter

idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Default_8x8_Intra[idx]	23	23	23	23	23	25	25	25	25	25	25	25	27	27	27	27
Default_8x8_Inter[idx]	21	21	21	21	21	22	22	22	22	22	22	22	24	24	24	24

Table 7-4 (continued) – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter

idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Default_8x8_Intra[idx]	27	27	27	27	29	29	29	29	29	29	29	31	31	31	31	31
Default_8x8_Inter[idx]	24	24	24	24	25	25	25	25	25	25	25	27	27	27	27	27

Table 7-4 (concluded) – Specification of default scaling lists Default_8x8_Intra and Default_8x8_Inter

idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Default_8x8_Intra[idx]	31	33	33	33	33	33	36	36	36	36	38	38	38	40	40	42
Default_8x8_Inter[idx]	27	28	28	28	28	28	30	30	30	30	32	32	32	33	33	35

log2_max_frame_num_minus4 specifies the value of the variable MaxFrameNum that is used in frame_num related derivations as follows:

$$MaxFrameNum = 2^{(log2_max_frame_num_minus4 + 4)} \quad (7\text{-}8)$$

The value of log2_max_frame_num_minus4 shall be in the range of 0 to 12, inclusive.

pic_order_cnt_type specifies the method to decode picture order count (as specified in subclause 8.2.1). The value of pic_order_cnt_type shall be in the range of 0 to 2, inclusive.

pic_order_cnt_type shall not be equal to 2 in a coded video sequence that contains any of the following

- an access unit containing a non-reference frame followed immediately by an access unit containing a non-reference picture;
- two access units each containing a field with the two fields together forming a complementary non-reference field pair followed immediately by an access unit containing a non-reference picture;
- an access unit containing a non-reference field followed immediately by an access unit containing another non-reference picture that does not form a complementary non-reference field pair with the first of the two access units.

log2_max_pic_order_cnt_lsb_minus4 specifies the value of the variable MaxPicOrderCntLsb that is used in the decoding process for picture order count as specified in subclause 8.2.1 as follows:

$$MaxPicOrderCntLsb = 2^{(log2_max_pic_order_cnt_lsb_minus4 + 4)} \quad (7\text{-}9)$$

The value of log2_max_pic_order_cnt_lsb_minus4 shall be in the range of 0 to 12, inclusive.

delta_pic_order_always_zero_flag equal to 1 specifies that delta_pic_order_cnt[0] and delta_pic_order_cnt[1] are not present in the slice headers of the sequence and shall be inferred to be equal to 0. delta_pic_order_always_zero_flag equal to 0 specifies that delta_pic_order_cnt[0] is present in the slice headers of the sequence and delta_pic_order_cnt[1] may be present in the slice headers of the sequence.

offset_for_non_ref_pic is used to calculate the picture order count of a non-reference picture as specified in 8.2.1. The value of offset_for_non_ref_pic shall be in the range of -2^{31} to $2^{31} - 1$, inclusive.

offset_for_top_to_bottom_field is used to calculate the picture order count of a bottom field as specified in subclause 8.2.1. The value of offset_for_top_to_bottom_field shall be in the range of -2^{31} to $2^{31} - 1$, inclusive.

num_ref_frames_in_pic_order_cnt_cycle is used in the decoding process for picture order count as specified in subclause 8.2.1. The value of num_ref_frames_in_pic_order_cnt_cycle shall be in the range of 0 to 255, inclusive.

offset_for_ref_frame[i **]** is an element of a list of num_ref_frames_in_pic_order_cnt_cycle values used in the decoding process for picture order count as specified in subclause 8.2.1. The value of offset_for_ref_frame[i] shall be in the range of -2^{31} to $2^{31} - 1$, inclusive.

num_ref_frames specifies the maximum number of short-term and long-term reference frames, complementary reference field pairs, and non-paired reference fields that may be used by the decoding process for inter prediction of any picture in the sequence. num_ref_frames also determines the size of the sliding window operation as specified in subclause 8.2.5.3. The value of num_ref_frames shall be in the range of 0 to MaxDpbSize (as specified in subclause A.3.1 or A.3.2), inclusive.

gaps_in_frame_num_value_allowed_flag specifies the allowed values of frame_num as specified in subclause 7.4.3 and the decoding process in case of an inferred gap between values of frame_num as specified in subclause 8.2.5.2.

pic_width_in_mbs_minus1 plus 1 specifies the width of each decoded picture in units of macroblocks.

The variable for the picture width in units of macroblocks is derived as follows

$$PicWidthInMbs = pic_width_in_mbs_minus1 + 1 \quad (7\text{-}10)$$

The variable for picture width for the luma component is derived as follows

$$PicWidthInSamples_L = PicWidthInMbs * 16 \quad (7\text{-}11)$$

The variable for picture width for the chroma components is derived as follows

$$PicWidthInSamples_C = PicWidthInMbs * MbWidthC \quad (7\text{-}12)$$

pic_height_in_map_units_minus1 plus 1 specifies the height in slice group map units of a decoded frame or field.

The variables PicHeightInMapUnits and PicSizeInMapUnits are derived as follows

$$PicHeightInMapUnits = pic_height_in_map_units_minus1 + 1 \quad (7\text{-}13)$$

$$PicSizeInMapUnits = PicWidthInMbs * PicHeightInMapUnits \quad (7\text{-}14)$$

frame_mbs_only_flag equal to 0 specifies that coded pictures of the coded video sequence may either be coded fields or coded frames. frame_mbs_only_flag equal to 1 specifies that every coded picture of the coded video sequence is a coded frame containing only frame macroblocks.

The allowed range of values for pic_width_in_mbs_minus1, pic_height_in_map_units_minus1, and frame_mbs_only_flag is specified by constraints in Annex A.

Depending on frame_mbs_only_flag, semantics are assigned to pic_height_in_map_units_minus1 as follows.

– If frame_mbs_only_flag is equal to 0, pic_height_in_map_units_minus1 plus 1 is the height of a field in units of macroblocks.

– Otherwise (frame_mbs_only_flag is equal to 1), pic_height_in_map_units_minus1 plus 1 is the height of a frame in units of macroblocks.

The variable FrameHeightInMbs is derived as follows

$$FrameHeightInMbs = (2 - frame_mbs_only_flag) * PicHeightInMapUnits \quad (7\text{-}15)$$

mb_adaptive_frame_field_flag equal to 0 specifies no switching between frame and field macroblocks within a picture. mb_adaptive_frame_field_flag equal to 1 specifies the possible use of switching between frame and field macroblocks within frames. When mb_adaptive_frame_field_flag is not present, it shall be inferred to be equal to 0.

direct_8x8_inference_flag specifies the method used in the derivation process for luma motion vectors for B_Skip, B_Direct_16x16 and B_Direct_8x8 as specified in subclause 8.4.1.2. When frame_mbs_only_flag is equal to 0, direct_8x8_inference_flag shall be equal to 1.

frame_cropping_flag equal to 1 specifies that the frame cropping offset parameters follow next in the sequence parameter set. frame_cropping_flag equal to 0 specifies that the frame cropping offset parameters are not present.

frame_crop_left_offset, frame_crop_right_offset, frame_crop_top_offset, frame_crop_bottom_offset specify the samples of the pictures in the coded video sequence that are output from the decoding process, in terms of a rectangular region specified in frame coordinates for output.

The variables CropUnitX and CropUnitY are derived as follows:

— If chroma_format_idc is equal to 0, CropUnitX and CropUnitY are derived as

$$CropUnitX = 1 \quad (7\text{-}16)$$
$$CropUnitY = 2 - frame_mbs_only_flag \quad (7\text{-}17)$$

— Otherwise (chroma_format_idc is equal to 1, 2, or 3), CropUnitX and CropUnitY are derived as

$$CropUnitX = SubWidthC \quad (7\text{-}18)$$
$$CropUnitY = SubHeightC * (2 - frame_mbs_only_flag) \quad (7\text{-}19)$$

The frame cropping rectangle contains luma samples with horizontal frame coordinates from CropUnitX * frame_crop_left_offset to $PicWidthInSamples_L$ – (CropUnitX * frame_crop_right_offset + 1) and vertical frame coordinates from CropUnitY * frame_crop_top_offset to (16 * FrameHeightInMbs) – (CropUnitY * frame_crop_bottom_offset + 1), inclusive. The value of frame_crop_left_offset shall be in the range of 0 to ($PicWidthInSamples_L$ / CropUnitX) – (frame_crop_right_offset + 1), inclusive; and the value of frame_crop_top_offset shall be in the range of 0 to (16 * FrameHeightInMbs / CropUnitY) – (frame_crop_bottom_offset + 1), inclusive.

When frame_cropping_flag is equal to 0, the values of frame_crop_left_offset, frame_crop_right_offset, frame_crop_top_offset, and frame_crop_bottom_offset shall be inferred to be equal to 0.

When chroma_format_idc is not equal to 0, the corresponding specified samples of the two chroma arrays are the samples having frame coordinates (x / SubWidthC, y / SubHeightC), where (x, y) are the frame coordinates of the specified luma samples.

For decoded fields, the specified samples of the decoded field are the samples that fall within the rectangle specified in frame coordinates.

vui_parameters_present_flag equal to 1 specifies that the vui_parameters( ) syntax structure as specified in Annex E is present. vui_parameters_present_flag equal to 0 specifies that the vui_parameters( ) syntax structure as specified in Annex E is not present.

7.4.2.1.1 Scaling list semantics

delta_scale is used to derive the j-th element of the scaling list for j in the range of 0 to sizeOfScalingList - 1, inclusive. The value of delta_scale shall be in the range of -128 to +127, inclusive.

When useDefaultScalingMatrixFlag is derived to be equal to 1, the scaling list shall be inferred to be equal to the default scaling list as specified in Table 7-2.

7.4.2.1.2 Sequence parameter set extension RBSP semantics

seq_parameter_set_id identifies the sequence parameter set associated with the sequence parameter set extension. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

aux_format_idc equal to 0 indicates that there are no auxiliary coded pictures in the coded video sequence. aux_format_idc equal to 1 indicates that exactly one auxiliary coded picture is present in each access unit of the coded video sequence, and that for alpha blending purposes the decoded samples of the associated primary coded picture in each access unit should be multiplied by the interpretation sample values of the auxiliary coded picture in the access unit in the display process after output from the decoding process. aux_format_idc equal to 2 indicates that exactly one auxiliary coded picture exists in each access unit of the coded video sequence, and that for alpha blending purposes the decoded samples of the associated primary coded picture in each access unit should not be multiplied by the interpretation sample values of the auxiliary coded picture in the access unit in the display process after output from the decoding process. aux_format_idc equal to 3 indicates that exactly one auxiliary coded picture exists in each access unit of the coded video sequence, and that the usage of the auxiliary coded pictures is unspecified. The value of aux_format_idc shall be in the range of 0 to 3, inclusive. Values greater than 3 for aux_format_idc are reserved to indicate the presence of exactly one auxiliary coded picture in each access unit of the coded video sequence for purposes to be specified in the future by ITU-T | ISO/IEC. When aux_format_idc is not present, it shall be inferred to be equal to 0.

NOTE 1 – Decoders conforming to this Recommendation | International Standard are not required to decode auxiliary coded pictures.

bit_depth_aux_minus8 specifies the bit depth of the samples of the sample array of the auxiliary coded picture. bit_depth_aux_minus8 shall be in the range of 0 to 4, inclusive.

alpha_incr_flag equal to 0 indicates that the interpretation sample value for each decoded auxiliary coded picture sample value is equal to the decoded auxiliary coded picture sample value for purposes of alpha blending. alpha_incr_flag equal to 1 indicates that, for purposes of alpha blending, after decoding the auxiliary coded picture samples, any auxiliary coded picture sample value that is greater than Min(alpha_opaque_value, alpha_transparent_value) should be increased by one to obtain the interpretation sample value for the auxiliary coded picture sample, and any auxiliary coded picture sample value that is less than or equal to Min(alpha_opaque_value, alpha_transparent_value) should be used without alteration as the interpretation sample value for the decoded auxiliary coded picture sample value.

alpha_opaque_value specifies the interpretation sample value of an auxiliary coded picture sample for which the associated luma and chroma samples of the same access unit are considered opaque for purposes of alpha blending. The number of bits used for the representation of the alpha_opaque_value syntax element is bit_depth_aux_minus8 + 9 bits.

alpha_transparent_value specifies the interpretation sample value of an auxiliary coded picture sample for which the associated luma and chroma samples of the same access unit are considered transparent for purposes of alpha blending. The number of bits used for the representation of the alpha_transparent_value syntax element is bit_depth_aux_minus8 + 9 bits.

When alpha_incr_flag is equal to 1, alpha_transparent_value shall not be equal to alpha_opaque_value and Log2(Abs(alpha_opaque_value – alpha_transparent_value)) shall have an integer value. A value of alpha_transparent_value that is equal to alpha_opaque_value indicates that the auxiliary coded picture is not intended for alpha blending purposes.

NOTE 2 – For alpha blending purposes, alpha_opaque_value may be greater than alpha_transparent_value, or it may be less than alpha_transparent_value. Interpretation sample values should be clipped to the range of alpha_opaque_value to alpha_transparent_value, inclusive.

The decoding of the sequence parameter set extension and the decoding of auxiliary coded pictures is not required for conformance with this Recommendation | International Standard.

The syntax of each coded slice of an auxiliary coded picture shall obey the same constraints as a coded slice of a redundant picture, with the following differences of constraints.

- The following applies in regard to whether the primary coded picture is an IDR picture.
  - If the primary coded picture is an IDR picture, the auxiliary coded slice syntax shall correspond to that of a slice having nal_unit_type equal to 5 (a slice of an IDR picture);
  - Otherwise (the primary coded picture is not an IDR picture), the auxiliary coded slice syntax shall correspond to that of a slice having nal_unit_type equal to 1 (a slice of a non-IDR picture).
- The slices of an auxiliary coded picture (when present) shall contain all macroblocks corresponding to those of the primary coded picture.
- redundant_pic_cnt shall be equal to 0 in all auxiliary coded slices.

The (optional) decoding process for the decoding of auxiliary coded pictures is the same as if the auxiliary coded pictures were primary coded pictures in a separate coded video stream that differs from the primary coded pictures in the current coded video stream in the following ways.

– The IDR or non-IDR status of each auxiliary coded picture shall be inferred to be the same as the IDR or non-IDR status of the primary picture in the same access unit, rather than being inferred from the value of nal_ref_idc.

– The value of chroma_format_idc shall be inferred to be equal to 0 for the decoding of the auxiliary coded pictures.

– The value of bit_depth_luma_minus8 shall be inferred to be equal to bit_depth_aux_minus8 for the decoding of the auxiliary coded pictures.

NOTE 3 – Alpha blending composition is normally performed with a background picture B, a foreground picture F, and a decoded auxiliary coded picture A, all of the same size. Assume for purposes of example illustration that the chroma resolution of B and F have been upsampled to the same resolution as the luma. Denote corresponding samples of B, F and A by b, f and a, respectively. Denote luma and chroma samples by subscripts Y, Cb and Cr.

Define the variables alphaRange, alphaFwt and alphaBwt as follows:

alphaRange = Abs(alpha_opaque_value - alpha_transparent_value)

alphaFwt = Abs(a - alpha_transparent_value)

alphaBwt = Abs(a - alpha_opaque_value)

Then, in alpha blending composition, samples d of the displayed picture D may be calculated as

$d_Y = (alphaFwt*f_Y + alphaBwt*b_Y + alphaRange/2) / alphaRange$

$d_{CB} = (alphaFwt*f_{CB} + alphaBwt*b_{CB} + alphaRange/2) / alphaRange$

$d_{CR} = (alphaFwt*f_{CR} + alphaBwt*b_{CR} + alphaRange/2) / alphaRange$

The samples of pictures D, F and B could also represent red, green, and blue component values (see subclause E.2.1). Here we have assumed Y, Cb and Cr component values. Each component, e.g. Y, is assumed for purposes of example illustration above to have the same bit depth in each of the pictures D, F and B. However, different components, e.g. Y and Cb, need not have the same bit depth in this example.

When aux_format_idc is equal to 1, F would be the decoded picture obtained from the decoded luma and chroma, and A would be the decoded picture obtained from the decoded auxiliary coded picture. In this case, the indicated example alpha blending composition involves multiplying the samples of F by factors obtained from the samples of A.

A picture format that is useful for editing or direct viewing, and that is commonly used, is called pre-multiplied-black video. If the foreground picture was F, then the pre-multiplied-black video S is given by

$s_Y = (alphaFwt*f_Y) / alphaRange$

$s_{CB} = (alphaFwt*f_{CB}) / alphaRange$

$s_{CR} = (alphaFwt*f_{CR}) / alphaRange$

Pre-multiplied-black video has the characteristic that the picture S will appear correct if displayed against a black background. For a non-black background B, the composition of the displayed picture D may be calculated as

$d_Y = s_Y + (alphaBwt*b_Y + alphaRange/2) / alphaRange$

$d_{CB} = s_{CB} + (alphaBwt*b_{CB} + alphaRange/2) / alphaRange$

$d_{CR} = s_{CR} + (alphaBwt*b_{CR} + alphaRange/2) / alphaRange$

When aux_format_idc is equal to 2, S would be the decoded picture obtained from the decoded luma and chroma, and A would again be the decoded picture obtained from the decoded auxiliary coded picture. In this case, alpha blending composition does not involve multiplication of the samples of S by factors obtained from the samples of A.

additional_extension_flag equal to 0 indicates that no additional data follows within the sequence parameter set extension syntax structure prior to the RBSP trailing bits. The value of additional_extension_flag shall be equal to 0. The value of 1 for additional_extension_flag is reserved for future use by ITU-T | ISO/IEC. Decoders that conform to this Recommendation | International Standard shall ignore all data that follows the value of 1 for additional_extension_flag in a sequence parameter set extension NAL unit.

7.4.2.2 Picture parameter set RBSP semantics

pic_parameter_set_id identifies the picture parameter set that is referred to in the slice header. The value of pic_parameter_set_id shall be in the range of 0 to 255, inclusive.

seq_parameter_set_id refers to the active sequence parameter set. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

entropy_coding_mode_flag selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows.

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see subclause 9.1 or CAVLC, see subclause 9.2).

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see subclause 9.3).

pic_order_present_flag equal to 1 specifies that the picture order count related syntax elements are present in the slice headers as specified in subclause 7.3.3. pic_order_present_flag equal to 0 specifies that the picture order count related syntax elements are not present in the slice headers.

num_slice_groups_minus1 plus 1 specifies the number of slice groups for a picture. When num_slice_groups_minus1 is equal to 0, all slices of the picture belong to the same slice group. The allowed range of num_slice_groups_minus1 is specified in Annex A.

slice_group_map_type specifies how the mapping of slice group map units to slice groups is coded. The value of slice_group_map_type shall be in the range of 0 to 6, inclusive.

slice_group_map_type equal to 0 specifies interleaved slice groups.

slice_group_map_type equal to 1 specifies a dispersed slice group mapping.

slice_group_map_type equal to 2 specifies one or more "foreground" slice groups and a "leftover" slice group.

slice_group_map_type values equal to 3, 4, and 5 specify changing slice groups. When num_slice_groups_minus1 is not equal to 1, slice_group_map_type shall not be equal to 3, 4, or 5.

slice_group_map_type equal to 6 specifies an explicit assignment of a slice group to each slice group map unit.

Slice group map units are specified as follows.

– If frame_mbs_only_flag is equal to 0 and mb_adaptive_frame_field_flag is equal to 1 and the coded picture is a frame, the slice group map units are macroblock pair units.

– Otherwise, if frame_mbs_only_flag is equal to 1 or a coded picture is a field, the slice group map units are units of macroblocks.

– Otherwise (frame_mbs_only_flag is equal to 0 and mb_adaptive_frame_field_flag is equal to 0 and the coded picture is a frame), the slice group map units are units of two macroblocks that are vertically contiguous as in a frame macroblock pair of an MBAFF frame.

run_length_minus1[i **]** is used to specify the number of consecutive slice group map units to be assigned to the i-th slice group in raster scan order of slice group map units. The value of run_length_minus1[i] shall be in the range of 0 to PicSizeInMapUnits - 1, inclusive.

top_left[i **]** and **bottom_right[** i **]** specify the top-left and bottom-right corners of a rectangle, respectively. top_left[i] and bottom_right[i] are slice group map unit positions in a raster scan of the picture for the slice group map units. For each rectangle i, all of the following constraints shall be obeyed by the values of the syntax elements top_left[i] and bottom_right[i]

– top_left[i] shall be less than or equal to bottom_right[i] and bottom_right[i] shall be less than PicSizeInMapUnits.

– (top_left[i] % PicWidthInMbs) shall be less than or equal to the value of (bottom_right[i] % PicWidthInMbs).

slice_group_change_direction_flag is used with slice_group_map_type to specify the refined map type when slice_group_map_type is 3, 4, or 5.

slice_group_change_rate_minus1 is used to specify the variable SliceGroupChangeRate. SliceGroupChangeRate specifies the multiple in number of slice group map units by which the size of a slice group can change from one picture to the next. The value of slice_group_change_rate_minus1 shall be in the range of 0 to PicSizeInMapUnits – 1, inclusive. The SliceGroupChangeRate variable is specified as follows:

SliceGroupChangeRate = slice_group_change_rate_minus1 + 1 (7-20)

pic_size_in_map_units_minus1 is used to specify the number of slice group map units in the picture. pic_size_in_map_units_minus1 shall be equal to PicSizeInMapUnits - 1.

slice_group_id[i **]** identifies a slice group of the i-th slice group map unit in raster scan order. The size of the slice_group_id[i] syntax element is Ceil(Log2(num_slice_groups_minus1 + 1)) bits. The value of slice_group_id[i] shall be in the range of 0 to num_slice_groups_minus1, inclusive.

num_ref_idx_l0_active_minus1 specifies the maximum reference index for reference picture list 0 that shall be used to decode each slice of the picture in which list 0 prediction is used when num_ref_idx_active_override_flag is equal to 0 for the slice. When MbaffFrameFlag is equal to 1, num_ref_idx_l0_active_minus1 is the maximum index value for the decoding of frame macroblocks and 2 * num_ref_idx_l0_active_minus1 + 1 is the maximum index value for the decoding of field macroblocks. The value of num_ref_idx_l0_active_minus1 shall be in the range of 0 to 31, inclusive.

num_ref_idx_l1_active_minus1 has the same semantics as num_ref_idx_l0_active_minus1 with l0 and list 0 replaced by l1 and list 1, respectively.

weighted_pred_flag equal to 0 specifies that weighted prediction shall not be applied to P and SP slices. weighted_pred_flag equal to 1 specifies that weighted prediction shall be applied to P and SP slices.

weighted_bipred_idc equal to 0 specifies that the default weighted prediction shall be applied to B slices. weighted_bipred_idc equal to 1 specifies that explicit weighted prediction shall be applied to B slices. weighted_bipred_idc equal to 2 specifies that implicit weighted prediction shall be applied to B slices. The value of weighted_bipred_idc shall be in the range of 0 to 2, inclusive.

pic_init_qp_minus26 specifies the initial value minus 26 of $SliceQP_Y$ for each slice. The initial value is modified at the slice layer when a non-zero value of slice_qp_delta is decoded, and is modified further when a non-zero value of mb_qp_delta is decoded at the macroblock layer. The value of pic_init_qp_minus26 shall be in the range of $-(26 + QpBdOffset_Y)$ to +25, inclusive.

pic_init_qs_minus26 specifies the initial value minus 26 of $SliceQS_Y$ for all macroblocks in SP or SI slices. The initial value is modified at the slice layer when a non-zero value of slice_qs_delta is decoded. The value of pic_init_qs_minus26 shall be in the range of -26 to +25, inclusive.

chroma_qp_index_offset specifies the offset that shall be added to QP_Y and QS_Y for addressing the table of QP_C values for the Cb chroma component. The value of chroma_qp_index_offset shall be in the range of -12 to +12, inclusive.

deblocking_filter_control_present_flag equal to 1 specifies that a set of syntax elements controlling the characteristics of the deblocking filter is present in the slice header. deblocking_filter_control_present_flag equal to 0 specifies that the set of syntax elements controlling the characteristics of the deblocking filter is not present in the slice headers and their inferred values are in effect.

constrained_intra_pred_flag equal to 0 specifies that intra prediction allows usage of residual data and decoded samples of neighbouring macroblocks coded using Inter macroblock prediction modes for the prediction of macroblocks coded using Intra macroblock prediction modes. constrained_intra_pred_flag equal to 1 specifies constrained intra prediction, in which case prediction of macroblocks coded using Intra macroblock prediction modes only uses residual data and decoded samples from I or SI macroblock types.

redundant_pic_cnt_present_flag equal to 0 specifies that the redundant_pic_cnt syntax element is not present in slice headers, data partitions B, and data partitions C that refer (either directly or by association with a corresponding data partition A) to the picture parameter set. redundant_pic_cnt_present_flag equal to 1 specifies that the redundant_pic_cnt syntax element is present in all slice headers, data partitions B, and data partitions C that refer (either directly or by association with a corresponding data partition A) to the picture parameter set.

transform_8x8_mode_flag equal to 1 specifies that the 8x8 transform decoding process may be in use (see subclause 8.5). transform_8x8_mode_flag equal to 0 specifies that the 8x8 transform decoding process is not in use. When transform_8x8_mode_flag is not present, it shall be inferred to be 0.

pic_scaling_matrix_present_flag equal to 1 specifies that parameters are present to modify the scaling lists specified in the sequence parameter set. pic_scaling_matrix_present_flag equal to 0 specifies that the scaling lists used for the picture shall be inferred to be equal to those specified by the sequence parameter set. When pic_scaling_matrix_present_flag is not present, it shall be inferred to be equal to 0.

pic_scaling_list_present_flag[i **]** equal to 1 specifies that the scaling list syntax structure is present to specify the scaling list for index i. pic_scaling_list_present_flag[i] equal to 0 specifies that the syntax structure for scaling list i is not present in the picture parameter set and that depending on the value of seq_scaling_matrix_present_flag, the following applies.

– If seq_scaling_matrix_present_flag is equal to 0, the scaling list fall-back rule set A as specified in Table 7-2 shall be used to derive the picture-level scaling list for index i.

– Otherwise (seq_scaling_matrix_present_flag is equal to 1), the scaling list fall-back rule set B as specified in Table 7-2 shall be used to derive the picture-level scaling list for index i.

second_chroma_qp_index_offset specifies the offset that shall be added to QP_Y and QS_Y for addressing the table of QP_C values for the Cr chroma component. The value of second_chroma_qp_index_offset shall be in the range of -12 to +12, inclusive.

When second_chroma_qp_index_offset is not present, it shall be inferred to be equal to chroma_qp_index_offset.

7.4.2.3 Supplemental enhancement information RBSP semantics

Supplemental Enhancement Information (SEI) contains information that is not necessary to decode the samples of coded pictures from VCL NAL units.

7.4.2.3.1 Supplemental enhancement information message semantics

An SEI NAL unit contains one or more SEI messages. Each SEI message consists of the variables specifying the type payloadType and size payloadSize of the SEI payload. SEI payloads are specified in Annex D. The derived SEI payload size payloadSize is specified in bytes and shall be equal to the number of bytes in the SEI payload.

ff_byte is a byte equal to 0xFF identifying a need for a longer representation of the syntax structure that it is used within.

last_payload_type_byte is the last byte of the payload type of an SEI message.

last_payload_size_byte is the last byte of the size of an SEI message.

7.4.2.4 Access unit delimiter RBSP semantics

The access unit delimiter may be used to indicate the type of slices present in a primary coded picture and to simplify the detection of the boundary between access units. There is no normative decoding process associated with the access unit delimiter.

primary_pic_type indicates that the slice_type values for all slices of the primary coded picture are members of the set listed in Table 7-5 for the given value of primary_pic_type.

Table 7-5 – Meaning of primary_pic_type

primary_pic_type	slice_type values that may be present in the primary coded picture
0	I
1	I, P
2	I, P, B
3	SI
4	SI, SP
5	I, SI
6	I, SI, P, SP
7	I, SI, P, SP, B

7.4.2.5 End of sequence RBSP semantics

The end of sequence RBSP specifies that the next subsequent access unit in the bitstream in decoding order (if any) shall be an IDR access unit. The syntax content of the SODB and RBSP for the end of sequence RBSP are empty. No normative decoding process is specified for an end of sequence RBSP.

7.4.2.6 End of stream RBSP semantics

The end of stream RBSP indicates that no additional NAL units shall be present in the bitstream that are subsequent to the end of stream RBSP in decoding order. The syntax content of the SODB and RBSP for the end of stream RBSP are empty. No normative decoding process is specified for an end of stream RBSP.

7.4.2.7 Filler data RBSP semantics

The filler data RBSP contains bytes whose value shall be equal to 0xFF. No normative decoding process is specified for a filler data RBSP.

ff_byte is a byte equal to 0xFF.

7.4.2.8 Slice layer without partitioning RBSP semantics

The slice layer without partitioning RBSP consists of a slice header and slice data.

7.4.2.9 Slice data partition RBSP semantics

7.4.2.9.1 Slice data partition A RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Partition A contains all syntax elements of category 2.

Category 2 syntax elements include all syntax elements in the slice header and slice data syntax structures other than the syntax elements in the residual( ) syntax structure.

slice_id identifies the slice associated with the data partition. Each slice shall have a unique slice_id value within the coded picture that contains the slice. When arbitrary slice order is not allowed as specified in Annex A, the first slice of a coded picture, in decoding order, shall have slice_id equal to 0 and the value of slice_id shall be incremented by one for each subsequent slice of the coded picture in decoding order.

The range of slice_id is specified as follows.

– If MbaffFrameFlag is equal to 0, slice_id shall be in the range of 0 to PicSizeInMbs - 1, inclusive.

– Otherwise (MbaffFrameFlag is equal to 1), slice_id shall be in the range of 0 to PicSizeInMbs / 2 - 1, inclusive.

7.4.2.9.2 Slice data partition B RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into one to three separate partitions. Slice data partition B contains all syntax elements of category 3.

Category 3 syntax elements include all syntax elements in the residual( ) syntax structure and in syntax structures used within that syntax structure for collective macroblock types I and SI as specified in Table 7-10.

slice_id has the same semantics as specified in subclause 7.4.2.9.1.

redundant_pic_cnt shall be equal to 0 for slices and slice data partitions belonging to the primary coded picture. The redundant_pic_cnt shall be greater than 0 for coded slices and coded slice data partitions in redundant coded pictures. When redundant_pic_cnt is not present, its value shall be inferred to be equal to 0. The value of redundant_pic_cnt shall be in the range of 0 to 127, inclusive.

The presence of a slice data partition B RBSP is specified as follows.

– If the syntax elements of a slice data partition A RBSP indicate the presence of any syntax elements of category 3 in the slice data for a slice, a slice data partition B RBSP shall be present having the same value of slice_id and redundant_pic_cnt as in the slice data partition A RBSP.

– Otherwise (the syntax elements of a slice data partition A RBSP do not indicate the presence of any syntax elements of category 3 in the slice data for a slice), no slice data partition B RBSP shall be present having the same value of slice_id and redundant_pic_cnt as in the slice data partition A RBSP.

7.4.2.9.3 Slice data partition C RBSP semantics

When slice data partitioning is in use, the coded data for a single slice is divided into three separate partitions. Slice data partition C contains all syntax elements of category 4.

Category 4 syntax elements include all syntax elements in the residual( ) syntax structure and in syntax structures used within that syntax structure for collective macroblock types P and B as specified in Table 7-10.

slice_id has the same semantics as specified in subclause 7.4.2.9.1.

redundant_pic_cnt has the same semantics as specified in subclause 7.4.2.9.2.

The presence of a slice data partition C RBSP is specified as follows.

– If the syntax elements of a slice data partition A RBSP indicate the presence of any syntax elements of category 4 in the slice data for a slice, a slice data partition C RBSP shall be present having the same value of slice_id and redundant_pic_cnt as in the slice data partition A RBSP.

– Otherwise (the syntax elements of a slice data partition A RBSP do not indicate the presence of any syntax elements of category 4 in the slice data for a slice), no slice data partition C RBSP shall be present having the same value of slice_id and redundant_pic_cnt as in the slice data partition A RBSP.

7.4.2.10 RBSP slice trailing bits semantics

cabac_zero_word is a byte-aligned sequence of two bytes equal to 0x0000.

Let NumBytesInVclNALunits be the sum of the values of NumBytesInNALunit for all VCL NAL units of a coded picture.

Let BinCountsInNALunits be the number of times that the parsing process function DecodeBin( ), specified in subclause 9.3.3.2, is invoked to decode the contents of all VCL NAL units of a coded picture. When entropy_coding_mode_flag is equal to 1, BinCountsInNALunits shall not exceed (32 ÷ 3) * NumBytesInVclNALunits + (RawMbBits * PicSizeInMbs) ÷ 32.

NOTE – The constraint on the maximum number of bins resulting from decoding the contents of the slice layer NAL units can be met by inserting a number of cabac_zero_word syntax elements to increase the value of NumBytesInVclNALunits. Each cabac_zero_word is represented in a NAL unit by the three-byte sequence 0x000003 (as a result of the constraints on NAL unit contents that result in requiring inclusion of an emulation_prevention_three_byte for each cabac_zero_word).

7.4.2.11 RBSP trailing bits semantics

rbsp_stop_one_bit shall be equal to 1.

rbsp_alignment_zero_bit shall be equal to 0.

7.4.3 Slice header semantics

When present, the value of the slice header syntax elements pic_parameter_set_id, frame_num, field_pic_flag, bottom_field_flag, idr_pic_id, pic_order_cnt_lsb, delta_pic_order_cnt_bottom, delta_pic_order_cnt[0], delta_pic_order_cnt[1], sp_for_switch_flag, and slice_group_change_cycle shall be the same in all slice headers of a coded picture.

first_mb_in_slice specifies the address of the first macroblock in the slice. When arbitrary slice order is not allowed as specified in Annex A, the value of first_mb_in_slice shall not be less than the value of first_mb_in_slice for any other slice of the current picture that precedes the current slice in decoding order.

The first macroblock address of the slice is derived as follows.

– If MbaffFrameFlag is equal to 0, first_mb_in_slice is the macroblock address of the first macroblock in the slice, and first_mb_in_slice shall be in the range of 0 to PicSizeInMbs - 1, inclusive.

– Otherwise (MbaffFrameFlag is equal to 1), first_mb_in_slice * 2 is the macroblock address of the first macroblock in the slice, which is the top macroblock of the first macroblock pair in the slice, and first_mb_in_slice shall be in the range of 0 to PicSizeInMbs / 2 - 1, inclusive.

slice_type specifies the coding type of the slice according to Table 7-6.

Table 7-6 – Name association to slice_type

slice_type	Name of slice_type
0	P (P slice)
1	B (B slice)
2	I (I slice)
3	SP (SP slice)
4	SI (SI slice)
5	P (P slice)
6	B (B slice)
7	I (I slice)
8	SP (SP slice)
9	SI (SI slice)

slice_type values in the range 5..9 specify, in addition to the coding type of the current slice, that all other slices of the current coded picture shall have a value of slice_type equal to the current value of slice_type or equal to the current value of slice_type – 5.

When nal_unit_type is equal to 5 (IDR picture), slice_type shall be equal to 2, 4, 7, or 9.

When num_ref_frames is equal to 0, slice_type shall be equal to 2, 4, 7, or 9.

pic_parameter_set_id specifies the picture parameter set in use. The value of pic_parameter_set_id shall be in the range of 0 to 255, inclusive.

frame_num is used as an identifier for pictures and shall be represented by log2_max_frame_num_minus4 + 4 bits in the bitstream. frame_num is constrained as follows:

The variable PrevRefFrameNum is derived as follows.

– If the current picture is an IDR picture, PrevRefFrameNum is set equal to 0.

– Otherwise (the current picture is not an IDR picture), PrevRefFrameNum is set as follows.

  – If the decoding process for gaps in frame_num specified in subclause 8.2.5.2 was invoked by the decoding process for an access unit that contained a non-reference picture that followed the previous access unit in decoding order that contained a reference picture, PrevRefFrameNum is set equal to the value of frame_num for the last of the "non-existing" reference frames inferred by the decoding process for gaps in frame_num specified in subclause 8.2.5.2.

  – Otherwise, PrevRefFrameNum is set equal to the value of frame_num for the previous access unit in decoding order that contained a reference picture.

The value of frame_num is constrained as follows.

– If the current picture is an IDR picture, frame_num shall be equal to 0.

– Otherwise (the current picture is not an IDR picture), referring to the primary coded picture in the previous access unit in decoding order that contains a reference picture as the preceding reference picture, the value of frame_num for the current picture shall not be equal to PrevRefFrameNum unless all of the following three conditions are true.

  – the current picture and the preceding reference picture belong to consecutive access units in decoding order

  – the current picture and the preceding reference picture are reference fields having opposite parity

  – one or more of the following conditions is true

    – the preceding reference picture is an IDR picture

    – the preceding reference picture includes a memory_management_control_operation syntax element equal to 5

      NOTE 1 – When the preceding reference picture includes a memory_management_control_operation syntax element equal to 5, PrevRefFrameNum is equal to 0.

    – there is a primary coded picture that precedes the preceding reference picture and the primary coded picture that precedes the preceding reference picture does not have frame_num equal to PrevRefFrameNum

    – there is a primary coded picture that precedes the preceding reference picture and the primary coded picture that precedes the preceding reference picture is not a reference picture

When the value of frame_num is not equal to PrevRefFrameNum, the following applies.

– There shall not be any previous field or frame in decoding order that is currently marked as "used for short-term reference" that has a value of frame_num equal to any value taken on by the variable UnusedShortTermFrameNum in the following:

```
UnusedShortTermFrameNum = ( PrevRefFrameNum + 1 ) % MaxFrameNum
while( UnusedShortTermFrameNum != frame_num )                    (7-21)
    UnusedShortTermFrameNum = ( UnusedShortTermFrameNum + 1 ) % MaxFrameNum
```

– The value of frame_num is constrained as follows.

  – If gaps_in_frame_num_value_allowed_flag is equal to 0, the value of frame_num for the current picture shall be equal to (PrevRefFrameNum + 1) % MaxFrameNum.

  – Otherwise (gaps_in_frame_num_value_allowed_flag is equal to 1), the following applies.

    – If frame_num is greater than PrevRefFrameNum, there shall not be any non-reference pictures in the bitstream that follow the previous reference picture and precede the current picture in decoding order in which either of the following conditions is true.

      – The value of frame_num for the non-reference picture is less than PrevRefFrameNum.

      – The value of frame_num for the non-reference picture is greater than the value of frame_num for the current picture.

– Otherwise (frame_num is less than PrevRefFrameNum), there shall not be any non-reference pictures in the bitstream that follow the previous reference picture and precede the current picture in decoding order in which both of the following conditions are true.
  – The value of frame_num for the non-reference picture is less than PrevRefFrameNum.
  – The value of frame_num for the non-reference picture is greater than the value of frame_num for the current picture.

A picture including a memory_management_control_operation equal to 5 shall have frame_num constraints as described above and, after the decoding of the current picture and the processing of the memory management control operations, the picture shall be inferred to have had frame_num equal to 0 for all subsequent use in the decoding process, except as specified in subclause 7.4.1.2.4.

NOTE 2 – When the primary coded picture is not an IDR picture and does not contain memory_management_control_operation syntax element equal to 5, the value of frame_num of a corresponding redundant coded picture is the same as the value of frame_num in the primary coded picture. Alternatively, the redundant coded picture includes a memory_management_control_operation syntax element equal to 5 and the corresponding primary coded picture is an IDR picture.

field_pic_flag equal to 1 specifies that the slice is a slice of a coded field. field_pic_flag equal to 0 specifies that the slice is a slice of a coded frame. When field_pic_flag is not present it shall be inferred to be equal to 0.

The variable MbaffFrameFlag is derived as follows.

MbaffFrameFlag = (mb_adaptive_frame_field_flag && !field_pic_flag) (7-22)

The variable for the picture height in units of macroblocks is derived as follows

PicHeightInMbs = FrameHeightInMbs / (1 + field_pic_flag) (7-23)

The variable for picture height for the luma component is derived as follows

$PicHeightInSamples_L$ = PicHeightInMbs * 16 (7-24)

The variable for picture height for the chroma component is derived as follows

$PicHeightInSamples_C$ = PicHeightInMbs * MbHeightC (7-25)

The variable PicSizeInMbs for the current picture is derived according to:

PicSizeInMbs = PicWidthInMbs * PicHeightInMbs (7-26)

The variable MaxPicNum is derived as follows.

– If field_pic_flag is equal to 0, MaxPicNum is set equal to MaxFrameNum.

– Otherwise (field_pic_flag is equal to 1), MaxPicNum is set equal to 2*MaxFrameNum.

The variable CurrPicNum is derived as follows.

– If field_pic_flag is equal to 0, CurrPicNum is set equal to frame_num.

– Otherwise (field_pic_flag is equal to 1), CurrPicNum is set equal to 2 * frame_num + 1.

bottom_field_flag equal to 1 specifies that the slice is part of a coded bottom field. bottom_field_flag equal to 0 specifies that the picture is a coded top field. When this syntax element is not present for the current slice, it shall be inferred to be equal to 0.

idr_pic_id identifies an IDR picture. The values of idr_pic_id in all the slices of an IDR picture shall remain unchanged. When two consecutive access units in decoding order are both IDR access units, the value of idr_pic_id in the slices of the first such IDR access unit shall differ from the idr_pic_id in the second such IDR access unit. The value of idr_pic_id shall be in the range of 0 to 65535, inclusive.

pic_order_cnt_lsb specifies the picture order count modulo MaxPicOrderCntLsb for the top field of a coded frame or for a coded field. The size of the pic_order_cnt_lsb syntax element is log2_max_pic_order_cnt_lsb_minus4 + 4 bits. The value of the pic_order_cnt_lsb shall be in the range of 0 to MaxPicOrderCntLsb – 1, inclusive.

delta_pic_order_cnt_bottom specifies the picture order count difference between the bottom field and the top field of a coded frame as follows.

– If the current picture includes a memory_management_control_operation equal to 5, the value of delta_pic_order_cnt_bottom shall be in the range of (1 – MaxPicOrderCntLsb) to 2^{31} - 1, inclusive.

– Otherwise (the current picture does not include a memory_management_control_operation equal to 5), the value of delta_pic_order_cnt_bottom shall be in the range of -2^{31} to 2^{31} - 1, inclusive.

When this syntax element is not present in the bitstream for the current slice, it shall be inferred to be equal to 0.

delta_pic_order_cnt[0 **]** specifies the picture order count difference from the expected picture order count for the top field of a coded frame or for a coded field as specified in subclause 8.2.1. The value of delta_pic_order_cnt[0] shall be in the range of -2^{31} to 2^{31} - 1, inclusive. When this syntax element is not present in the bitstream for the current slice, it shall be inferred to be equal to 0.

delta_pic_order_cnt[1 **]** specifies the picture order count difference from the expected picture order count for the bottom field of a coded frame specified in subclause 8.2.1. The value of delta_pic_order_cnt[1] shall be in the range of -2^{31} to 2^{31} - 1, inclusive. When this syntax element is not present in the bitstream for the current slice, it shall be inferred to be equal to 0.

redundant_pic_cnt shall be equal to 0 for slices and slice data partitions belonging to the primary coded picture. The value of redundant_pic_cnt shall be greater than 0 for coded slices or coded slice data partitions of a redundant coded picture. When redundant_pic_cnt is not present in the bitstream, its value shall be inferred to be equal to 0. The value of redundant_pic_cnt shall be in the range of 0 to 127, inclusive.

NOTE 3 – Any area of the decoded primary picture and the corresponding area that would result from application of the decoding process specified in clause 8 for any redundant picture in the same access unit should be visually similar in appearance.

The value of pic_parameter_set_id in a coded slice or coded slice data partition of a redundant coded picture shall be such that the value of pic_order_present_flag in the picture parameter set in use in a redundant coded picture is equal to the value of pic_order_present_flag in the picture parameter set in use in the corresponding primary coded picture.

When present in the primary coded picture and any redundant coded picture, the following syntax elements shall have the same value: field_pic_flag, bottom_field_flag, and idr_pic_id.

When the value of nal_ref_idc in one VCL NAL unit of an access unit is equal to 0, the value of nal_ref_idc in all other VCL NAL units of the same access unit shall be equal to 0.

NOTE 4 – The above constraint also has the following implications. If the value of nal_ref_idc for the VCL NAL units of the primary coded picture is equal to 0, the value of nal_ref_idc for the VCL NAL units of any corresponding redundant coded picture are equal to 0; otherwise (the value of nal_ref_idc for the VCL NAL units of the primary coded picture is greater than 0), the value of nal_ref_idc for the VCL NAL units of any corresponding redundant coded picture are also greater than 0.

The marking status of reference pictures and the value of frame_num after the decoded reference picture marking process as specified in subclause 8.2.5 is invoked for the primary coded picture or any redundant coded picture of the same access unit shall be identical regardless whether the primary coded picture or any redundant coded picture (instead of the primary coded picture) of the access unit would be decoded.

NOTE 5 – The above constraint also has the following implications.

If a primary coded picture is not an IDR picture, the contents of the dec_ref_pic_marking( ) syntax structure must be identical in all slice headers of the primary coded picture and all redundant coded pictures corresponding to the primary coded picture.

Otherwise (a primary coded picture is an IDR picture), the following applies.

If a redundant coded picture corresponding to the primary coded picture is an IDR picture, the contents of the dec_ref_pic_marking( ) syntax structure must be identical in all slice headers of the primary coded picture and the redundant coded picture corresponding to the primary coded picture.

Otherwise (a redundant picture corresponding to the primary coded picture is not an IDR picture), all slice headers of the redundant picture must contain a dec_ref_pic_marking syntax( ) structure including a memory_management_control_operation syntax element equal to 5, and the following applies.

If the value of long_term_reference_flag in the primary coded picture is equal to 0, the dec_ref_pic_marking syntax structure of the redundant coded picture must not include a memory_management_control_operation syntax element equal to 6.

Otherwise (the value of long_term_reference_flag in the primary coded picture is equal to 1), the dec_ref_pic_marking syntax structure of the redundant coded picture must include memory_management_control_operation syntax elements equal to 5, 4, and 6 in decoding order, and the value of max_long_term_frame_idx_plus1 must be equal to 1, and the value of long_term_frame_idx must be equal to 0.

The values of TopFieldOrderCnt and BottomFieldOrderCnt (if applicable) that result after completion of the decoding process for any redundant coded picture or the primary coded picture of the same access unit shall be identical regardless whether the primary coded picture or any redundant coded picture (instead of the primary coded picture) of the access unit would be decoded.

There is no required decoding process for a coded slice or coded slice data partition of a redundant coded picture. When the redundant_pic_cnt in the slice header of a coded slice is greater than 0, the decoder may discard the coded slice. However, a coded slice or coded slice data partition of any redundant coded picture shall obey the same constraints as a coded slice or coded slice data partition of a primary picture.

NOTE 6 – When some of the samples in the decoded primary picture cannot be correctly decoded due to errors or losses in transmission of the sequence and a coded redundant slice can be correctly decoded, the decoder should replace the samples of the decoded primary picture with the corresponding samples of the decoded redundant slice. When more than one redundant slice covers the relevant region of the primary picture, the redundant slice having the lowest value of redundant_pic_cnt should be used.

Redundant slices and slice data partitions having the same value of redundant_pic_cnt belong to the same redundant picture. Decoded slices within the same redundant picture need not cover the entire picture area and shall not overlap.

direct_spatial_mv_pred_flag specifies the method used in the decoding process to derive motion vectors and reference indices for inter prediction as follows.

– If direct_spatial_mv_pred_flag is equal to 1, the derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 in subclause 8.4.1.2 shall use spatial direct mode prediction as specified in subclause 8.4.1.2.2.

– Otherwise (direct_spatial_mv_pred_flag is equal to 0), the derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 in subclause 8.4.1.2 shall use temporal direct mode prediction as specified in subclause 8.4.1.2.3.

num_ref_idx_active_override_flag equal to 0 specifies that the values of the syntax elements num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 specified in the referred picture parameter set are in effect. num_ref_idx_active_override_flag equal to 1 specifies that the num_ref_idx_l0_active_minus1 and num_ref_idx_l1_active_minus1 specified in the referred picture parameter set are overridden for the current slice (and only for the current slice) by the following values in the slice header.

When the current slice is a P, SP, or B slice and field_pic_flag is equal to 0 and the value of num_ref_idx_l0_active_minus1 in the picture parameter set exceeds 15, num_ref_idx_active_override_flag shall be equal to 1.

When the current slice is a B slice and field_pic_flag is equal to 0 and the value of num_ref_idx_l1_active_minus1 in the picture parameter set exceeds 15, num_ref_idx_active_override_flag shall be equal to 1.

num_ref_idx_l0_active_minus1 specifies the maximum reference index for reference picture list 0 that shall be used to decode the slice.

The range of num_ref_idx_l0_active_minus1 is specified as follows.

– If field_pic_flag is equal to 0, num_ref_idx_l0_active_minus1 shall be in the range of 0 to 15, inclusive. When MbaffFrameFlag is equal to 1, num_ref_idx_l0_active_minus1 is the maximum index value for the decoding of frame macroblocks and 2 * num_ref_idx_l0_active_minus1 + 1 is the maximum index value for the decoding of field macroblocks.

– Otherwise (field_pic_flag is equal to 1), num_ref_idx_l0_active_minus1 shall be in the range of 0 to 31, inclusive.

num_ref_idx_l1_active_minus1 has the same semantics as num_ref_idx_l0_active_minus1 with l0 and list 0 replaced by l1 and list 1, respectively.

cabac_init_idc specifies the index for determining the initialisation table used in the initialisation process for context variables. The value of cabac_init_idc shall be in the range of 0 to 2, inclusive.

slice_qp_delta specifies the initial value of QP_Y to be used for all the macroblocks in the slice until modified by the value of mb_qp_delta in the macroblock layer. The initial QP_Y quantisation parameter for the slice is computed as:

$$SliceQP_Y = 26 + pic_init_qp_minus26 + slice_qp_delta \quad (7\text{-}27)$$

The value of slice_qp_delta shall be limited such that $SliceQP_Y$ is in the range of $-QpBdOffset_Y$ to +51, inclusive.

sp_for_switch_flag specifies the decoding process to be used to decode P macroblocks in an SP slice as follows.

– If sp_for_switch_flag is equal to 0, the P macroblocks in the SP slice shall be decoded using the SP decoding process for non-switching pictures as specified in subclause 8.6.1.

– Otherwise (sp_for_switch_flag is equal to 1), the P macroblocks in the SP slice shall be decoded using the SP and SI decoding process for switching pictures as specified in subclause 8.6.2.

slice_qs_delta specifies the value of QS_Y for all the macroblocks in SP and SI slices. The QS_Y quantisation parameter for the slice is computed as:

$$QS_Y = 26 + pic_init_qs_minus26 + slice_qs_delta \quad (7\text{-}28)$$

The value of slice_qs_delta shall be limited such that QS_Y is in the range of 0 to 51, inclusive. This value of QS_Y is used for the decoding of all macroblocks in SI slices with mb_type equal to SI and all macroblocks in SP slices with prediction mode equal to inter.

disable_deblocking_filter_idc specifies whether the operation of the deblocking filter shall be disabled across some block edges of the slice and specifies for which edges the filtering is disabled. When disable_deblocking_filter_idc is not present in the slice header, the value of disable_deblocking_filter_idc shall be inferred to be equal to 0.

The value of disable_deblocking_filter_idc shall be in the range of 0 to 2, inclusive.

slice_alpha_c0_offset_div2 specifies the offset used in accessing the α and t_{C0} deblocking filter tables for filtering operations controlled by the macroblocks within the slice. From this value, the offset that shall be applied when addressing these tables shall be computed as:

$$FilterOffsetA = slice_alpha_c0_offset_div2 << 1 \quad (7\text{-}29)$$

The value of slice_alpha_c0_offset_div2 shall be in the range of -6 to +6, inclusive. When slice_alpha_c0_offset_div2 is not present in the slice header, the value of slice_alpha_c0_offset_div2 shall be inferred to be equal to 0.

slice_beta_offset_div2 specifies the offset used in accessing the β deblocking filter table for filtering operations controlled by the macroblocks within the slice. From this value, the offset that is applied when addressing the β table of the deblocking filter shall be computed as:

$$FilterOffsetB = slice_beta_offset_div2 << 1 \quad (7\text{-}30)$$

The value of slice_beta_offset_div2 shall be in the range of -6 to +6, inclusive. When slice_beta_offset_div2 is not present in the slice header the value of slice_beta_offset_div2 shall be inferred to be equal to 0.

slice_group_change_cycle is used to derive the number of slice group map units in slice group 0 when slice_group_map_type is equal to 3, 4, or 5, as specified by

$$MapUnitsInSliceGroup0 = Min(slice_group_change_cycle * SliceGroupChangeRate, PicSizeInMapUnits) \quad (7\text{-}31)$$

The value of slice_group_change_cycle is represented in the bitstream by the following number of bits

$$Ceil(Log2(PicSizeInMapUnits \div SliceGroupChangeRate + 1)) \quad (7\text{-}32)$$

The value of slice_group_change_cycle shall be in the range of 0 to Ceil(PicSizeInMapUnits÷SliceGroupChangeRate), inclusive.

7.4.3.1 Reference picture list reordering semantics

The syntax elements reordering_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_num specify the change from the initial reference picture lists to the reference picture lists to be used for decoding the slice.

ref_pic_list_reordering_flag_l0 equal to 1 specifies that the syntax element reordering_of_pic_nums_idc is present for specifying reference picture list 0. ref_pic_list_reordering_flag_l0 equal to 0 specifies that this syntax element is not present.

When ref_pic_list_reordering_flag_l0 is equal to 1, the number of times that reordering_of_pic_nums_idc is not equal to 3 following ref_pic_list_reordering_flag_l0 shall not exceed num_ref_idx_l0_active_minus1 + 1.

When RefPicList0[num_ref_idx_l0_active_minus1] in the initial reference picture list produced as specified in subclause 8.2.4.2 is equal to "no reference picture", ref_pic_list_reordering_flag_l0 shall be equal to 1 and reordering_of_pic_nums_idc shall not be equal to 3 until RefPicList0[num_ref_idx_l0_active_minus1] in the reordered list produced as specified in subclause 8.2.4.3 is not equal to "no reference picture".

ref_pic_list_reordering_flag_l1 equal to 1 specifies that the syntax element reordering_of_pic_nums_idc is present for specifying reference picture list 1. ref_pic_list_reordering_flag_l1 equal to 0 specifies that this syntax element is not present.

When ref_pic_list_reordering_flag_l1 is equal to 1, the number of times that reordering_of_pic_nums_idc is not equal to 3 following ref_pic_list_reordering_flag_l1 shall not exceed num_ref_idx_l1_active_minus1 + 1.

When decoding a B slice and RefPicList1[num_ref_idx_l1_active_minus1] in the initial reference picture list produced as specified in subclause 8.2.4.2 is equal to "no reference picture", ref_pic_list_reordering_flag_l1 shall be equal to 1 and reordering_of_pic_nums_idc shall not be equal to 3 until RefPicList1[num_ref_idx_l1_active_minus1] in the reordered list produced as specified in subclause 8.2.4.3 is not equal to "no reference picture".

reordering_of_pic_nums_idc together with abs_diff_pic_num_minus1 or long_term_pic_num specifies which of the reference pictures are re-mapped. The values of reordering_of_pic_nums_idc are specified in Table 7-7. The value of the first reordering_of_pic_nums_idc that follows immediately after ref_pic_list_reordering_flag_l0 or ref_pic_list_reordering_flag_l1 shall not be equal to 3.

Table 7-7 – reordering_of_pic_nums_idc operations for reordering of reference picture lists

reordering_of_pic_nums_idc	Reordering specified
0	abs_diff_pic_num_minus1 is present and corresponds to a difference to subtract from a picture number prediction value
1	abs_diff_pic_num_minus1 is present and corresponds to a difference to add to a picture number prediction value
2	long_term_pic_num is present and specifies the long-term picture number for a reference picture
3	End loop for reordering of the initial reference picture list

abs_diff_pic_num_minus1 plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value. abs_diff_pic_num_minus1 shall be in the range of 0 to MaxPicNum – 1. The allowed values of abs_diff_pic_num_minus1 are further restricted as specified in subclause 8.2.4.3.1.

long_term_pic_num specifies the long-term picture number of the picture being moved to the current index in the list. When decoding a coded frame, long_term_pic_num shall be equal to a LongTermPicNum assigned to one of the reference frames or complementary reference field pairs marked as "used for long-term reference". When decoding a coded field, long_term_pic_num shall be equal to a LongTermPicNum assigned to one of the reference fields marked as "used for long-term reference".

7.4.3.2 Prediction weight table semantics

luma_log2_weight_denom is the base 2 logarithm of the denominator for all luma weighting factors. The value of luma_log2_weight_denom shall be in the range of 0 to 7, inclusive.

chroma_log2_weight_denom is the base 2 logarithm of the denominator for all chroma weighting factors. The value of chroma_log2_weight_denom shall be in the range of 0 to 7, inclusive.

luma_weight_l0_flag equal to 1 specifies that weighting factors for the luma component of list 0 prediction are present. luma_weight_l0_flag equal to 0 specifies that these weighting factors are not present.

luma_weight_l0[i **]** is the weighting factor applied to the luma prediction value for list 0 prediction using RefPicList0[i]. When luma_weight_l0_flag is equal to 1, the value of luma_weight_l0[i] shall be in the range of –128 to 127, inclusive. When luma_weight_l0_flag is equal to 0, luma_weight_l0[i] shall be inferred to be equal to $2^{\text{luma_log2_weight_denom}}$ for RefPicList0[i].

luma_offset_l0[i **]** is the additive offset applied to the luma prediction value for list 0 prediction using RefPicList0[i]. The value of luma_offset_l0[i] shall be in the range of –128 to 127, inclusive. When luma_weight_l0_flag is equal to 0, luma_offset_l0[i] shall be inferred as equal to 0 for RefPicList0[i].

chroma_weight_l0_flag equal to 1 specifies that weighting factors for the chroma prediction values of list 0 prediction are present. chroma_weight_l0_flag equal to 0 specifies that these weighting factors are not present.

chroma_weight_l0[i][j] is the weighting factor applied to the chroma prediction values for list 0 prediction using RefPicList0[i] with j equal to 0 for Cb and j equal to 1 for Cr. When chroma_weight_l0_flag is equal to 1, the value of chroma_weight_l0[i][j] shall be in the range of –128 to 127, inclusive. When chroma_weight_l0_flag is equal to 0**,** chroma_weight_l0[i][j] shall be inferred to be equal to $2^{\text{chroma_log2_weight_denom}}$ for RefPicList0[i].

chroma_offset_l0[i][j] is the additive offset applied to the chroma prediction values for list 0 prediction using RefPicList0[i] with j equal to 0 for Cb and j equal to 1 for Cr. The value of chroma_offset_l0[i][j] shall be in the range of -128 to 127, inclusive. When chroma_weight_l0_flag is equal to 0**,** chroma_offset_l0[i][j] shall be inferred to be equal to 0 for RefPicList0[i].

luma_weight_l1_flag, luma_weight_l1, **luma_offset_l1**, **chroma_weight_l1_flag**, **chroma_weight_l1**, **chroma_offset_l1** have the same semantics as luma_weight_l0_flag, luma_weight_l0, luma_offset_l0, chroma_weight_l0_flag, chroma_weight_l0, chroma_offset_l0, respectively, with l0, list 0, and List0 replaced by l1, list 1, and List1, respectively.

7.4.3.3 Decoded reference picture marking semantics

The syntax elements no_output_of_prior_pics_flag, long_term_reference_flag, adaptive_ref_pic_marking_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_frame_idx, long_term_pic_num, and max_long_term_frame_idx_plus1 specify marking of the reference pictures.

The marking of a reference picture can be "unused for reference", "used for short-term reference", or "used for long-term reference", but only one among these three. When a reference picture is referred to as being marked as "used for reference", this collectively refers to the picture being marked as "used for short-term reference" or "used for long-term reference" (but not both). A reference picture that is marked as "used for short-term reference" is referred to as a short-term reference picture. A reference picture that is marked as "used for long-term reference" is referred to as a long-term reference picture.

The syntax element adaptive_ref_pic_marking_mode_flag and the content of the decoded reference picture marking syntax structure shall be identical for all coded slices of a coded picture.

The syntax category of the decoded reference picture marking syntax structure shall be inferred as follows.

– If the decoded reference picture marking syntax structure is in a slice header, the syntax category of the decoded reference picture marking syntax structure shall be inferred to be equal to 2.

– Otherwise (the decoded reference picture marking syntax structure is in a decoded reference picture marking repetition SEI message as specified in Annex D), the syntax category of the decoded reference picture marking syntax structure shall be inferred to be equal to 5.

no_output_of_prior_pics_flag specifies how the previously-decoded pictures in the decoded picture buffer are treated after decoding of an IDR picture. See Annex C. When the IDR picture is the first IDR picture in the bitstream, the value of no_output_of_prior_pics_flag has no effect on the decoding process. When the IDR picture is not the first IDR picture in the bitstream and the value of PicWidthInMbs, FrameHeightInMbs, or max_dec_frame_buffering derived from the active sequence parameter set is different from the value of PicWidthInMbs, FrameHeightInMbs, or max_dec_frame_buffering derived from the sequence parameter set active for the preceding sequence, no_output_of_prior_pics_flag equal to 1 may be inferred by the decoder, regardless of the actual value of no_output_of_prior_pics_flag.

long_term_reference_flag equal to 0 specifies that the MaxLongTermFrameIdx variable is set equal to “no long-term frame indices” and that the IDR picture is marked as “used for short-term reference”. long_term_reference_flag equal to 1 specifies that the MaxLongTermFrameIdx variable is set equal to 0 and that the current IDR picture is marked “used for long-term reference” and is assigned LongTermFrameIdx equal to 0. When num_ref_frames is equal to 0, long_term_reference_flag shall be equal to 0.

adaptive_ref_pic_marking_mode_flag selects the reference picture marking mode of the currently decoded picture as specified in Table 7-8. adaptive_ref_pic_marking_mode_flag shall be equal to 1 when the number of frames, complementary field pairs, and non-paired fields that are currently marked as "used for long-term reference" is equal to Max(num_ref_frames, 1).

Table 7-8 – Interpretation of adaptive_ref_pic_marking_mode_flag

adaptive_ref_pic_marking_mode_flag	Reference picture marking mode specified
0	Sliding window reference picture marking mode: A marking mode providing a first-in first-out mechanism for short-term reference pictures.
1	Adaptive reference picture marking mode: A reference picture marking mode providing syntax elements to specify marking of reference pictures as "unused for reference" and to assign long-term frame indices.

memory_management_control_operation specifies a control operation to be applied to affect the reference picture marking. The memory_management_control_operation syntax element is followed by data necessary for the operation specified by the value of memory_management_control_operation. The values and control operations associated with memory_management_control_operation are specified in Table 7-9. The memory_management_control_operation syntax elements are processed by the decoding process in the order in which they appear in the slice header, and the semantics constraints expressed for each memory_management_control_operation apply at the specific position in that order at which that individual memory_management_control_operation is processed.

For interpretation of memory_management_control_operation, the term reference picture is interpreted as follows.

– If the current picture is a frame, the term reference picture refers either to a reference frame or a complementary reference field pair.

– Otherwise (the current picture is a field), the term reference picture refers either to a reference field or a field of a reference frame.

memory_management_control_operation shall not be equal to 1 in a slice header unless the specified reference picture is marked as "used for short-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 2 in a slice header unless the specified long-term picture number refers to a reference picture that is marked as "used for long-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 3 in a slice header unless the specified reference picture is marked as "used for short-term reference" when the memory_management_control_operation is processed by the decoding process.

memory_management_control_operation shall not be equal to 3 or 6 if the value of the variable MaxLongTermFrameIdx is equal to "no long-term frame indices" when the memory_management_control_operation is processed by the decoding process.

Not more than one memory_management_control_operation equal to 4 shall be present in a slice header.

Not more than one memory_management_control_operation equal to 5 shall be present in a slice header.

Not more than one memory_management_control_operation equal to 6 shall be present in a slice header.

memory_management_control_operation shall not be equal to 5 in a slice header unless no memory_management_control_operation in the range of 1 to 3 is present in the same decoded reference picture marking syntax structure.

A memory_management_control_operation equal to 5 shall not follow a memory_management_control_operation equal to 6 in the same slice header.

When a memory_management_control_operation equal to 6 is present, any memory_management_control_operation equal to 2, 3, or 4 that follows the memory_management_control_operation equal to 6 within the same slice header shall not specify the current picture to be marked as "unused for reference".

NOTE 1 – These constraints prohibit any combination of multiple memory_management_control_operation syntax elements that would specify the current picture to be marked as "unused for reference". However, some other combinations of memory_management_control_operation syntax elements are permitted that may affect the marking status of other reference pictures more than once in the same slice header. In particular, it is permitted for a memory_management_control_operation equal to 3 that specifies a long-term frame index to be assigned to a particular short-term reference picture to be followed in the same slice header by a memory_management_control_operation equal to 2, 3, 4 or 6 that specifies the same reference picture to subsequently be marked as "unused for reference".

Table 7-9 – Memory management control operation (memory_management_control_operation) values

memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation syntax element loop
1	Mark a short-term reference picture as “unused for reference”
2	Mark a long-term reference picture as “unused for reference”
3	Mark a short-term reference picture as "used for long-term reference" and assign a long-term frame index to it
4	Specify the maximum long-term frame index and mark all long-term reference pictures having long-term frame indices greater than the maximum value as "unused for reference"
5	Mark all reference pictures as "unused for reference" and set the MaxLongTermFrameIdx variable to "no long-term frame indices"
6	Mark the current picture as "used for long-term reference" and assign a long-term frame index to it

When decoding a field and a memory_management_control_operation command equal to 3 is present that assigns a long-term frame index to a field that is part of a short-term reference frame or part of a short-term complementary reference field pair, another memory_management_control_operation command to assign the same long-term frame index to the other field of the same frame or complementary reference field pair shall be present in the same decoded reference picture marking syntax structure.

NOTE 2 – The above requirement must be fulfilled even when the field referred to by the memory_management_control_operation equal to 3 is subsequently marked as "unused for reference" (for example when a memory_management_control_operation equal to 2 is present in the same slice header that causes the field to be marked as "unused for reference").

When the first field (in decoding order) of a complementary reference field pair includes a long_term_reference_flag equal to 1 or a memory_management_control_operation command equal to 6, the decoded reference picture marking syntax structure for the other field of the complementary reference field pair shall contain a memory_management_control_operation command equal to 6 that assigns the same long-term frame index to the other field.

NOTE 3 – The above requirement must be fulfilled even when the first field of the complementary reference field pair is subsequently marked as "unused for reference" (for example, when a memory_management_control_operation equal to 2 is present in the slice header of the second field that causes the first field to be marked as "unused for reference").

difference_of_pic_nums_minus1 is used (with memory_management_control_operation equal to 3 or 1) to assign a long-term frame index to a short-term reference picture or to mark a short-term reference picture as “unused for reference”. When the associated memory_management_control_operation is processed by the decoding process, the resulting picture number derived from difference_of_pic_nums_minus1 shall be a picture number assigned to one of the reference pictures marked as "used for reference" and not previously assigned to a long-term frame index.

The resulting picture number is constrained as follows.

– If field_pic_flag is equal to 0, the resulting picture number shall be one of the set of picture numbers assigned to reference frames or complementary reference field pairs.

  NOTE 4 – When field_pic_flag is equal to 0, the resulting picture number must be a picture number assigned to a complementary reference field pair in which both fields are marked as "used for reference" or a frame in which both fields are marked as "used for reference". In particular, when field_pic_flag is equal to 0, the marking of a non-paired field or a frame in which a single field is marked as "used for reference" cannot be affected by a memory_management_control_operation equal to 1.

– Otherwise (field_pic_flag is equal to 1), the resulting picture number shall be one of the set of picture numbers assigned to reference fields.

long_term_pic_num is used (with memory_management_control_operation equal to 2) to mark a long-term reference picture as "unused for reference". When the associated memory_management_control_operation is processed by the decoding process, long_term_pic_num shall be equal to a long-term picture number assigned to one of the reference pictures that is currently marked as "used for long-term reference".

The resulting long-term picture number is constrained as follows.

– If field_pic_flag is equal to 0, the resulting long-term picture number shall be one of the set of long-term picture numbers assigned to reference frames or complementary reference field pairs.

> NOTE 5 – When field_pic_flag is equal to 0, the resulting long-term picture number must be a long-term picture number assigned to a complementary reference field pair in which both fields are marked as "used for reference" or a frame in which both fields are marked as "used for reference". In particular, when field_pic_flag is equal to 0, the marking of a non-paired field or a frame in which a single field is marked as "used for reference" cannot be affected by a memory_management_control_operation equal to 2.

– Otherwise (field_pic_flag is equal to 1), the resulting long-term picture number shall be one of the set of long-term picture numbers assigned to reference fields.

long_term_frame_idx is used (with memory_management_control_operation equal to 3 or 6) to assign a long-term frame index to a picture. When the associated memory_management_control_operation is processed by the decoding process, the value of long_term_frame_idx shall be in the range of 0 to MaxLongTermFrameIdx, inclusive.

max_long_term_frame_idx_plus1 minus 1 specifies the maximum value of long-term frame index allowed for long-term reference pictures (until receipt of another value of max_long_term_frame_idx_plus1). The value of max_long_term_frame_idx_plus1 shall be in the range of 0 to num_ref_frames, inclusive.

7.4.4 Slice data semantics

cabac_alignment_one_bit is a bit equal to 1.

mb_skip_run specifies the number of consecutive skipped macroblocks for which, when decoding a P or SP slice, mb_type shall be inferred to be P_Skip and the macroblock type is collectively referred to as a P macroblock type, or for which, when decoding a B slice, mb_type shall be inferred to be B_Skip and the macroblock type is collectively referred to as a B macroblock type. The value of mb_skip_run shall be in the range of 0 to PicSizeInMbs – CurrMbAddr, inclusive.

mb_skip_flag equal to 1 specifies that for the current macroblock, when decoding a P or SP slice, mb_type shall be inferred to be P_Skip and the macroblock type is collectively referred to as P macroblock type, or for which, when decoding a B slice, mb_type shall be inferred to be B_Skip and the macroblock type is collectively referred to as B macroblock type. mb_skip_flag equal to 0 specifies that the current macroblock is not skipped.

mb_field_decoding_flag equal to 0 specifies that the current macroblock pair is a frame macroblock pair. mb_field_decoding_flag equal to 1 specifies that the macroblock pair is a field macroblock pair. Both macroblocks of a frame macroblock pair are referred to in the text as frame macroblocks, whereas both macroblocks of a field macroblock pair are referred to in the text as field macroblocks.

When mb_field_decoding_flag is not present for either macroblock of a macroblock pair, the value of mb_field_decoding_flag is derived as follows.

– If there is a neighbouring macroblock pair immediately to the left of the current macroblock pair in the same slice, the value of mb_field_decoding_flag shall be inferred to be equal to the value of mb_field_decoding_flag for the neighbouring macroblock pair immediately to the left of the current macroblock pair,

– Otherwise, if there is no neighbouring macroblock pair immediately to the left of the current macroblock pair in the same slice and there is a neighbouring macroblock pair immediately above the current macroblock pair in the same slice, the value of mb_field_decoding_flag shall be inferred to be equal to the value of mb_field_decoding_flag for the neighbouring macroblock pair immediately above the current macroblock pair,

– Otherwise (there is no neighbouring macroblock pair either immediately to the left or immediately above the current macroblock pair in the same slice), the value of mb_field_decoding_flag shall be inferred to be equal to 0.

end_of_slice_flag equal to 0 specifies that another macroblock is following in the slice. end_of_slice_flag equal to 1 specifies the end of the slice and that no further macroblock follows.

The function NextMbAddress( ) used in the slice data syntax table is specified in subclause 8.2.2.

7.4.5 Macroblock layer semantics

mb_type specifies the macroblock type. The semantics of mb_type depend on the slice type.

Tables and semantics are specified for the various macroblock types for I, SI, P, SP, and B slices. Each table presents the value of mb_type, the name of mb_type, the number of macroblock partitions used (given by the NumMbPart(mb_type) function), the prediction mode of the macroblock (when it is not partitioned) or the first partition (given by the MbPartPredMode(mb_type, 0) function) and the prediction mode of the second partition (given by the MbPartPredMode(mb_type, 1) function). When a value is not applicable it is designated by "na". In the text, the value of mb_type may be referred to as the macroblock type and a value X of MbPartPredMode( ) may be referred to in the text by "X macroblock (partition) prediction mode" or as "X prediction macroblocks".

Table 7-10 shows the allowed collective macroblock types for each slice_type.

NOTE 1 – There are some macroblock types with Pred_L0 prediction mode that are classified as B macroblock types.

Table 7-10 – Allowed collective macroblock types for slice_type

slice_type	allowed collective macroblock types
I (slice)	I (see Table 7-11) (macroblock types)
P (slice)	P (see Table 7-13) and I (see Table 7-11) (macroblock types)
B (slice)	B (see Table 7-14) and I (see Table 7-11) (macroblock types)
SI (slice)	SI (see Table 7-12) and I (see Table 7-11) (macroblock types)
SP (slice)	P (see Table 7-13) and I (see Table 7-11) (macroblock types)

transform_size_8x8_flag equal to 1 specifies that for the current macroblock the transform coefficient decoding process and picture construction process prior to deblocking filter process for residual 8x8 blocks shall be invoked for luma samples. transform_size_8x8_flag equal to 0 specifies that for the current macroblock the transform coefficient decoding process and picture construction process prior to deblocking filter process for residual 4x4 blocks shall be invoked for luma samples. When transform_size_8x8_flag is not present in the bitstream, it shall be inferred to be equal to 0.

NOTE 2 – When the current macroblock prediction mode MbPartPredMode(mb_type, 0) is equal to Intra_16x16, transform_size_8x8_flag is not present in the bitstream and then inferred to be equal to 0.

When sub_mb_type[mbPartIdx] (see subclause 7.4.5.2) is present in the bitstream for all 8x8 blocks indexed by mbPartIdx = 0..3, the variable noSubMbPartSizeLessThan8x8Flag indicates whether for each of the four 8x8 blocks the corresponding SubMbPartWidth(sub_mb_type[mbPartIdx]) and SubMbPartHeight(sub_mb_type[mbPartIdx]) are both equal to 8.

NOTE 3 – When noSubMbPartSizeLessThan8x8Flag is equal to 0 and the current macroblock type is not equal to I_NxN, transform_size_8x8_flag is not present in the bitstream and then inferred to be equal to 0.

Macroblock types that may be collectively referred to as I macroblock types are specified in Table 7-11.

The macroblock types for I slices are all I macroblock types.

Table 7-11 – Macroblock types for I slices

mb_type	Name of mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	I_NxN	0	Intra_4x4	na	Equation 7-33	Equation 7-33
0	I_NxN	1	Intra_8x8	na	Equation 7-33	Equation 7-33
1	I_16x16_0_0_0	na	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15
25	I_PCM	na	na	na	na	na

The following semantics are assigned to the macroblock types in Table 7-11.

I_NxN: A mnemonic name for mb_type equal to 0 with MbPartPredMode(mb_type, 0) equal to Intra_4x4 or Intra_8x8.

I_16x16_0_0_0, I_16x16_1_0_0, I_16x16_2_0_0, I_16x16_3_0_0, I_16x16_0_1_0, I_16x16_1_1_0, I_16x16_2_1_0, I_16x16_3_1_0, I_16x16_0_2_0, I_16x16_1_2_0, I_16x16_2_2_0, I_16x16_3_2_0, I_16x16_0_0_1, I_16x16_1_0_1, I_16x16_2_0_1, I_16x16_3_0_1, I_16x16_0_1_1, I_16x16_1_1_1, I_16x16_2_1_1, I_16x16_3_1_1, I_16x16_0_2_1, I_16x16_1_2_1, I_16x16_2_2_1, I_16x16_3_2_1: the macroblock is coded as an Intra_16x16 prediction mode macroblock.

To each Intra_16x16 prediction macroblock, an Intra16x16PredMode is assigned, which specifies the Intra_16x16 prediction mode. CodedBlockPatternChroma contains the coded block pattern value for chroma as specified in Table 7-15. When chroma_format_idc is equal to 0, CodedBlockPatternChroma shall be equal to 0. CodedBlockPatternLuma specifies whether, for the luma component, non-zero AC transform coefficient levels are present. CodedBlockPatternLuma equal to 0 specifies that all AC transform coefficient levels in the luma component of the macroblock are equal to 0. CodedBlockPatternLuma equal to 15 specifies that at least one of the AC transform coefficient levels in the luma component of the macroblock is non-zero, requiring scanning of AC transform coefficient levels for all 16 of the 4x4 blocks in the 16x16 block.

Intra_4x4 specifies the macroblock prediction mode and specifies that the Intra_4x4 prediction process is invoked as specified in subclause 8.3.1. Intra_4x4 is an Intra macroblock prediction mode.

Intra_8x8 specifies the macroblock prediction mode and specifies that the Intra_8x8 prediction process is invoked as specified in subclause 8.3.2. Intra_8x8 is an Intra macroblock prediction mode.

Intra_16x16 specifies the macroblock prediction mode and specifies that the Intra_16x16 prediction process is invoked as specified in subclause 8.3.3. Intra_16x16 is an Intra macroblock prediction mode.

For a macroblock coded with mb_type equal to I_PCM, the Intra macroblock prediction mode shall be inferred.

A macroblock type that may be referred to as SI macroblock type is specified in Table 7-12.

The macroblock types for SI slices are specified in Tables 7-12 and 7-11. The mb_type value 0 is specified in Table 7-12 and the mb_type values 1 to 26 are specified in Table 7-11, indexed by subtracting 1 from the value of mb_type.

Table 7-12 – Macroblock type with value 0 for SI slices

mb_type	Name of mb_type	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	SI	Intra_4x4	na	Equation 7-33	Equation 7-33

The following semantics are assigned to the macroblock type in Table 7-12. The SI macroblock is coded as Intra_4x4 prediction macroblock.

Macroblock types that may be collectively referred to as P macroblock types are specified in Table 7-13.

The macroblock types for P and SP slices are specified in Tables 7-13 and 7-11. mb_type values 0 to 4 are specified in Table 7-13 and mb_type values 5 to 30 are specified in Table 7-11, indexed by subtracting 5 from the value of mb_type.

Table 7-13 – Macroblock type values 0 to 4 for P and SP slices

mb_type	Name of mb_type	NumMbPart (mb_type)	MbPartPredMode (mb_type, 0)	MbPartPredMode (mb_type, 1)	MbPartWidth (mb_type)	MbPartHeight (mb_type)
0	P_L0_16x16	1	Pred_L0	na	16	16
1	P_L0_L0_16x8	2	Pred_L0	Pred_L0	16	8
2	P_L0_L0_8x16	2	Pred_L0	Pred_L0	8	16
3	P_8x8	4	na	na	8	8
4	P_8x8ref0	4	na	na	8	8
inferred	P_Skip	1	Pred_L0	na	16	16

The following semantics are assigned to the macroblock types in Table 7-13.

– P_L0_16x16: the samples of the macroblock are predicted with one luma macroblock partition of size 16x16 luma samples and associated chroma samples.

– P_L0_L0_MxN, with MxN being replaced by 16x8 or 8x16: the samples of the macroblock are predicted using two luma partitions of size MxN equal to 16x8, or two luma partitions of size MxN equal to 8x16, and associated chroma samples, respectively.

– P_8x8: for each sub-macroblock an additional syntax element (sub_mb_type) is present in the bitstream that specifies the type of the corresponding sub-macroblock (see subclause 7.4.5.2).

– P_8x8ref0: has the same semantics as P_8x8 but no syntax element for the reference index (ref_idx_l0) is present in the bitstream and ref_idx_l0[mbPartIdx] shall be inferred to be equal to 0 for all sub-macroblocks of the macroblock (with indices mbPartIdx equal to 0..3).

– P_Skip: no further data is present for the macroblock in the bitstream.

The following semantics are assigned to the macroblock prediction modes (MbPartPredMode( )) in Table 7-13.

– Pred_L0: specifies that the inter prediction process is invoked using list 0 prediction. Pred_L0 is an Inter macroblock prediction mode.

Macroblock types that may be collectively referred to as B macroblock types are specified in Table 7-14.

The macroblock types for B slices are specified in Tables 7-14 and 7-11. The mb_type values 0 to 22 are specified in Table 7-14 and the mb_type values 23 to 48 are specified in Table 7-11, indexed by subtracting 23 from the value of mb_type.

Table 7-14 – Macroblock type values 0 to 22 for B slices

mb_type	Name of mb_type	NumMbPart (mb_type)	MbPartPredMode (mb_type, 0)	MbPartPredMode (mb_type, 1)	MbPartWidth (mb_type)	MbPartHeight (mb_type)
0	B_Direct_16x16	na	Direct	na	8	8
1	B_L0_16x16	1	Pred_L0	na	16	16
2	B_L1_16x16	1	Pred_L1	na	16	16
3	B_Bi_16x16	1	BiPred	na	16	16
4	B_L0_L0_16x8	2	Pred_L0	Pred_L0	16	8
5	B_L0_L0_8x16	2	Pred_L0	Pred_L0	8	16
6	B_L1_L1_16x8	2	Pred_L1	Pred_L1	16	8
7	B_L1_L1_8x16	2	Pred_L1	Pred_L1	8	16
8	B_L0_L1_16x8	2	Pred_L0	Pred_L1	16	8
9	B_L0_L1_8x16	2	Pred_L0	Pred_L1	8	16
10	B_L1_L0_16x8	2	Pred_L1	Pred_L0	16	8
11	B_L1_L0_8x16	2	Pred_L1	Pred_L0	8	16
12	B_L0_Bi_16x8	2	Pred_L0	BiPred	16	8
13	B_L0_Bi_8x16	2	Pred_L0	BiPred	8	16
14	B_L1_Bi_16x8	2	Pred_L1	BiPred	16	8
15	B_L1_Bi_8x16	2	Pred_L1	BiPred	8	16
16	B_Bi_L0_16x8	2	BiPred	Pred_L0	16	8
17	B_Bi_L0_8x16	2	BiPred	Pred_L0	8	16
18	B_Bi_L1_16x8	2	BiPred	Pred_L1	16	8
19	B_Bi_L1_8x16	2	BiPred	Pred_L1	8	16
20	B_Bi_Bi_16x8	2	BiPred	BiPred	16	8
21	B_Bi_Bi_8x16	2	BiPred	BiPred	8	16
22	B_8x8	4	na	na	8	8
inferred	B_Skip	na	Direct	na	8	8

The following semantics are assigned to the macroblock types in Table 7-14:

– B_Direct_16x16: no motion vector differences or reference indices are present for the macroblock in the bitstream. The functions MbPartWidth(B_Direct_16x16), and MbPartHeight(B_Direct_16x16) are used in the derivation process for motion vectors and reference frame indices in subclause 8.4.1 for direct mode prediction.

– B_X_16x16 with X being replaced by L0, L1, or Bi: the samples of the macroblock are predicted with one luma macroblock partition of size 16x16 luma samples and associated chroma samples. For a macroblock with type B_X_16x16 with X being replaced by either L0 or L1, one motion vector difference and one reference index is

present in the bitstream for the macroblock. For a macroblock with type B_X_16x16 with X being replaced by Bi, two motion vector differences and two reference indices are present in the bitstream for the macroblock.

– B_X0_X1_MxN, with X0, X1 referring to the first and second macroblock partition and being replaced by L0, L1, or Bi, and MxN being replaced by 16x8 or 8x16: the samples of the macroblock are predicted using two luma partitions of size MxN equal to 16x8, or two luma partitions of size MxN equal to 8x16, and associated chroma samples, respectively. For a macroblock partition X0 or X1 with X0 or X1 being replaced by either L0 or L1, one motion vector difference and one reference index is present in the bitstream. For a macroblock partition X0 or X1 with X0 or X1 being replaced by Bi, two motion vector differences and two reference indices are present in the bitstream for the macroblock partition.

– B_8x8: for each sub-macroblock an additional syntax element (sub_mb_type) is present in the bitstream that specifies the type of the corresponding sub-macroblock (see subclause 7.4.5.2).

– B_Skip: no further data is present for the macroblock in the bitstream. The functions MbPartWidth(B_Skip), and MbPartHeight(B_Skip) are used in the derivation process for motion vectors and reference frame indices in subclause 8.4.1 for direct mode prediction.

The following semantics are assigned to the macroblock prediction modes (MbPartPredMode( )) in Table 7-14.

– Direct: no motion vector differences or reference indices are present for the macroblock (in case of B_Skip or B_Direct_16x16) in the bitstream. Direct is an Inter macroblock prediction mode.

– Pred_L0: see semantics for Table 7-13.

– Pred_L1: specifies that the Inter prediction process is invoked using list 1 prediction. Pred_L1 is an Inter macroblock prediction mode.

– BiPred: specifies that the Inter prediction process is invoked using list 0 and list 1 prediction. BiPred is an Inter macroblock prediction mode.

pcm_alignment_zero_bit is a bit equal to 0.

pcm_sample_luma[i] is a sample value. The first pcm_sample_luma[i] values represent luma sample values in the raster scan within the macroblock. The number of bits used to represent each of these samples is $BitDepth_Y$. When profile_idc is not equal to 100, 110, 122, or 144, pcm_sample_luma[i] shall not be equal to 0.

pcm_sample_chroma[i] is a sample value. The first MbWidthC * MbHeightC pcm_sample_chroma[i] values represent Cb sample values in the raster scan within the macroblock and the remaining MbWidthC * MbHeightC pcm_sample_chroma[i] values represent Cr sample values in the raster scan within the macroblock. The number of bits used to represent each of these samples is $BitDepth_C$. When profile_idc is not equal to 100, 110, 122, or 144, pcm_sample_chroma[i] shall not be equal to 0.

coded_block_pattern specifies which of the four 8x8 luma blocks and associated chroma blocks of a macroblock may contain non-zero transform coefficient levels. For macroblocks with prediction mode not equal to Intra_16x16, coded_block_pattern is present in the bitstream and the variables CodedBlockPatternLuma and CodedBlockPatternChroma are derived as follows.

```
CodedBlockPatternLuma = coded_block_pattern % 16
CodedBlockPatternChroma = coded_block_pattern / 16                    (7-33)
```

When coded_block_pattern is present, CodedBlockPatternLuma specifies, for each of the four 8x8 luma blocks of the macroblock, one of the following cases.

– All transform coefficient levels of the four 4x4 luma blocks in the 8x8 luma block are equal to zero

– One or more transform coefficient levels of one or more of the 4x4 luma blocks in the 8x8 luma block shall be non-zero valued.

The meaning of CodedBlockPatternChroma is specified in Table 7-15.

Table 7-15 – Specification of CodedBlockPatternChroma values

CodedBlockPatternChroma	Description
0	All chroma transform coefficient levels are equal to 0.
1	One or more chroma DC transform coefficient levels shall be non-zero valued. All chroma AC transform coefficient levels are equal to 0.
2	Zero or more chroma DC transform coefficient levels are non-zero valued. One or more chroma AC transform coefficient levels shall be non-zero valued.

mb_qp_delta can change the value of QP_Y in the macroblock layer. The decoded value of mb_qp_delta shall be in the range of $-(26 + QpBdOffset_Y / 2)$ to $+(25 + QpBdOffset_Y / 2)$, inclusive. mb_qp_delta shall be inferred to be equal to 0 when it is not present for any macroblock (including P_Skip and B_Skip macroblock types).

The value of QP_Y is derived as

$$QP_Y = ((QP_{Y,PREV} + mb_qp_delta + 52 + 2 * QpBdOffset_Y) \% (52 + QpBdOffset_Y)) - QpBdOffset_Y \quad (7\text{-}34)$$

where $QP_{Y,PREV}$ is the luma quantisation parameter, QP_Y, of the previous macroblock in decoding order in the current slice. For the first macroblock in the slice $QP_{Y,PREV}$ is initially set equal to $SliceQP_Y$ derived in Equation 7-27 at the start of each slice.

The value of QP'_Y is derived as

$$QP'_Y = QP_Y + QpBdOffset_Y \quad (7\text{-}35)$$

7.4.5.1 Macroblock prediction semantics

All samples of the macroblock are predicted. The prediction modes are derived using the following syntax elements.

prev_intra4x4_pred_mode_flag[luma4x4BlkIdx **]** and **rem_intra4x4_pred_mode[** luma4x4BlkIdx **]** specify the Intra_4x4 prediction of the 4x4 luma block with index luma4x4BlkIdx = 0..15.

prev_intra8x8_pred_mode_flag[luma8x8BlkIdx] and **rem_intra8x8_pred_mode[** luma8x8BlkIdx] specify the Intra_8x8 prediction of the 8x8 luma block with index luma8x8BlkIdx = 0..3.

intra_chroma_pred_mode specifies the type of spatial prediction used for chroma in macroblocks using Intra_4x4 or Intra_16x16 prediction, as shown in Table 7-16. The value of intra_chroma_pred_mode shall be in the range of 0 to 3, inclusive.

Table 7-16 – Relationship between intra_chroma_pred_mode and spatial prediction modes

intra_chroma_pred_mode	Intra Chroma Prediction Mode
0	DC
1	Horizontal
2	Vertical
3	Plane

ref_idx_l0[mbPartIdx **]** when present, specifies the index in reference picture list 0 of the reference picture to be used for prediction.

The range of ref_idx_l0[mbPartIdx], the index in list 0 of the reference picture, and, if applicable, the parity of the field within the reference picture used for prediction are specified as follows.

– If MbaffFrameFlag is equal to 0 or mb_field_decoding_flag is equal to 0, the value of ref_idx_l0[mbPartIdx] shall be in the range of 0 to num_ref_idx_l0_active_minus1, inclusive.

– Otherwise (MbaffFrameFlag is equal to 1 and mb_field_decoding_flag is equal to 1), the value of ref_idx_l0[mbPartIdx] shall be in the range of 0 to 2 * num_ref_idx_l0_active_minus1 + 1, inclusive.

When only one reference picture is used for inter prediction, the values of ref_idx_l0[mbPartIdx] shall be inferred to be equal to 0.

ref_idx_l1[mbPartIdx **]** has the same semantics as ref_idx_l0, with l0 and list 0 replaced by l1 and list 1, respectively.

mvd_l0[mbPartIdx **][** 0 **][** compIdx **]** specifies the difference between a vector component to be used and its prediction. The index mbPartIdx specifies to which macroblock partition mvd_l0 is assigned. The partitioning of the macroblock is specified by mb_type. The horizontal motion vector component difference is decoded first in decoding order and is assigned CompIdx = 0. The vertical motion vector component is decoded second in decoding order and is assigned CompIdx = 1. The range of the components of mvd_l0[mbPartIdx][0][compIdx] is specified by constraints on the motion vector variable values derived from it as specified in Annex A.

mvd_l1[mbPartIdx **][** 0 **][** compIdx **]** has the same semantics as mvd_l0, with l0 and L0 replaced by l1 and L1, respectively.

7.4.5.2 Sub-macroblock prediction semantics

sub_mb_type[mbPartIdx **]** specifies the sub-macroblock types.

Tables and semantics are specified for the various sub-macroblock types for P, and B macroblock types. Each table presents the value of sub_mb_type, the name of sub_mb_type, the number of sub-macroblock partitions used (given by the NumSubMbPart(sub_mb_type) function), and the prediction mode of the sub-macroblock (given by the SubMbPredMode(sub_mb_type) function). In the text, the value of sub_mb_type may be referred to by "sub-macroblock type". In the text, the value of SubMbPredMode( ) may be referred to by "sub-macroblock prediction mode".

The interpretation of sub_mb_type[mbPartIdx] for P macroblock types is specified in Table 7-17, where the row for "inferred" specifies values inferred when sub_mb_type[mbPartIdx] is not present.

Table 7-17 – Sub-macroblock types in P macroblocks

sub_mb_type[mbPartIdx]	Name of sub_mb_type[mbPartIdx]	NumSubMbPart (sub_mb_type[mbPartIdx])	SubMbPredMode (sub_mb_type[mbPartIdx])	SubMbPartWidth (sub_mb_type[mbPartIdx])	SubMbPartHeight (sub_mb_type[mbPartIdx])
inferred	na	na	na	na	na
0	P_L0_8x8	1	Pred_L0	8	8
1	P_L0_8x4	2	Pred_L0	8	4
2	P_L0_4x8	2	Pred_L0	4	8
3	P_L0_4x4	4	Pred_L0	4	4

The following semantics are assigned to the sub-macroblock types in Table 7-17.

– P_L0_MxN, with MxN being replaced by 8x8, 8x4, 4x8, or 4x4: the samples of the sub-macroblock are predicted using one luma partition of size MxN equal to 8x8, two luma partitions of size MxN equal to 8x4, or two luma partitions of size MxN equal to 4x8, or four luma partitions of size MxN equal to 4x4, and associated chroma samples, respectively.

The following semantics are assigned to the sub-macroblock prediction modes (SubMbPredMode( )) in Table 7-17.

– Pred_L0: see semantics for Table 7-13.

The interpretation of sub_mb_type[mbPartIdx] for B macroblock types is specified in Table 7-18, where the row for "inferred" specifies values inferred when sub_mb_type[mbPartIdx] is not present, and the inferred value "mb_type" specifies that the name of sub_mb_type[mbPartIdx] is the same as the name of mb_type for this case.

Table 7-18 – Sub-macroblock types in B macroblocks

sub_mb_type[mbPartIdx]	Name of sub_mb_type[mbPartIdx]	NumSubMbPart (sub_mb_type[mbPartIdx])	SubMbPredMode (sub_mb_type[mbPartIdx])	SubMbPartWidth (sub_mb_type[mbPartIdx])	SubMbPartHeight (sub_mb_type[mbPartIdx])
inferred	mb_type	4	Direct	4	4
0	B_Direct_8x8	4	Direct	4	4
1	B_L0_8x8	1	Pred_L0	8	8
2	B_L1_8x8	1	Pred_L1	8	8
3	B_Bi_8x8	1	BiPred	8	8
4	B_L0_8x4	2	Pred_L0	8	4
5	B_L0_4x8	2	Pred_L0	4	8
6	B_L1_8x4	2	Pred_L1	8	4
7	B_L1_4x8	2	Pred_L1	4	8
8	B_Bi_8x4	2	BiPred	8	4
9	B_Bi_4x8	2	BiPred	4	8
10	B_L0_4x4	4	Pred_L0	4	4
11	B_L1_4x4	4	Pred_L1	4	4
12	B_Bi_4x4	4	BiPred	4	4

The following semantics are assigned to the sub-macroblock types in Table 7-18:

– B_Skip and B_Direct_16x16: no motion vector differences or reference indices are present for the sub-macroblock in the bitstream. The functions SubMbPartWidth( ) and SubMbPartHeight( ) are used in the derivation process for motion vectors and reference frame indices in subclause 8.4.1 for direct mode prediction.

– B_Direct_8x8: no motion vector differences or reference indices are present for the sub-macroblock in the bitstream. The functions SubMbPartWidth(B_Direct_8x8) and SubMbPartHeight(B_Direct_8x8) are used in the derivation process for motion vectors and reference frame indices in subclause 8.4.1 for direct mode prediction.

– B_X_MxN, with X being replaced by L0, L1, or Bi, and MxN being replaced by 8x8, 8x4, 4x8 or 4x4: the samples of the sub-macroblock are predicted using one luma partition of size MxN equal to 8x8, or the samples of the sub-macroblock are predicted using two luma partitions of size MxN equal to 8x4, or the samples of the sub-macroblock are predicted using two luma partitions of size MxN equal to 4x8, or the samples of the sub-macroblock are predicted using four luma partitions of size MxN equal to 4x4, and associated chroma samples, respectively. All sub-macroblock partitions share the same reference index. For an MxN sub-macroblock partition in a sub-macroblock with sub_mb_type being B_X_MxN with X being replaced by either L0 or L1, one motion vector difference is present in the bitstream. For an MxN sub-macroblock partition in a sub-macroblock with sub_mb_type being B_Bi_MxN, two motion vector difference are present in the bitstream.

The following semantics are assigned to the sub-macroblock prediction modes (SubMbPredMode( )) in Table 7-18.

– Direct: see semantics for Table 7-14.

– Pred_L0: see semantics for Table 7-13.

– Pred_L1: see semantics for Table 7-14.

– BiPred: see semantics for Table 7-14.

ref_idx_l0[mbPartIdx **]** has the same semantics as ref_idx_l0 in subclause 7.4.5.1.

ref_idx_l1[mbPartIdx **]** has the same semantics as ref_idx_l1 in subclause 7.4.5.1.

mvd_l0[mbPartIdx **][** subMbPartIdx **][** compIdx **]** has the same semantics as mvd_l0 in subclause 7.4.5.1, except that it is applied to the sub-macroblock partition index with subMbPartIdx. The indices mbPartIdx and subMbPartIdx specify to which macroblock partition and sub-macroblock partition mvd_l0 is assigned.

mvd_l1[mbPartIdx **][** subMbPartIdx **][** compIdx **]** has the same semantics as mvd_l1 in subclause 7.4.5.1.

7.4.5.3 Residual data semantics

The syntax structure residual_block( ), which is used for parsing the transform coefficient levels, is assigned as follows.

– If entropy_coding_mode_flag is equal to 0, residual_block is set equal to residual_block_cavlc, which is used for parsing the syntax elements for transform coefficient levels.

– Otherwise (entropy_coding_mode_flag is equal to 1), residual_block is set equal to residual_block_cabac, which is used for parsing the syntax elements for transform coefficient levels.

Depending on mb_type, luma or chroma, and chroma format, the syntax structure residual_block(coeffLevel, maxNumCoeff) is used with the arguments coeffLevel, which is a list containing the maxNumCoeff transform coefficient levels that are parsed in residual_block( ) and maxNumCoeff as follows.

– Depending on MbPartPredMode(mb_type, 0), the following applies.
  – If MbPartPredMode(mb_type, 0) is equal to Intra_16x16, the transform coefficient levels are parsed into the list Intra16x16DCLevel and into the 16 lists Intra16x16ACLevel[i]. Intra16x16DCLevel contains the 16 transform coefficient levels of the DC transform coefficient levels for each 4x4 luma block. For each of the 16 4x4 luma blocks indexed by i = 0..15, the 15 AC transform coefficients levels of the i-th block are parsed into the i-th list Intra16x16ACLevel[i].
  – Otherwise (MbPartPredMode(mb_type, 0) is not equal to Intra_16x16), the following applies.
    – If transform_size_8x8_flag is equal to 0, for each of the 16 4x4 luma blocks indexed by i = 0..15, the 16 transform coefficient levels of the i-th block are parsed into the i-th list LumaLevel[i].
    – Otherwise (transform_size_8x8_flag is equal to 1), for each of the 4 8x8 luma blocks indexed by i8x8 = 0..3, the following applies.
      – If entropy_coding_mode_flag is equal to 0, first for each of the 4 4x4 luma blocks indexed by i4x4 = 0..3, the 16 transform coefficient levels of the i4x4-th block are parsed into the (i8x8 * 4 + i4x4)-th list LumaLevel[i8x8 * 4 + i4x4]. Then, the 64 transform coefficient levels of the i8x8-th 8x8 luma block which are indexed by 4 * i + i4x4, where i = 0..15 and i4x4 = 0..3, are derived as LumaLevel8x8[i8x8][4 * i + i4x4] = LumaLevel[i8x8 * 4 + i4x4][i].

        NOTE – The 4x4 luma blocks with luma4x4BlkIdx = i8x8 * 4 + i4x4 containing every fourth transform coefficient level of the corresponding i8x8-th 8x8 luma block with offset i4x4 are assumed to represent spatial locations given by the inverse 4x4 luma block scanning process in subclause 6.4.3.

      – Otherwise (entropy_coding_mode_flag is equal to 1), the 64 transform coefficient levels of the i8x8-th block are parsed into the i8x8-th list LumaLevel8x8[i8x8].

– For each chroma component, indexed by iCbCr = 0..1, the DC transform coefficient levels of the 4 * NumC8x8 4x4 chroma blocks are parsed into the iCbCr-th list ChromaDCLevel[iCbCr].

– For each of the 4x4 chroma blocks, indexed by i4x4 = 0..3 and i8x8 = 0..NumC8x8 – 1, of each chroma component, indexed by iCbCr = 0..1, the 15 AC transform coefficient levels are parsed into the (i8x8*4 + i4x4)-th list of the iCbCr-th chroma component ChromaACLevel[iCbCr][i8x8*4 + i4x4].

7.4.5.3.1 Residual block CAVLC semantics

The function TotalCoeff(coeff_token) that is used in subclause 7.3.5.3.1 returns the number of non-zero transform coefficient levels derived from coeff_token.

The function TrailingOnes(coeff_token) that is used in subclause 7.3.5.3.1 returns the trailing ones derived from coeff_token.

coeff_token specifies the total number of non-zero transform coefficient levels and the number of trailing one transform coefficient levels in a transform coefficient level scan. A trailing one transform coefficient level is one of up to three consecutive non-zero transform coefficient levels having an absolute value equal to 1 at the end of a scan of non-zero transform coefficient levels. The range of coeff_token is specified in subclause 9.2.1.

trailing_ones_sign_flag specifies the sign of a trailing one transform coefficient level as follows.

– If trailing_ones_sign_flag is equal to 0, the corresponding transform coefficient level is decoded as +1.

– Otherwise (trailing_ones_sign_flag equal to 1), the corresponding transform coefficient level is decoded as -1.

level_prefix and **level_suffix** specify the value of a non-zero transform coefficient level. The range of level_prefix and level_suffix is specified in subclause 9.2.2.

total_zeros specifies the total number of zero-valued transform coefficient levels that are located before the position of the last non-zero transform coefficient level in a scan of transform coefficient levels. The range of total_zeros is specified in subclause 9.2.3.

run_before specifies the number of consecutive transform coefficient levels in the scan with zero value before a non-zero valued transform coefficient level. The range of run_before is specified in subclause 9.2.3.

coeffLevel contains maxNumCoeff transform coefficient levels for the current list of transform coefficient levels.

7.4.5.3.2 Residual block CABAC semantics

coded_block_flag specifies whether the block contains non-zero transform coefficient levels as follows.

- If coded_block_flag is equal to 0, the block contains no non-zero transform coefficient levels.

- Otherwise (coded_block_flag is equal to 1), the block contains at least one non-zero transform coefficient level.

significant_coeff_flag[i **]** specifies whether the transform coefficient level at scanning position i is non-zero as follows.

– If significant_coeff_flag[i] is equal to 0, the transform coefficient level at scanning position i is set equal to 0;

– Otherwise (significant_coeff_flag[i] is equal to 1), the transform coefficient level at scanning position i has a non-zero value.

last_significant_coeff_flag[i **]** specifies for the scanning position i whether there are non-zero transform coefficient levels for subsequent scanning positions i + 1 to maxNumCoeff – 1 as follows.

– If last_significant_coeff_flag[i] is equal to 1, all following transform coefficient levels (in scanning order) of the block have value equal to 0.

– Otherwise (last_significant_coeff_flag[i] is equal to 0), there are further non-zero transform coefficient levels along the scanning path.

coeff_abs_level_minus1[i **]** is the absolute value of a transform coefficient level minus 1. The value of coeff_abs_level_minus1 is constrained by the limits in subclause 8.5.

coeff_sign_flag[i **]** specifies the sign of a transform coefficient level as follows.

– If coeff_sign_flag is equal to 0, the corresponding transform coefficient level has a positive value.

– Otherwise (coeff_sign_flag is equal to 1), the corresponding transform coefficient level has a negative value.

coeffLevel contains maxNumCoeff transform coefficient levels for the current list of transform coefficient levels.

8 Decoding process

Outputs of this process are decoded samples of the current picture (sometimes referred to by the variable CurrPic).

This clause describes the decoding process, given syntax elements and upper-case variables from clause 7.

The decoding process is specified such that all decoders shall produce numerically identical results. Any decoding process that produces identical results to the process described here conforms to the decoding process requirements of this Recommendation | International Standard.

Each picture referred to in this clause is a primary picture. Each slice referred to in this clause is a slice of a primary picture. Each slice data partition referred to in this clause is a slice data partition of a primary picture.

An overview of the decoding process is given as follows.

- The decoding of NAL units is specified in subclause 8.1.
- The processes in subclause 8.2 specify decoding processes using syntax elements in the slice layer and above.
  - Variables and functions relating to picture order count are derived in subclause 8.2.1. (only needed to be invoked for one slice of a picture)
  - Variables and functions relating to the macroblock to slice group map are derived in subclause 8.2.2. (only needed to be invoked for one slice of a picture)
  - The method of combining the various partitions when slice data partitioning is used is described in subclause 8.2.3.
  - When the frame_num of the current picture is not equal to PrevRefFrameNum and is not equal to (PrevRefFrameNum + 1) % MaxFrameNum, the decoding process for gaps in frame_num is performed according to subclause 8.2.5.2 prior to the decoding of any slices of the current picture.
  - At the beginning of the decoding process for each P, SP, or B slice, the decoding process for reference picture lists construction specified in 8.2.4 performed for derivation of reference picture list 0 (RefPicList0), and when decoding a B slice, reference picture list 1 (RefPicList1).
  - When the current picture is a reference picture and after all slices of the current picture have been decoded, the decoded reference picture marking process in subclause 8.2.5 specifies how the current picture is used in the decoding process of inter prediction in later decoded pictures.
- The processes in subclauses 8.3, 8.4, 8.5, 8.6, and 8.7 specify decoding processes using syntax elements in the macroblock layer and above.
  - The intra prediction process for I and SI macroblocks, except for I_PCM macroblocks as specified in subclause 8.3, has intra prediction samples as its output. For I_PCM macroblocks subclause 8.3 directly specifies a picture construction process. The output are the constructed samples prior to the deblocking filter process.
  - The inter prediction process for P and B macroblocks is specified in subclause 8.4 with inter prediction samples being the output.
  - The transform coefficient decoding process and picture construction process prior to deblocking filter process are specified in subclause 8.5. That process derives samples for I and B macroblocks and for P macroblocks in P slices. The output are constructed samples prior to the deblocking filter process.
  - The decoding process for P macroblocks in SP slices or SI macroblocks is specified in subclause 8.6. That process derives samples for P macroblocks in SP slices and for SI macroblocks. The output are constructed samples prior to the deblocking filter process.
  - The constructed samples prior to the deblocking filter process that are next to the edges of blocks and macroblocks are processed by a deblocking filter as specified in subclause 8.7 with the output being the decoded samples.

8.1 NAL unit decoding process

Inputs to this process are NAL units.

Outputs of this process are the RBSP syntax structures encapsulated within the NAL units.

The decoding process for each NAL unit extracts the RBSP syntax structure from the NAL unit and then operates the decoding processes specified for the RBSP syntax structure in the NAL unit as follows.

Subclause 8.2 describes the decoding process for NAL units with nal_unit_type equal to 1 through 5.

Subclauses 8.3 describes the decoding process for a macroblock or part of a macroblock coded in NAL units with nal_unit_type equal to 1, 2, and 5.

Subclause 8.4 describes the decoding process for a macroblock or part of a macroblock coded in NAL units with nal_unit_type equal to 1 and 2.

Subclause 8.5 describes the decoding process for a macroblock or part of a macroblock coded in NAL units with nal_unit_type equal to 1 and 3 to 5.

Subclause 8.6 describes the decoding process for a macroblock or part of a macroblock coded in NAL units with nal_unit_type equal to 1 and 3 to 5.

Subclause 8.7 describes the decoding process for a macroblock or part of a macroblock coded in NAL units with nal_unit_type equal to 1 to 5.

NAL units with nal_unit_type equal to 7 and 8 contain sequence parameter sets and picture parameter sets, respectively. Picture parameter sets are used in the decoding processes of other NAL units as determined by reference to a picture parameter set within the slice headers of each picture. Sequence parameter sets are used in the decoding processes of other NAL units as determined by reference to a sequence parameter set within the picture parameter sets of each sequence.

No normative decoding process is specified for NAL units with nal_unit_type equal to 6, 9, 10, 11, and 12.

8.2 Slice decoding process

8.2.1 Decoding process for picture order count

Outputs of this process are TopFieldOrderCnt (if applicable) and BottomFieldOrderCnt (if applicable).

Picture order counts are used to determine initial picture orderings for reference pictures in the decoding of B slices (see subclauses 8.2.4.2.3 and 8.2.4.2.4), to represent picture order differences between frames or fields for motion vector derivation in temporal direct mode (see subclause 8.4.1.2.3), for implicit mode weighted prediction in B slices (see subclause 8.4.2.3.2), and for decoder conformance checking (see subclause C.4).

Picture order count information is derived for every frame, field (whether decoded from a coded field or as a part of a decoded frame), or complementary field pair as follows:

– Each coded frame is associated with two picture order counts, called TopFieldOrderCnt and BottomFieldOrderCnt for its top field and bottom field, respectively.

– Each coded field is associated with a picture order count, called TopFieldOrderCnt for a coded top field and BottomFieldOrderCnt for a bottom field.

– Each complementary field pair is associated with two picture order counts, which are the TopFieldOrderCnt for its coded top field and the BottomFieldOrderCnt for its coded bottom field, respectively.

TopFieldOrderCnt and BottomFieldOrderCnt indicate the picture order of the corresponding top field or bottom field relative to the first output field of the previous IDR picture or the previous reference picture including a memory_management_control_operation equal to 5 in decoding order.

TopFieldOrderCnt and BottomFieldOrderCnt are derived by invoking one of the decoding processes for picture order count type 0, 1, and 2 in subclauses 8.2.1.1, 8.2.1.2, and 8.2.1.3, respectively. When the current picture includes a memory management control operation equal to 5, after the decoding of the current picture, tempPicOrderCnt is set equal to PicOrderCnt(CurrPic), TopFieldOrderCnt of the current picture (if any) is set equal to TopFieldOrderCnt - tempPicOrderCnt, and BottomFieldOrderCnt of the current picture (if any) is set equal to BottomFieldOrderCnt - tempPicOrderCnt.

The bitstream shall not contain data that results in Min(TopFieldOrderCnt, BottomFieldOrderCnt) not equal to 0 for a coded IDR frame, TopFieldOrderCnt not equal to 0 for a coded IDR top field, or BottomFieldOrderCnt not equal to 0 for a coded IDR bottom field. Thus, at least one of TopFieldOrderCnt and BottomFieldOrderCnt shall be equal to 0 for the fields of a coded IDR frame.

When the current picture is not an IDR picture, the following applies.

– Consider the list variable listD containing as elements the TopFieldOrderCnt and BottomFieldOrderCnt values associated with the list of pictures including all of the following
  – the first picture in the list is the previous picture of any of the following types
    – an IDR picture
    – a picture containing a memory_management_control_operation equal to 5
  – the following additional pictures.

– If pic_order_cnt_type is equal to 0, all other pictures that follow in decoding order after the first picture in the list and are not "non-existing" frames inferred by the decoding process for gaps in frame_num specified in subclause 8.2.5.2 and either precede the current picture in decoding order or are the current picture. When pic_order_cnt_type is equal to 0 and the current picture is not a "non-existing" frame inferred by the decoding process for gaps in frame_num specified in subclause 8.2.5.2, the current picture is included in listD prior to the invoking of the decoded reference picture marking process.

  Otherwise (pic_order_cnt_type is not equal to 0), all other pictures that follow in decoding order after the first picture in the list and either precede the current picture in decoding order or are the current picture. When pic_order_cnt_type is not equal to 0, the current picture is included in listD prior to the invoking of the decoded reference picture marking process.

– Consider the list variable listO which contains the elements of listD sorted in ascending order. listO shall not contain any of the following.
  – a pair of TopFieldOrderCnt and BottomFieldOrderCnt for a frame or complementary field pair that are not at consecutive positions in listO.
  – a TopFieldOrderCnt that has a value equal to another TopFieldOrderCnt.
  – a BottomFieldOrderCnt that has a value equal to another BottomFieldOrderCnt.
  – a BottomFieldOrderCnt that has a value equal to a TopFieldOrderCnt unless the BottomFieldOrderCnt and TopFieldOrderCnt belong to the same coded frame or complementary field pair.

The bitstream shall not contain data that results in values of TopFieldOrderCnt, BottomFieldOrderCnt, PicOrderCntMsb, or FrameNumOffset used in the decoding process as specified in subclauses 8.2.1.1 to 8.2.1.3 that exceed the range of values from -2^{31} to $2^{31}-1$, inclusive.

The function PicOrderCnt(picX) is specified as follows:

```
if( picX is a frame or a complementary field pair )
   PicOrderCnt( picX ) = Min( TopFieldOrderCnt, BottomFieldOrderCnt ) of the frame or complementary field
pair picX
else if( picX is a top field )
   PicOrderCnt( picX ) = TopFieldOrderCnt of field picX                                        (8-1)
else if( picX is a bottom field )
   PicOrderCnt( picX ) = BottomFieldOrderCnt of field picX
```

Then DiffPicOrderCnt(picA, picB) is specified as follows:

```
DiffPicOrderCnt( picA, picB ) = PicOrderCnt( picA ) - PicOrderCnt( picB )                      (8-2)
```

The bitstream shall not contain data that results in values of DiffPicOrderCnt(picA, picB) used in the decoding process that exceed the range of -2^{15} to $2^{15} - 1$, inclusive.

NOTE 1 – Let X be the current picture and Y and Z be two other pictures in the same sequence, Y and Z are considered to be in the same output order direction from X when both DiffPicOrderCnt(X, Y) and DiffPicOrderCnt(X, Z) are positive or both are negative.

NOTE 2 – Many applications assign PicOrderCnt(X) proportional to the sampling time of the picture X relative to the sampling time of an IDR picture.

When the current picture includes a memory_management_control_operation equal to 5, PicOrderCnt(CurrPic) shall be greater than PicOrderCnt(any other picture in listD).

8.2.1.1 Decoding process for picture order count type 0

This process is invoked when pic_order_cnt_type is equal to 0.

Input to this process is PicOrderCntMsb of the previous reference picture in decoding order as specified in this subclause.

Outputs of this process are either or both TopFieldOrderCnt or BottomFieldOrderCnt.

The variables prevPicOrderCntMsb and prevPicOrderCntLsb are derived as follows.

– If the current picture is an IDR picture, prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to 0.

– Otherwise (the current picture is not an IDR picture), the following applies.

– If the previous reference picture in decoding order included a memory_management_control_operation equal to 5, the following applies.
  – If the previous reference picture in decoding order is not a bottom field, prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to the value of TopFieldOrderCnt for the previous reference picture in decoding order.
  – Otherwise (the previous reference picture in decoding order is a bottom field), prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to 0.
– Otherwise (the previous reference picture in decoding order did not include a memory_management_control_operation equal to 5), prevPicOrderCntMsb is set equal to PicOrderCntMsb of the previous reference picture in decoding order and prevPicOrderCntLsb is set equal to the value of pic_order_cnt_lsb of the previous reference picture in decoding order.

PicOrderCntMsb of the current picture is derived as follows:

```
if( ( pic_order_cnt_lsb  <  prevPicOrderCntLsb )  &&
    ( ( prevPicOrderCntLsb – pic_order_cnt_lsb )  >=  ( MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb                          (8-3)
else if( ( pic_order_cnt_lsb  >  prevPicOrderCntLsb )  &&
        ( ( pic_order_cnt_lsb – prevPicOrderCntLsb )  >  ( MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb – MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb
```

When the current picture is not a bottom field, TopFieldOrderCnt is derived as follows:

```
if( !field_pic_flag  ||  !bottom_field_flag )
    TopFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb                            (8-4)
```

When the current picture is not a top field, BottomFieldOrderCnt is derived as follows:

```
if( !field_pic_flag )
    BottomFieldOrderCnt = TopFieldOrderCnt + delta_pic_order_cnt_bottom
else if( bottom_field_flag )                                                         (8-5)
    BottomFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb
```

8.2.1.2 Decoding process for picture order count type 1

This process is invoked when pic_order_cnt_type is equal to 1.

Input to this process is FrameNumOffset of the previous picture in decoding order as specified in this subclause.

Outputs of this process are either or both TopFieldOrderCnt or BottomFieldOrderCnt.

The values of TopFieldOrderCnt and BottomFieldOrderCnt are derived as specified in this subclause. Let prevFrameNum be equal to the frame_num of the previous picture in decoding order.

When the current picture is not an IDR picture, the variable prevFrameNumOffset is derived as follows.

– If the previous picture in decoding order included a memory_management_control_operation equal to 5, prevFrameNumOffset is set equal to 0.
– Otherwise (the previous picture in decoding order did not include a memory_management_control_operation equal to 5), prevFrameNumOffset is set equal to the value of FrameNumOffset of the previous picture in decoding order.

NOTE – When gaps_in_frame_num_value_allowed_flag is equal to 1, the previous picture in decoding order may be a "non-existing" frame inferred by the decoding process for gaps in frame_num specified in subclause 8.2.5.2.

The derivation proceeds in the following ordered steps.

1. The variable FrameNumOffset is derived as follows:

```
if( nal_unit_type  =  =  5 )
    FrameNumOffset = 0
else if( prevFrameNum > frame_num )                                                  (8-6)
    FrameNumOffset = prevFrameNumOffset + MaxFrameNum
```

```
else
    FrameNumOffset = prevFrameNumOffset
```

2. The variable absFrameNum is derived as follows:

```
if( num_ref_frames_in_pic_order_cnt_cycle  !=  0 )
    absFrameNum = FrameNumOffset + frame_num
else                                                                    (8-7)
    absFrameNum = 0
if( nal_ref_idc  = =  0  &&  absFrameNum  >  0 )
    absFrameNum = absFrameNum – 1
```

3. When absFrameNum > 0, picOrderCntCycleCnt and frameNumInPicOrderCntCycle are derived as follows:

```
if( absFrameNum  >  0 ) {
    picOrderCntCycleCnt = ( absFrameNum – 1 ) / num_ref_frames_in_pic_order_cnt_cycle
    frameNumInPicOrderCntCycle = ( absFrameNum – 1 ) % num_ref_frames_in_pic_order_cnt_cycle      (8-8)
}
```

4. The variable expectedDeltaPerPicOrderCntCycle is derived as follows:

```
expectedDeltaPerPicOrderCntCycle = 0
for( i = 0;  i < num_ref_frames_in_pic_order_cnt_cycle;  i++ )
    expectedDeltaPerPicOrderCntCycle += offset_for_ref_frame[ i ]       (8-9)
```

5. The variable expectedPicOrderCnt is derived as follows:

```
if( absFrameNum > 0 ){
    expectedPicOrderCnt = picOrderCntCycleCnt * expectedDeltaPerPicOrderCntCycle
    for( i = 0; i <= frameNumInPicOrderCntCycle; i++ )
        expectedPicOrderCnt = expectedPicOrderCnt + offset_for_ref_frame[ i ]
} else
    expectedPicOrderCnt = 0
if( nal_ref_idc  = =  0 )                                               (8-10)
    expectedPicOrderCnt = expectedPicOrderCnt + offset_for_non_ref_pic
```

6. The variables TopFieldOrderCnt or BottomFieldOrderCnt are derived as follows:

```
if( !field_pic_flag ) {
    TopFieldOrderCnt = expectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
    BottomFieldOrderCnt = TopFieldOrderCnt +
            offset_for_top_to_bottom_field + delta_pic_order_cnt[ 1 ]   (8-11)
} else if( !bottom_field_flag )
    TopFieldOrderCnt = expectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
else
    BottomFieldOrderCnt = expectedPicOrderCnt + offset_for_top_to_bottom_field + delta_pic_order_cnt[ 0 ]
```

8.2.1.3 Decoding process for picture order count type 2

This process is invoked when pic_order_cnt_type is equal to 2.

Outputs of this process are either or both TopFieldOrderCnt or BottomFieldOrderCnt.

Let prevFrameNum be equal to the frame_num of the previous picture in decoding order.

When the current picture is not an IDR picture, the variable prevFrameNumOffset is derived as follows.

– If the previous picture in decoding order included a memory_management_control_operation equal to 5, prevFrameNumOffset is set equal to 0.

– Otherwise (the previous picture in decoding order did not include a memory_management_control_operation equal to 5), prevFrameNumOffset is set equal to the value of FrameNumOffset of the previous picture in decoding order.

NOTE 1 – When gaps_in_frame_num_value_allowed_flag is equal to 1, the previous picture in decoding order may be a "non-existing" frame inferred by the decoding process for gaps in frame_num specified in subclause 8.2.5.2.

The variable FrameNumOffset is derived as follows.

```
if( nal_unit_type == 5 )
    FrameNumOffset = 0
else if( prevFrameNum > frame_num )                                    (8-12)
    FrameNumOffset = prevFrameNumOffset + MaxFrameNum
else
    FrameNumOffset = prevFrameNumOffset
```

The variable tempPicOrderCnt is derived as follows:

```
if( nal_unit_type == 5 )
    tempPicOrderCnt = 0
else if( nal_ref_idc == 0 )                                            (8-13)
    tempPicOrderCnt = 2 * ( FrameNumOffset + frame_num ) – 1
else
    tempPicOrderCnt = 2 * ( FrameNumOffset + frame_num )
```

The variables TopFieldOrderCnt or BottomFieldOrderCnt are derived as follows:

```
if( !field_pic_flag ) {
    TopFieldOrderCnt = tempPicOrderCnt
    BottomFieldOrderCnt = tempPicOrderCnt                              (8-14)
} else if( bottom_field_flag )
    BottomFieldOrderCnt = tempPicOrderCnt
else
    TopFieldOrderCnt = tempPicOrderCnt
```

NOTE 2 – Picture order count type 2 cannot be used in a coded video sequence that contains consecutive non-reference pictures that would result in more than one of these pictures having the same value of TopFieldOrderCnt or more than one of these pictures having the same value of BottomFieldOrderCnt.

NOTE 3 – Picture order count type 2 results in an output order that is the same as the decoding order.

8.2.2 Decoding process for macroblock to slice group map

Inputs to this process are the active picture parameter set and the slice header of the slice to be decoded.

Output of this process is a macroblock to slice group map MbToSliceGroupMap.

This process is invoked at the start of every slice.

NOTE – The output of this process is equal for all slices of a picture.

When num_slice_groups_minus1 is equal to 1 and slice_group_map_type is equal to 3, 4, or 5, slice groups 0 and 1 have a size and shape determined by slice_group_change_direction_flag as shown in Table 8-1 and specified in subclauses 8.2.2.4 to 8.2.2.6.

Table 8-1 – Refined slice group map type

slice_group_map_type	slice_group_change_direction_flag	refined slice group map type
3	0	Box-out clockwise
3	1	Box-out counter-clockwise
4	0	Raster scan
4	1	Reverse raster scan
5	0	Wipe right
5	1	Wipe left

In such a case, MapUnitsInSliceGroup0 slice group map units in the specified growth order are allocated for slice group 0 and the remaining PicSizeInMapUnits – MapUnitsInSliceGroup0 slice group map units of the picture are allocated for slice group 1.

When num_slice_groups_minus1 is equal to 1 and slice_group_map_type is equal to 4 or 5, the variable sizeOfUpperLeftGroup is defined as follows:

```
sizeOfUpperLeftGroup = ( slice_group_change_direction_flag ?
                         ( PicSizeInMapUnits – MapUnitsInSliceGroup0 ) : MapUnitsInSliceGroup0 )   (8-15)
```

The variable mapUnitToSliceGroupMap is derived as follows.

– If num_slice_groups_minus1 is equal to 0, the map unit to slice group map is generated for all i ranging from 0 to PicSizeInMapUnits – 1, inclusive, as specified by:

```
mapUnitToSliceGroupMap[ i ] = 0                                                                    (8-16)
```

– Otherwise (num_slice_groups_minus1 is not equal to 0), mapUnitToSliceGroupMap is derived as follows.
  – If slice_group_map_type is equal to 0, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.1 applies.
  – Otherwise, if slice_group_map_type is equal to 1, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.2 applies.
  – Otherwise, if slice_group_map_type is equal to 2, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.3 applies.
  – Otherwise, if slice_group_map_type is equal to 3, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.4 applies.
  – Otherwise, if slice_group_map_type is equal to 4, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.5 applies.
  – Otherwise, if slice_group_map_type is equal to 5, the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.6 applies.
  – Otherwise (slice_group_map_type is equal to 6), the derivation of mapUnitToSliceGroupMap as specified in subclause 8.2.2.7 applies.

After derivation of the mapUnitToSliceGroupMap, the process specified in subclause 8.2.2.8 is invoked to convert the map unit to slice group map mapUnitToSliceGroupMap to the macroblock to slice group map MbToSliceGroupMap. After derivation of the macroblock to slice group map as specified in subclause 8.2.2.8, the function NextMbAddress(n) is defined as the value of the variable nextMbAddress derived as specified by:

```
i = n + 1
while( i < PicSizeInMbs  &&  MbToSliceGroupMap[ i ]  !=  MbToSliceGroupMap[ n ] )
    i++;
nextMbAddress = i                                                                                  (8-17)
```

8.2.2.1 Specification for interleaved slice group map type

The specifications in this subclause apply when slice_group_map_type is equal to 0.

The map unit to slice group map is generated as specified by:

```
i = 0
do
    for( iGroup = 0; iGroup <= num_slice_groups_minus1 && i < PicSizeInMapUnits;
         i += run_length_minus1[ iGroup++ ] + 1 )
        for( j = 0; j <= run_length_minus1[ iGroup ] && i + j < PicSizeInMapUnits; j++ )
            mapUnitToSliceGroupMap[ i + j ] = iGroup                                               (8-18)
while( i < PicSizeInMapUnits )
```

8.2.2.2 Specification for dispersed slice group map type

The specifications in this subclause apply when slice_group_map_type is equal to 1.

The map unit to slice group map is generated as specified by:

```
for( i = 0; i < PicSizeInMapUnits; i++ )
    mapUnitToSliceGroupMap[ i ] = ( ( i % PicWidthInMbs ) +
                                    ( ( ( i / PicWidthInMbs ) * ( num_slice_groups_minus1 + 1 ) ) / 2 ) )
                                  % ( num_slice_groups_minus1 + 1 )                                    (8-19)
```

8.2.2.3 Specification for foreground with left-over slice group map type

The specifications in this subclause apply when slice_group_map_type is equal to 2.

The map unit to slice group map is generated as specified by:

```
for( i = 0; i < PicSizeInMapUnits; i++ )
    mapUnitToSliceGroupMap[ i ] = num_slice_groups_minus1
for( iGroup = num_slice_groups_minus1 – 1; iGroup >= 0; iGroup-- ) {
    yTopLeft = top_left[ iGroup ] / PicWidthInMbs
    xTopLeft = top_left[ iGroup ] % PicWidthInMbs
    yBottomRight = bottom_right[ iGroup ] / PicWidthInMbs
    xBottomRight = bottom_right[ iGroup ] % PicWidthInMbs
    for( y = yTopLeft; y <= yBottomRight; y++ )
        for( x = xTopLeft; x <= xBottomRight; x++ )
            mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] = iGroup                                   (8-20)
}
```

NOTE – The rectangles may overlap. Slice group 0 contains the macroblocks that are within the rectangle specified by top_left[0] and bottom_right[0]. A slice group having slice group ID greater than 0 and less than num_slice_groups_minus1 contains the macroblocks that are within the specified rectangle for that slice group that are not within the rectangle specified for any slice group having a smaller slice group ID. The slice group with slice group ID equal to num_slice_groups_minus1 contains the macroblocks that are not in the other slice groups.

8.2.2.4 Specification for box-out slice group map types

The specifications in this subclause apply when slice_group_map_type is equal to 3.

The map unit to slice group map is generated as specified by:

```
for( i = 0; i < PicSizeInMapUnits; i++ )
    mapUnitToSliceGroupMap[ i ] = 1
x = ( PicWidthInMbs – slice_group_change_direction_flag ) / 2
y = ( PicHeightInMapUnits – slice_group_change_direction_flag ) / 2
( leftBound, topBound ) = ( x, y )
( rightBound, bottomBound ) = ( x, y )
( xDir, yDir ) = ( slice_group_change_direction_flag – 1, slice_group_change_direction_flag )
for( k = 0; k < MapUnitsInSliceGroup0; k += mapUnitVacant ) {
    mapUnitVacant = ( mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] == 1 )
    if( mapUnitVacant )
        mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] = 0                                            (8-21)
    if( xDir == –1 && x == leftBound ) {
        leftBound = Max( leftBound – 1, 0 )
        x = leftBound
        ( xDir, yDir ) = ( 0, 2 * slice_group_change_direction_flag – 1 )
    } else if( xDir == 1 && x == rightBound ) {
        rightBound = Min( rightBound + 1, PicWidthInMbs – 1 )
        x = rightBound
        ( xDir, yDir ) = ( 0, 1 – 2 * slice_group_change_direction_flag )
    } else if( yDir == –1 && y == topBound ) {
        topBound = Max( topBound – 1, 0 )
        y = topBound
        ( xDir, yDir ) = ( 1 – 2 * slice_group_change_direction_flag, 0 )
    } else if( yDir == 1 && y == bottomBound ) {
        bottomBound = Min( bottomBound + 1, PicHeightInMapUnits – 1 )
        y = bottomBound
        ( xDir, yDir ) = ( 2 * slice_group_change_direction_flag – 1, 0 )
    } else
```

```
        ( x, y ) = ( x + xDir, y + yDir )
}
```

8.2.2.5 Specification for raster scan slice group map types

The specifications in this subclause apply when slice_group_map_type is equal to 4.

The map unit to slice group map is generated as specified by:

```
for( i = 0; i < PicSizeInMapUnits; i++ )
    if( i < sizeOfUpperLeftGroup )
        mapUnitToSliceGroupMap[ i ] = slice_group_change_direction_flag
    else                                                                    (8-22)
        mapUnitToSliceGroupMap[ i ] = 1 – slice_group_change_direction_flag
```

8.2.2.6 Specification for wipe slice group map types

The specifications in this subclause apply when slice_group_map_type is equal to 5.

The map unit to slice group map is generated as specified by:

```
k = 0;
for( j = 0; j < PicWidthInMbs; j++ )
    for( i = 0; i < PicHeightInMapUnits; i++ )
        if( k++ < sizeOfUpperLeftGroup )
            mapUnitToSliceGroupMap[ i * PicWidthInMbs + j ] = slice_group_change_direction_flag
        else                                                                (8-23)
            mapUnitToSliceGroupMap[ i * PicWidthInMbs + j ] = 1 – slice_group_change_direction_flag
```

8.2.2.7 Specification for explicit slice group map type

The specifications in this subclause apply when slice_group_map_type is equal to 6.

The map unit to slice group map is generated as specified by:

```
mapUnitToSliceGroupMap[ i ] = slice_group_id[ i ]                           (8-24)
```

for all i ranging from 0 to PicSizeInMapUnits – 1, inclusive.

8.2.2.8 Specification for conversion of map unit to slice group map to macroblock to slice group map

For each value of i ranging from 0 to PicSizeInMbs – 1, inclusive, the macroblock to slice group map is specified as follows.

— If frame_mbs_only_flag is equal to 1 or field_pic_flag is equal to 1, the macroblock to slice group map is specified by:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ i ]                        (8-25)
```

– Otherwise, if MbaffFrameFlag is equal to 1, the macroblock to slice group map is specified by:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ i / 2 ]                    (8-26)
```

– Otherwise (frame_mbs_only_flag is equal to 0 and mb_adaptive_frame_field_flag is equal to 0 and field_pic_flag is equal to 0), the macroblock to slice group map is specified by:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ ( i / ( 2 * PicWidthInMbs ) ) * PicWidthInMbs
                                                 + ( i % PicWidthInMbs ) ]  (8-27)
```

8.2.3 Decoding process for slice data partitioning

Inputs to this process are

– a slice data partition A layer RBSP,

– when syntax elements of category 3 are present in the slice data, a slice data partition B layer RBSP having the same slice_id as in the slice data partition A layer RBSP, and

– when syntax elements of category 4 are present in the slice data, a slice data partition C layer RBSP having the same slice_id as in the slice data partition A layer RBSP.

NOTE 1 – The slice data partition B layer RBSP and slice data partition C layer RBSP need not be present.

Output of this process is a coded slice.

When slice data partitioning is not used, coded slices are represented by a slice layer without partitioning RBSP that contains a slice header followed by a slice data syntax structure that contains all the syntax elements of categories 2, 3, and 4 (see category column in subclause 7.3) of the macroblock data for the macroblocks of the slice.

When slice data partitioning is used, the macroblock data of a slice is partitioned into one to three partitions contained in separate NAL units. Partition A contains a slice data partition A header, and all syntax elements of category 2. Partition B, when present, contains a slice data partition B header and all syntax elements of category 3. Partition C, when present, contains a slice data partition C header and all syntax elements of category 4.

When slice data partitioning is used, the syntax elements of each category are parsed from a separate NAL unit, which need not be present when no symbols of the respective category exist. The decoding process shall process the slice data partitions of a coded slice in a manner equivalent to processing a corresponding slice layer without partitioning RBSP by extracting each syntax element from the slice data partition in which the syntax element appears depending on the slice data partition assignment in the syntax tables in subclause 7.3.

NOTE 2 – Syntax elements of category 3 are relevant to the decoding of residual data of I and SI macroblock types. Syntax elements of category 4 are relevant to the decoding of residual data of P and B macroblock types. Category 2 encompasses all other syntax elements related to the decoding of macroblocks, and their information is often denoted as header information. The slice data partition A header contains all the syntax elements of the slice header, and additionally a slice_id that are used to associate the slice data partitions B and C with the slice data partition A. The slice data partition B and C headers contain the slice_id syntax element that establishes their association with the slice data partition A of the slice.

8.2.4 Decoding process for reference picture lists construction

This process is invoked at the beginning of decoding of each P, SP, or B slice.

Decoded reference pictures are marked as "used for short-term reference" or "used for long-term reference" as specified by the bitstream and specified in subclause 8.2.5. Short-term reference pictures are identified by the value of frame_num. Long-term reference pictures are assigned a long-term frame index as specified by the bitstream and specified in subclause 8.2.5.

Subclause 8.2.4.1 is invoked to specify

– the assignment of variables FrameNum, FrameNumWrap, and PicNum to each of the short-term reference pictures, and

– the assignment of variable LongTermPicNum to each of the long-term reference pictures.

Reference pictures are addressed through reference indices as specified in subclause 8.4.2.1. A reference index is an index into a reference picture list. When decoding a P or SP slice, there is a single reference picture list RefPicList0. When decoding a B slice, there is a second independent reference picture list RefPicList1 in addition to RefPicList0.

At the beginning of decoding of each slice, reference picture list RefPicList0, and for B slices RefPicList1, are derived as follows.

– An initial reference picture list RefPicList0 and for B slices RefPicList1 are derived as specified in subclause 8.2.4.2.

– The initial reference picture list RefPicList0 and for B slices RefPicList1 are modified as specified in subclause 8.2.4.3.

NOTE – The reordering process for reference picture lists specified in subclause 8.2.4.3 allows the contents of RefPicList0 and for B slices RefPicList1 to be modified in a flexible fashion. In particular, it is possible for a picture that is currently marked "used for reference" to be inserted into RefPicList0 and for B slices RefPicList1 even when the picture is not in the initial reference picture list derived as specified in subclause 8.2.4.2.

The number of entries in the modified reference picture list RefPicList0 is num_ref_idx_l0_active_minus1 + 1, and for B slices the number of entries in the modified reference picture list RefPicList1 is num_ref_idx_l1_active_minus1 + 1. A reference picture may appear at more than one index in the modified reference picture lists RefPicList0 or RefPicList1.

8.2.4.1 Decoding process for picture numbers

This process is invoked when the decoding process for reference picture lists construction specified in subclause 8.2.4 or the decoded reference picture marking process specified in subclause 8.2.5 is invoked.

The variables FrameNum, FrameNumWrap, PicNum, LongTermFrameIdx, and LongTermPicNum are used for the initialisation process for reference picture lists in subclause 8.2.4.2, the modification process for reference picture lists in subclause 8.2.4.3, and for the decoded reference picture marking process in subclause 8.2.5.

To each short-term reference picture the variables FrameNum and FrameNumWrap are assigned as follows. First, FrameNum is set equal to the syntax element frame_num that has been decoded in the slice header(s) of the corresponding short-term reference picture. Then the variable FrameNumWrap is derived as

```
if( FrameNum > frame_num )
    FrameNumWrap = FrameNum – MaxFrameNum                    (8-28)
else
    FrameNumWrap = FrameNum
```

where the value of frame_num used in Equation 8-28 is the frame_num in the slice header(s) for the current picture.

Each long-term reference picture has an associated value of LongTermFrameIdx (that was assigned to it as specified in subclause 8.2.5).

To each short-term reference picture a variable PicNum is assigned, and to each long-term reference picture a variable LongTermPicNum is assigned. The values of these variables depend on the value of field_pic_flag and bottom_field_flag for the current picture and they are set as follows.

– If field_pic_flag is equal to 0, the following applies.
  – For each short-term reference frame or complementary reference field pair:

    PicNum = FrameNumWrap (8-29)

  – For each long-term reference frame or long-term complementary reference field pair:

    LongTermPicNum = LongTermFrameIdx (8-30)

    NOTE – When decoding a frame the value of MbaffFrameFlag has no influence on the derivations in subclauses 8.2.4.2, 8.2.4.3, and 8.2.5.

– Otherwise (field_pic_flag is equal to 1), the following applies.
  – For each short-term reference field the following applies.
    – If the reference field has the same parity as the current field

    PicNum = 2 * FrameNumWrap + 1 (8-31)

    – Otherwise (the reference field has the opposite parity of the current field),

    PicNum = 2 * FrameNumWrap (8-32)

  – For each long-term reference field the following applies.
    – If the reference field has the same parity as the current field

    LongTermPicNum = 2 * LongTermFrameIdx + 1 (8-33)

    – Otherwise (the reference field has the opposite parity of the current field),

    LongTermPicNum = 2 * LongTermFrameIdx (8-34)

8.2.4.2 Initialisation process for reference picture lists

This initialisation process is invoked when decoding a P, SP, or B slice header.

RefPicList0 and RefPicList1 have initial entries as specified in subclauses 8.2.4.2.1 through 8.2.4.2.5.

When the number of entries in the initial RefPicList0 or RefPicList1 produced as specified in subclauses 8.2.4.2.1 through 8.2.4.2.5 is greater than num_ref_idx_l0_active_minus1 + 1 or num_ref_idx_l1_active_minus1 + 1, respectively, the extra entries past position num_ref_idx_l0_active_minus1 or num_ref_idx_l1_active_minus1 are discarded from the initial reference picture list.

When the number of entries in the initial RefPicList0 or RefPicList1 produced as specified in subclauses 8.2.4.2.1 through 8.2.4.2.5 is less than num_ref_idx_l0_active_minus1 + 1 or num_ref_idx_l1_active_minus1 + 1, respectively, the remaining entries in the initial reference picture list are set equal to "no reference picture".

8.2.4.2.1 Initialisation process for the reference picture list for P and SP slices in frames

This initialisation process is invoked when decoding a P or SP slice in a coded frame.

When this process is invoked, there shall be at least one reference frame or complementary reference field pair that is currently marked as "used for short-term reference" or "used for long-term reference".

The reference picture list RefPicList0 is ordered so that short-term reference frames and short-term complementary reference field pairs have lower indices than long-term reference frames and long-term complementary reference field pairs.

The short-term reference frames and complementary reference field pairs are ordered starting with the frame or complementary field pair with the highest PicNum value and proceeding through in descending order to the frame or complementary field pair with the lowest PicNum value.

The long-term reference frames and complementary reference field pairs are ordered starting with the frame or complementary field pair with the lowest LongTermPicNum value and proceeding through in ascending order to the frame or complementary field pair with the highest LongTermPicNum value.

NOTE – A non-paired reference field is not used for inter prediction for decoding a frame, regardless of the value of MbaffFrameFlag.

For example, when three reference frames are marked as "used for short-term reference" with PicNum equal to 300, 302, and 303 and two reference frames are marked as "used for long-term reference" with LongTermPicNum equal to 0 and 3, the initial index order is:

– RefPicList0[0] is set equal to the short-term reference picture with PicNum = 303,

– RefPicList0[1] is set equal to the short-term reference picture with PicNum = 302,

– RefPicList0[2] is set equal to the short-term reference picture with PicNum = 300,

– RefPicList0[3] is set equal to the long-term reference picture with LongTermPicNum = 0, and

– RefPicList0[4] is set equal to the long-term reference picture with LongTermPicNum = 3.

8.2.4.2.2 Initialisation process for the reference picture list for P and SP slices in fields

This initialisation process is invoked when decoding a P or SP slice in a coded field.

Each field included in the reference picture list RefPicList0 has a separate index in the reference picture list RefPicList0.

NOTE – When decoding a field, there are effectively at least twice as many pictures available for referencing as there would be when decoding a frame at the same position in decoding order.

Two ordered lists of reference frames, refFrameList0ShortTerm and refFrameList0LongTerm, are derived as follows. For purposes of the formation of this list of frames, decoded reference frames, complementary reference field pairs, non-paired reference fields and reference frames in which a single field is marked "used for short-term reference" or "used for long-term reference" are all considered reference frames.

– All frames having one or more fields marked "used for short-term reference" are included in the list of short-term reference frames refFrameList0ShortTerm. When the current field is the second field (in decoding order) of a complementary reference field pair and the first field is marked as "used for short-term reference", the first field is included in the list of short-term reference frames refFrameList0ShortTerm. refFrameList0ShortTerm is ordered starting with the reference frame with the highest FrameNumWrap value and proceeding through in descending order to the reference frame with the lowest FrameNumWrap value.

– All frames having one or more fields marked "used for long-term reference" are included in the list of long-term reference frames refFrameList0LongTerm. When the current field is the second field (in decoding order) of a complementary reference field pair and the first field is marked as "used for long-term reference, the first field is included in the list of long-term reference frames refFrameList0LongTerm. refFrameList0LongTerm is ordered

starting with the reference frame with the lowest LongTermFrameIdx value and proceeding through in ascending order to the reference frame with the highest LongTermFrameIdx value.

The process specified in subclause 8.2.4.2.5 is invoked with refFrameList0ShortTerm and refFrameList0LongTerm given as input and the output is assigned to RefPicList0.

8.2.4.2.3 Initialisation process for reference picture lists for B slices in frames

This initialisation process is invoked when decoding a B slice in a coded frame.

For purposes of the formation of the reference picture lists RefPicList0 and RefPicList1 the term reference entry refers in the following to decoded reference frames or complementary reference field pairs.

When this process is invoked, there shall be at least one reference entry that is currently marked as "used for short-term reference" or "used for long-term reference".

For B slices, the order of short-term reference entries in the reference picture lists RefPicList0 and RefPicList1 depends on output order, as given by PicOrderCnt( ). When pic_order_cnt_type is equal to 0, reference pictures that are marked as "non-existing" as specified in subclause 8.2.5.2 are not included in either RefPicList0 or RefPicList1.

NOTE 1 – When gaps_in_frame_num_value_allowed_flag is equal to 1, encoders should use reference picture list reordering to ensure proper operation of the decoding process (particularly when pic_order_cnt_type is equal to 0, in which case PicOrderCnt( ) is not inferred for "non-existing" frames).

The reference picture list RefPicList0 is ordered such that short-term reference entries have lower indices than long-term reference entries. It is ordered as follows.

– Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference". When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) less than PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refPicList0 in descending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refPicList0 in ascending order of PicOrderCnt(entryShortTerm).

– The long-term reference entries are ordered starting with the long-term reference entry that has the lowest LongTermPicNum value and proceeding through in ascending order to the long-term reference entry that has the highest LongTermPicNum value.

The reference picture list RefPicList1 is ordered so that short-term reference entries have lower indices than long-term reference entries. It is ordered as follows.

– Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference". When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) greater than PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refPicList1 in ascending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refPicList1 in descending order of PicOrderCnt(entryShortTerm).

– Long-term reference entries are ordered starting with the long-term reference frame or complementary reference field pair that has the lowest LongTermPicNum value and proceeding through in ascending order to the long-term reference entry that has the highest LongTermPicNum value.

– When the reference picture list RefPicList1 has more than one entry and RefPicList1 is identical to the reference picture list RefPicList0, the first two entries RefPicList1[0] and RefPicList1[1] are switched.

NOTE 2 – A non-paired reference field is not used for inter prediction of frames (independent of the value of MbaffFrameFlag).

8.2.4.2.4 Initialisation process for reference picture lists for B slices in fields

This initialisation process is invoked when decoding a B slice in a coded field.

When decoding a field, each field of a stored reference frame is identified as a separate reference picture with a unique index. The order of short-term reference pictures in the reference picture lists RefPicList0 and RefPicList1 depend on output order, as given by PicOrderCnt( ). When pic_order_cnt_type is equal to 0, reference pictures that are marked as "non-existing" as specified in subclause 8.2.5.2 are not included in either RefPicList0 or RefPicList1.

NOTE 1 – When gaps_in_frame_num_value_allowed_flag is equal to 1, encoders should use reference picture list reordering to ensure proper operation of the decoding process (particularly when pic_order_cnt_type is equal to 0, in which case PicOrderCnt( ) is not inferred for "non-existing" frames).

NOTE 2 – When decoding a field, there are effectively at least twice as many pictures available for referencing as there would be when decoding a frame at the same position in decoding order.

Three ordered lists of reference frames, refFrameList0ShortTerm, refFrameList1ShortTerm and refFrameListLongTerm, are derived as follows. For purposes of the formation of these lists of frames the term reference entry refers in the following to decoded reference frames, complementary reference field pairs, or non-paired reference

fields. When pic_order_cnt_type is equal to 0, the term reference entry does not refer to frames that are marked as "non-existing" as specified in subclause 8.2.5.2.

– Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference". When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) less than or equal to PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refFrameList0ShortTerm in descending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refFrameList0ShortTerm in ascending order of PicOrderCnt(entryShortTerm).

  NOTE 3 – When the current field follows in decoding order a coded field fldPrev with which together it forms a complementary reference field pair, fldPrev is included into the list refFrameList0ShortTerm using PicOrderCnt(fldPrev) and the ordering method described in the previous sentence is applied.

– Let entryShortTerm be a variable ranging over all reference entries that are currently marked as "used for short-term reference". When some values of entryShortTerm are present having PicOrderCnt(entryShortTerm) greater than PicOrderCnt(CurrPic), these values of entryShortTerm are placed at the beginning of refFrameList1ShortTerm in ascending order of PicOrderCnt(entryShortTerm). All of the remaining values of entryShortTerm (when present) are then appended to refFrameList1ShortTerm in descending order of PicOrderCnt(entryShortTerm).

  NOTE 4 – When the current field follows in decoding order a coded field fldPrev with which together it forms a complementary reference field pair, fldPrev is included into the list refFrameList1ShortTerm using PicOrderCnt(fldPrev) and the ordering method described in the previous sentence is applied.

– refFrameListLongTerm is ordered starting with the reference entry having the lowest LongTermFrameIdx value and proceeding through in ascending order to the reference entry having highest LongTermFrameIdx value.

  NOTE 5 – When the complementary field of the current picture is marked "used for long-term reference" it is included into the list refFrameListLongTerm. A reference entry in which only one field is marked as “used for long-term reference” is included into the list refFrameListLongTerm.

The process specified in subclause 8.2.4.2.5 is invoked with refFrameList0ShortTerm and refFrameListLongTerm given as input and the output is assigned to RefPicList0.

The process specified in subclause 8.2.4.2.5 is invoked with refFrameList1ShortTerm and refFrameListLongTerm given as input and the output is assigned to RefPicList1.

When the reference picture list RefPicList1 has more than one entry and RefPicList1 is identical to the reference picture list RefPicList0, the first two entries RefPicList1[0] and RefPicList1[1] are switched.

8.2.4.2.5 Initialisation process for reference picture lists in fields

Inputs of this process are the reference frame lists refFrameListXShortTerm (with X may be 0 or 1) and refFrameListLongTerm.

The reference picture list RefPicListX is a list ordered such that short-term reference fields have lower indices than long-term reference fields. Given the reference frame lists refFrameListXShortTerm and refFrameListLongTerm, it is derived as follows.

– Short-term reference fields are ordered by selecting reference fields from the ordered list of frames refFrameListXShortTerm by alternating between fields of differing parity, starting with a field that has the same parity as the current field (when present). When one field of a reference frame was not decoded or is not marked as “used for short-term reference”, the missing field is ignored and instead the next available stored reference field of the chosen parity from the ordered list of frames refFrameListXShortTerm is inserted into RefPicListX. When there are no more short-term reference fields of the alternate parity in the ordered list of frames refFrameListXShortTerm, the next not yet indexed fields of the available parity are inserted into RefPicListX in the order in which they occur in the ordered list of frames refFrameListXShortTerm.

– Long-term reference fields are ordered by selecting reference fields from the ordered list of frames refFrameListLongTerm by alternating between fields of differing parity, starting with a field that has the same parity as the current field (when present). When one field of a reference frame was not decoded or is not marked as “used for long-term reference”, the missing field is ignored and instead the next available stored reference field of the chosen parity from the ordered list of frames refFrameListLongTerm is inserted into RefPicListX. When there are no more long-term reference fields of the alternate parity in the ordered list of frames refFrameListLongTerm, the next not yet indexed fields of the available parity are inserted into RefPicListX in the order in which they occur in the ordered list of frames refFrameListLongTerm.

8.2.4.3 Reordering process for reference picture lists

When ref_pic_list_reordering_flag_l0 is equal to 1, the following applies.

– Let refIdxL0 be an index into the reference picture list RefPicList0. It is initially set equal to 0.

– The corresponding syntax elements reordering_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.
  – If reordering_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause 8.2.4.3.1 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.
  – Otherwise, if reordering_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.4.3.2 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.
  – Otherwise (reordering_of_pic_nums_idc is equal to 3), the reordering process for reference picture list RefPicList0 is finished.

When ref_pic_list_reordering_flag_l1 is equal to 1, the following applies.

– Let refIdxL1 be an index into the reference picture list RefPicList1. It is initially set equal to 0.
– The corresponding syntax elements reordering_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.
  – If reordering_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause 8.2.4.3.1 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.
  – Otherwise, if reordering_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.4.3.2 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.
  – Otherwise (reordering_of_pic_nums_idc is equal to 3), the reordering process for reference picture list RefPicList1 is finished.

8.2.4.3.1 Reordering process of reference picture lists for short-term reference pictures

Input to this process is an index refIdxLX (with X being 0 or 1).

Output of this process is an incremented index refIdxLX.

The variable picNumLXNoWrap is derived as follows.

– If reordering_of_pic_nums_idc is equal to 0

```
if( picNumLXPred – ( abs_diff_pic_num_minus1 + 1 ) < 0 )
    picNumLXNoWrap = picNumLXPred – ( abs_diff_pic_num_minus1 + 1 ) + MaxPicNum      (8-35)
else
    picNumLXNoWrap = picNumLXPred – ( abs_diff_pic_num_minus1 + 1 )
```

– Otherwise (reordering_of_pic_nums_idc is equal to 1),

```
if( picNumLXPred + ( abs_diff_pic_num_minus1 + 1 ) >= MaxPicNum )
    picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 ) – MaxPicNum      (8-36)
else
    picNumLXNoWrap = picNumLXPred + ( abs_diff_pic_num_minus1 + 1 )
```

picNumLXPred is the prediction value for the variable picNumLXNoWrap. When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of reordering_of_pic_nums_idc equal to 0 or 1 in the ref_pic_list_reordering( ) syntax), picNumL0Pred and picNumL1Pred are initially set equal to CurrPicNum. After each assignment of picNumLXNoWrap, the value of picNumLXNoWrap is assigned to picNumLXPred.

The variable picNumLX is derived as follows

```
if( picNumLXNoWrap > CurrPicNum )
    picNumLX = picNumLXNoWrap – MaxPicNum                                          (8-37)
else
    picNumLX = picNumLXNoWrap
```

picNumLX shall be equal to the PicNum of a reference picture that is marked as “used for short-term reference” and shall not be equal to the PicNum of a short-term reference picture that is marked as "non-existing".

The following procedure is conducted to place the picture with short-term picture number picNumLX into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx – 1]
RefPicListX[ refIdxLX++ ] = short-term reference picture with PicNum equal to picNumLX
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )                    (8-38)
    if( PicNumF( RefPicListX[ cIdx ] ) != picNumLX )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

where the function PicNumF(RefPicListX[cIdx]) is derived as follows:

– If the picture RefPicListX[cIdx] is marked as "used for short-term reference", PicNumF(RefPicListX[cIdx]) is the PicNum of the picture RefPicListX[cIdx].

– Otherwise (the picture RefPicListX[cIdx] is not marked as "used for short-term reference"), PicNumF(RefPicListX[cIdx]) is equal to MaxPicNum.

NOTE 1 – A value of MaxPicNum can never be equal to picNumLX.

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

8.2.4.3.2 Reordering process of reference picture lists for long-term reference pictures

Input to this process is an index refIdxLX (with X being 0 or 1).

Output of this process is an incremented index refIdxLX.

The following procedure is conducted to place the picture with long-term picture number long_term_pic_num into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.

```
for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx – 1]
RefPicListX[ refIdxLX++ ] = long-term reference picture with LongTermPicNum equal to long_term_pic_num
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )                    (8-39)
    if( LongTermPicNumF( RefPicListX[ cIdx ] ) != long_term_pic_num )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]
```

where the function LongTermPicNumF(RefPicListX[cIdx]) is derived as follows:

– If the picture RefPicListX[cIdx] is marked as "used for long-term reference", LongTermPicNumF(RefPicListX[cIdx]) is the LongTermPicNum of the picture RefPicListX[cIdx].

– Otherwise (the picture RefPicListX[cIdx] is not marked as "used for long-term reference"), LongTermPicNumF(RefPicListX[cIdx]) is equal to 2 * (MaxLongTermFrameIdx + 1).

NOTE 1 – A value of 2 * (MaxLongTermFrameIdx + 1) can never be equal to long_term_pic_num.

NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.

8.2.5 Decoded reference picture marking process

This process is invoked for decoded pictures when nal_ref_idc is not equal to 0.

NOTE – The decoding process for gaps in frame_num that is specified in subclause 8.2.5.2 may also be invoked when nal_ref_idc is equal to 0, as specified in clause 8.

A decoded picture with nal_ref_idc not equal to 0, referred to as a reference picture, is marked as “used for short-term reference” or "used for long-term reference". For a decoded reference frame, both of its fields are marked the same as the frame. For a complementary reference field pair, the pair is marked the same as both of its fields. A picture that is marked as "used for short-term reference" is identified by its FrameNum and, when it is a field, by its parity. A picture that is marked as "used for long-term reference" is identified by its LongTermFrameIdx and, when it is a field, by its parity.

Frames or complementary field pairs marked as “used for short-term reference” or as "used for long-term reference" can be used as a reference for inter prediction when decoding a frame until the frame, the complementary field pair, or one of its constituent fields is marked as “unused for reference”. A field marked as “used for short-term reference” or as

"used for long-term reference" can be used as a reference for inter prediction when decoding a field until marked as "unused for reference".

A picture can be marked as "unused for reference" by the sliding window reference picture marking process, a first-in, first-out mechanism specified in subclause 8.2.5.3 or by the adaptive memory control reference picture marking process, a customised adaptive marking operation specified in subclause 8.2.5.4.

A short-term reference picture is identified for use in the decoding process by its variables FrameNum and FrameNumWrap and its picture number PicNum, and a long-term reference picture is identified for use in the decoding process by its long-term picture number LongTermPicNum. When the current picture is not an IDR picture, subclause 8.2.4.1 is invoked to specify the assignment of the variables FrameNum, FrameNumWrap, PicNum and LongTermPicNum.

8.2.5.1 Sequence of operations for decoded reference picture marking process

Decoded reference picture marking proceeds in the following ordered steps.

1. All slices of the current picture are decoded.
2. Depending on whether the current picture is an IDR picture, the following applies.
   - If the current picture is an IDR picture, the following applies.
     - All reference pictures are marked as "unused for reference"
     - Depending on long_term_reference_flag, the following applies.
       - If long_term_reference_flag is equal to 0, the IDR picture is marked as "used for short-term reference" and MaxLongTermFrameIdx is set equal to "no long-term frame indices".
       - Otherwise (long_term_reference_flag is equal to 1), the IDR picture is marked as "used for long-term reference", the LongTermFrameIdx for the IDR picture is set equal to 0, and MaxLongTermFrameIdx is set equal to 0.
   - Otherwise (the current picture is not an IDR picture), the following applies.
     - If adaptive_ref_pic_marking_mode_flag is equal to 0, the process specified in subclause 8.2.5.3 is invoked.
     - Otherwise (adaptive_ref_pic_marking_mode_flag is equal to 1), the process specified in subclause 8.2.5.4 is invoked.
3. When the current picture is not an IDR picture and it was not marked as "used for long-term reference" by memory_management_control_operation equal to 6, it is marked as "used for short-term reference".

After marking the current decoded reference picture, the total number of frames with at least one field marked as "used for reference", plus the number of complementary field pairs with at least one field marked as "used for reference", plus the number of non-paired fields marked as "used for reference" shall not be greater than Max(num_ref_frames, 1).

8.2.5.2 Decoding process for gaps in frame_num

This process is invoked when frame_num is not equal to PrevRefFrameNum and is not equal to (PrevRefFrameNum + 1) % MaxFrameNum.

NOTE 1 – Although this process is specified as a subclause within subclause 8.2.5 (which defines a process that is invoked only when nal_ref_idc is not equal to 0), this process may also be invoked when nal_ref_idc is equal to 0 (as specified in clause 8). The reasons for the location of this subclause within the structure of this Recommendation | International Standard are historical.

NOTE 2 – This process can only be invoked for a conforming bitstream when gaps_in_frame_num_value_allowed_flag is equal to 1. When gaps_in_frame_num_value_allowed_flag is equal to 0 and frame_num is not equal to PrevRefFrameNum and is not equal to (PrevRefFrameNum + 1) % MaxFrameNum, the decoding process should infer an unintentional loss of pictures.

When this process is invoked, a set of values of frame_num pertaining to "non-existing" pictures is derived as all values taken on by UnusedShortTermFrameNum in Equation 7-21 except the value of frame_num for the current picture.

The decoding process generates and marks a frame for each of the values of frame_num pertaining to "non-existing" pictures, in the order in which the values of UnusedShortTermFrameNum are generated by Equation 7-21, using the "sliding window" picture marking process as specified in subclause 8.2.5.3. The generated frames are also marked as "non-existing" and "used for short-term reference". The sample values of the generated frames may be set to any value. The bitstream shall not contain data that results in a reference to these generated frames which are marked as "non-existing" in the inter prediction process, a reference to these frames in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or a reference to these frames in the assignment process of a LongTermFrameIdx to a short-term reference picture (subclause 8.2.5.4.3).

When pic_order_cnt_type is not equal to 0, TopFieldOrderCnt and BottomFieldOrderCnt are derived for each of the "non-existing" frames by invoking the decoding process for picture order count in subclause 8.2.1. When invoking the process in subclause 8.2.1 for a particular "non-existing" frame, the current picture is considered to be a picture considered having frame_num inferred to be equal to UnusedShortTermFrameNum, nal_ref_idc inferred to be not equal to 0, nal_unit_type inferred to be not equal to 5, field_pic_flag inferred to be equal to 0, adaptive_ref_pic_marking_mode_flag inferred to be equal to 0, delta_pic_order_cnt[0] (if needed) inferred to be equal to 0, and delta_pic_order_cnt[1] (if needed) inferred to be equal to 0.

NOTE 3 – The decoding process should infer an unintentional picture loss when any of these values of frame_num pertaining to "non-existing" pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or is referred to in the assignment process of a LongTermFrameIdx to a short-term reference picture (subclause 8.2.5.4.3). The decoding process should not infer an unintentional picture loss when a memory management control operation not equal to 3 is applied to a frame marked as "non-existing".

8.2.5.3 Sliding window decoded reference picture marking process

This process is invoked when adaptive_ref_pic_marking_mode_flag is equal to 0.

Depending on the properties of the current picture as specified below, the following applies.

– If the current picture is a coded field that is the second field in decoding order of a complementary reference field pair, and the first field has been marked as "used for short-term reference", the current picture is also marked as "used for short-term reference".

– Otherwise, the following applies.

  – Let numShortTerm be the total number of reference frames, complementary reference field pairs and non-paired reference fields for which at least one field is marked as "used for short-term reference". Let numLongTerm be the total number of reference frames, complementary reference field pairs and non-paired reference fields for which at least one field is marked as "used for long-term reference".

  – When numShortTerm + numLongTerm is equal to Max(num_ref_frames, 1), the condition that numShortTerm is greater than 0 shall be fulfilled, and the short-term reference frame, complementary reference field pair or non-paired reference field that has the smallest value of FrameNumWrap is marked as "unused for reference". When it is a frame or a complementary field pair, both of its fields are also marked as "unused for reference".

8.2.5.4 Adaptive memory control decoded reference picture marking process

This process is invoked when adaptive_ref_pic_marking_mode_flag is equal to 1.

The memory_management_control_operation commands with values of 1 to 6 are processed in the order they occur in the bitstream after the current picture has been decoded. For each of these memory_management_control_operation commands, one of the processes specified in subclauses 8.2.5.4.1 to 8.2.5.4.5 is invoked depending on the value of memory_management_control_operation. The memory_management_control_operation command with value of 0 specifies the end of memory_management_control_operation commands.

Memory management control operations are applied to pictures as follows.

– If field_pic_flag is equal to 0, memory_management_control_operation commands are applied to the frames or complementary reference field pairs specified.

– Otherwise (field_pic_flag is equal to 1), memory_management_control_operation commands are applied to the individual reference fields specified.

8.2.5.4.1 Marking process of a short-term reference picture as "unused for reference"

This process is invoked when memory_management_control_operation is equal to 1.

Let picNumX be specified by

$$picNumX = CurrPicNum - (difference_of_pic_nums_minus1 + 1). \quad (8\text{-}40)$$

Depending on field_pic_flag the value of picNumX is used to mark a short-term reference picture as "unused for reference" as follows.

– If field_pic_flag is equal to 0, the short-term reference frame or short-term complementary reference field pair specified by picNumX and both of its fields are marked as "unused for reference".

– Otherwise (field_pic_flag is equal to 1), the short-term reference field specified by picNumX is marked as "unused for reference". When that reference field is part of a reference frame or a complementary reference field pair, the

frame or complementary field pair is also marked as "unused for reference", but the marking of the other field is not changed.

8.2.5.4.2 Marking process of a long-term reference picture as "unused for reference"

This process is invoked when memory_management_control_operation is equal to 2.

Depending on field_pic_flag the value of LongTermPicNum is used to mark a long-term reference picture as "unused for reference" as follows.

– If field_pic_flag is equal to 0, the long-term reference frame or long-term complementary reference field pair having LongTermPicNum equal to long_term_pic_num and both of its fields are marked as "unused for reference".

– Otherwise (field_pic_flag is equal to 1), the long-term reference field specified by LongTermPicNum equal to long_term_pic_num is marked as "unused for reference". When that reference field is part of a reference frame or a complementary reference field pair, the frame or complementary field pair is also marked as "unused for reference", but the marking of the other field is not changed.

8.2.5.4.3 Assignment process of a LongTermFrameIdx to a short-term reference picture

This process is invoked when memory_management_control_operation is equal to 3.

Given the syntax element difference_of_pic_nums_minus1, the variable picNumX is obtained as specified in subclause 8.2.5.4.1. picNumX shall refer to a frame or complementary reference field pair or non-paired reference field marked as "used for short-term reference" and not marked as "non-existing".

When LongTermFrameIdx equal to long_term_frame_idx is already assigned to a long-term reference frame or a long-term complementary reference field pair, that frame or complementary field pair and both of its fields are marked as "unused for reference". When LongTermFrameIdx is already assigned to a non-paired reference field, and the field is not the complementary field of the picture specified by picNumX, that field is marked as "unused for reference".

Depending on field_pic_flag the value of LongTermFrameIdx is used to mark a picture from "used for short-term reference" to "used for long-term reference" as follows.

– If field_pic_flag is equal to 0, the marking of the short-term reference frame or short-term complementary reference field pair specified by picNumX and both of its fields are changed from "used for short-term reference" to "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

– Otherwise (field_pic_flag is equal to 1), the marking of the short-term reference field specified by picNumX is changed from "used for short-term reference" to "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx. When the field is part of a reference frame or a complementary reference field pair, and the other field of the same reference frame or complementary reference field pair is also marked as "used for long-term reference", the reference frame or complementary reference field pair is also marked as "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

8.2.5.4.4 Decoding process for MaxLongTermFrameIdx

This process is invoked when memory_management_control_operation is equal to 4.

All pictures for which LongTermFrameIdx is greater than max_long_term_frame_idx_plus1 – 1 and that are marked as "used for long-term reference" are marked as "unused for reference".

The variable MaxLongTermFrameIdx is derived as follows.

– If max_long_term_frame_idx_plus1 is equal to 0, MaxLongTermFrameIdx is set equal to "no long-term frame indices".

– Otherwise (max_long_term_frame_idx_plus1 is greater than 0), MaxLongTermFrameIdx is set equal to max_long_term_frame_idx_plus1 – 1.

NOTE – The memory_management_control_operation command equal to 4 can be used to mark long-term reference pictures as "unused for reference". The frequency of transmitting max_long_term_frame_idx_plus1 is not specified by this Recommendation | International Standard. However, the encoder should send a memory_management_control_operation command equal to 4 upon receiving an error message, such as an intra refresh request message.

8.2.5.4.4.1 Marking process of all reference pictures as "unused for reference" and setting MaxLongTermFrameIdx to "no long-term frame indices"

This process is invoked when memory_management_control_operation is equal to 5.

All reference pictures are marked as “unused for reference” and the variable MaxLongTermFrameIdx is set equal to “no long-term frame indices”.

8.2.5.4.5 Process for assigning a long-term frame index to the current picture

This process is invoked when memory_management_control_operation is equal to 6.

When a variable LongTermFrameIdx equal to long_term_frame_idx is already assigned to a long-term reference frame or a long-term complementary reference field pair, that frame or complementary field pair and both of its fields are marked as "unused for reference". When LongTermFrameIdx is already assigned to a non-paired reference field, and the field is not the complementary field of the current picture, that field is marked as “unused for reference”.

The current picture is marked as "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

When field_pic_flag is equal to 0, both its fields are also marked as "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

When field_pic_flag is equal to 1 and the current picture is the second field (in decoding order) of a complementary reference field pair, and the first field of the complementary reference field pair is also currently marked as "used for long-term reference), the complementary reference field pair is also marked as "used for long-term reference" and assigned LongTermFrameIdx equal to long_term_frame_idx.

After marking the current decoded reference picture, the total number of frames with at least one field marked as “used for reference”, plus the number of complementary field pairs with at least one field marked as “used for reference”, plus the number of non-paired fields marked as “used for reference” shall not be greater than Max(num_ref_frames, 1).

NOTE – Under some circumstances, the above statement may impose a constraint on the order in which a memory_management_control_operation syntax element equal to 6 can appear in the decoded reference picture marking syntax relative to a memory_management_control_operation syntax element equal to 1, 2, or 4.

8.3 Intra prediction process

This process is invoked for I and SI macroblock types.

Inputs to this process are constructed samples prior to the deblocking filter process and, for Intra_NxN prediction modes (where NxN is equal to 4x4 or 8x8), the values of IntraNxNPredMode from neighbouring macroblocks.

Outputs of this process are specified as follows.

- If the macroblock prediction mode is Intra_4x4 or Intra_8x8, the outputs are constructed luma samples prior to the deblocking filter process and (when chroma_format_idc is not equal to 0) chroma prediction samples of the macroblock $pred_C$, where C is equal to Cb and Cr.
- Otherwise, if mb_type is not equal to I_PCM, the outputs are luma prediction samples of the macroblock $pred_L$ and (when chroma_format_idc is not equal to 0) chroma prediction samples of the macroblock $pred_C$, where C is equal to Cb and Cr.
- Otherwise (mb_type is equal to I_PCM), the outputs are constructed luma and (when chroma_format_idc is not equal to 0) chroma samples prior to the deblocking filter process.

The variable MvCnt is set equal to 0.

Depending on the value of mb_type the following applies.

- If mb_type is equal to I_PCM, the process specified in subclause 8.3.5 is invoked.
- Otherwise (mb_type is not equal to I_PCM), the following applies.
  - The decoding processes for Intra prediction modes are described for the luma component as follows.
    - If the macroblock prediction mode is equal to Intra_4x4, the specification in subclause 8.3.1 applies.
    - Otherwise, if the macroblock prediction mode is equal to Intra_8x8, the specification in subclause 8.3.2 applies.
    - Otherwise (the macroblock prediction mode is equal to Intra_16x16), the specification in subclause 8.3.3 applies.
  - The decoding processes for Intra prediction modes for the chroma components are described in subclause 8.3.4. This process is only invoked when chroma_format_idc is not equal to 0 (monochrome).

Samples used in the Intra prediction process are the sample values prior to alteration by any deblocking filter operation.

8.3.1 Intra_4x4 prediction process for luma samples

This process is invoked when the macroblock prediction mode is equal to Intra_4x4.

Inputs to this process are the values of Intra4x4PredMode (if available) or Intra8x8PredMode (if available) from neighbouring macroblocks or macroblock pairs.

The luma component of a macroblock consists of 16 blocks of 4x4 luma samples. These blocks are inverse scanned using the 4x4 luma block inverse scanning process as specified in subclause 6.4.3.

For all 4x4 luma blocks of the luma component of a macroblock with luma4x4BlkIdx = 0..15, the derivation process for the Intra4x4PredMode as specified in subclause 8.3.1.1 is invoked with luma4x4BlkIdx as well as Intra4x4PredMode and Intra8x8PredMode that are previously (in decoding order) derived for adjacent macroblocks as the input and the variable Intra4x4PredMode[luma4x4BlkIdx] as the output.

For each luma block of 4x4 samples indexed using luma4x4BlkIdx = 0..15,

1. The Intra_4x4 sample prediction process in subclause 8.3.1.2 is invoked with luma4x4BlkIdx and constructed samples prior (in decoding order) to the deblocking filter process from adjacent luma blocks as the input and the output are the Intra_4x4 luma prediction samples $pred4x4_L[x, y]$ with x, y = 0..3.
2. The position of the upper-left sample of a 4x4 luma block with index luma4x4BlkIdx inside the current macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (xO, yO) and x, y = 0..3.

$$pred_L[xO + x, yO + y] = pred4x4_L[x, y] \quad (8\text{-}41)$$

3. The transform coefficient decoding process and picture construction process prior to deblocking filter process in subclause 8.5 is invoked with $pred_L$ and luma4x4BlkIdx as the input and the constructed samples for the current 4x4 luma block S'_L as the output.

8.3.1.1 Derivation process for the Intra4x4PredMode

Inputs to this process are the index of the 4x4 luma block luma4x4BlkIdx and variable arrays Intra4x4PredMode (if available) and Intra8x8PredMode (if available) that are previously (in decoding order) derived for adjacent macroblocks.

Output of this process is the variable Intra4x4PredMode[luma4x4BlkIdx].

Table 8-2 specifies the values for Intra4x4PredMode[luma4x4BlkIdx] and the associated names.

Table 8-2 – Specification of Intra4x4PredMode[luma4x4BlkIdx] and associated names

Intra4x4PredMode[luma4x4BlkIdx]	Name of Intra4x4PredMode[luma4x4BlkIdx]
0	Intra_4x4_Vertical (prediction mode)
1	Intra_4x4_Horizontal (prediction mode)
2	Intra_4x4_DC (prediction mode)
3	Intra_4x4_Diagonal_Down_Left (prediction mode)
4	Intra_4x4_Diagonal_Down_Right (prediction mode)
5	Intra_4x4_Vertical_Right (prediction mode)
6	Intra_4x4_Horizontal_Down (prediction mode)
7	Intra_4x4_Vertical_Left (prediction mode)
8	Intra_4x4_Horizontal_Up (prediction mode)

Intra4x4PredMode[luma4x4BlkIdx] labelled 0, 1, 3, 4, 5, 6, 7, and 8 represent directions of predictions as illustrated in Figure 8-1.

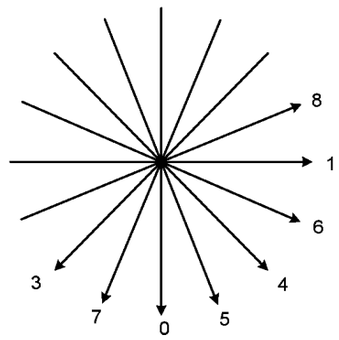


Figure 8-1 – Intra_4x4 prediction mode directions (informative)

Intra4x4PredMode[luma4x4BlkIdx] is derived as follows.

– The process specified in subclause 6.4.8.3 is invoked with luma4x4BlkIdx given as input and the output is assigned to mbAddrA, luma4x4BlkIdxA, mbAddrB, and luma4x4BlkIdxB.

– The variable dcPredModePredictedFlag is derived as follows.

  – If any of the following conditions are true, dcPredModePredictedFlag is set equal to 1

    – the macroblock with address mbAddrA is not available

    – the macroblock with address mbAddrB is not available

    – the macroblock with address mbAddrA is available and coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1

    – the macroblock with address mbAddrB is available and coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1

  – Otherwise, dcPredModePredictedFlag is set equal to 0.

– For N being either replaced by A or B, the variables intraMxMPredModeN are derived as follows.

  – If dcPredModePredictedFlag is equal to 1 or the macroblock with address mbAddrN is not coded in Intra_4x4 or Intra_8x8 macroblock prediction mode, intraMxMPredModeN is set equal to 2 (Intra_4x4_DC prediction mode).

  – Otherwise (dcPredModePredictedFlag is equal to 0 and (the macroblock with address mbAddrN is coded in Intra_4x4 macroblock prediction mode or the macroblock with address mbAddrN is coded in Intra_8x8 macroblock prediction mode)), the following applies.

    – If the macroblock with address mbAddrN is coded in Intra_4x4 macroblock mode, intraMxMPredModeN is set equal to Intra4x4PredMode[luma4x4BlkIdxN], where Intra4x4PredMode is the variable array assigned to the macroblock mbAddrN.

    – Otherwise (the macroblock with address mbAddrN is coded in Intra_8x8 macroblock mode), intraMxMPredModeN is set equal to Intra8x8PredMode[luma4x4BlkIdxN >> 2], where Intra8x8PredMode is the variable array assigned to the macroblock mbAddrN.

– Intra4x4PredMode[luma4x4BlkIdx] is derived by applying the following procedure.

```
predIntra4x4PredMode = Min( intraMxMPredModeA, intraMxMPredModeB )
if( prev_intra4x4_pred_mode_flag[ luma4x4BlkIdx ] )
    Intra4x4PredMode[ luma4x4BlkIdx ] = predIntra4x4PredMode
else                                                                    (8-42)
    if( rem_intra4x4_pred_mode[ luma4x4BlkIdx ] < predIntra4x4PredMode )
        Intra4x4PredMode[ luma4x4BlkIdx ] = rem_intra4x4_pred_mode[ luma4x4BlkIdx ]
    else
        Intra4x4PredMode[ luma4x4BlkIdx ] = rem_intra4x4_pred_mode[ luma4x4BlkIdx ] + 1
```

8.3.1.2 Intra_4x4 sample prediction

This process is invoked for each 4x4 luma block of a macroblock with prediction mode equal to Intra_4x4 followed by the transform decoding process and picture construction process prior to deblocking for each 4x4 luma block.

Input to this process is the index of a 4x4 luma block luma4x4BlkIdx.

Output of this process are the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 for the 4x4 luma block with index luma4x4BlkIdx.

The position of the upper-left sample of a 4x4 luma block with index luma4x4BlkIdx inside the current macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (xO, yO).

The 13 neighbouring samples p[x, y] that are constructed luma samples prior to the deblocking filter process, with x = -1, y = -1..3 and x = 0..7, y = -1, are derived as follows.

– The luma location (xN, yN) is specified by

$$xN = xO + x \tag{8-43}$$

$$yN = yO + y \tag{8-44}$$

– The derivation process for neighbouring locations in subclause 6.4.9 is invoked for luma locations with (xN, yN) as input and mbAddrN and (xW, yW) as output.

– Each sample p[x, y] with x = -1, y = -1..3 and x = 0..7, y = -1 is derived as follows.

  – If any of the following conditions is true, the sample p[x, y] is marked as "not available for Intra_4x4 prediction"

    – mbAddrN is not available,

    – the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

    – the macroblock mbAddrN has mb_type equal to SI and constrained_intra_pred_flag is equal to 1 and the current macroblock does not have mb_type equal to SI.

    – x is greater than 3 and luma4x4BlkIdx is equal to 3 or 11

  – Otherwise, the sample p[x, y] is marked as "available for Intra_4x4 prediction" and the luma sample at luma location (xW, yW) inside the macroblock mbAddrN is assigned to p[x, y].

When samples p[x, -1], with x = 4..7 are marked as "not available for Intra_4x4 prediction," and the sample p[3, -1] is marked as "available for Intra_4x4 prediction," the sample value of p[3, -1] is substituted for sample values p[x, -1], with x = 4..7 and samples p[x, -1], with x = 4..7 are marked as "available for Intra_4x4 prediction".

NOTE – Each block is assumed to be constructed into a picture array prior to decoding of the next block.

Depending on Intra4x4PredMode[luma4x4BlkIdx], one of the Intra_4x4 prediction modes specified in subclauses 8.3.1.2.1 to 8.3.1.2.9 is invoked.

8.3.1.2.1 Specification of Intra_4x4_Vertical prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 0.

This mode shall be used only when the samples p[x, -1] with x = 0..3 are marked as "available for Intra_4x4 prediction".

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived by

$$pred4x4_L[x, y] = p[x, -1], \text{ with } x, y = 0..3 \tag{8-45}$$

8.3.1.2.2 Specification of Intra_4x4_Horizontal prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 1.

This mode shall be used only when the samples p[-1, y], with y = 0..3 are marked as "available for Intra_4x4 prediction".

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived by

$$pred4x4_L[x, y] = p[-1, y], \text{ with } x,y = 0..3 \quad (8\text{-}46)$$

8.3.1.2.3 Specification of Intra_4x4_DC prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 2.

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If all samples p[x, -1], with x = 0..3 and p[-1, y], with y = 0..3 are marked as "available for Intra_4x4 prediction", the values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived by

$$pred4x4_L[x, y] = (p[0, -1] + p[1, -1] + p[2, -1] + p[3, -1] + p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + 4) >> 3 \quad (8\text{-}47)$$

– Otherwise, if any samples p[x, -1], with x = 0..3 are marked as "not available for Intra_4x4 prediction" and all samples p[-1, y], with y = 0..3 are marked as "available for Intra_4x4 prediction", the values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived by

$$pred4x4_L[x, y] = (p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + 2) >> 2 \quad (8\text{-}48)$$

– Otherwise, if any samples p[-1, y], with y = 0..3 are marked as "not available for Intra_4x4 prediction" and all samples p[x, -1], with x = 0 .. 3 are marked as "available for Intra_4x4 prediction", the values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0 .. 3 are derived by

$$pred4x4_L[x, y] = (p[0, -1] + p[1, -1] + p[2, -1] + p[3, -1] + 2) >> 2 \quad (8\text{-}49)$$

– Otherwise (some samples p[x, -1], with x = 0..3 and some samples p[-1, y], with y = 0..3 are marked as "not available for Intra_4x4 prediction"), the values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived by

$$pred4x4_L[x, y] = (1 << (BitDepth_Y - 1)) \quad (8\text{-}50)$$

NOTE – A 4x4 luma block can always be predicted using this mode.

8.3.1.2.4 Specification of Intra_4x4_Diagonal_Down_Left prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 3.

This mode shall be used only when the samples p[x, -1] with x = 0..7 are marked as "available for Intra_4x4 prediction".

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If x is equal to 3 and y is equal to 3,

$$pred4x4_L[x, y] = (p[6, -1] + 3 * p[7, -1] + 2) >> 2 \quad (8\text{-}51)$$

– Otherwise (x is not equal to 3 or y is not equal to 3),

$$pred4x4_L[x, y] = (p[x + y, -1] + 2 * p[x + y + 1, -1] + p[x + y + 2, -1] + 2) >> 2 \quad (8\text{-}52)$$

8.3.1.2.5 Specification of Intra_4x4_Diagonal_Down_Right prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 4.

This mode shall be used only when the samples p[x, -1] with x = 0..3 and p[-1, y] with y = -1..3 are marked as "available for Intra_4x4 prediction".

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If x is greater than y,

$$pred4x4_L[x, y] = (p[x - y - 2, -1] + 2 * p[x - y - 1, -1] + p[x - y, -1] + 2) >> 2 \quad (8\text{-}53)$$

– Otherwise if x is less than y,

$$pred4x4_L[x, y] = (p[-1, y - x - 2] + 2 * p[-1, y - x - 1] + p[-1, y - x] + 2) >> 2 \quad (8\text{-}54)$$

– Otherwise (x is equal to y),

$$pred4x4_L[x, y] = (p[0, -1] + 2 * p[-1, -1] + p[-1, 0] + 2) >> 2 \quad (8\text{-}55)$$

8.3.1.2.6 Specification of Intra_4x4_Vertical_Right prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 5.

This mode shall be used only when the samples p[x, -1] with x = 0..3 and p[-1, y] with y = -1..3 are marked as "available for Intra_4x4 prediction".

Let the variable zVR be set equal to 2 * x – y.

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If zVR is equal to 0, 2, 4, or 6,

$$pred4x4_L[x, y] = (p[x - (y >> 1) - 1, -1] + p[x - (y >> 1), -1] + 1) >> 1 \quad (8\text{-}56)$$

– Otherwise, if zVR is equal to 1, 3, or 5,

$$pred4x4_L[x, y] = (p[x - (y >> 1) - 2, -1] + 2 * p[x - (y >> 1) - 1, -1] + p[x - (y >> 1), -1] + 2) >> 2 \quad (8\text{-}57)$$

– Otherwise, if zVR is equal to -1,

$$pred4x4_L[x, y] = (p[-1, 0] + 2 * p[-1, -1] + p[0, -1] + 2) >> 2 \quad (8\text{-}58)$$

– Otherwise (zVR is equal to -2 or -3),

$$pred4x4_L[x, y] = (p[-1, y - 1] + 2 * p[-1, y - 2] + p[-1, y - 3] + 2) >> 2 \quad (8\text{-}59)$$

8.3.1.2.7 Specification of Intra_4x4_Horizontal_Down prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 6.

This mode shall be used only when the samples p[x, -1] with x = 0..3 and p[-1, y] with y = -1..3 are marked as "available for Intra_4x4 prediction".

Let the variable zHD be set equal to 2 * y – x.

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If zHD is equal to 0, 2, 4, or 6,

$$pred4x4_L[x, y] = (p[-1, y -(x >> 1) - 1] + p[-1, y - (x >> 1)] + 1) >> 1 \quad (8\text{-}60)$$

– Otherwise, if zHD is equal to 1, 3, or 5,

$$pred4x4_L[x, y] = (p[-1, y - (x >> 1) - 2] + 2 * p[-1, y - (x >> 1) - 1] + p[-1, y - (x >> 1)] + 2) >> 2 \quad (8\text{-}61)$$

– Otherwise, if zHD is equal to -1,

$$pred4x4_L[x, y] = (p[-1, 0] + 2 * p[-1, -1] + p[0, -1] + 2) >> 2 \quad (8\text{-}62)$$

– Otherwise (zHD is equal to -2 or -3),

$$pred4x4_L[x, y] = (p[x - 1, -1] + 2 * p[x - 2, -1] + p[x - 3, -1] + 2) >> 2 \quad (8\text{-}63)$$

8.3.1.2.8 Specification of Intra_4x4_Vertical_Left prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 7.

This mode shall be used only when the samples p[x, -1] with x = 0..7 are marked as "available for Intra_4x4 prediction".

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows.

– If y is equal to 0 or 2,

$$pred4x4_L[x, y] = (p[x + (y >> 1), -1] + p[x + (y >> 1) + 1, -1] + 1) >> 1 \quad (8\text{-}64)$$

– Otherwise (y is equal to 1 or 3),

$$pred4x4_L[x, y] = (p[x + (y >> 1), -1] + 2 * p[x + (y >> 1) + 1, -1] + p[x + (y >> 1) + 2, -1] + 2) >> 2 \quad (8\text{-}65)$$

8.3.1.2.9 Specification of Intra_4x4_Horizontal_Up prediction mode

This Intra_4x4 prediction mode is invoked when Intra4x4PredMode[luma4x4BlkIdx] is equal to 8.

This mode shall be used only when the samples p[-1, y] with y = 0..3 are marked as "available for Intra_4x4 prediction".

Let the variable zHU be set equal to x + 2 * y.

The values of the prediction samples $pred4x4_L[x, y]$, with x, y = 0..3 are derived as follows:

– If zHU is equal to 0, 2, or 4

$$pred4x4_L[x, y] = (p[-1, y + (x >> 1)] + p[-1, y + (x >> 1) + 1] + 1) >> 1 \quad (8\text{-}66)$$

– Otherwise, if zHU is equal to 1 or 3

$$pred4x4_L[x, y] = (p[-1, y + (x >> 1)] + 2 * p[-1, y + (x >> 1) + 1] + p[-1, y + (x >> 1) + 2] + 2) >> 2 \quad (8\text{-}67)$$

– Otherwise, if zHU is equal to 5,

$$pred4x4_L[x, y] = (p[-1, 2] + 3 * p[-1, 3] + 2) >> 2 \quad (8\text{-}68)$$

– Otherwise (zHU is greater than 5),

$$pred4x4_L[x, y] = p[-1, 3] \quad (8\text{-}69)$$

8.3.2 Intra_8x8 prediction process for luma samples

This process is invoked when the macroblock prediction mode is equal to Intra_8x8.

Inputs to this process are the values of Intra4x4PredMode (if available) or Intra8x8PredMode (if available) from the neighbouring macroblocks or macroblock pairs.

Outputs of this process are 8x8 luma sample arrays as part of the 16x16 luma array of prediction samples of the macroblock $pred_L$.

The luma component of a macroblock consists of 4 blocks of 8x8 luma samples. These blocks are inverse scanned using the inverse 8x8 luma block scanning process as specified in subclause 6.4.4.

For all 8x8 luma blocks of the luma component of a macroblock with luma8x8BlkIdx = 0..3, the derivation process for Intra8x8PredMode as specified in subclause 8.3.2.1 is invoked with luma8x8BlkIdx as well as Intra4x4PredMode and

Intra8x8PredMode that are previously (in decoding order) derived for adjacent macroblocks as the input and the variable Intra8x8PredMode[luma8x8BlkIdx] as the output.

For each luma block of 8x8 samples indexed using luma8x8BlkIdx = 0..3, the following applies.

– The Intra_8x8 sample prediction process in subclause 8.3.2.2 is invoked with luma8x8BlkIdx and constructed samples prior (in decoding order) to the deblocking filter process from adjacent luma blocks as the input and the output are the Intra_8x8 luma prediction samples $pred8x8_L[x, y]$ with x, y = 0..7.

– The position of the upper-left sample of an 8x8 luma block with index luma8x8BlkIdx inside the current macroblock is derived by invoking the inverse 8x8 luma block scanning process in subclause 6.4.4 with luma8x8BlkIdx as the input and the output being assigned to (xO, yO) and x, y = 0..7.

$$pred_L[xO + x, yO + y] = pred8x8_L[x, y] \quad (8\text{-}70)$$

– The transform coefficient decoding process and picture construction process prior to deblocking filter process in subclause 8.5 is invoked with $pred_L$ and luma8x8BlkIdx as the input and the constructed samples for the current 8x8 luma block S'_L as the output.

8.3.2.1 Derivation process for Intra8x8PredMode

Inputs to this process are the index of the 8x8 luma block luma8x8BlkIdx and variable arrays Intra4x4PredMode (if available) and Intra8x8PredMode (if available) that are previously (in decoding order) derived for adjacent macroblocks.

Output of this process is the variable Intra8x8PredMode[luma8x8BlkIdx].

Table 8-3 specifies the values for Intra8x8PredMode[luma8x8BlkIdx] and the associated mnemonic names.

Table 8-3 – Specification of Intra8x8PredMode[luma8x8BlkIdx] and associated names

Intra8x8PredMode[luma8x8BlkIdx]	Name of Intra8x8PredMode[luma8x8BlkIdx]
0	Intra_8x8_Vertical (prediction mode)
1	Intra_8x8_Horizontal (prediction mode)
2	Intra_8x8_DC (prediction mode)
3	Intra_8x8_Diagonal_Down_Left (prediction mode)
4	Intra_8x8_Diagonal_Down_Right (prediction mode)
5	Intra_8x8_Vertical_Right (prediction mode)
6	Intra_8x8_Horizontal_Down (prediction mode)
7	Intra_8x8_Vertical_Left (prediction mode)
8	Intra_8x8_Horizontal_Up (prediction mode)

Intra8x8PredMode[luma8x8BlkIdx] is derived as follows.

– The process specified in subclause 6.4.8.2 is invoked with luma8x8BlkIdx given as input and the output is assigned to mbAddrA, luma8x8BlkIdxA, mbAddrB, and luma8x8BlkIdxB.

– The variable dcPredModePredictedFlag is derived as follows.

  – If any of the following conditions are true, dcPredModePredictedFlag is set equal to 1

    – the macroblock with address mbAddrA is not available

    – the macroblock with address mbAddrB is not available

    – the macroblock with address mbAddrA is available and coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1

- the macroblock with address mbAddrB is available and coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1

- Otherwise, dcPredModePredictedFlag is set equal to 0.

– For N being either replaced by A or B, the variables intraMxMPredModeN are derived as follows.

  - If dcPredModePredictedFlag is equal to 1 or (the macroblock with address mbAddrN is not coded in Intra_4x4 macroblock prediction mode and the macroblock with address mbAddrN is not coded in Intra_8x8 macroblock prediction mode), intraMxMPredModeN is set equal to 2 (Intra_8x8_DC prediction mode).

  - Otherwise (dcPredModePredictedFlag is equal to 0 and (the macroblock with address mbAddrN is coded in Intra_4x4 macroblock prediction mode or the macroblock with address mbAddrN is coded in Intra_8x8 macroblock prediction mode)), the following applies.

    - If the macroblock with address mbAddrN is coded in Intra_8x8 macroblock mode, intraMxMPredModeN is set equal to Intra8x8PredMode[luma8x8BlkIdxN], where Intra8x8PredMode is the variable array assigned to the macroblock mbAddrN.

    - Otherwise (the macroblock with address mbAddrN is coded in Intra_4x4 macroblock mode), intraMxMPredModeN is derived by the following procedure, where Intra4x4PredMode is the variable array assigned to the macroblock mbAddrN.

      intraMxMPredModeN = Intra4x4PredMode[luma8x8BlkIdxN * 4 + n] (8-71)

      where the variable n is derived as follows

      - If N is equal to A, depending on the variable MbaffFrameFlag, the variable luma8x8BlkIdx, the current macroblock, and the macroblock mbAddrN, the following applies.

        - If MbaffFrameFlag is equal to 1, the current macroblock is a frame coded macroblock, the macroblock mbAddrN is a field coded macroblock, and luma8x8BlkIdx is equal to 2, n is set equal to 3.

        - Otherwise (MbaffFrameFlag is equal to 0 or the current macroblock is a field coded macroblock or the macroblock mbAddrN is a frame coded macroblock or luma8x8BlkIdx is not equal to 2), n is set equal to 1.

      - Otherwise (N is equal to B), n is set equal to 2.

– Finally, given intraMxMPredModeA and intraMxMPredModeB, the variable Intra8x8PredMode[luma8x8BlkIdx] is derived by applying the following procedure.

```
predIntra8x8PredMode = Min( intraMxMPredModeA, intraMxMPredModeB )
if( prev_intra8x8_pred_mode_flag[ luma8x8BlkIdx ] )
    Intra8x8PredMode[ luma8x8BlkIdx ] = predIntra8x8PredMode
else                                                                    (8-72)
    if( rem_intra8x8_pred_mode[ luma8x8BlkIdx ] < predIntra8x8PredMode )
        Intra8x8PredMode[ luma8x8BlkIdx ] = rem_intra8x8_pred_mode[ luma8x8BlkIdx ]
    else
        Intra8x8PredMode[ luma8x8BlkIdx ] = rem_intra8x8_pred_mode[ luma8x8BlkIdx ] + 1
```

8.3.2.2 Intra_8x8 sample prediction

This process is invoked for each 8x8 luma block of a macroblock with prediction mode equal to Intra_8x8 followed by the transform decoding process and picture construction process prior to deblocking for each 8x8 luma block.

Input to this process is the index of an 8x8 luma block luma8x8BlkIdx.

Output of this process are the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 for the 8x8 luma block with index luma8x8BlkIdx.

The position of the upper-left sample of an 8x8 luma block with index luma8x8BlkIdx inside the current macroblock is derived by invoking the inverse 8x8 luma block scanning process in subclause 6.4.4 with luma8x8BlkIdx as the input and the output being assigned to (xO, yO).

The 25 neighbouring samples p[x, y] that are constructed luma samples prior to the deblocking filter process, with x = -1, y = -1..7 and x = 0..15, y = -1, are derived as follows.

– The luma location (xN, yN) is specified by

$$xN = xO + x \tag{8-73}$$

$$yN = yO + y \tag{8-74}$$

– The derivation process for neighbouring locations in subclause 6.4.9 is invoked for luma locations with (xN, yN) as input and mbAddrN and (xW, yW) as output.

– Each sample p[x, y] with x = -1, y = -1..7 and x = 0..15, y = -1 is derived as follows.

  – If any of the following conditions is true, the sample p[x, y] is marked as "not available for Intra_8x8 prediction"

    – mbAddrN is not available,

    – the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1,

  – Otherwise, the sample p[x, y] is marked as "available for Intra_8x8 prediction" and the luma sample at luma location (xW, yW) inside the macroblock mbAddrN is assigned to p[x, y].

When samples p[x, -1], with x = 8..15 are marked as "not available for Intra_8x8 prediction," and the sample p[7, -1] is marked as "available for Intra_8x8 prediction," the sample value of p[7, -1] is substituted for sample values p[x, -1], with x = 8..15 and samples p[x, -1], with x = 8..15 are marked as "available for Intra_8x8 prediction".

NOTE – Each block is assumed to be constructed into a picture array prior to decoding of the next block.

The reference sample filtering process for Intra_8x8 sample prediction in subclause 8.3.2.2.1 is invoked with the samples p[x, y] with x = -1, y = -1..7 and x = 0..15, y = -1 (if available) as input and p'[x, y] with x = -1, y = -1..7 and x = 0..15, y = -1 as output.

Depending on Intra8x8PredMode[luma8x8BlkIdx], one of the Intra_8x8 prediction modes specified in subclauses 8.3.2.2.2 to 8.3.2.2.10 is invoked.

8.3.2.2.1 Reference sample filtering process for Intra_8x8 sample prediction

Inputs to this process are the reference samples p[x, y] with x = -1, y = -1..7 and x = 0..15, y = -1 (if available) for Intra_8x8 sample prediction.

Outputs of this process are the filtered reference samples p'[x, y] with x = -1, y = -1..7 and x = 0..15, y = -1 for Intra_8x8 sample prediction.

When all samples p[x, -1] with x = 0..7 are marked as "available for Intra_8x8 prediction", the following applies.

– The value of p'[0, -1] is derived as follows.

  – If p[-1, -1] is marked as "available for Intra_8x8 prediction", p'[0, -1] is derived by

$$p'[\,0, -1\,] = (\, p[\,-1, -1\,] + 2 * p[\,0, -1\,] + p[\,1, -1\,] + 2\,) >> 2 \tag{8-75}$$

  – Otherwise (p[-1, -1] is marked as "not available for Intra_8x8 prediction"), p'[0, -1] is derived by

$$p'[\,0, -1\,] = (\, 3 * p[\,0, -1\,] + p[\,1, -1\,] + 2\,) >> 2 \tag{8-76}$$

– The values of p'[x, -1], with x = 1..7 are derived by

$$p'[\,x, -1\,] = (\, p[\,x-1, -1\,] + 2 * p[\,x, -1\,] + p[\,x+1, -1\,] + 2\,) >> 2 \tag{8-77}$$

When all samples p[x, -1] with x = 7..15 are marked as "available for Intra_8x8 prediction", the following applies.

– The values of p'[x, -1], with x = 8..14 are derived by

$$p'[\,x, -1\,] = (\, p[\,x-1, -1\,] + 2 * p[\,x, -1\,] + p[\,x+1, -1\,] + 2\,) >> 2 \tag{8-78}$$

– The value of p'[15, -1] is derived by

$$p'[\,15, -1\,] = (\, p[\,14, -1\,] + 3 * p[\,15, -1\,] + 2\,) >> 2 \tag{8-79}$$

When the sample p[-1, -1] is marked as "available for Intra_8x8 prediction", the value of p'[-1, -1] is derived as follows.

– If the sample p[0, -1] is marked as "not available for Intra_8x8 prediction" or the sample p[-1, 0] is marked as "not available for Intra_8x8 prediction", the following applies.

  – If the sample p[0, -1] is marked as "available for Intra_8x8 prediction", p'[-1, -1] is derived by

$$p'[-1, -1] = (3 * p[-1, -1] + p[0, -1] + 2) >> 2 \quad (8\text{-}80)$$

  – Otherwise (the sample p[0, -1] is marked as "not available for Intra_8x8 prediction" and the sample p[-1, 0] is marked as "available for Intra_8x8 prediction"), p'[-1, -1] is derived by

$$p'[-1, -1] = (3 * p[-1, -1] + p[-1, 0] + 2) >> 2 \quad (8\text{-}81)$$

– Otherwise (the sample p[0, -1] is marked as "available for Intra_8x8 prediction" and the sample p[-1, 0] is marked as "available for Intra_8x8 prediction"), p'[-1, -1] is derived by

$$p'[-1, -1] = (p[0, -1] + 2 * p[-1, -1] + p[-1, 0] + 2) >> 2 \quad (8\text{-}82)$$

When all samples p[-1, y] with y = 0..7 are marked as "available for Intra_8x8 prediction", the following applies.

– The value of p'[-1, 0] is derived as follows.

  – If p[-1, -1] is marked as "available for Intra_8x8 prediction", p'[-1, 0] is derived by

$$p'[-1, 0] = (p[-1, -1] + 2 * p[-1, 0] + p[-1, 1] + 2) >> 2 \quad (8\text{-}83)$$

  – Otherwise (if p[-1, -1] is marked as "not available for Intra_8x8 prediction"), p'[-1, 0] is derived by

$$p'[-1, 0] = (3 * p[-1, 0] + p[-1, 1] + 2) >> 2 \quad (8\text{-}84)$$

– The values of p'[-1, y], with y = 1..6 are derived by

$$p'[-1, y] = (p[-1, y-1] + 2 * p[-1, y] + p[-1, y+1] + 2) >> 2 \quad (8\text{-}85)$$

– The value of p'[-1, 7] is derived by

$$p'[-1, 7] = (p[-1, 6] + 3 * p[-1, 7] + 2) >> 2 \quad (8\text{-}86)$$

8.3.2.2.2 Specification of Intra_8x8_Vertical prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 0.

This mode shall be used only when the samples p[x, -1] with x = 0..7 are marked as "available for Intra_8x8 prediction".

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = p'[x, -1], \text{ with } x, y = 0..7 \quad (8\text{-}87)$$

8.3.2.2.3 Specification of Intra_8x8_Horizontal prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 1.

This mode shall be used only when the samples p[-1, y], with y = 0..7 are marked as "available for Intra_8x8 prediction".

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = p'[-1, y], \text{ with } x, y = 0..7 \quad (8\text{-}88)$$

8.3.2.2.4 Specification of Intra_8x8_DC prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 2.

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If all samples p[x, -1], with x = 0..7 and p[-1, y], with y = 0..7 are marked as "available for Intra_8x8 prediction," the values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = (\sum_{x'=0}^{7} p'[x',-1] + \sum_{y'=0}^{7} p'[-1, y'] + 8) >> 4 \quad (8\text{-}89)$$

– Otherwise, if any samples p[x, -1], with x = 0..7 are marked as "not available for Intra_8x8 prediction" and all samples p[-1, y], with y = 0..7 are marked as "available for Intra_8x8 prediction", the values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = (\sum_{y'=0}^{7} p'[-1, y'] + 4) >> 3 \quad (8\text{-}90)$$

– Otherwise, if any samples p[-1, y], with y = 0..7 are marked as "not available for Intra_8x8 prediction" and all samples p[x, -1], with x = 0..7 are marked as "available for Intra_8x8 prediction", the values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = (\sum_{x'=0}^{7} p'[x',-1] + 4) >> 3 \quad (8\text{-}91)$$

– Otherwise (some samples p[x, -1], with x = 0..7 and some samples p[-1, y], with y = 0..7 are marked as "not available for Intra_8x8 prediction"), the values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived by

$$pred8x8_L[x, y] = (1 << (BitDepth_Y - 1)) \quad (8\text{-}92)$$

NOTE – An 8x8 luma block can always be predicted using this mode.

8.3.2.2.5 Specification of Intra_8x8_Diagonal_Down_Left prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 3.

This mode shall be used only when the samples p[x, -1] with x = 0..15 are marked as "available for Intra_8x8 prediction".

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If x is equal to 7 and y is equal to 7,

$$pred8x8_L[x, y] = (p'[14, -1] + 3 * p'[15, -1] + 2) >> 2 \quad (8\text{-}93)$$

– Otherwise (x is not equal to 7 or y is not equal to 7),

$$pred8x8_L[x, y] = (p'[x + y, -1] + 2 * p'[x + y + 1, -1] + p'[x + y + 2, -1] + 2) >> 2 \quad (8\text{-}94)$$

8.3.2.2.6 Specification of Intra_8x8_Diagonal_Down_Right prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 4.

This mode shall be used only when the samples p[x, -1] with x = 0..7 and p[-1, y] with y = -1..7 are marked as "available for Intra_8x8 prediction".

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If x is greater than y,

$$pred8x8_L[x, y] = (p'[x - y - 2, -1] + 2 * p'[x - y - 1, -1] + p'[x - y, -1] + 2) >> 2 \quad (8\text{-}95)$$

– Otherwise if x is less than y,

$$pred8x8_L[x, y] = (p'[-1, y - x - 2] + 2 * p'[-1, y - x - 1] + p'[-1, y - x] + 2) >> 2 \quad (8\text{-}96)$$

– Otherwise (x is equal to y),

$$pred8x8_L[x, y] = (p'[0, -1] + 2 * p'[-1, -1] + p'[-1, 0] + 2) >> 2 \quad (8\text{-}97)$$

8.3.2.2.7 Specification of Intra_8x8_Vertical_Right prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 5.

This mode shall be used only when the samples p[x, -1] with x = 0..7 and p[-1, y] with y = -1..7 are marked as "available for Intra_8x8 prediction".

Let the variable zVR be set equal to 2 * x – y.

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If zVR is equal to 0, 2, 4, 6, 8, 10, 12, or 14

$$pred8x8_L[x, y] = (p'[x - (y >> 1) - 1, -1] + p'[x - (y >> 1), -1] + 1) >> 1 \quad (8\text{-}98)$$

– Otherwise, if zVR is equal to 1, 3, 5, 7, 9, 11, or 13

$$pred8x8_L[x, y] = (p'[x - (y >> 1) - 2, -1] + 2 * p'[x - (y >> 1) - 1, -1] + p'[x - (y >> 1), -1] + 2) >> 2 \quad (8\text{-}99)$$

– Otherwise, if zVR is equal to -1,

$$pred8x8_L[x, y] = (p'[-1, 0] + 2 * p'[-1, -1] + p'[0, -1] + 2) >> 2 \quad (8\text{-}100)$$

– Otherwise (zVR is equal to -2, -3, -4, -5, -6, or -7),

$$pred8x8_L[x, y] = (p'[-1, y - 2*x - 1] + 2 * p'[-1, y - 2*x - 2] + p'[-1, y - 2*x - 3] + 2) >> 2 \quad (8\text{-}101)$$

8.3.2.2.8 Specification of Intra_8x8_Horizontal_Down prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 6.

This mode shall be used only when the samples p[x, -1] with x = 0..7 and p[-1, y] with y = -1..7 are marked as "available for Intra_8x8 prediction".

Let the variable zHD be set equal to 2 * y – x.

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If zHD is equal to 0, 2, 4, 6, 8, 10, 12, or 14

$$pred8x8_L[x, y] = (p'[-1, y -(x >> 1) - 1] + p'[-1, y - (x >> 1)] + 1) >> 1 \quad (8\text{-}102)$$

– Otherwise, if zHD is equal to 1, 3, 5, 7, 9, 11, or 13

$$pred8x8_L[x, y] = (p'[-1, y - (x >> 1) - 2] + 2 * p'[-1, y - (x >> 1) - 1] + p'[-1, y - (x >> 1)] + 2) >> 2 \quad (8\text{-}103)$$

– Otherwise, if zHD is equal to -1,

$$pred8x8_L[x, y] = (p'[-1, 0] + 2 * p'[-1, -1] + p'[0, -1] + 2) >> 2 \quad (8\text{-}104)$$

– Otherwise (zHD is equal to –2, -3, -4, -5, -6, -7),

$$pred8x8_L[x, y] = (p'[x - 2*y - 1, -1] + 2 * p'[x - 2*y - 2, -1] + p'[x - 2*y - 3, -1] + 2) >> 2 \quad (8\text{-}105)$$

8.3.2.2.9 Specification of Intra_8x8_Vertical_Left prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 7.

This mode shall be used only when the samples p[x, -1] with x = 0..15 are marked as "available for Intra_8x8 prediction".

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows.

– If y is equal to 0, 2, 4 or 6

$$pred8x8_L[x, y] = (p'[x + (y >> 1), -1] + p'[x + (y >> 1) + 1, -1] + 1) >> 1 \quad (8\text{-}106)$$

– Otherwise (y is equal to 1, 3, 5, 7),

$$pred8x8_L[x, y] = (p'[x + (y >> 1), -1] + 2 * p'[x + (y >> 1) + 1, -1] + p'[x + (y >> 1) + 2, -1] + 2) >>2 \quad (8\text{-}107)$$

8.3.2.2.10 Specification of Intra_8x8_Horizontal_Up prediction mode

This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 8.

This mode shall be used only when the samples p[-1, y] with y = 0..7 are marked as "available for Intra_8x8 prediction".

Let the variable zHU be set equal to x + 2 * y.

The values of the prediction samples $pred8x8_L[x, y]$, with x, y = 0..7 are derived as follows:

– If zHU is equal to 0, 2, 4, 6, 8, 10, or 12

$$pred8x8_L[x, y] = (p'[-1, y + (x >> 1)] + p'[-1, y + (x >> 1) + 1] + 1) >> 1 \quad (8\text{-}108)$$

– Otherwise, if zHU is equal to 1, 3, 5, 7, 9, or 11

$$pred8x8_L[x, y] = (p'[-1, y + (x >> 1)] + 2 * p'[-1, y + (x >> 1) + 1] + p'[-1, y + (x >> 1) + 2] + 2) >>2 \quad (8\text{-}109)$$

– Otherwise, if zHU is equal to 13,

$$pred8x8_L[x, y] = (p'[-1, 6] + 3 * p'[-1, 7] + 2) >> 2 \quad (8\text{-}110)$$

– Otherwise (zHU is greater than 13),

$$pred8x8_L[x, y] = p'[-1, 7] \quad (8\text{-}111)$$

8.3.3 Intra_16x16 prediction process for luma samples

This process is invoked when the macroblock prediction mode is equal to Intra_16x16. It specifies how the Intra prediction luma samples for the current macroblock are derived.

Outputs of this process are Intra prediction luma samples for the current macroblock $pred_L[x, y]$.

The 33 neighbouring samples p[x, y] that are constructed luma samples prior to the deblocking filter process, with x = -1, y = -1..15 and with x = 0..15, y = -1, are derived as follows.

– The derivation process for neighbouring locations in subclause 6.4.9 is invoked for luma locations with (x, y) assigned to (xN, yN) as input and mbAddrN and (xW, yW) as output.

– Each sample p[x, y] with x = -1, y = -1..15 and with x = 0..15, y = -1 is derived as follows.

  – If any of the following conditions is true, the sample p[x, y] is marked as "not available for Intra_16x16 prediction"

    – mbAddrN is not available,

    – the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

– the macroblock mbAddrN has mb_type equal to SI and constrained_intra_pred_flag is equal to 1.

– Otherwise, the sample p[x, y] is marked as "available for Intra_16x16 prediction" and the luma sample at luma location (xW, yW) inside the macroblock mbAddrN is assigned to p[x, y].

Let $pred_L$[x, y] with x, y = 0..15 denote the prediction samples for the 16x16 luma block samples.

Intra_16x16 prediction modes are specified in Table 8-4.

Table 8-4 – Specification of Intra16x16PredMode and associated names

Intra16x16PredMode	Name of Intra16x16PredMode
0	Intra_16x16_Vertical (prediction mode)
1	Intra_16x16_Horizontal (prediction mode)
2	Intra_16x16_DC (prediction mode)
3	Intra_16x16_Plane (prediction mode)

Depending on Intra16x16PredMode, one of the Intra_16x16 prediction modes specified in subclauses 8.3.3.1 to 8.3.3.4 is invoked.

8.3.3.1 Specification of Intra_16x16_Vertical prediction mode

This Intra_16x16 prediction mode shall be used only when the samples p[x, -1] with x = 0..15 are marked as "available for Intra_16x16 prediction".

$$pred_L[x, y] = p[x, -1], \text{ with } x, y = 0..15 \quad (8\text{-}112)$$

8.3.3.2 Specification of Intra_16x16_Horizontal prediction mode

This Intra_16x16 prediction mode shall be used only when the samples p[-1, y] with y = 0..15 are marked as "available for Intra_16x16 prediction".

$$pred_L[x, y] = p[-1, y], \text{ with } x, y = 0..15 \quad (8\text{-}113)$$

8.3.3.3 Specification of Intra_16x16_DC prediction mode

This Intra_16x16 prediction mode operates, depending on whether the neighbouring samples are marked as "available for Intra_16x16 prediction", as follows.

– If all neighbouring samples p[x, -1], with x = 0..15 and p[-1, y], with y = 0..15 are marked as "available for Intra_16x16 prediction", the prediction for all luma samples in the macroblock is given by:

$$pred_L[x, y] = (\sum_{x'=0}^{15} p[x',-1] + \sum_{y'=0}^{15} p[-1,y'] + 16) >> 5, \text{ with } x, y = 0..15 \quad (8\text{-}114)$$

– Otherwise, if any of the neighbouring samples p[x, -1], with x = 0..15 are marked as "not available for Intra_16x16 prediction" and all of the neighbouring samples p[-1, y], with y = 0..15 are marked as "available for Intra_16x16 prediction", the prediction for all luma samples in the macroblock is given by:

$$pred_L[x, y] = (\sum_{y'=0}^{15} p[-1,y'] + 8) >> 4, \text{ with } x, y = 0..15 \quad (8\text{-}115)$$

– Otherwise, if any of the neighbouring samples p[-1, y], with y = 0..15 are marked as "not available for Intra_16x16 prediction" and all of the neighbouring samples p[x, -1], with x = 0..15 are marked as "available for Intra_16x16 prediction", the prediction for all luma samples in the macroblock is given by:

$$pred_L[x, y] = (\sum_{x'=0}^{15} p[x',-1] + 8) >> 4, \text{ with } x, y = 0..15 \quad (8\text{-}116)$$

– Otherwise (some of the neighbouring samples p[x, -1], with x = 0..15 and some of the neighbouring samples p[-1, y], with y = 0..15 are marked as "not available for Intra_16x16 prediction"), the prediction for all luma samples in the macroblock is given by:

$$pred_L[x, y] = (1 << (BitDepth_Y - 1)), \text{ with } x, y = 0..15 \quad (8\text{-}117)$$

8.3.3.4 Specification of Intra_16x16_Plane prediction mode

This Intra_16x16 prediction mode shall be used only when the samples p[x, -1] with x = -1..15 and p[-1, y] with y = 0..15 are marked as "available for Intra_16x16 prediction".

$$pred_L[x, y] = Clip1_Y((a + b * (x - 7) + c * (y - 7) + 16) >> 5), \text{ with } x, y = 0..15, \quad (8\text{-}118)$$

where:

$$a = 16 * (p[-1, 15] + p[15, -1]) \quad (8\text{-}119)$$

$$b = (5 * H + 32) >> 6 \quad (8\text{-}120)$$

$$c = (5 * V + 32) >> 6 \quad (8\text{-}121)$$

and H and V are specified in Equations 8-122 and 8-123.

$$H = \sum_{x'=0}^{7} (x'+1) * (p[8+x', -1] - p[6-x', -1]) \quad (8\text{-}122)$$

$$V = \sum_{y'=0}^{7} (y'+1) * (p[-1, 8+y'] - p[-1, 6-y']) \quad (8\text{-}123)$$

8.3.4 Intra prediction process for chroma samples

This process is invoked for I and SI macroblock types. It specifies how the Intra prediction chroma samples for the current macroblock are derived.

Outputs of this process are Intra prediction chroma samples for the current macroblock $pred_{Cb}[x, y]$ and $pred_{Cr}[x, y]$.

Both chroma blocks (Cb and Cr) of the macroblock use the same prediction mode. The prediction mode is applied to each of the chroma blocks separately. The process specified in this subclause is invoked for each chroma block. In the remainder of this subclause, chroma block refers to one of the two chroma blocks and the subscript C is used as a replacement of the subscript Cb or Cr.

The neighbouring samples p[x, y] that are constructed chroma samples prior to the deblocking filter process, with x = -1, y = -1..MbHeightC - 1 and with x = 0..MbWidthC - 1, y = -1, are derived as follows.

– The derivation process for neighbouring locations in subclause 6.4.9 is invoked for chroma locations with (x, y) assigned to (xN, yN) as input and mbAddrN and (xW, yW) as output.

– Each sample p[x, y] is derived as follows.

  – If any of the following conditions is true, the sample p[x, y] is marked as "not available for Intra chroma prediction"

    – mbAddrN is not available,

    – the macroblock mbAddrN is coded in Inter prediction mode and constrained_intra_pred_flag is equal to 1.

    – the macroblock mbAddrN has mb_type equal to SI and constrained_intra_pred_flag is equal to 1 and the current macroblock does not have mb_type equal to SI.

  – Otherwise, the sample p[x, y] is marked as "available for Intra chroma prediction" and the chroma sample of component C at chroma location (xW, yW) inside the macroblock mbAddrN is assigned to p[x, y].

Let $pred_C[x, y]$ with x = 0..MbWidthC - 1, y = 0..MbHeightC - 1 denote the prediction samples for the chroma block samples.

Intra chroma prediction modes are specified in Table 8-5.

Table 8-5 – Specification of Intra chroma prediction modes and associated names

intra_chroma_pred_mode	Name of intra_chroma_pred_mode
0	Intra_Chroma_DC (prediction mode)
1	Intra_Chroma_Horizontal (prediction mode)
2	Intra_Chroma_Vertical (prediction mode)
3	Intra_Chroma_Plane (prediction mode)

Depending on intra_chroma_pred_mode, one of the Intra chroma prediction modes specified in subclauses 8.3.4.1 to 8.3.4.4 is invoked.

8.3.4.1 Specification of Intra_Chroma_DC prediction mode

This Intra chroma prediction mode is invoked when intra_chroma_pred_mode is equal to 0.

For each chroma block of 4x4 samples indexed by chroma4x4BlkIdx = 0..(1 << (chroma_format_idc + 1)) – 1, the following applies.

– Depending on chroma_format_idc, the position of the upper-left sample of a 4x4 chroma block with index chroma4x4BlkIdx is derived as follows

  – If chroma_format_idc is equal to 1 or 2, the following applies

$$xO = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 0) \quad (8\text{-}124)$$

$$yO = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 1) \quad (8\text{-}125)$$

  – Otherwise (chroma_format_idc is equal to 3), the following applies

$$xO = InverseRasterScan(chroma4x4BlkIdx / 4, 8, 8, 16, 0) + InverseRasterScan(chroma4x4BlkIdx \% 4, 4, 4, 8, 0) \quad (8\text{-}126)$$

$$yO = InverseRasterScan(chroma4x4BlkIdx / 4, 8, 8, 16, 1) + InverseRasterScan(chroma4x4BlkIdx \% 4, 4, 4, 8, 1) \quad (8\text{-}127)$$

– If (xO, yO) is equal to (0, 0) or xO and yO are greater than 0, the values of the prediction samples $pred_C[x + xO, y + yO]$ with x, y = 0..3 are derived as follows.

  – If all samples p[x + xO, -1], with x = 0..3 and p[-1, y +yO], with y = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C[x + xO, y + yO]$, with x, y = 0..3 are derived as

$$pred_C[x + xO, y + yO] = \left(\sum_{x'=0}^{3} p[x'+xO,-1] + \sum_{y'=0}^{3} p[-1, y'+yO] + 4 \right) >> 3 \text{ , with x, y = 0..3.} \quad (8\text{-}128)$$

  – Otherwise, if any samples p[x + xO, -1], with x = 0..3 are marked as "not available for Intra chroma prediction" and all samples p[-1, y +yO], with y = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C[x + xO, y + yO]$, with x, y = 0..3 are derived as

$$pred_C[x + xO, y + yO] = \left(\sum_{y'=0}^{3} p[-1, y'+yO] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}129)$$

– Otherwise, if any samples p[-1, y +yO], with y = 0..3 are marked as "not available for Intra chroma prediction" and all samples p[x + xO, -1], with x = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = \left(\sum_{x'=0}^{3} p[x'+xO, -1] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}130)$$

– Otherwise (some samples p[x + xO, -1], with x = 0..3 and some samples p[-1, y +yO], with y = 0..3 are marked as "not available for Intra chroma prediction"), the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = (1 << (BitDepth_C - 1)), \text{ with x, y = 0..3.} \quad (8\text{-}131)$$

– Otherwise, if xO is greater than 0 and yO is equal to 0, the values of the prediction samples $pred_C$[x + xO, y + yO] with x, y = 0..3 are derived as follows.

– If all samples p[x + xO, -1], with x = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = \left(\sum_{x'=0}^{3} p[x'+xO, -1] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}132)$$

– Otherwise, if all samples p[-1, y +yO], with y = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = \left(\sum_{y'=0}^{3} p[-1, y'+yO] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}133)$$

– Otherwise (some samples p[x + xO, -1], with x = 0..3 and some samples p[-1, y +yO], with y = 0..3 are marked as "not available for Intra chroma prediction"), the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = (1 << (BitDepth_C - 1)), \text{ with x, y = 0..3.} \quad (8\text{-}134)$$

– Otherwise (xO is equal to 0 and yO is greater than 0), the values of the prediction samples $pred_C$[x + xO, y + yO] with x, y = 0..3 are derived as follows.

– If all samples p[-1, y +yO], with y = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = \left(\sum_{y'=0}^{3} p[-1, y'+yO] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}135)$$

– Otherwise, if all samples p[x + xO, -1], with x = 0..3 are marked as "available for Intra chroma prediction", the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = \left(\sum_{x'=0}^{3} p[x'+xO, -1] + 2 \right) >> 2 \text{ , with x, y = 0..3.} \quad (8\text{-}136)$$

– Otherwise (some samples p[x + xO, -1], with x = 0..3 and some samples p[-1, y +yO], with y = 0..3 are marked as "not available for Intra chroma prediction"), the values of the prediction samples $pred_C$[x + xO, y + yO], with x, y = 0..3 are derived as

$$pred_C[\, x + xO, y + yO\,] = (1 << (BitDepth_C - 1)), \text{ with x, y = 0..3.} \quad (8\text{-}137)$$

8.3.4.2 Specification of Intra_Chroma_Horizontal prediction mode

This Intra chroma prediction mode is invoked when intra_chroma_pred_mode is equal to 1.

This mode shall be used only when the samples p[-1, y] with y = 0..MbHeightC - 1 are marked as "available for Intra chroma prediction".

The values of the prediction samples $pred_C$[x, y] are derived as follows.

$pred_C$[x, y] = p[-1, y], with x = 0..MbWidthC - 1 and y = 0..MbHeightC - 1 (8-138)

8.3.4.3 Specification of Intra_Chroma_Vertical prediction mode

This Intra chroma prediction mode is invoked when intra_chroma_pred_mode is equal to 2.

This mode shall be used only when the samples p[x, -1] with x = 0..MbWidthC - 1 are marked as "available for Intra chroma prediction".

The values of the prediction samples $pred_C$[x, y] are derived as follows.

$pred_C$[x, y] = p[x, -1], with x = 0..MbWidthC - 1 and y = 0..MbHeightC - 1 (8-139)

8.3.4.4 Specification of Intra_Chroma_Plane prediction mode

This Intra chroma prediction mode is invoked when intra_chroma_pred_mode is equal to 3.

This mode shall be used only when the samples p[x, -1], with x = 0..MbWidthC - 1 and p[-1, y], with y = -1..MbHeightC - 1 are marked as "available for Intra chroma prediction".

The values of the prediction samples $pred_C$[x, y] are derived as follows.

Let the variable xCF be set equal to 4 * (chroma_format_idc == 3) and let the variable yCF be set equal to 4 * (chroma_format_idc != 1).

$pred_C$[x, y] = $Clip1_C$((a + b * (x – 3 – xCF) + c * (y – 3 – yCF) + 16) >> 5),
with x = 0..MbWidthC - 1 and y = 0..MbHeightC - 1 (8-140)

where:

a = 16 * (p[-1, MbHeightC - 1] + p[MbWidthC - 1, -1]) (8-141)

b = ((34 – 29 * (chroma_format_idc == 3)) * H + 32) >> 6 (8-142)

c = ((34 – 29 * (chroma_format_idc != 1)) * V + 32) >> 6 (8-143)

and H and V are specified as

$$H = \sum_{x'=0}^{3+xCF} (x'+1) * (p[4 + xCF + x', -1] - p[2 + xCF - x', -1]) \quad (8\text{-}144)$$

$$V = \sum_{y'=0}^{3+yCF} (y'+1) * (p[-1, 4 + yCF + y'] - p[-1, 2 + yCF - y']) \quad (8\text{-}145)$$

8.3.5 Sample construction process for I_PCM macroblocks

This process is invoked when mb_type is equal to I_PCM.

The variable dy is derived as follows.

– If MbaffFrameFlag is equal to 1 and the current macroblock is a field macroblock, dy is set equal to 2.

– Otherwise (MbaffFrameFlag is equal to 0 or the current macroblock is a frame macroblock), dy is set equal to 1.

The position of the upper-left luma sample of the current macroblock is derived by invoking the inverse macroblock scanning process in subclause 6.4.1 with CurrMbAddr as input and the output being assigned to (xP, yP).

The constructed luma samples prior to the deblocking process are generated as specified by:

$$S'_L[\, xP + (i \,\%\, 16),\ yP + dy * (i / 16))] = pcm_sample_luma[\, i \,] \tag{8-146}$$

When chroma_format_idc is not equal to 0 (monochrome), the constructed chroma samples prior to the deblocking process are generated as specified by:

$$\begin{aligned} &S'_{Cb}[\, (xP / SubWidthC) + (i \,\%\, MbWidthC),\\ &\qquad ((yP + SubHeightC - 1) / SubHeightC) + dy * (i / MbWidthC)] =\\ &\qquad pcm_sample_chroma[\, i \,]\\ &S'_{Cr}[\, (xP / SubWidthC) + (i \,\%\, MbWidthC),\\ &\qquad ((yP + SubHeightC - 1) / SubHeightC) + dy * (i / MbWidthC)] =\\ &\qquad pcm_sample_chroma[\, i + MbWidthC * MbHeightC \,] \end{aligned} \tag{8-147}$$

for(i = 0; i < MbWidthC * MbHeightC; i++) {
}

8.4 Inter prediction process

This process is invoked when decoding P and B macroblock types.

Outputs of this process are Inter prediction samples for the current macroblock that are a 16x16 array $pred_L$ of luma samples and when chroma_format_idc is not equal to 0 (monochrome) two 8x8 arrays $pred_{Cr}$ and $pred_{Cb}$ of chroma samples, one for each of the chroma components Cb and Cr.

The partitioning of a macroblock is specified by mb_type. Each macroblock partition is referred to by mbPartIdx. When the macroblock partitioning consists of partitions that are equal to sub-macroblocks, each sub-macroblock can be further partitioned into sub-macroblock partitions as specified by sub_mb_type. Each sub-macroblock partition is referred to by subMbPartIdx. When the macroblock partitioning does not consist of sub-macroblocks, subMbPartIdx is set equal to 0.

The following steps are specified for each macroblock partition or for each sub-macroblock partition.

The functions MbPartWidth( ), MbPartHeight( ), SubMbPartWidth( ), and SubMbPartHeight( ) describing the width and height of macroblock partitions and sub-macroblock partitions are specified in Tables 7-13, 7-14, 7-17, and 7-18.

The range of the macroblock partition index mbPartIdx is derived as follows.

– If mb_type is equal to B_Skip or B_Direct_16x16, mbPartIdx proceeds over values 0..3.

– Otherwise (mb_type is not equal to B_Skip or B_Direct_16x16), mbPartIdx proceeds over values 0..NumMbPart(mb_type) – 1.

For each value of mbPartIdx, the variables partWidth and partHeight for each macroblock partition or sub-macroblock partition in the macroblock are derived as follows.

– If mb_type is not equal to P_8x8, P_8x8ref0, B_Skip, B_Direct_16x16, or B_8x8, subMbPartIdx is set equal to 0, and partWidth and partHeight are derived as

$$partWidth = MbPartWidth(mb_type) \tag{8-148}$$

$$partHeight = MbPartHeight(mb_type) \tag{8-149}$$

– Otherwise, if mb_type is equal to P_8x8 or P_8x8ref0, or mb_type is equal to B_8x8 and sub_mb_type[mbPartIdx] is not equal to B_Direct_8x8, subMbPartIdx proceeds over values 0..NumSubMbPart(sub_mb_type) – 1, and partWidth and partHeight are derived as

$$partWidth = SubMbPartWidth(sub_mb_type[\, mbPartIdx \,]) \tag{8-150}$$

$$partHeight = SubMbPartHeight(sub_mb_type[\, mbPartIdx \,]). \tag{8-151}$$

– Otherwise (mb_type is equal to B_Skip or B_Direct_16x16, or mb_type is equal to B_8x8 and sub_mb_type[mbPartIdx] is equal to B_Direct_8x8), subMbPartIdx proceeds over values 0..3, and partWidth and partHeight are derived as

$$partWidth = 4 \quad (8\text{-}152)$$

$$partHeight = 4 \quad (8\text{-}153)$$

When chroma_format_idc is not equal to 0 (monochrome) the variables partWidthC and partHeightC are derived as

$$partWidthC = partWidth / SubWidthC \quad (8\text{-}154)$$
$$partHeightC = partHeight / SubHeightC \quad (8\text{-}155)$$

Let the variable MvCnt be initially set equal to 0 before any invocation of subclause 8.4.1 for the macroblock.

The Inter prediction process for a macroblock partition mbPartIdx and a sub-macroblock partition subMbPartIdx consists of the following ordered steps

1. Derivation process for motion vector components and reference indices as specified in subclause 8.4.1.

   Inputs to this process are

   – a macroblock partition mbPartIdx,

   – a sub-macroblock partition subMbPartIdx.

   Outputs of this process are

   – luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) the chroma motion vectors mvCL0 and mvCL1

   – reference indices refIdxL0 and refIdxL1

   – prediction list utilization flags predFlagL0 and predFlagL1

   – the sub-macroblock partition motion vector count subMvCnt.

2. The variable MvCnt is incremented by subMvCnt.

3. Decoding process for Inter prediction samples as specified in subclause 8.4.2.

   Inputs to this process are

   – a macroblock partition mbPartIdx,

   – a sub-macroblock partition subMbPartIdx.

   – variables specifying partition width and height for luma and chroma (if available), partWidth, partHeight, partWidthC (if available), and partHeightC (if available)

   – luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) the chroma motion vectors mvCL0 and mvCL1

   – reference indices refIdxL0 and refIdxL1

   – prediction list utilization flags predFlagL0 and predFlagL1

   Outputs of this process are

   – inter prediction samples (pred); which are a (partWidth)x(partHeight) array $predPart_L$ of prediction luma samples and when chroma_format_idc is not equal to 0 (monochrome) two (partWidthC)x(partHeightC) arrays $predPart_{Cr}$, and $predPart_{Cb}$ of prediction chroma samples, one for each of the chroma components Cb and Cr.

For use in derivation processes of variables invoked later in the decoding process, the following assignments are made:

$$MvL0[mbPartIdx][subMbPartIdx] = mvL0 \quad (8\text{-}156)$$

$$MvL1[mbPartIdx][subMbPartIdx] = mvL1 \quad (8\text{-}157)$$

RefIdxL0[mbPartIdx] = refIdxL0 (8-158)

RefIdxL1[mbPartIdx] = refIdxL1 (8-159)

PredFlagL0[mbPartIdx] = predFlagL0 (8-160)

PredFlagL1[mbPartIdx] = predFlagL1 (8-161)

The location of the upper-left sample of the partition relative to the upper-left sample of the macroblock is derived by invoking the inverse macroblock partition scanning process as described in subclause 6.4.2.1 with mbPartIdx as the input and (xP, yP) as the output.

The location of the upper-left sample of the macroblock sub-partition relative to the upper-left sample of the macroblock partition is derived by invoking the inverse sub-macroblock partition scanning process as described in subclause 6.4.2.2 with subMbPartIdx as the input and (xS, yS) as the output.

The macroblock prediction is formed by placing the partition or sub-macroblock partition prediction samples in their correct relative positions in the macroblock, as follows.

The variable $pred_L$[xP + xS + x, yP + yS + y] with x = 0 .. partWidth – 1, y = 0 .. partHeight – 1 is derived by

$pred_L$[xP + xS + x, yP + yS + y] = $predPart_L$[x, y] (8-162)

When chroma_format_idc is not equal to 0 (monochrome) the variable $pred_C$ with x = 0..partWidthC – 1, y = 0..partHeightC – 1, and C in $pred_C$ and $predPart_C$ being replaced by Cb or Cr is derived by

$pred_C$[xP / SubWidthC + xS / SubWidthC + x, yP / SubHeightC + yS / SubHeightC + y] = $predPart_C$[x, y] (8-163)

8.4.1 Derivation process for motion vector components and reference indices

Inputs to this process are

– a macroblock partition mbPartIdx,

– a sub-macroblock partition subMbPartIdx.

Outputs of this process are

– luma motion vectors mvL0 and mvL1 as well as the chroma motion vectors mvCL0 and mvCL1

– reference indices refIdxL0 and refIdxL1

– prediction list utilization flags predFlagL0 and predFlagL1

– a sub-partition macroblock motion vector count variable subMvCnt

For the derivation of the variables mvL0 and mvL1 as well as refIdxL0 and refIdxL1, the following applies.

– If mb_type is equal to P_Skip, the derivation process for luma motion vectors for skipped macroblocks in P and SP slices in subclause 8.4.1.1 is invoked with the output being the luma motion vectors mvL0 and reference indices refIdxL0, and predFlagL0 is set equal to 1. mvL1 and refIdxL1 are marked as not available and predFlagL1 is set equal to 0. The sub-partition motion vector count variable subMvCnt is set equal to 1.

– Otherwise, if mb_type is equal to B_Skip or B_Direct_16x16 or sub_mb_type[mbPartIdx] is equal to B_Direct_8x8, the derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 in B slices in subclause 8.4.1.2 is invoked with mbPartIdx and subMbPartIdx as the input and the output being the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1, the sub-partition motion vector count subMvCnt, and the prediction utilization flags predFlagL0 and predFlagL1.

– Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX, refIdxLX, and in Pred_LX and in the syntax elements ref_idx_lX and mvd_lX, the following applies.

1. The variables refIdxLX and predFlagLX are derived as follows.
   – If MbPartPredMode(mb_type, mbPartIdx) or SubMbPredMode(sub_mb_type[mbPartIdx]) is equal to Pred_LX or to BiPred,

     refIdxLX = ref_idx_lX[mbPartIdx] (8-164)

     predFlagLX = 1 (8-165)

   – Otherwise, the variables refIdxLX and predFlagLX are specified by

     refIdxLX = -1 (8-166)

     predFlagLX = 0 (8-167)

2. The variable subMvCnt for sub-partition motion vector count is set equal to predFlagL0 + predFlagL1.
3. The variable currSubMbType is derived as follows.
   - If the macroblock type is equal to B_8x8, currSubMbType is set equal to sub_mb_type[mbPartIdx].
   - Otherwise (the macroblock type is not equal to B_8x8), currSubMbType is set equal to "na".
4. When predFlagLX is equal to 1, the derivation process for luma motion vector prediction in subclause 8.4.1.3 is invoked with mbPartIdx subMbPartIdx, refIdxLX, and currSubMbType as the inputs and the output being mvpLX. The luma motion vectors are derived by

   mvLX[0] = mvpLX[0] + mvd_lX[mbPartIdx][subMbPartIdx][0] (8-168)

   mvLX[1] = mvpLX[1] + mvd_lX[mbPartIdx][subMbPartIdx][1] (8-169)

For the derivation of the variables for the chroma motion vectors, the following applies. When predFlagLX (with X being either 0 or 1) is equal to 1, the derivation process for chroma motion vectors in subclause 8.4.1.4 is invoked with mvLX and refIdxLX as input and the output being mvCLX.

8.4.1.1 Derivation process for luma motion vectors for skipped macroblocks in P and SP slices

This process is invoked when mb_type is equal to P_Skip.

Outputs of this process are the motion vector mvL0 and the reference index refIdxL0.

The reference index refIdxL0 for a skipped macroblock is derived as follows.

refIdxL0 = 0. (8-170)

For the derivation of the motion vector mvL0 of a P_Skip macroblock type, the following applies.

– The process specified in subclause 8.4.1.3.2 is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType set equal to "na", and listSuffixFlag set equal to 0 as input and the output is assigned to mbAddrA, mbAddrB, mvL0A, mvL0B, refIdxL0A, and refIdxL0B.
– The variable mvL0 is specified as follows.
  – If any of the following conditions are true, both components of the motion vector mvL0 are set equal to 0.
    – mbAddrA is not available
    – mbAddrB is not available
    – refIdxL0A is equal to 0 and both components of mvL0A are equal to 0
    – refIdxL0B is equal to 0 and both components of mvL0B are equal to 0

– Otherwise, the derivation process for luma motion vector prediction as specified in subclause 8.4.1.3 is invoked with mbPartIdx = 0, subMbPartIdx = 0, refIdxL0, and currSubMbType = "na" as inputs and the output is assigned to mvL0.

NOTE – The output is directly assigned to mvL0, since the predictor is equal to the actual motion vector.

8.4.1.2 Derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8

This process is invoked when mb_type is equal to B_Skip or B_Direct_16x16, or sub_mb_type[mbPartIdx] is equal to B_Direct_8x8.

Inputs to this process are mbPartIdx and subMbPartIdx.

Outputs of this process are the reference indices refIdxL0, refIdxL1, the motion vectors mvL0 and mvL1, the sub-partition motion vector count subMvCnt, and the prediction list utilization flags, predFlagL0 and predFlagL1.

The derivation process depends on the value of direct_spatial_mv_pred_flag, which is present in the bitstream in the slice header syntax as specified in subclause 7.3.3, and is specified as follows.

– If direct_spatial_mv_pred_flag is equal to 1, the mode in which the outputs of this process are derived is referred to as spatial direct prediction mode.

– Otherwise (direct_spatial_mv_pred_flag is equal to 0), mode in which the outputs of this process are derived is referred to as temporal direct prediction mode.

Both spatial and temporal direct prediction mode use the co-located motion vectors and reference indices as specified in subclause 8.4.1.2.1.

The motion vectors and reference indices are derived as follows.

– If spatial direct prediction mode is used, the direct motion vector and reference index prediction mode specified in subclause 8.4.1.2.2 is used, with subMvCnt being an output.

– Otherwise (temporal direct prediction mode is used), the direct motion vector and reference index prediction mode specified in subclause 8.4.1.2.3 is used and the variable subMvCnt is derived as follows.

  – If subMbPartIdx is equal to 0, subMvCnt is set equal to 2.

  – Otherwise (subMbPartIdx is not equal to 0), subMvCnt is set equal to 0.

8.4.1.2.1 Derivation process for the co-located 4x4 sub-macroblock partitions

Inputs to this process are mbPartIdx and subMbPartIdx.

Outputs of this process are the picture colPic, the co-located macroblock mbAddrCol, the motion vector mvCol, the reference index refIdxCol, and the variable vertMvScale (which can be One_To_One, Frm_To_Fld or Fld_To_Frm).

When RefPicList1[0] is a frame or a complementary field pair, let firstRefPicL1Top and firstRefPicL1Bottom be the top and bottom fields of RefPicList1[0], respectively, and let the following variables be specified as

topAbsDiffPOC = Abs(DiffPicOrderCnt(firstRefPicL1Top, CurrPic)) (8-171)

bottomAbsDiffPOC = Abs(DiffPicOrderCnt(firstRefPicL1Bottom, CurrPic)) (8-172)

The variable colPic specifies the picture that contains the co-located macroblock as specified in Table 8-6.

Table 8-6 – Specification of the variable colPic

field_pic_flag	RefPicList1[0] is …	mb_field_decoding_flag	additional condition	colPic
1	a field of a decoded frame			the frame containing RefPicList1[0]
	a decoded field			RefPicList1[0]
0	a decoded frame			RefPicList1[0]
	a complementary field pair	0	topAbsDiffPOC < bottomAbsDiffPOC	firstRefPicL1Top
			topAbsDiffPOC >= bottomAbsDiffPOC	firstRefPicL1Bottom
		1	(CurrMbAddr & 1) == 0	firstRefPicL1Top
			(CurrMbAddr & 1) != 0	firstRefPicL1Bottom

When direct_8x8_inference_flag is equal to 1, subMbPartIdx is set as follows.

subMbPartIdx = mbPartIdx (8-173)

Let PicCodingStruct(X) be a function with the argument X being either CurrPic or colPic. It is specified in Table 8-7.

Table 8-7 – Specification of PicCodingStruct(X)

X is coded with field_pic_flag equal to …	mb_adaptive_frame_field_flag	PicCodingStruct(X)
1		FLD
0	0	FRM
0	1	AFRM

With luma4x4BlkIdx = mbPartIdx * 4 + subMbPartIdx, the inverse 4x4 luma block scanning process as specified in subclause 6.4.3 is invoked with luma4x4BlkIdx as the input and (x, y) assigned to (xCol, yCol) as the output.

Table 8-8 specifies the co-located macroblock address mbAddrCol, yM, and the variable vertMvScale in two steps:

1. Specification of a macroblock address mbAddrX depending on PicCodingStruct(CurrPic), and PicCodingStruct(colPic).

   NOTE – It is not possible for CurrPic and colPic picture coding types to be either (FRM, AFRM) or (AFRM, FRM) because these picture coding types must be separated by an IDR picture.

2. Specification of mbAddrCol, yM, and vertMvScale depending on mb_field_decoding_flag and the variable fieldDecodingFlagX, which is derived as follows.
   - If the macroblock mbAddrX in the picture colPic is a field macroblock, fieldDecodingFlagX is set equal to 1
   - Otherwise (the macroblock mbAddrX in the picture colPic is a frame macroblock), fieldDecodingFlagX is set equal to 0.

Unspecified values in Table 8-8 indicate that the value of the corresponding variable is not relevant for the current table row.

mbAddrCol is set equal to CurrMbAddr or to one of the following values.

mbAddrCol1 = 2 * PicWidthInMbs * (CurrMbAddr / PicWidthInMbs) +
(CurrMbAddr % PicWidthInMbs) + PicWidthInMbs * (yCol / 8) (8-174)

mbAddrCol2 = 2 * CurrMbAddr + (yCol / 8) (8-175)

mbAddrCol3 = 2 * CurrMbAddr + bottom_field_flag (8-176)

mbAddrCol4 = PicWidthInMbs * (CurrMbAddr / (2 * PicWidthInMbs)) +
(CurrMbAddr % PicWidthInMbs) (8-177)

mbAddrCol5 = CurrMbAddr / 2 (8-178)

mbAddrCol6 = 2 * (CurrMbAddr / 2) + ((topAbsDiffPOC < bottomAbsDiffPOC) ? 0 : 1) (8-179)

mbAddrCol7 = 2 * (CurrMbAddr / 2) + (yCol / 8) (8-180)

Table 8-8 – Specification of mbAddrCol, yM, and vertMvScale

PicCodingStruct(CurrPic)	PicCodingStruct(colPic)	mbAddrX	mb_field_decoding_flag	fieldDecodingFlagX	mbAddrCol	yM	vertMvScale
FLD	FLD				CurrMbAddr	yCol	One_To_One
	FRM				mbAddrCol1	(2 * yCol) % 16	Frm_To_Fld
	AFRM	2*CurrMbAddr		0	mbAddrCol2	(2 * yCol) % 16	Frm_To_Fld
				1	mbAddrCol3	yCol	One_To_One
FRM	FLD				mbAddrCol4	8 * ((CurrMbAddr / PicWidthInMbs) % 2) + 4 * (yCol / 8)	Fld_To_Frm
	FRM				CurrMbAddr	yCol	One_To_One
AFRM	FLD		0		mbAddrCol5	8 * (CurrMbAddr % 2) +4 * (yCol / 8)	Fld_To_Frm
			1		mbAddrCol5	yCol	One_To_One
	AFRM	CurrMbAddr	0	0	CurrMbAddr	yCol	One_To_One
				1	mbAddrCol6	8 * (CurrMbAddr % 2) + 4 * (yCol / 8)	Fld_To_Frm
		CurrMbAddr	1	0	mbAddrCol7	(2 * yCol) % 16	Frm_To_Fld
				1	CurrMbAddr	yCol	One_To_One

Let mbPartIdxCol be the macroblock partition index of the co-located partition and subMbPartIdxCol the sub-macroblock partition index of the co-located sub-macroblock partition. The partition in the macroblock mbAddrCol inside the picture colPic covering the sample (xCol, yM) is assigned to mbPartIdxCol and the sub-macroblock partition inside the partition mbPartIdxCol covering the sample (xCol, yM) in the macroblock mbAddrCol inside the picture colPic is assigned to subMbPartIdxCol.

The prediction utilization flags predFlagL0Col and predFlagL1Col are set equal to PredFlagL0[mbPartIdxCol] and PredFlagL1[mbPartIdxCol], respectively, which are the prediction utilization flags that have been assigned to the macroblock partition mbAddrCol\mbPartIdxCol inside the picture colPic.

The motion vector mvCol and the reference index refIdxCol are derived as follows.

– If the macroblock mbAddrCol is coded in Intra macroblock prediction mode or both prediction utilization flags, predFlagL0Col and predFlagL1Col are equal to 0, both components of mvCol are set equal to 0 and refIdxCol is set equal to –1.

– Otherwise, the following applies.

  – If predFlagL0Col is equal to 1, the motion vector mvCol and the reference index refIdxCol are set equal to MvL0[mbPartIdxCol][subMbPartIdxCol] and RefIdxL0[mbPartIdxCol], respectively, which are the motion vector mvL0 and the reference index refIdxL0 that have been assigned to the (sub-)macroblock partition mbAddrCol\mbPartIdxCol\subMbPartIdxCol inside the picture colPic.

  – Otherwise (predFlagL0Col is equal to 0 and predFlagL1Col is equal to 1), the motion vector mvCol and the reference index refIdxCol are set equal to MvL1[mbPartIdxCol][subMbPartIdxCol] and RefIdxL1[mbPartIdxCol], respectively, which are the motion vector mvL1 and the reference index refIdxL1 that have been assigned to the (sub-)macroblock partition mbAddrCol\mbPartIdxCol\subMbPartIdxCol inside the picture colPic.

8.4.1.2.2 Derivation process for spatial direct luma motion vector and reference index prediction mode

This process is invoked when direct_spatial_mv_pred_flag is equal to 1 and any of the following conditions is true.

– mb_type is equal to B_Skip

– mb_type is equal to B_Direct_16x16

– sub_mb_type[mbPartIdx] is equal to B_Direct_8x8.

Inputs to this process are mbPartIdx, subMbPartIdx.

Outputs of this process are the reference indices refIdxL0, refIdxL1, the motion vectors mvL0 and mvL1, the sub-partition motion vector count subMvCnt, and the prediction list utilization flags, predFlagL0 and predFlagL1.

The reference indices refIdxL0 and refIdxL1 and the variable directZeroPredictionFlag are derived by applying the following ordered steps.

1. Let the variable currSubMbType be set equal to sub_mb_type[mbPartIdx].

2. The process specified in subclause 8.4.1.3.2 is invoked with mbPartIdx = 0, subMbPartIdx = 0, currSubMbType, and listSuffixFlag = 0 as inputs and the output is assigned to the motion vectors mvL0N and the reference indices refIdxL0N with N being replaced by A, B, or C.

3. The process specified in subclause 8.4.1.3.2 is invoked with mbPartIdx = 0, subMbPartIdx = 0, currSubMbType, and listSuffixFlag = 1 as inputs and the output is assigned to the motion vectors mvL1N and the reference indices refIdxL1N with N being replaced by A, B, or C.

   NOTE 1 – The motion vectors mvL0N, mvL1N and the reference indices refIdxL0N, refIdxL1N are identical for all 4x4 sub-macroblock partitions of a macroblock.

4. The reference indices refIdxL0, refIdxL1, and directZeroPredictionFlag are derived by

$$refIdxL0 = MinPositive(refIdxL0A, MinPositive(refIdxL0B, refIdxL0C)) \quad (8\text{-}181)$$

$$refIdxL1 = MinPositive(refIdxL1A, MinPositive(refIdxL1B, refIdxL1C)) \quad (8\text{-}182)$$

$$directZeroPredictionFlag = 0 \quad (8\text{-}183)$$

where

$$MinPositive(x, y) = \begin{cases} Min(x, y) & \text{if } x >= 0 \text{ and } y >= 0 \\ Max(x, y) & \text{otherwise} \end{cases} \quad (8\text{-}184)$$

5. When both reference indices refIdxL0 and refIdxL1 are less than 0,

refIdxL0 = 0 (8-185)
refIdxL1 = 0 (8-186)
directZeroPredictionFlag = 1 (8-187)

The process specified in subclause 8.4.1.2.1 is invoked with mbPartIdx, subMbPartIdx given as input and the output is assigned to refIdxCol and mvCol.

The variable colZeroFlag is derived as follows.

– If all of the following conditions are true, colZeroFlag is set equal to 1.
  – RefPicList1[0] is currently marked as "used for short-term reference".
  – refIdxCol is equal to 0
  – both motion vector components mvCol[0] and mvCol[1] lie in the range of -1 to 1 in units specified as follows.
    – If the co-located macroblock is a frame macroblock, the units of mvCol[0] and mvCol[1] are units of quarter luma frame samples.
    – Otherwise (the co-located macroblock is a field macroblock), the units of mvCol[0] and mvCol[1] are units of quarter luma field samples.

NOTE 2 – For purposes of determining the condition above, the value mvCol[1] is not scaled to use the units of a motion vector for the current macroblock in cases when the current macroblock is a frame macroblock and the co-located macroblock is a field macroblock or when the current macroblock is a field macroblock and the co-located macroblock is a frame macroblock. This aspect differs from the use of mvCol[1] in the temporal direct mode as specified in subclause 8.4.1.2.3, which applies scaling to the motion vector of the co-located macroblock to use the same units as the units of a motion vector for the current macroblock, using Equation 8-190 or Equation 8-191 in these cases.

– Otherwise, colZeroFlag is set equal to 0.

The motion vectors mvLX (with X being 0 or 1) are derived as follows.

– If any of the following conditions is true, both components of the motion vector mvLX are set equal to 0.
  – directZeroPredictionFlag is equal to 1
  – refIdxLX is less than 0
  – refIdxLX is equal to 0 and colZeroFlag is equal to 1
– Otherwise, the process specified in subclause 8.4.1.3 is invoked with mbPartIdx = 0, subMbPartIdx = 0, refIdxLX, and currSubMbType as inputs and the output is assigned to mvLX.

NOTE 3 – The motion vector mvLX returned from subclause 8.4.1.3 is identical for all 4x4 sub-macroblock partitions of a macroblock for which the process is invoked.

The prediction utilization flags predFlagL0 and predFlagL1 are derived as specified using Table 8-9.

Table 8-9 – Assignment of prediction utilization flags

refIdxL0	refIdxL1	predFlagL0	predFlagL1
>= 0	>= 0	1	1
>= 0	< 0	1	0
< 0	>= 0	0	1

The variable subMvCnt is derived as follows.

– If subMbPartIdx is not equal to 0 or direct_8x8_inference_flag is equal to 0, subMvCnt is set equal to 0.
– Otherwise (subMbPartIdx is equal to 0 and direct_8x8_inference_flag is equal to 1), subMvCnt is set equal to predFlagL0 + predFLagL1.

8.4.1.2.3 Derivation process for temporal direct luma motion vector and reference index prediction mode

This process is invoked when direct_spatial_mv_pred_flag is equal to 0 and any of the following conditions is true.

- mb_type is equal to B_Skip
- mb_type is equal to B_Direct_16x16
- sub_mb_type[mbPartIdx] is equal to B_Direct_8x8.

Inputs to this process are mbPartIdx and subMbPartIdx.

Outputs of this process are the motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, and the prediction list utilization flags, predFlagL0 and predFlagL1.

The process specified in subclause 8.4.1.2.1 is invoked with mbPartIdx, subMbPartIdx given as input and the output is assigned to colPic, mbAddrCol, mvCol, refIdxCol, and vertMvScale.

The reference indices refIdxL0 and refIdxL1 are derived as follows.

$$refIdxL0 = ((refIdxCol < 0) ? 0 : MapColToList0(refIdxCol)) \quad (8\text{-}188)$$

$$refIdxL1 = 0 \quad (8\text{-}189)$$

NOTE 1 – If the current macroblock is a field macroblock, refIdxL0 and refIdxL1 index a list of fields; otherwise (the current macroblock is a frame macroblock), refIdxL0 and refIdxL1 index a list of frames or complementary reference field pairs.

Let refPicCol be a frame, a field, or a complementary field pair that was referred by the reference index refIdxCol when decoding the co-located macroblock mbAddrCol inside the picture colPic. The function MapColToList0(refIdxCol) is specified as follows.

- If vertMvScale is equal to One_To_One, the following applies.
  - If field_pic_flag is equal to 0 and the current macroblock is a field macroblock, the following applies.
    - Let refIdxL0Frm be the lowest valued reference index in the current reference picture list RefPicList0 that references the frame or complementary field pair that contains the field refPicCol. RefPicList0 shall contain a frame or complementary field pair that contains the field refPicCol. The return value of MapColToList0( ) is specified as follows.
      - If the field referred to by refIdxCol has the same parity as the current macroblock, MapColToList0(refIdxCol) returns the reference index (refIdxL0Frm << 1).
      - Otherwise (the field referred by refIdxCol has the opposite parity of the current macroblock), MapColToList0(refIdxCol) returns the reference index ((refIdxL0Frm << 1) + 1).
  - Otherwise (field_pic_flag is equal to 1 or the current macroblock is a frame macroblock), MapColToList0(refIdxCol) returns the lowest valued reference index refIdxL0 in the current reference picture list RefPicList0 that references refPicCol. RefPicList0 shall contain refPicCol.
- Otherwise, if vertMvScale is equal to Frm_To_Fld, the following applies.
  - If field_pic_flag is equal to 0, let refIdxL0Frm be the lowest valued reference index in the current reference picture list RefPicList0 that references refPicCol. MapColToList0(refIdxCol) returns the reference index (refIdxL0Frm << 1). RefPicList0 shall contain refPicCol.
  - Otherwise (field_pic_flag is equal to 1), MapColToList0(refIdxCol) returns the lowest valued reference index refIdxL0 in the current reference picture list RefPicList0 that references the field of refPicCol with the same parity as the current picture CurrPic. RefPicList0 shall contain the field of refPicCol with the same parity as the current picture CurrPic.
- Otherwise (vertMvScale is equal to Fld_To_Frm), MapColToList0(refIdxCol) returns the lowest valued reference index refIdxL0 in the current reference picture list RefPicList0 that references the frame or complementary field pair that contains refPicCol. RefPicList0 shall contain a frame or complementary field pair that contains the field refPicCol.

NOTE 2 – A decoded reference picture that was marked as "used for short-term reference" when it was referenced in the decoding process of the picture containing the co-located macroblock may have been modified to be marked as "used for long-term reference" before being used for reference for inter prediction using the direct prediction mode for the current macroblock.

Depending on the value of vertMvScale the vertical component of mvCol is modified as follows.

– If vertMvScale is equal to Frm_To_Fld

$$mvCol[1] = mvCol[1] / 2 \quad (8\text{-}190)$$

– Otherwise, if vertMvScale is equal to Fld_To_Frm

$$mvCol[1] = mvCol[1] * 2 \quad (8\text{-}191)$$

– Otherwise (vertMvScale is equal to One_To_One), mvCol[1] remains unchanged.

The variables currPicOrField, pic0, and pic1, are derived as follows.

– If field_pic_flag is equal to 0 and the current macroblock is a field macroblock, the following applies.
  – currPicOrField is the field of the current picture CurrPic that has the same parity as the current macroblock.
  – pic1 is the field of RefPicList1[0] that has the same parity as the current macroblock.
  – The variable pic0 is derived as follows.
    – If refIdxL0 % 2 is equal to 0, pic0 is the field of RefPicList0[refIdxL0 / 2] that has the same parity as the current macroblock.
    – Otherwise (refIdxL0 % 2 is not equal to 0), pic0 is the field of RefPicList0[refIdxL0 / 2] that has the opposite parity of the current macroblock.
– Otherwise (field_pic_flag is equal to 1 or the current macroblock is a frame macroblock), currPicOrField is the current picture CurrPic, pic1 is the decoded reference picture RefPicList1[0], and pic0 is the decoded reference picture RefPicList0[refIdxL0].

The two motion vectors mvL0 and mvL1 for each 4x4 sub-macroblock partition of the current macroblock are derived as follows:

NOTE 3 – It is often the case that many of the 4x4 sub-macroblock partitions share the same motion vectors and reference pictures. In these cases, temporal direct mode motion compensation can calculate the inter prediction sample values in larger units than 4x4 luma sample blocks. For example, when direct_8x8_inference_flag is equal to 1, at least each 8x8 luma sample quadrant of the macroblock shares the same motion vectors and reference pictures.

– If the reference index refIdxL0 refers to a long-term reference picture, or DiffPicOrderCnt(pic1, pic0) is equal to 0, the motion vectors mvL0, mvL1 for the direct mode partition are derived by

$$mvL0 = mvCol \quad (8\text{-}192)$$

$$mvL1 = 0 \quad (8\text{-}193)$$

– Otherwise, the motion vectors mvL0, mvL1 are derived as scaled versions of the motion vector mvCol of the co-located sub-macroblock partition as specified below (see Figure 8-2)

$$tx = (16\,384 + Abs(td / 2)) / td \quad (8\text{-}194)$$

$$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) \quad (8\text{-}195)$$

$$mvL0 = (DistScaleFactor * mvCol + 128) >> 8 \quad (8\text{-}196)$$

$$mvL1 = mvL0 - mvCol \quad (8\text{-}197)$$

where tb and td are derived as follows.

$$tb = Clip3(-128, 127, DiffPicOrderCnt(currPicOrField, pic0)) \quad (8\text{-}198)$$

$$td = Clip3(-128, 127, DiffPicOrderCnt(pic1, pic0)) \quad (8\text{-}199)$$

NOTE 4 – mvL0 and mvL1 cannot exceed the ranges specified in Annex A.

The prediction utilization flags predFlagL0 and predFlagL1 are both set equal to 1.

Figure 8-2 illustrates the temporal direct-mode motion vector inference when the current picture is temporally between the reference picture from reference picture list 0 and the reference picture from reference picture list 1.

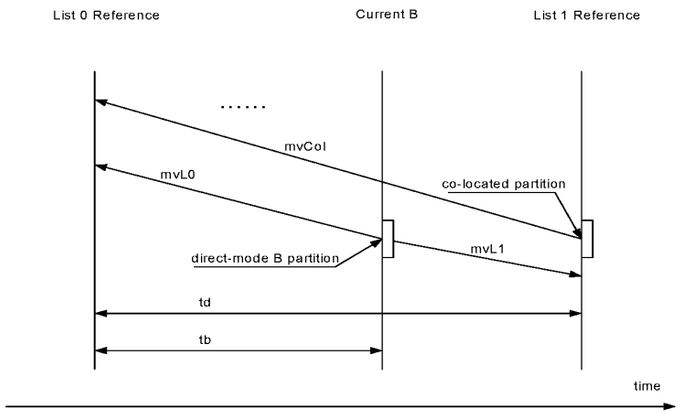


Figure 8-2 – Example for temporal direct-mode motion vector inference (informative)

8.4.1.3 Derivation process for luma motion vector prediction

Inputs to this process are

– the macroblock partition index mbPartIdx,

– the sub-macroblock partition index subMbPartIdx,

– the reference index of the current partition refIdxLX (with X being 0 or 1),

– the variable currSubMbType.

Output of this process is the prediction mvpLX of the motion vector mvLX (with X being 0 or 1).

The derivation process for the neighbouring blocks for motion data in subclause 8.4.1.3.2 is invoked with mbPartIdx, subMbPartIdx, currSubMbType, and listSuffixFlag = X (with X being 0 or 1 for refIdxLX being refIdxL0 or refIdxL1, respectively) as the input and with mbAddrN\mbPartIdxN\subMbPartIdxN, reference indices refIdxLXN and the motion vectors mvLXN with N being replaced by A, B, or C as the output.

The derivation process for median luma motion vector prediction in subclause 8.4.1.3.1 is invoked with mbAddrN\mbPartIdxN\subMbPartIdxN, mvLXN, refIdxLXN with N being replaced by A, B, or C and refIdxLX as the input and mvpLX as the output, unless one of the following is true.

– MbPartWidth(mb_type) is equal to 16, MbPartHeight(mb_type) is equal to 8, mbPartIdx is equal to 0, and refIdxLXB is equal to refIdxLX,

$$mvpLX = mvLXB \tag{8-200}$$

– MbPartWidth(mb_type) is equal to 16, MbPartHeight(mb_type) is equal to 8, mbPartIdx is equal to 1, and refIdxLXA is equal to refIdxLX,

$$mvpLX = mvLXA \tag{8-201}$$

– MbPartWidth(mb_type) is equal to 8, MbPartHeight(mb_type) is equal to 16, mbPartIdx is equal to 0, and refIdxLXA is equal to refIdxLX,

$$mvpLX = mvLXA \tag{8-202}$$

– MbPartWidth(mb_type) is equal to 8, MbPartHeight(mb_type) is equal to 16, mbPartIdx is equal to 1, and refIdxLXC is equal to refIdxLX,

mvpLX = mvLXC (8-203)

Figure 8-3 illustrates the non-median prediction as described above.

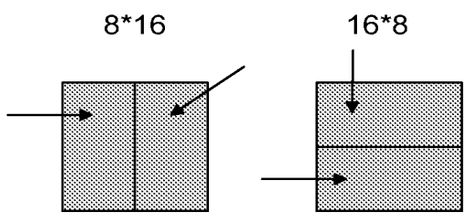


Figure 8-3 – Directional segmentation prediction (informative)

8.4.1.3.1 Derivation process for median luma motion vector prediction

Inputs to this process are

– the neighbouring partitions mbAddrN\mbPartIdxN\subMbPartIdxN (with N being replaced by A, B, or C),

– the motion vectors mvLXN (with N being replaced by A, B, or C) of the neighbouring partitions,

– the reference indices refIdxLXN (with N being replaced by A, B, or C) of the neighbouring partitions, and

– the reference index refIdxLX of the current partition.

Output of this process is the motion vector prediction mvpLX.

The variable mvpLX is derived as follows:

– When both partitions mbAddrB\mbPartIdxB\subMbPartIdxB and mbAddrC\mbPartIdxC\subMbPartIdxC are not available and mbAddrA\mbPartIdxA\subMbPartIdxA is available,

mvLXB = mvLXA (8-204)

mvLXC = mvLXA (8-205)

refIdxLXB = refIdxLXA (8-206)

refIdxLXC = refIdxLXA (8-207)

– Depending on reference indices refIdxLXA, refIdxLXB, or refIdxLXC, the following applies.

  – If one and only one of the reference indices refIdxLXA, refIdxLXB, or refIdxLXC is equal to the reference index refIdxLX of the current partition, the following applies. Let refIdxLXN be the reference index that is equal to refIdxLX, the motion vector mvLXN is assigned to the motion vector prediction mvpLX:

  mvpLX = mvLXN (8-208)

  – Otherwise, each component of the motion vector prediction mvpLX is given by the median of the corresponding vector components of the motion vector mvLXA, mvLXB, and mvLXC:

  mvpLX[0] = Median(mvLXA[0], mvLXB[0], mvLXC[0]) (8-209)

mvpLX[1] = Median(mvLXA[1], mvLXB[1], mvLXC[1]) (8-210)

8.4.1.3.2 Derivation process for motion data of neighbouring partitions

Inputs to this process are

– the macroblock partition index mbPartIdx,

– the sub-macroblock partition index subMbPartIdx,

– the current sub-macroblock type currSubMbType,

– the list suffix flag listSuffixFlag

Outputs of this process are (with N being replaced by A, B, or C)

– mbAddrN\mbPartIdxN\subMbPartIdxN specifying neighbouring partitions,

– the motion vectors mvLXN of the neighbouring partitions, and

– the reference indices refIdxLXN of the neighbouring partitions.

Variable names that include the string "LX" are interpreted with the X being equal to listSuffixFlag.

The partitions mbAddrN\mbPartIdxN\subMbPartIdxN with N being either A, B, or C are derived in the following ordered steps.

1. Let mbAddrD\mbPartIdxD\subMbPartIdxD be variables specifying an additional neighbouring partition.
2. The process in subclause 6.4.8.5 is invoked with mbPartIdx, currSubMbType, and subMbPartIdx as input and the output is assigned to mbAddrN\mbPartIdxN\subMbPartIdxN with N being replaced by A, B, C, or D.
3. When the partition mbAddrC\mbPartIdxC\subMbPartIdxC is not available, the following applies

   mbAddrC = mbAddrD (8-211)

   mbPartIdxC = mbPartIdxD (8-212)

   subMbPartIdxC = subMbPartIdxD (8-213)

The motion vectors mvLXN and reference indices refIdxLXN (with N being A, B, or C) are derived as follows.

– If the macroblock partition or sub-macroblock partition mbAddrN\mbPartIdxN\subMbPartIdxN is not available or mbAddrN is coded in Intra prediction mode or predFlagLX of mbAddrN\mbPartIdxN\subMbPartIdxN is equal to 0, both components of mvLXN are set equal to 0 and refIdxLXN is set equal to –1.

– Otherwise, the following applies.

  – The motion vector mvLXN and reference index refIdxLXN are set equal to MvLX[mbPartIdxN][subMbPartIdxN] and RefIdxLX[mbPartIdxN], respectively, which are the motion vector mvLX and reference index refIdxLX that have been assigned to the (sub-)macroblock partition mbAddrN\mbPartIdxN\subMbPartIdxN.

  – The variables mvLXN[1] and refIdxLXN are further processed as follows.

    – If the current macroblock is a field macroblock and the macroblock mbAddrN is a frame macroblock

      mvLXN[1] = mvLXN[1] / 2 (8-214)

      refIdxLXN = refIdxLXN * 2 (8-215)

    – Otherwise, if the current macroblock is a frame macroblock and the macroblock mbAddrN is a field macroblock

      mvLXN[1] = mvLXN[1] * 2 (8-216)

refIdxLXN = refIdxLXN / 2 (8-217)

– Otherwise, the vertical motion vector component mvLXN[1] and the reference index refIdxLXN remain unchanged.

8.4.1.4 Derivation process for chroma motion vectors

This process is only invoked when chroma_format_idc is not equal to 0 (monochrome).

Inputs to this process are a luma motion vector mvLX and a reference index refIdxLX.

Output of this process is a chroma motion vector mvCLX.

A chroma motion vector is derived from the corresponding luma motion vector.

The precision of the chroma motion vector components is 1 ÷ (4 * SubWidthC) horizontally and 1 ÷ (4 * SubHeightC) vertically.

NOTE – For example, when using the 4:2:0 chroma format, since the units of luma motion vectors are one-quarter luma sample units and chroma has half horizontal and vertical resolution compared to luma, the units of chroma motion vectors are one-eighth chroma sample units, i.e., a value of 1 for the chroma motion vector refers to a one-eighth chroma sample displacement. For example, when the luma vector applies to 8x16 luma samples, the corresponding chroma vector in 4:2:0 chroma format applies to 4x8 chroma samples and when the luma vector applies to 4x4 luma samples, the corresponding chroma vector in 4:2:0 chroma format applies to 2x2 chroma samples.

For the derivation of the motion vector mvCLX, the following applies.

– If chroma_format_idc is not equal to 1 or the current macroblock is a frame macroblock, the horizontal and vertical components of the chroma motion vector mvCLX are derived as

mvCLX[0] = mvLX[0] (8-218)
mvCLX[1] = mvLX[1] (8-219)

– Otherwise (chroma_format_idc is equal to 1 and the current macroblock is a field macroblock), only the horizontal component of the chroma motion vector mvCLX[0] is derived using Equation 8-218. The vertical component of the chroma motion vector mvCLX[1] is dependent on the parity of the current field or the current macroblock and the reference picture, which is referred by the reference index refIdxLX. mvCLX[1] is derived from mvLX[1] according to Table 8-10.

Table 8-10 – Derivation of the vertical component of the chroma vector in field coding mode

Parity conditions		**mvCLX[1]**
Reference picture (refIdxLX)	Current field (picture/macroblock)	
Top field	Bottom field	mvLX[1] + 2
Bottom field	Top field	mvLX[1] – 2
Otherwise		mvLX[1]

8.4.2 Decoding process for Inter prediction samples

Inputs to this process are

– a macroblock partition mbPartIdx,

– a sub-macroblock partition subMbPartIdx.

– variables specifying partition width and height for luma and chroma (if available), partWidth, partHeight, partWidthC (if available) and partHeightC (if available)

– luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) chroma motion vectors mvCL0 and mvCL1

– reference indices refIdxL0 and refIdxL1

– prediction list utilization flags, predFlagL0 and predFlagL1

Outputs of this process are

– the Inter prediction samples predPart, which are a (partWidth)x(partHeight) array $predPart_L$ of prediction luma samples, and when chroma_format_idc is not equal to 0 (monochrome) two (partWidthC)x(partHeightC) arrays $predPart_{Cb}$, $predPart_{Cr}$ of prediction chroma samples, one for each of the chroma components Cb and Cr.

Let $predPartL0_L$ and $predPartL1_L$ be (partWidth)x(partHeight) arrays of predicted luma sample values and when chroma_format_idc is not equal to 0 (monochrome) $predPartL0_{Cb}$, $predPartL1_{Cb}$, $predPartL0_{Cr}$, and $predPartL1_{Cr}$ be (partWidthC)x(partHeightC) arrays of predicted chroma sample values.

For LX being replaced by either L0 or L1 in the variables predFlagLX, RefPicListX, refIdxLX, refPicLX, predPartLX, the following is specified.

When predFlagLX is equal to 1, the following applies.

– The reference picture consisting of an ordered two-dimensional array $refPicLX_L$ of luma samples and when chroma_format_idc is not equal to 0 (monochrome) two ordered two-dimensional arrays $refPicLX_{Cb}$ and $refPicLX_{Cr}$ of chroma samples is derived by invoking the process specified in subclause 8.4.2.1 with refIdxLX and RefPicListX given as input.

– The array $predPartLX_L$ and when chroma_format_idc is not equal to 0 (monochrome) the arrays $predPartLX_{Cb}$ and $predPartLX_{Cr}$ are derived by invoking the process specified in subclause 8.4.2.2 with the current partition specified by mbPartIdx\subMbPartIdx, the motion vectors mvLX, mvCLX (if available), and the reference arrays with $refPicLX_L$, $refPicLX_{Cb}$ (if available), and $refPicLX_{Cr}$ (if available) given as input.

For C being replaced by L, Cb (if available), or Cr (if available), the array $predPart_C$ of the prediction samples of component C is derived by invoking the process specified in subclause 8.4.2.3 with the current partition specified by mbPartIdx and subMbPartIdx and the array $predPartL0_C$ and $predPartL1_C$ as well as predFlagL0 and predFlagL1 given as input.

8.4.2.1 Reference picture selection process

Input to this process is a reference index refIdxLX.

Output of this process is a reference picture consisting of a two-dimensional array of luma samples $refPicLX_L$ and two two-dimensional arrays of chroma samples $refPicLX_{Cb}$ and $refPicLX_{Cr}$.

Depending on field_pic_flag, the reference picture list RefPicListX (which has been derived as specified in subclause 8.2.4) consists of the following.

– If field_pic_flag is equal to 1, each entry of RefPicListX is a reference field or a field of a reference frame.

– Otherwise (field_pic_flag is equal to 0), each entry of RefPicListX is a reference frame or a complementary reference field pair.

For the derivation of the reference picture, the following applies.

– If field_pic_flag is equal to 1, the reference field or field of a reference frame RefPicListX[refIdxLX] is the output. The output reference field or field of a reference frame consists of a ($PicWidthInSamples_L$)x($PicHeightInSamples_L$) array of luma samples $refPicLX_L$ and, when chroma_format_idc is not equal to 0 (monochrome), two ($PicWidthInSamples_C$)x($PicHeightInSamples_C$) arrays of chroma samples $refPicLX_{Cb}$ and $refPicLX_{Cr}$.

– Otherwise (field_pic_flag is equal to 0), the following applies.

  – If the current macroblock is a frame macroblock, the reference frame or complementary reference field pair RefPicListX[refIdxLX] is the output. The output reference frame or complementary reference field pair consists of a ($PicWidthInSamples_L$)x($PicHeightInSamples_L$) array of luma samples $refPicLX_L$ and, when chroma_format_idc is not equal to 0 (monochrome), two ($PicWidthInSamples_C$)x($PicHeightInSamples_C$) arrays of chroma samples $refPicLX_{Cb}$ and $refPicLX_{Cr}$.

  – Otherwise (the current macroblock is a field macroblock), the following applies.

    – Let refFrame be the reference frame or complementary reference field pair RefPicListX[refIdxLX / 2].

    – The field of refFrame is selected as follows.

      – If refIdxLX % 2 is equal to 0, the field of refFrame that has the same parity as the current macroblock is the output.

– Otherwise (refIdxLX % 2 is equal to 1), the field of refFrame that has the opposite parity as the current macroblock is the output.

– The output reference field or field of a reference frame consists of a (PicWidthInSamples$_L$)x(PicHeightInSamples$_L$ / 2) array of luma samples refPicLX$_L$ and, when chroma_format_idc is not equal to 0 (monochrome), two (PicWidthInSamples$_C$)x(PicHeightInSamples$_C$ / 2) arrays of chroma samples refPicLX$_{Cb}$ and refPicLX$_{Cr}$.

The reference picture sample arrays refPicLX$_L$, refPicLX$_{Cb}$ (if available), and refPicLX$_{Cr}$ (if available) correspond to decoded sample arrays S_L, S_{Cb} (if available), S_{Cr} (if available) derived in subclause 8.7 for a previously-decoded reference field or reference frame or complementary reference field pair or field of a reference frame.

8.4.2.2 Fractional sample interpolation process

Inputs to this process are

– the current partition given by its partition index mbPartIdx and its sub-macroblock partition index subMbPartIdx,

– the width and height partWidth, partHeight of this partition in luma-sample units,

– a luma motion vector mvLX given in quarter-luma-sample units,

– a chroma motion vector mvCLX given in eighth-chroma-sample units, and

– the selected reference picture sample arrays refPicLX$_L$, refPicLX$_{Cb}$, and refPicLX$_{Cb}$

Outputs of this process are

– a (partWidth)x(partHeight) array predPartLX$_L$ of prediction luma sample values and

– when chroma_format_idc is not equal to 0 (monochrome) two (partWidthC)x(partHeightC) arrays predPartLX$_{Cb}$, and predPartLX$_{Cr}$ of prediction chroma sample values.

Let (xA$_L$, yA$_L$) be the location given in full-sample units of the upper-left luma sample of the current partition given by mbPartIdx\subMbPartIdx relative to the upper-left luma sample location of the given two-dimensional array of luma samples.

Let (xInt$_L$, yInt$_L$) be a luma location given in full-sample units and (xFrac$_L$, yFrac$_L$) be an offset given in quarter-sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays refPicLX$_L$, refPicLX$_{Cb}$ (if available), and refPicLX$_{Cr}$ (if available).

For each luma sample location (0 <= x_L < partWidth, 0 <= y_L < partHeight) inside the prediction luma sample array predPartLX$_L$, the corresponding prediction luma sample value predPartLX$_L$[x_L, y_L] is derived as follows:

– The variables xInt$_L$, yInt$_L$, xFrac$_L$, and yFrac$_L$ are derived by

$$xInt_L = xA_L + (mvLX[0] >> 2) + x_L \quad (8\text{-}220)$$
$$yInt_L = yA_L + (mvLX[1] >> 2) + y_L \quad (8\text{-}221)$$

$$xFrac_L = mvLX[0] \& 3 \quad (8\text{-}222)$$
$$yFrac_L = mvLX[1] \& 3 \quad (8\text{-}223)$$

– The prediction luma sample value predPartLX$_L$[x_L, y_L] is derived by invoking the process specified in subclause 8.4.2.2.1 with (xInt$_L$, yInt$_L$), (xFrac$_L$, yFrac$_L$) and refPicLX$_L$ given as input.

When chroma_format_idc is not equal to 0 (monochrome), the following applies.

Let (xInt$_C$, yInt$_C$) be a chroma location given in full-sample units and (xFrac$_C$, yFrac$_C$) be an offset given in one-eighth sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays refPicLX$_{Cb}$, and refPicLX$_{Cr}$.

For each chroma sample location (0 <= x_C < partWidthC, 0 <= y_C < partHeightC) inside the prediction chroma sample arrays predPartLX$_{Cb}$ and predPartLX$_{Cr}$, the corresponding prediction chroma sample values predPartLX$_{Cb}$[x_C, y_C] and predPartLX$_{Cr}$[x_C, y_C] are derived as follows:

– Depending on chroma_format_idc, the variables xInt$_C$, yInt$_C$, xFrac$_C$, and yFrac$_C$ are derived as follows.

  – If chroma_format_idc is equal to 1,

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] >> 3) + x_C \quad (8\text{-}224)$$
$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] >> 3) + y_C \quad (8\text{-}225)$$

$$xFrac_C = mvCLX[0] \& 7 \quad (8\text{-}226)$$
$$yFrac_C = mvCLX[1] \& 7 \quad (8\text{-}227)$$

– Otherwise, if chroma_format_idc is equal to 2,

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] >> 3) + x_C \quad (8\text{-}228)$$
$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] >> 2) + y_C \quad (8\text{-}229)$$

$$xFrac_C = mvCLX[0] \& 7 \quad (8\text{-}230)$$
$$yFrac_C = (mvCLX[1] \& 3) << 1 \quad (8\text{-}231)$$

– Otherwise (chroma_format_idc is equal to 3),

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] >> 2) + x_C \quad (8\text{-}232)$$
$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] >> 2) + y_C \quad (8\text{-}233)$$

$$xFrac_C = (mvCLX[0] \& 3) << 1 \quad (8\text{-}234)$$
$$yFrac_C = (mvCLX[1] \& 3) << 1 \quad (8\text{-}235)$$

– The prediction sample value $predPartLX_{Cb}[x_C, y_C]$ is derived by invoking the process specified in subclause 8.4.2.2.2 with ($xInt_C$, $yInt_C$), ($xFrac_C$, $yFrac_C$) and $refPicLX_{Cb}$ given as input.

– The prediction sample value $predPartLX_{Cr}[x_C, y_C]$ is derived by invoking the process specified in subclause 8.4.2.2.2 with ($xInt_C$, $yInt_C$), ($xFrac_C$, $yFrac_C$) and $refPicLX_{Cr}$ given as input.

8.4.2.2.1 Luma sample interpolation process

Inputs to this process are

– a luma location in full-sample units ($xInt_L$, $yInt_L$),

– a luma location offset in fractional-sample units ($xFrac_L$, $yFrac_L$), and

– the luma sample array of the selected reference picture $refPicLX_L$

Output of this process is a predicted luma sample value $predPartLX_L[x_L, y_L]$.

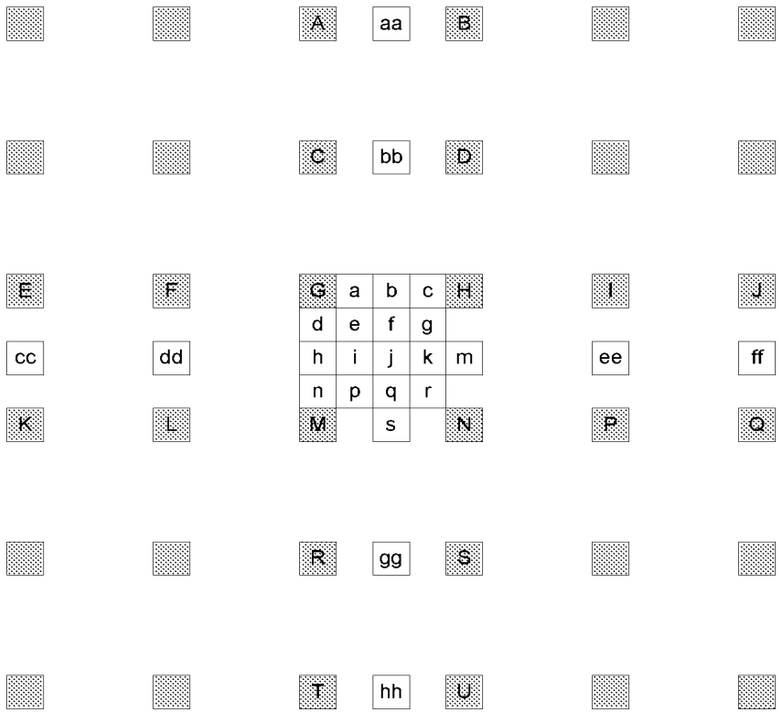


Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation

The variable $refPicHeightEffective_L$, which is the height of the effective reference picture luma array, is derived as follows.

– If MbaffFrameFlag is equal to 0 or mb_field_decoding_flag is equal to 0, $refPicHeightEffective_L$ is set equal to $PicHeightInSamples_L$.

– Otherwise (MbaffFrameFlag is equal to 1 and mb_field_decoding_flag is equal to 1), $refPicHeightEffective_L$ is set equal to $PicHeightInSamples_L$ / 2.

In Figure 8-4, the positions labelled with upper-case letters within shaded blocks represent luma samples at full-sample locations inside the given two-dimensional array $refPicLX_L$ of luma samples. These samples may be used for generating the predicted luma sample value $predPartLX_L[x_L, y_L]$. The locations (xZ_L, yZ_L) for each of the corresponding luma samples Z, where Z may be A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, S, T, or U, inside the given array $refPicLX_L$ of luma samples are derived as follows:

$$xZ_L = Clip3(0, PicWidthInSamples_L - 1, xInt_L + xDZ_L)$$
$$yZ_L = Clip3(0, refPicHeightEffective_L - 1, yInt_L + yDZ_L) \quad (8\text{-}236)$$

Table 8-11 specifies (xDZ_L, yDZ_L) for different replacements of Z.

Table 8-11 – Differential full-sample luma locations

Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T	U
xDZ_L	0	1	0	1	-2	-1	0	1	2	3	-2	-1	0	1	2	3	0	1	0	1
yDZ_L	-2	-2	-1	-1	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3

Given the luma samples 'A' to 'U' at full-sample locations (xA_L, yA_L) to (xU_L, yU_L), the luma samples 'a' to 's' at fractional sample positions are derived by the following rules. The luma prediction values at half sample positions are derived by applying a 6-tap filter with tap values (1, -5, 20, 20, -5, 1). The luma prediction values at quarter sample positions are derived by averaging samples at full and half sample positions. The process for each fractional position is described below.

– The samples at half sample positions labelled b are derived by first calculating intermediate values denoted as b_1 by applying the 6-tap filter to the nearest integer position samples in the horizontal direction. The samples at half sample positions labelled h are derived by first calculating intermediate values denoted as h_1 by applying the 6-tap filter to the nearest integer position samples in the vertical direction:

$$b_1 = (E - 5 * F + 20 * G + 20 * H - 5 * I + J) \quad (8\text{-}237)$$
$$h_1 = (A - 5 * C + 20 * G + 20 * M - 5 * R + T) \quad (8\text{-}238)$$

The final prediction values b and h are derived using:

$$b = Clip1_Y((b_1 + 16) >> 5) \quad (8\text{-}239)$$
$$h = Clip1_Y((h_1 + 16) >> 5) \quad (8\text{-}240)$$

– The samples at half sample position labelled as j are derived by first calculating intermediate value denoted as j_1 by applying the 6-tap filter to the intermediate values of the closest half sample positions in either the horizontal or vertical direction because these yield an equal result.

$$j_1 = cc - 5 * dd + 20 * h_1 + 20 * m_1 - 5 * ee + ff, \text{ or} \quad (8\text{-}241)$$
$$j_1 = aa - 5 * bb + 20 * b_1 + 20 * s_1 - 5 * gg + hh \quad (8\text{-}242)$$

where intermediate values denoted as aa, bb, gg, s_1 and hh are derived by applying the 6-tap filter horizontally in the same manner as the derivation of b_1 and intermediate values denoted as cc, dd, ee, m_1 and ff are derived by applying the 6-tap filter vertically in the same manner as the derivation of h_1. The final prediction value j are derived using:

$$j = Clip1_Y((j_1 + 512) >> 10) \quad (8\text{-}243)$$

– The final prediction values s and m are derived from s_1 and m_1 in the same manner as the derivation of b and h, as given by:

$$s = Clip1_Y((s_1 + 16) >> 5) \quad (8\text{-}244)$$
$$m = Clip1_Y((m_1 + 16) >> 5) \quad (8\text{-}245)$$

– The samples at quarter sample positions labelled as a, c, d, n, f, i, k, and q are derived by averaging with upward rounding of the two nearest samples at integer and half sample positions using:

$$a = (G + b + 1) >> 1 \quad (8\text{-}246)$$
$$c = (H + b + 1) >> 1 \quad (8\text{-}247)$$
$$d = (G + h + 1) >> 1 \quad (8\text{-}248)$$
$$n = (M + h + 1) >> 1 \quad (8\text{-}249)$$
$$f = (b + j + 1) >> 1 \quad (8\text{-}250)$$
$$i = (h + j + 1) >> 1 \quad (8\text{-}251)$$
$$k = (j + m + 1) >> 1 \quad (8\text{-}252)$$
$$q = (j + s + 1) >> 1. \quad (8\text{-}253)$$

– The samples at quarter sample positions labelled as e, g, p, and r are derived by averaging with upward rounding of the two nearest samples at half sample positions in the diagonal direction using

$$e = (b + h + 1) >> 1 \quad (8\text{-}254)$$
$$g = (b + m + 1) >> 1 \quad (8\text{-}255)$$
$$p = (h + s + 1) >> 1 \quad (8\text{-}256)$$
$$r = (m + s + 1) >> 1. \quad (8\text{-}257)$$

The luma location offset in fractional-sample units ($xFrac_L$, $yFrac_L$) specifies which of the generated luma samples at full-sample and fractional-sample locations is assigned to the predicted luma sample value $predPartLX_L[x_L, y_L]$. This assignment is done according to Table 8-12. The value of $predPartLX_L[x_L, y_L]$ is the output.

Table 8-12 – Assignment of the luma prediction sample predPartLX$_L$[x_L, y_L]

$xFrac_L$	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
$yFrac_L$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predPartLX$_L$[x_L, y_L]	G	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

8.4.2.2.2 Chroma sample interpolation process

This process shall only be invoked when chroma_format_idc is not equal to 0 (monochrome).

Inputs to this process are

– a chroma location in full-sample units ($xInt_C$, $yInt_C$),

– a chroma location offset in fractional-sample units ($xFrac_C$, $yFrac_C$), and

– chroma component samples from the selected reference picture refPicLX$_C$.

Output of this process is a predicted chroma sample value predPartLX$_C$[x_C, y_C].

In Figure 8-5, the positions labelled with A, B, C, and D represent chroma samples at full-sample locations inside the given two-dimensional array refPicLX$_C$ of chroma samples.

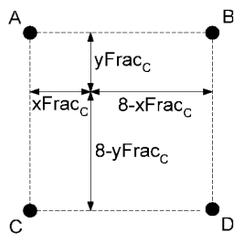


Figure 8-5 – Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D

The variable refPicHeightEffective$_C$, which is the height of the effective reference picture chroma array, is derived as follows.

– If MbaffFrameFlag is equal to 0 or mb_field_decoding_flag is equal to 0, refPicHeightEffective$_C$ is set equal to PicHeightInSamples$_C$.

– Otherwise (MbaffFrameFlag is equal to 1 and mb_field_decoding_flag is equal to 1), refPicHeightEffective$_C$ is set equal to PicHeightInSamples$_C$ / 2.

The sample coordinates specified in Equations 8-258 through 8-265 are used for generating the predicted chroma sample value predPartLX$_C$[x_C, y_C].

$$xA_C = Clip3(0, PicWidthInSamples_C – 1, xInt_C) \quad (8\text{-}258)$$
$$xB_C = Clip3(0, PicWidthInSamples_C – 1, xInt_C + 1) \quad (8\text{-}259)$$
$$xC_C = Clip3(0, PicWidthInSamples_C – 1, xInt_C) \quad (8\text{-}260)$$
$$xD_C = Clip3(0, PicWidthInSamples_C – 1, xInt_C + 1) \quad (8\text{-}261)$$

$$yA_C = Clip3(0, refPicHeightEffective_C – 1, yInt_C) \quad (8\text{-}262)$$
$$yB_C = Clip3(0, refPicHeightEffective_C – 1, yInt_C) \quad (8\text{-}263)$$
$$yC_C = Clip3(0, refPicHeightEffective_C – 1, yInt_C + 1) \quad (8\text{-}264)$$
$$yD_C = Clip3(0, refPicHeightEffective_C – 1, yInt_C + 1) \quad (8\text{-}265)$$

Given the chroma samples A, B, C, and D at full-sample locations specified in Equations 8-258 through 8-265, the predicted chroma sample value $predPartLX_C[x_C, y_C]$ is derived as follows:

$$predPartLX_C[x_C, y_C] = ((8 - xFrac_C) * (8 - yFrac_C) * A + xFrac_C * (8 - yFrac_C) * B + (8 - xFrac_C) * yFrac_C * C + xFrac_C * yFrac_C * D + 32) >> 6 \quad (8\text{-}266)$$

8.4.2.3 Weighted sample prediction process

Inputs to this process are

- mbPartIdx: the current partition given by the partition index
- subMbPartIdx: the sub-macroblock partition index
- predFlagL0 and predFlagL1: prediction list utilization flags
- $predPartLX_L$: a (partWidth)x(partHeight) array of prediction luma samples (with LX being replaced by L0 or L1 depending on predFlagL0 and predFlagL1)
- when chroma_format_idc is not equal to 0 (monochrome), $predPartLX_{Cb}$ and $predPartLX_{Cr}$: (partWidthC)x(partHeightC) arrays of prediction chroma samples, one for each of the chroma components Cb and Cr (with LX being replaced by L0 or L1 depending on predFlagL0 and predFlagL1)

Outputs of this process are

- $predPart_L$: a (partWidth)x(partHeight) array of prediction luma samples and
- when chroma_format_idc is not equal to 0 (monochrome), $predPart_{Cb}$, and $predPart_{Cr}$: (partWidthC)x(partHeightC) arrays of prediction chroma samples, one for each of the chroma components Cb and Cr.

For macroblocks or partitions with predFlagL0 equal to 1 in P and SP slices, the following applies.

- If weighted_pred_flag is equal to 0, the default weighted sample prediction process as described in subclause 8.4.2.3.1 is invoked with the same inputs and outputs as the process described in this subclause.
- Otherwise (weighted_pred_flag is equal to 1), the explicit weighted prediction process as described in subclause 8.4.2.3.2 is invoked with the same inputs and outputs as the process described in this subclause.

For macroblocks or partitions with predFlagL0 or predFlagL1 equal to 1 in B slices, the following applies.

- If weighted_bipred_idc is equal to 0, the default weighted sample prediction process as described in subclause 8.4.2.3.1 is invoked with the same inputs and outputs as the process described in this subclause.
- Otherwise, if weighted_bipred_idc is equal to 1, the explicit weighted sample prediction process as described in subclause 8.4.2.3.2, for macroblocks or partitions with predFlagL0 or predFlagL1 equal to 1 with the same inputs and outputs as the process described in this subclause.
- Otherwise (weighted_bipred_idc is equal to 2), the following applies.
  - If predFlagL0 is equal to 1 and predFlagL1 is equal to 1, the implicit weighted sample prediction as described in subclause 8.4.2.3.2 is invoked with the same inputs and outputs as the process described in this subclause.
  - Otherwise (predFlagL0 or predFlagL1 are equal to 1 but not both), the default weighted sample prediction process as described in subclause 8.4.2.3.1 is invoked with the same inputs and outputs as the process described in this subclause.

8.4.2.3.1 Default weighted sample prediction process

Input to this process are the same as specified in subclause 8.4.2.3.

Output of this process are the same as specified in subclause 8.4.2.3.

Depending on the available component for which the prediction block is derived, the following applies.

- If the luma sample prediction values $predPart_L[x, y]$ are derived, the following applies with C set equal to L, x set equal to 0 .. partWidth - 1, and y set equal to 0 .. partHeight - 1.
- Otherwise, if the chroma Cb component sample prediction values $predPart_{Cb}[x, y]$ are derived, the following applies with C set equal to Cb, x set equal to 0 .. partWidthC - 1, and y set equal to 0 .. partHeightC - 1.
- Otherwise (the chroma Cr component sample prediction values $predPart_{Cr}[x, y]$ are derived), the following applies with C set equal to Cr, x set equal to 0 .. partWidthC - 1, and y set equal to 0 .. partHeightC - 1.

The prediction sample values are derived as follows.

– If predFlagL0 is equal to 1 and predFlagL1 is equal to 0 for the current partition

$$predPart_C[x, y] = predPartL0_C[x, y] \quad (8\text{-}267)$$

– Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1 for the current partition

$$predPart_C[x, y] = predPartL1_C[x, y] \quad (8\text{-}268)$$

– Otherwise (predFlagL0 and predFlagL1 are equal to 1 for the current partition),

$$predPart_C[x, y] = (predPartL0_C[x, y] + predPartL1_C[x, y] + 1) >> 1. \quad (8\text{-}269)$$

8.4.2.3.2 Weighted sample prediction process

Input to this process are the same as specified in subclause 8.4.2.3.

Output of this process are the same as specified in subclause 8.4.2.3.

Depending on the available component for which the prediction block is derived, the following applies.

– If the luma sample prediction values $predPart_L[x, y]$ are derived, the following applies with C set equal to L, x set equal to 0 .. partWidth - 1, and y set equal to 0 .. partHeight - 1.

– Otherwise, if the chroma Cb component sample prediction values $predPart_{Cb}[x, y]$ are derived, the following applies with C set equal to Cb, x set equal to 0 .. partWidthC - 1, and y set equal to 0 .. partHeightC - 1.

– Otherwise (the chroma Cr component sample prediction values $predPart_{Cr}[x, y]$ are derived), the following applies with C set equal to Cr, x set equal to 0 .. partWidthC - 1, and y set equal to 0 .. partHeightC - 1.

The prediction sample values are derived as follows

– If the partition mbPartIdx\subMbPartIdx has predFlagL0 equal to 1 and predFlagL1 equal to 0, the final predicted sample values $predPart_C[x, y]$ are derived by

$$\begin{aligned} &\text{if(logWD >= 1)} \\ &\quad predPart_C[x, y] = Clip1_C(((predPartL0_C[x, y] * w_0 + 2^{logWD - 1}) >> logWD) + o_0) \\ &\text{else} \\ &\quad predPart_C[x, y] = Clip1_C(predPartL0_C[x, y] * w_0 + o_0) \end{aligned} \quad (8\text{-}270)$$

– Otherwise, if the partition mbPartIdx\subMbPartIdx has predFlagL0 equal to 0 and predFlagL1 equal to 1, the final predicted sample values $predPart_C[x, y]$ are derived by

$$\begin{aligned} &\text{if(logWD >= 1)} \\ &\quad predPart_C[x, y] = Clip1_C(((predPartL1_C[x, y] * w_1 + 2^{logWD - 1}) >> logWD) + o_1) \\ &\text{else} \\ &\quad predPart_C[x, y] = Clip1_C(predPartL1_C[x, y] * w_1 + o_1) \end{aligned} \quad (8\text{-}271)$$

– Otherwise (the partition mbPartIdx\subMbPartIdx has both predFlagL0 and predFlagL1 equal to 1), the final predicted sample values $predPart_C[x, y]$ are derived by

$$\begin{aligned} predPart_C[x, y] = Clip1_C(((predPartL0_C[x, y] * w_0 + predPartL1_C[x, y] * w_1 + 2^{logWD}) >> \\ (logWD + 1)) + ((o_0 + o_1 + 1) >> 1)) \end{aligned} \quad (8\text{-}272)$$

The variables in the above derivation for the prediction samples are derived as follows.

– If weighted_bipred_idc is equal to 2 and the slice_type is equal to B, implicit mode weighted prediction is used as follows.

$$logWD = 5 \quad (8\text{-}273)$$

$o_0 = 0$ (8-274)

$o_1 = 0$ (8-275)

and w_0 and w_1 are derived as follows.

– The variables currPicOrField, pic0, and pic1 are derived as follows:
  – If field_pic_flag is equal to 0 and the current macroblock is a field macroblock, the following applies.
    – currPicOrField is the field of the current picture CurrPic that has the same parity as the current macroblock.
    – The variable pic0 is derived as follows.
      – If refIdxL0 % 2 is equal to 0, pic0 is the field of RefPicList0[refIdxL0 / 2] that has the same parity as the current macroblock.
      – Otherwise (refIdxL0 % 2 is not equal to 0), pic0 is the field of RefPicList0[refIdxL0 / 2] that has the opposite parity of the current macroblock.
    – The variable pic1 is derived as follows.
      – If refIdxL1 % 2 is equal to 0, pic1 is the field of RefPicList1[refIdxL1 / 2] that has the same parity as the current macroblock.
      – Otherwise (refIdxL1 % 2 is not equal to 0), pic1 is the field of RefPicList1[refIdxL1 / 2] that has the opposite parity of the current macroblock.
  – Otherwise (field_pic_flag is equal to 1 or the current macroblock is a frame macroblock), currPicOrField is the current picture CurrPic, pic1 is RefPicList1[refIdxL1], and pic0 is RefPicList0[refIdxL0].
– The variables tb, td, tx, and DistScaleFactor are derived from the values of currPicOrField, pic0, pic1 using Equations 8-198, 8-199, 8-194, and 8-195, respectively.
– If DiffPicOrderCnt(pic1, pic0) is equal to 0 or one or both of pic1 and pic0 is marked as "used for long-term reference" or (DistScaleFactor >> 2) < -64 or (DistScaleFactor >> 2) > 128, w_0 and w_1 are derived as

  $w_0 = 32$ (8-276)

  $w_1 = 32$ (8-277)

- Otherwise,

  $w_0 = 64 - (\text{DistScaleFactor} >> 2)$ (8-278)

  $w_1 = \text{DistScaleFactor} >> 2$ (8-279)

– Otherwise (weighted_pred_flag is equal to 1 in P or SP slices or weighted_bipred_idc equal to 1 in B slices), explicit mode weighted prediction is used as follows.
  – The variables refIdxL0WP and refIdxL1WP are derived as follows.
    – If MbaffFrameFlag is equal to 1 and the current macroblock is a field macroblock

      refIdxL0WP = refIdxL0 >> 1 (8-280)

      refIdxL1WP = refIdxL1 >> 1 (8-281)

    – Otherwise (MbaffFrameFlag is equal to 0 or the current macroblock is a frame macroblock),

      refIdxL0WP = refIdxL0 (8-282)

refIdxL1WP = refIdxL1 (8-283)

– The variables logWD, w_0, w_1, o_0, and o_1 are derived as follows.

– If C in $predPart_C[x, y]$ is replaced by L for luma samples

logWD = luma_log2_weight_denom (8-284)

w_0 = luma_weight_l0[refIdxL0WP] (8-285)

w_1 = luma_weight_l1[refIdxL1WP] (8-286)

o_0 = luma_offset_l0[refIdxL0WP] * (1 << ($BitDepth_Y$ – 8)) (8-287)

o_1 = luma_offset_l1[refIdxL1WP] * (1 << ($BitDepth_Y$ – 8)) (8-288)

– Otherwise (C in $predPart_C[x, y]$ is replaced by Cb or Cr for chroma samples, with iCbCr = 0 for Cb, iCbCr = 1 for Cr),

logWD = chroma_log2_weight_denom (8-289)

w_0 = chroma_weight_l0[refIdxL0WP][iCbCr] (8-290)

w_1 = chroma_weight_l1[refIdxL1WP][iCbCr] (8-291)

o_0 = chroma_offset_l0[refIdxL0WP][iCbCr] * (1 << ($BitDepth_C$ – 8)) (8-292)

o_1 = chroma_offset_l1[refIdxL1WP][iCbCr] * (1 << ($BitDepth_C$ – 8)) (8-293)

When explicit mode weighted prediction is used and the partition mbPartIdx\subMbPartIdx has both predFlagL0 and predFlagL1 equal to 1, the following constraint shall be obeyed

-128 <= w_0 + w_1 <= ((logWD == 7) ? 127 : 128) (8-294)

NOTE – For implicit mode weighted prediction, weights w_0 and w_1 are each guaranteed to be in the range of -64..128 and the constraint expressed in Equation 8-294, although not explicitly imposed, will always be met. For explicit mode weighted prediction with logWD equal to 7, when one of the two weights w_0 or w_1 is inferred to be equal to 128 (as a consequence of luma_weight_l0_flag, luma_weight_l1_flag, chroma_weight_l0_flag, or chroma_weight_l1_flag equal to 0), the other weight (w_1 or w_0) must have a negative value in order for the constraint expressed in Equation 8-294 to hold (and therefore the other flag luma_weight_l0_flag, luma_weight_l1_flag, chroma_weight_l0_flag, or chroma_weight_l1_flag must be equal to 1).

8.5 Transform coefficient decoding process and picture construction process prior to deblocking filter process

Inputs to this process are Intra16x16DCLevel (if available), Intra16x16ACLevel (if available), LumaLevel (if available), LumaLevel8x8 (if available), ChromaDCLevel (if available), ChromaACLevel (if available), and available Inter or Intra prediction sample arrays for the current macroblock for the applicable components $pred_L$, $pred_{Cb}$, or $pred_{Cr}$.

NOTE 1 – When decoding a macroblock in Intra_4x4 (or Intra_8x8) prediction mode, the luma component of the macroblock prediction array may not be complete, since for each 4x4 (or 8x8) luma block, the Intra_4x4 (or Intra_8x8) prediction process for luma samples as specified in subclause 8.3.1 (or 8.3.2) and the process specified in this subclause are iterated.

Outputs of this process are the constructed sample arrays prior to the deblocking filter process for the applicable components S'_L, S'_{Cb}, or S'_{Cr}.

NOTE 2 – When decoding a macroblock in Intra_4x4 (or Intra_8x8) prediction mode, the luma component of the macroblock constructed sample arrays prior to the deblocking filter process may not be complete, since for each 4x4 (or 8x8) luma block, the

Intra_4x4 (or Intra_8x8) prediction process for luma samples as specified in subclause 8.3.1 (or 8.3.2) and the process specified in this subclause are iterated.

This subclause specifies transform coefficient decoding and picture construction prior to the deblocking filter process.

When the current macroblock is coded as P_Skip or B_Skip, all values of LumaLevel, LumaLevel8x8, ChromaDCLevel, ChromaACLevel are set equal to 0 for the current macroblock.

When residual_colour_transform_flag is equal to 1, the residual colour transform process as specified in subclause 8.5.13 is invoked.

8.5.1 Specification of transform decoding process for 4x4 luma residual blocks

This specification applies when transform_size_8x8_flag is equal to 0.

When the current macroblock prediction mode is not equal to Intra_16x16, the variable LumaLevel contains the levels for the luma transform coefficients. For a 4x4 luma block indexed by luma4x4BlkIdx = 0..15, the following ordered steps are specified.

1. The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with LumaLevel[luma4x4BlkIdx] as the input and the two-dimensional array c as the output.
2. The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c as the input and r as the output.
3. When residual_colour_transform_flag is equal to 1, the variable $R_{Y,ij}$ is set equal to r_{ij} with i, j = 0..3 and this process is suspended until after completion of the residual colour transform process as specified in subclause 8.5.13, after the completion of which, the variable r_{ij} is set equal to $R_{G,ij}$ with i, j = 0..3 and this process is continued.
4. The position of the upper-left sample of a 4x4 luma block with index luma4x4BlkIdx inside the macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (xO, yO).
5. The 4x4 array u with elements u_{ij} for i, j = 0..3 is derived as

   $$u_{ij} = \text{Clip1}_Y(\text{pred}_L[xO + j, yO + i] + r_{ij}) \quad (8\text{-}295)$$

   When qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the bitstream shall not contain data that result in a value of u_{ij} as computed by Equation 8-295 that is not equal to $\text{pred}_L[xO + j, yO + i] + r_{ij}$.
6. The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with luma4x4BlkIdx and u as the inputs.

8.5.2 Specification of transform decoding process for luma samples of Intra_16x16 macroblock prediction mode

When the current macroblock prediction mode is equal to Intra_16x16, the variables Intra16x16DCLevel and Intra16x16ACLevel contain the levels for the luma transform coefficients. The transform coefficient decoding proceeds in the following ordered steps:

1. The 4x4 luma DC transform coefficients of all 4x4 luma blocks of the macroblock are decoded.
   a. The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with Intra16x16DCLevel as the input and the two-dimensional array c as the output.
   b. The scaling and transformation process for luma DC transform coefficients for Intra_16x16 macroblock type as specified in subclause 8.5.8 is invoked with c as the input and dcY as the output.
2. For a 4x4 luma block indexed by luma4x4BlkIdx = 0..15, the following ordered steps are specified.
   a. The variable lumaList, which is a list of 16 entries, is derived. The first entry of lumaList is the corresponding value from the array dcY. Figure 8-6 shows the assignment of the indices of the array dcY to the luma4x4BlkIdx. The two numbers in the small squares refer to indices i and j in dcY_{ij}, and the numbers in large squares refer to luma4x4BlkIdx.

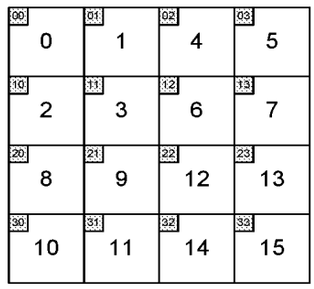


Figure 8-6 – Assignment of the indices of dcY to luma4x4BlkIdx

The elements in lumaList with index k = 1..15 are specified as

lumaList[k] = Intra16x16ACLevel[luma4x4BlkIdx][k - 1] (8-296)

b. The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with lumaList as the input and the two-dimensional array c as the output.

c. The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c as the input and r as the output.

d. When residual_colour_transform_flag is equal to 1, the variable $R_{Y,ij}$ is set equal to r_{ij} with i, j = 0..3 and this process is suspended until after completion of the residual colour transform process as specified in subclause 8.5.13, after the completion of which, the variable r_{ij} is set equal to $R_{G,ij}$ with i, j = 0..3 and this process is continued.

e. The position of the upper-left sample of a 4x4 luma block with index luma4x4BlkIdx inside the macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (xO, yO).

f. The 4x4 array u with elements u_{ij} for i, j = 0..3 is derived as

$$u_{ij} = Clip1_Y(pred_L[xO + j, yO + i] + r_{ij}) \quad (8\text{-}297)$$

When qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the bitstream shall not contain data that result in a value of u_{ij} as computed by 8-297 that is not equal to $pred_L[xO + j, yO + i] + r_{ij}$.

g. The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with luma4x4BlkIdx and u as the inputs.

8.5.3 Specification of transform decoding process for 8x8 luma residual blocks

This specification applies when transform_size_8x8_flag is equal to 1.

The variable LumaLevel8x8[luma8x8BlkIdx] with luma8x8BlkIdx = 0..3 contains the levels for the luma transform coefficients for the luma 8x8 block with index luma8x8BlkIdx.

For an 8x8 luma block indexed by luma8x8BlkIdx = 0..3, the following ordered steps are specified.

1. The inverse scanning process for 8x8 luma transform coefficients as described in subclause 8.5.6 is invoked with LumaLevel8x8[luma8x8BlkIdx] as the input and the two-dimensional array c as the output.

2. The scaling and transformation process for residual 8x8 blocks as specified in subclause 8.5.11 is invoked with c as the input and r as the output.

3. When residual_colour_transform_flag is equal to 1, the variable $R_{Y,ij}$ is set equal to r_{ij} with i, j = 0..7 and this process is suspended until after completion of the residual colour transform process as specified in subclause 8.5.13, after the completion of which, the variable r_{ij} is set equal to $R_{G,ij}$ with i, j = 0..7 and this process is continued.

4. The position of the upper-left sample of an 8x8 luma block with index luma8x8BlkIdx inside the macroblock is derived by invoking the inverse 8x8 luma block scanning process in subclause 6.4.4 with luma8x8BlkIdx as the input and the output being assigned to (xO, yO).

5. The 8x8 array u with elements u_{ij} for i, j = 0..7 is derived as

$$u_{ij} = Clip1_Y(pred_L[xO + j, yO + i] + r_{ij}) \quad (8\text{-}298)$$

When qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the bitstream shall not contain data that result in a value of u_{ij} as computed by Equation 8-298 that is not equal to $pred_L[xO + j, yO + i] + r_{ij}$.

6. The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with luma8x8BlkIdx and u as the inputs.

8.5.4 Specification of transform decoding process for chroma samples

This process is invoked for each chroma component Cb and Cr seperately.

For each chroma component, the variables ChromaDCLevel[iCbCr] and ChromaACLevel[iCbCr], with iCbCr set equal to 0 for Cb and iCbCr set equal to 1 for Cr, contain the levels for both components of the chroma transform coefficients.

Let the variable numChroma4x4Blks be set equal to (MbWidthC / 4) * (MbHeightC / 4).

For each chroma component, the transform decoding proceeds separately in the following ordered steps:

1. The numChroma4x4Blks chroma DC transform coefficients of the 4x4 chroma blocks of the component indexed by iCbCr of the macroblock are decoded.

   a. Depending on the variable chroma_format_idc, the following applies.

   – If chroma_format_idc is equal to 1, the 2x2 array c is derived using the inverse raster scanning process applied to ChromaDCLevel as follows

$$c = \begin{bmatrix} ChromaDCLevel[iCbCr][0] & ChromaDCLevel[iCbCr][1] \\ ChromaDCLevel[iCbCr][2] & ChromaDCLevel[iCbCr][3] \end{bmatrix} \quad (8\text{-}299)$$

   – Otherwise, if chroma_format_idc is equal to 2, the 2x4 array c is derived using the inverse raster scanning process applied to ChromaDCLevel as follows

$$c = \begin{bmatrix} ChromaDCLevel[iCbCr][0] & ChromaDCLevel[iCbCr][2] \\ ChromaDCLevel[iCbCr][1] & ChromaDCLevel[iCbCr][5] \\ ChromaDCLevel[iCbCr][3] & ChromaDCLevel[iCbCr][6] \\ ChromaDCLevel[iCbCr][4] & ChromaDCLevel[iCbCr][7] \end{bmatrix} \quad (8\text{-}300)$$

   – Otherwise (chroma_format_idc is equal to 3), the inverse scanning process for transform coefficients as specified in subclause 8.5.5 is invoked with ChromaDCLevel[iCbCr] as the input and the two-dimensional 4x4 array c as the output.

   b. The scaling and transformation process for chroma DC transform coefficients as specified in subclause 8.5.9 is invoked with c as the input and dcC as the output.

2. For each 4x4 chroma block indexed by chroma4x4BlkIdx = 0..numChroma4x4Blks – 1 of the component indexed by iCbCr, the following ordered steps are specified.

   a. The variable chromaList, which is a list of 16 entries, is derived. The first entry of chromaList is the corresponding value from the array dcC. Figure 8-7 shows the assignment of the indices of the array dcC to the chroma4x4BlkIdx. The two numbers in the small squares refer to indices i and j in dcC_{ij}, and the numbers in large squares refer to chroma4x4BlkIdx.

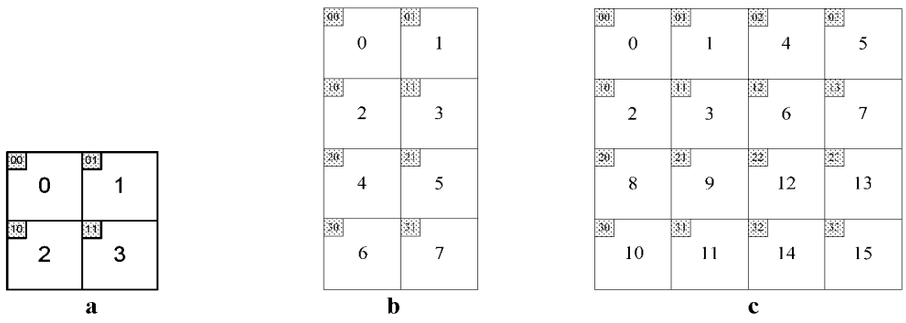


Figure 8-7 – Assignment of the indices of dcC to chroma4x4BlkIdx:
(a) chroma_format_idc equal to 1, (b) chroma_format_idc equal to 2, (c) chroma_format_idc equal to 3

The elements in chromaList with index k = 1..15 are specified as

chromaList[k] = ChromaACLevel[chroma4x4BlkIdx][k – 1] (8-301)

b. The inverse scanning process for transform coefficients as specified in subclause 8.5.9 is invoked with chromaList as the input and the two-dimensional array c as the output.

c. The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c as the input and r as the output.

d. Depending on the variable chroma_format_idc, the position of the upper-left sample of a 4x4 chroma block with index chroma4x4BlkIdx inside the macroblock is derived as follows.

– If chroma_format_idc is equal to 1 or 2, the following applies.

xO = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 0) (8-302)

yO = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 1) (8-303)

– Otherwise (chroma_format_idc is equal to 3), the following applies.

xO = InverseRasterScan(chroma4x4BlkIdx / 4, 8, 8, 16, 0) +
InverseRasterScan(chroma4x4BlkIdx % 4, 4, 4, 8, 0) (8-304)

yO =InverseRasterScan(chroma4x4BlkIdx / 4, 8, 8, 16, 1) +
InverseRasterScan(chroma4x4BlkIdx % 4, 4, 4, 8, 1) (8-305)

e. When residual_colour_transform_flag is equal to 1, the variable xO' is set equal to xO % (4 << transform_size_8x8_flag), the variable yO' is set equal to yO % (4 << transform_size_8x8_flag), and the following applies.

– If this process is invoked for the chroma component Cb, the variable $R_{Cb,mn}$ is set equal to r_{ij} with i, j = 0..3, m = xO' + i, n = yO' + j, and this process is suspended until after completion of the residual colour transform process as specified in subclause 8.5.13, after which the variable r_{ij} is set equal to $R_{B,mn}$ with i, j = 0..3, m = xO' + i, n = yO' + j and this process is continued.

– Otherwise (this process is invoked for the chroma component Cr), the variable $R_{Cr,mn}$ is set equal to r_{ij} with i, j = 0..3, m = xO' + i, n = yO' + j and this process is suspended until after the completion of the residual colour transform process as specified in subclause 8.5.13, after which the variable r_{ij} is set equal to $R_{R,mn}$ with i, j = 0..3, m = xO' + i, n = yO' + j and this process is continued.

f. The 4x4 array u with elements u_{ij} for i, j = 0..3 is derived as

$$u_{ij} = Clip1_C(pred_C[xO + j, yO + i] + r_{ij}) \quad (8\text{-}306)$$

When qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the bitstream shall not contain data that result in a value of u_{ij} as computed by Equation 8-306 that is not equal to $pred_C[xO + j, yO + i] + r_{ij}$.

g. The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with chroma4x4BlkIdx and u as the inputs.

8.5.5 Inverse scanning process for transform coefficients

Input to this process is a list of 16 values.

Output of this process is a variable c containing a two-dimensional array of 4x4 values. In the case of transform coefficients, these 4x4 values represent levels assigned to locations in the transform block. In the case of applying the inverse scanning process to a scaling list, the output variable c contains a two-dimensional array representing a 4x4 scaling matrix.

The inverse scanning process for transform coefficients maps the sequence of transform coefficient levels to the transform coefficient level positions. Table 8-13 specifies the two mappings: inverse zig-zag scan and inverse field scan. The inverse zig-zag scan is used for transform coefficients in frame macroblocks and the inverse field scan is used for transform coefficients in field macroblocks.

The inverse scanning process for scaling lists maps the sequence of scaling list entries to the positions in the corresponding scaling matrix. For this mapping, the inverse zig-zag scan is used.

Figure 8-8 illustrates the scans.

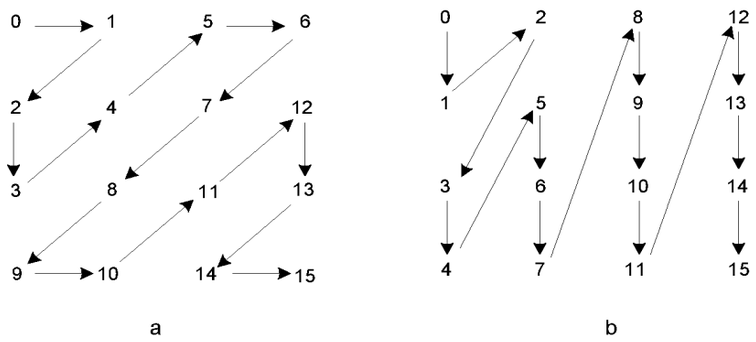


Figure 8-8 – 4x4 block scans. (a) Zig-zag scan. (b) Field scan (informative)

Table 8-13 provides the mapping from the index idx of input list of 16 elements to indices i and j of the two-dimensional array c.

Table 8-13 – Specification of mapping of idx to c_{ij} for zig-zag and field scan

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
zig-zag	c_{00}	c_{01}	c_{10}	c_{20}	c_{11}	c_{02}	c_{03}	c_{12}	c_{21}	c_{30}	c_{31}	c_{22}	c_{13}	c_{23}	c_{32}	c_{33}
field	c_{00}	c_{10}	c_{01}	c_{20}	c_{30}	c_{11}	c_{21}	c_{31}	c_{02}	c_{12}	c_{22}	c_{32}	c_{03}	c_{13}	c_{23}	c_{33}

8.5.6 Inverse scanning process for 8x8 luma transform coefficients

Input to this process is a list of 64 values.

Output of this process is a variable c containing a two-dimensional array of 8x8 values. In the case of transform coefficients, these 8x8 values represent levels assigned to locations in the transform block. In the case of applying the inverse scanning process to a scaling list, the output variable c contains a two-dimensional array representing an 8x8 scaling matrix.

The inverse scanning process for transform coefficients maps the sequence of transform coefficient levels to the transform coefficient level positions. Table 8-14 specifies the two mappings: inverse 8x8 zig-zag scan and inverse 8x8 field scan. The inverse 8x8 zig-zag scan is used for transform coefficients in frame macroblocks and the inverse 8x8 field scan is used for transform coefficients in field macroblocks.

The inverse scanning process for scaling lists maps the sequence of scaling list entries to the positions in the corresponding scaling matrix. For this mapping, the inverse zig-zag scan is used.

Figure 8-9 illustrates the scans.

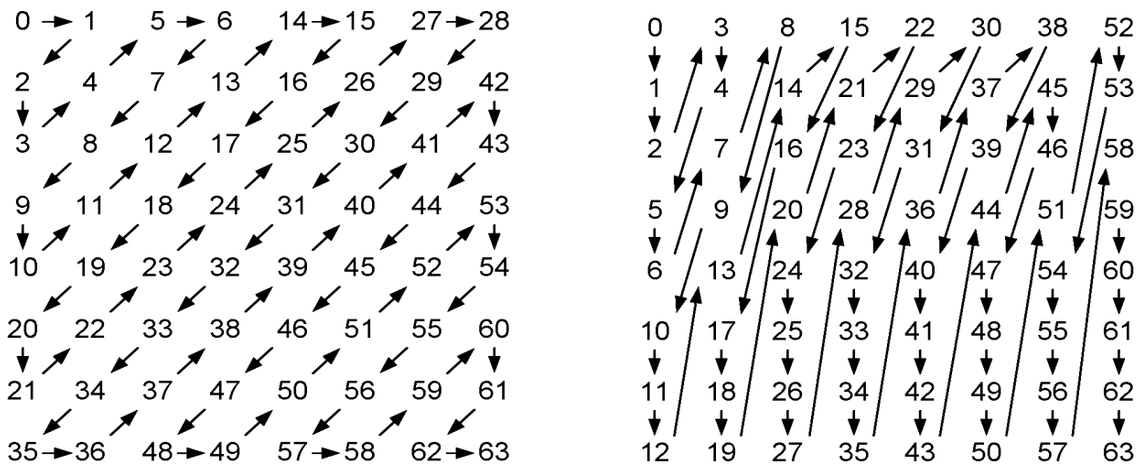


Figure 8-9 – 8x8 block scans. (a) 8x8 zig-zag scan. (b) 8x8 field scan (informative)

Table 8-14 provides the mapping from the index idx of the input list of 64 elements to indices i and j of the two-dimensional array c.

Table 8-14 – Specification of mapping of idx to c_{ij} for 8x8 zig-zag and 8x8 field scan

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
zig-zag	c_{00}	c_{01}	c_{10}	c_{20}	c_{11}	c_{02}	c_{03}	c_{12}	c_{21}	c_{30}	c_{40}	c_{31}	c_{22}	c_{13}	c_{04}	c_{05}
field	c_{00}	c_{10}	c_{20}	c_{01}	c_{11}	c_{30}	c_{40}	c_{21}	c_{02}	c_{31}	c_{50}	c_{60}	c_{70}	c_{41}	c_{12}	c_{03}

Table 8-14 (continued) – Specification of mapping of idx to cij for 8x8 zig-zag and 8x8 field scan

idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
zig-zag	c_{14}	c_{23}	c_{32}	c_{41}	c_{50}	c_{60}	c_{51}	c_{42}	c_{33}	c_{24}	c_{15}	c_{06}	c_{07}	c_{16}	c_{25}	c_{34}
field	c_{22}	c_{51}	c_{61}	c_{71}	c_{32}	c_{13}	c_{04}	c_{23}	c_{42}	c_{52}	c_{62}	c_{72}	c_{33}	c_{14}	c_{05}	c_{24}

Table 8-14 (continued) – Specification of mapping of idx to cij for 8x8 zig-zag and 8x8 field scan

idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
zig-zag	c_{43}	c_{52}	c_{61}	c_{70}	c_{71}	c_{62}	c_{53}	c_{44}	c_{35}	c_{26}	c_{17}	c_{27}	c_{36}	c_{45}	c_{54}	c_{63}
field	c_{43}	c_{53}	c_{63}	c_{73}	c_{34}	c_{15}	c_{06}	c_{25}	c_{44}	c_{54}	c_{64}	c_{74}	c_{35}	c_{16}	c_{26}	c_{45}

Table 8-14 (concluded) – Specification of mapping of idx to cij for 8x8 zig-zag and 8x8 field scan

idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
zig-zag	c_{72}	c_{73}	c_{64}	c_{55}	c_{46}	c_{37}	c_{47}	c_{56}	c_{65}	c_{74}	c_{75}	c_{66}	c_{57}	c_{67}	c_{76}	c_{77}
field	c_{55}	c_{65}	c_{75}	c_{36}	c_{07}	c_{17}	c_{46}	c_{56}	c_{66}	c_{76}	c_{27}	c_{37}	c_{47}	c_{57}	c_{67}	c_{77}

8.5.7 Derivation process for the chroma quantisation parameters and scaling function

Outputs of this process are:

– QP_C: the chroma quantisation parameter for each chroma component Cb and Cr

– QS_C: the additional chroma quantisation parameter for each chroma component Cb and Cr required for decoding SP and SI slices (if applicable)

NOTE 1 – QP quantisation parameter values QP_Y and QS_Y are always in the range of $-QpBdOffset_Y$ to 51, inclusive. QP quantisation parameter values QP_C and QS_C are always in the range of $-QpBdOffset_C$ to 51, inclusive.

The value of QP_C for a chroma component is determined from the current value of QP_Y and the value of chroma_qp_index_offset (for Cb) or second_chroma_qp_index_offset (for Cr).

NOTE 2 – The scaling equations are specified such that the equivalent transform coefficient level scaling factor doubles for every increment of 6 in QP_Y. Thus, there is an increase in the factor used for scaling of approximately 12 % for each increase of 1 in the value of QP_Y.

The value of QP_C for each chroma component is determined as specified in Table 8-15 based on the index denoted as qP_I.

The variable qP_{Offset} for each chroma component is derived as follows.

– If the chroma component is the Cb component, qP_{Offset} is specified as

$$qP_{Offset} = \text{chroma_qp_index_offset} \tag{8-307}$$

– Otherwise (the chroma component is the Cr component), qP_{Offset} is specified as

$$qP_{Offset} = \text{second_chroma_qp_index_offset} \tag{8-308}$$

The value of qP_I for each chroma component is derived as

$$qP_I = \text{Clip3}(-QpBdOffset_C, 51, QP_Y + qP_{Offset}) \tag{8-309}$$

The value of QP'_C for the chroma components is derived as

$$QP'_C = QP_C + QpBdOffset_C \tag{8-310}$$

The value of $BitDepth'_C$ for the chroma components is derived as

$$BitDepth'_C = BitDepth_C + residual_colour_transform_flag \tag{8-311}$$

Table 8-15 – Specification of QP_C as a function of qP_I

qP_I	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
QP_C	$=qP_I$	29	30	31	32	32	33	34	34	35	35	36	36	37	37	37	38	38	38	39	39	39	39

When the current slice is an SP or SI slice, QS_C is derived using the above process, substituting QP_Y with QS_Y and QP_C with QS_C.

The function LevelScale(m, i, j) is specified as follows.

– The 4x4 matrix weightScale(i, j) is specified as follows.
  – The variable mbIsInterFlag is derived as follows.
    – If the current macroblock is coded using Inter macroblock prediction modes, mbIsInterFlag is set equal to 1.
    – Otherwise (the current macroblock is coded using Intra macroblock prediction modes), mbIsInterFlag is set equal to 0.
  – The variable iYCbCr derived as follows.
    – If the input array c relates to a luma residual block, iYCbCr is set equal to 0.
    – Otherwise, if the input array c relates to a chroma residual block and the chroma component is equal to Cb, iYCbCr is set equal to 1.
    – Otherwise (the input array c relates to a chroma residual block and the chroma component is equal to Cr), iYCbCr is set equal to 2.
  – The inverse scanning process for transform coefficients as specified in subclause 8.5.5 is invoked with ScalingList4x4[iYCbCr + ((mbIsInterFlag == 1) ? 3 : 0)] as the input and the output is assigned to the 4x4 matrix weightScale.

$$LevelScale(m, i, j) = weightScale(i, j) * normAdjust(m, i, j) \tag{8-312}$$

where

$$normAdjust(m,i,j) = \begin{cases} v_{m0} & \text{for } (i \% 2, j \% 2) \text{ equal to } (0,0), \\ v_{m1} & \text{for } (i \% 2, j \% 2) \text{ equal to } (1,1), \\ v_{m2} & \text{otherwise;} \end{cases} \tag{8-313}$$

where the first and second subscripts of v are row and column indices, respectively, of the matrix specified as:

$$v = \begin{bmatrix} 10 & 16 & 13 \\ 11 & 18 & 14 \\ 13 & 20 & 16 \\ 14 & 23 & 18 \\ 16 & 25 & 20 \\ 18 & 29 & 23 \end{bmatrix}. \tag{8-314}$$

The function LevelScale8x8(m, i, j) is specified as follows:

– The 8x8 matrix weightScale8x8(i, j) is specified as follows.
  – The variable mbIsInterFlag is derived as follows.
    – If the current macroblock is coded using Inter macroblock prediction modes, mbIsInterFlag is set equal to 1.
    – Otherwise (the current macroblock is coded using Intra macroblock prediction modes), mbIsInterFlag is set equal to 0.
  – The inverse scanning process for 8x8 luma transform coefficients as specified in subclause 8.5.6 is invoked with ScalingList8x8[mbIsInterFlag] as the input and the output is assigned to the 8x8 matrix weightScale8x8.

$$\text{LevelScale8x8}(m, i, j) = \text{weightScale8x8}(i, j) * \text{normAdjust8x8}(m, i, j) \tag{8-315}$$

where

$$\text{normAdjust8x8}(m,i,j) = \begin{cases} v_{m0} & \text{for } (i\,\%\,4, j\,\%\,4) \text{ equal to } (0,0), \\ v_{m1} & \text{for } (i\,\%\,2, j\,\%\,2) \text{ equal to } (1,1), \\ v_{m2} & \text{for } (i\,\%\,4, j\,\%\,4) \text{ equal to } (2,2), \\ v_{m3} & \text{for } (i\,\%\,4, j\,\%\,2) \text{ equal to } (0,1) \text{ or } (i\,\%\,2, j\,\%\,4) \text{ equal to } (1,0), \\ v_{m4} & \text{for } (i\,\%\,4, j\,\%\,4) \text{ equal to } (0,2) \text{ or } (i\,\%\,4, j\,\%\,4) \text{ equal to } (2,0), \\ v_{m5} & \text{otherwise;} \end{cases} \tag{8-316}$$

where the first and second subscripts of v are row and column indices, respectively, of the matrix specified as:

$$v = \begin{bmatrix} 20 & 18 & 32 & 19 & 25 & 24 \\ 22 & 19 & 35 & 21 & 28 & 26 \\ 26 & 23 & 42 & 24 & 33 & 31 \\ 28 & 25 & 45 & 26 & 35 & 33 \\ 32 & 28 & 51 & 30 & 40 & 38 \\ 36 & 32 & 58 & 34 & 46 & 43 \end{bmatrix}. \tag{8-317}$$

8.5.8 Scaling and transformation process for luma DC transform coefficients for Intra_16x16 macroblock type

Inputs to this process are transform coefficient level values for luma DC transform coefficients of Intra_16x16 macroblocks as a 4x4 array c with elements c_{ij}, where i and j form a two-dimensional frequency index.

Outputs of this process are 16 scaled DC values for luma 4x4 blocks of Intra_16x16 macroblocks as a 4x4 array dcY with elements dcY_{ij}.

Depending on the values of qpprime_y_zero_transform_bypass_flag and QP'_Y, the following applies.

– If qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the output dcY is derived as

$$dcY_{ij} = c_{ij} \quad \text{with} \quad i, j = 0..3 \tag{8-318}$$

– Otherwise (qpprime_y_zero_transform_bypass_flag is equal to 0 or QP'_Y is not equal to 0), the following text of this process specifies the output.

The inverse transform for the 4x4 luma DC transform coefficients is specified by:

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{8-319}$$

The bitstream shall not contain data that results in any element f_{ij} of f with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)}-1$, inclusive.

After the inverse transform, scaling is performed as follows.

– If QP'_Y is greater than or equal to 36, the scaled result is derived as

$$dcY_{ij} = (f_{ij} * \text{LevelScale}(QP'_Y \% 6, 0, 0)) << (QP'_Y / 6 - 6), \quad \text{with} \quad i, j = 0..3 \tag{8-320}$$

– Otherwise (QP'_Y is less than 36), the scaled result is derived as

$$dcY_{ij} = (f_{ij} * \text{LevelScale}(QP'_Y \% 6, 0, 0) + 2^{5 - QP'_Y / 6}) >> (6 - QP'_Y / 6), \quad \text{with} \quad i, j = 0..3 \tag{8-321}$$

The bitstream shall not contain data that results in any element dcY_{ij} of dcY with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)}-1$, inclusive.

NOTE 1 – When entropy_coding_mode_flag is equal to 0 and QP'_Y is less than 10 and profile_idc is equal to 66, 77, or 88, the range of values that can be represented for the elements c_{ij} of c is not sufficient to represent the full range of values of the elements dcY_{ij} of dcY that could be necessary to form a close approximation of the content of any possible source picture by use of the Intra_16x16 macroblock type.

NOTE 2 – Since the range limit imposed on the elements dcY_{ij} of dcY is imposed after the right shift in Equation 8-321, a larger range of values must be supported in the decoder prior to the right shift.

8.5.9 Scaling and transformation process for chroma DC transform coefficients

Inputs to this process are transform coefficient level values for chroma DC transform coefficients of one chroma component of the macroblock as an (MbWidthC / 4)x(MbHeightC / 4) array c with elements c_{ij}, where i and j form a two-dimensional frequency index.

Outputs of this process are the scaled DC values as an (MbWidthC / 4)x(MbHeightC / 4) array dcC with elements dcC_{ij}.

Depending on the values of qpprime_y_zero_transform_bypass_flag and QP'_Y, the following applies.

– If qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the output dcC is derived as

$$dcC_{ij} = c_{ij} \quad \text{with} \quad i = 0..(\text{MbWidthC} / 4) - 1 \text{ and } j = 0..(\text{MbHeightC} / 4) - 1. \tag{8-322}$$

– Otherwise (qpprime_y_zero_transform_bypass_flag is equal to 0 or QP'_Y is not equal to 0), the following text of this process specifies the output.

Depending on the variable chroma_format_idc, the inverse transform is specified as follows.

– If chroma_format_idc is equal to 1, the inverse transform for the 2x2 chroma DC transform coefficients is specified as

$$f = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8-323}$$

– Otherwise, if chroma_format_idc is equal to 2, the inverse transform for the 2x4 chroma DC transform coefficients is specified as

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ c_{20} & c_{21} \\ c_{30} & c_{31} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8-324}$$

– Otherwise (chroma_format_idc is equal to 3), the inverse transform for the 4x4 chroma DC transform coefficients is specified as follows.
  – If residual_colour_transform_flag is equal to 1 and the current macroblock prediction mode MbPartPredMode(mb_type, 0) is Intra_4x4 or Intra_8x8, the inverse transform for the 4x4 chroma DC transform coefficients is specified as

$$f_{ij} = c_{ij} << 2 \quad \text{with} \quad i, j = 0..3 \tag{8-325}$$

  – Otherwise, the inverse transform for the 4x4 chroma DC transform coefficients is specified as

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \tag{8-326}$$

The bitstream shall not contain data that results in any element f_{ij} of f with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}'_C)}$ to $2^{(7 + \text{BitDepth}'_C)}-1$, inclusive.

After the inverse transform, scaling is performed depending on the variable chroma_format_idc as follows.

– If chroma_format_idc is equal to 1, the scaled result is derived as

$$dcC_{ij} = ((f_{ij} * \text{LevelScale}(QP'_C \,\%\, 6, 0, 0)) << (QP'_C / 6)) >> 5, \quad \textit{with} \quad i, j = 0, 1 \tag{8-327}$$

– If chroma_format_idc is equal to 2, the following applies.
  – The variable $QP'_{C,DC}$ is derived as

$$QP'_{C,DC} = QP'_C + 3 \tag{8-328}$$

  – Depending on the value of $QP'_{C,DC}$, the following applies.
    – If $QP'_{C,DC}$ is greater than or equal to 36, the scaled result is derived as

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_{C,DC} \,\%6, 0, 0)) << (QP'_{C,DC} / 6 - 6), \quad \text{with } i = 0..3, j = 0, 1 \tag{8-329}$$

    – Otherwise ($QP'_{C,DC}$ is less than 36), the scaled result is derived as

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_{C,DC} \,\%\, 6, 0, 0) + 2^{5 - QP'_{C,DC}/6}) >> (6 - QP'_{C,DC} / 6), \quad \text{with} \quad i = 0..3, j = 0, 1 \tag{8-330}$$

– Otherwise (chroma_format_idc is equal to 3), the following applies.
  – If QP'_C is greater than or equal to 36, the scaled result is derived as

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_C \,\%6, 0, 0)) << (QP'_C / 6 - 6), \quad \text{with} \quad i, j = 0..3. \tag{8-331}$$

  – Otherwise (QP'_C is less than 36), the scaled result is derived as

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_C \,\%\, 6, 0, 0) + 2^{5 - QP'_C / 6}) >> (6 - QP'_C / 6), \text{ with } i, j, = 0..3 \tag{8-332}$$

The bitstream shall not contain data that results in any element dcC_{ij} of dcC with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}'_C)}$ to $2^{(7 + \text{BitDepth}'_C)}-1$, inclusive.

NOTE 1 – When entropy_coding_mode_flag is equal to 0 and QP'_C is less than 4 and profile_idc is equal to 66, 77, or 88, the range of values that can be represented for the elements cij of c may not be sufficient to represent the full range of values of the elements dcCij of dcC that could be necessary to form a close approximation of the content of any possible source picture.

NOTE 2 – Since the range limit imposed on the elements dcCij of dcC is imposed after the right shift in Equation 8-327, 8-330, or 8-332, a larger range of values must be supported in the decoder prior to the right shift.

8.5.10 Scaling and transformation process for residual 4x4 blocks

Input to this process is a 4x4 array c with elements c_{ij} which is either an array relating to a residual block of the luma component or an array relating to a residual block of a chroma component.

Outputs of this process are residual sample values as 4x4 array r with elements r_{ij}.

Depending on the values of qpprime_y_zero_transform_bypass_flag and QP'_Y, the following applies.

– If qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the output r is derived as

$$r_{ij} = c_{ij} \text{ with } i, j = 0..3 \tag{8-333}$$

– Otherwise (qpprime_y_zero_transform_bypass_flag is equal to 0 or QP'_Y is not equal to 0), the following text of this process specifies the output.

The variable bitDepth is derived as follows.

– If the input array c relates to a luma residual block, bitDepth is set equal to $BitDepth_Y$.

– Otherwise (the input array c relates to a chroma residual block), bitDepth is set equal to $BitDepth'_C$.

The bitstream shall not contain data that results in any element c_{ij} of c with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + bitDepth)}$ to $2^{(7 + bitDepth)}-1$, inclusive.

The variable sMbFlag is derived as follows.

- If mb_type is equal to SI or the macroblock prediction mode is equal to Inter in an SP slice, sMbFlag is set equal to 1,
- Otherwise (mb_type not equal to SI and the macroblock prediction mode is not equal to Inter in an SP slice), sMbFlag is set equal to 0.

The variable qP is derived as follows.

– If the input array c relates to a luma residual block and sMbFlag is equal to 0

$$qP = QP'_Y \tag{8-334}$$

– Otherwise, if the input array c relates to a luma residual block and sMbFlag is equal to 1

$$qP = QS_Y \tag{8-335}$$

– Otherwise, if the input array c relates to a chroma residual block and sMbFlag is equal to 0

$$qP = QP'_C \tag{8-336}$$

– Otherwise (the input array c relates to a chroma residual block and sMbFlag is equal to 1),

$$qP = QS_C \tag{8-337}$$

Scaling of 4x4 block transform coefficient levels c_{ij} proceeds as follows.

– If all of the following conditions are true
  – i is equal to 0
  – j is equal to 0
  – c relates to a luma residual block coded using Intra_16x16 prediction mode or c relates to a chroma residual block

the variable d_{00} is derived by

$$d_{00} = c_{00} \tag{8-338}$$

– Otherwise, the following applies.

  – If qP is greater than or equal to 24, the scaled result is derived as follows

$$d_{ij} = (c_{ij} * \text{LevelScale}(qP \,\%\, 6, i, j)) << (qP / 6 - 4), \text{ with } i,j = 0..3 \text{ except as noted above} \tag{8-339}$$

  – Otherwise (qP is less than 24), the scaled result is derived as follows

$$d_{ij} = (c_{ij} * \text{LevelScale}(qP \,\%\, 6, i, j) + 2^{3-qP/6}) >> (4 - qP / 6), \text{ with } i, j = 0..3 \text{ except as noted above} \tag{8-340}$$

The bitstream shall not contain data that results in any element d_{ij} of d with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{bitDepth})}$ to $2^{(7 + \text{bitDepth})} - 1$, inclusive.

The transform process shall convert the block of scaled transform coefficients to a block of output samples in a manner mathematically equivalent to the following.

First, each (horizontal) row of scaled transform coefficients is transformed using a one-dimensional inverse transform as follows.

A set of intermediate values is computed as follows.

$$e_{i0} = d_{i0} + d_{i2}, \quad \text{with} \quad i = 0..3 \tag{8-341}$$

$$e_{i1} = d_{i0} - d_{i2}, \quad \text{with} \quad i = 0..3 \tag{8-342}$$

$$e_{i2} = (d_{i1} >> 1) - d_{i3}, \quad \text{with} \quad i = 0..3 \tag{8-343}$$

$$e_{i3} = d_{i1} + (d_{i3} >> 1), \quad \text{with} \quad i = 0..3 \tag{8-344}$$

The bitstream shall not contain data that results in any element e_{ij} of e with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{bitDepth})}$ to $2^{(7 + \text{bitDepth})} - 1$, inclusive.

Then, the transformed result is computed from these intermediate values as follows.

$$f_{i0} = e_{i0} + e_{i3}, \quad \text{with} \quad i = 0..3 \tag{8-345}$$

$$f_{i1} = e_{i1} + e_{i2}, \quad \text{with} \quad i = 0..3 \tag{8-346}$$

$$f_{i2} = e_{i1} - e_{i2}, \quad \text{with} \quad i = 0..3 \tag{8-347}$$

$$f_{i3} = e_{i0} - e_{i3}, \quad \text{with} \quad i = 0..3 \tag{8-348}$$

The bitstream shall not contain data that results in any element f_{ij} of f with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{bitDepth})}$ to $2^{(7 + \text{bitDepth})} - 1$, inclusive.

Then, each (vertical) column of the resulting matrix is transformed using the same one-dimensional inverse transform as follows.

A set of intermediate values is computed as follows.

$$g_{0j} = f_{0j} + f_{2j}, \quad \text{with} \quad j = 0..3 \tag{8-349}$$

$$g_{1j} = f_{0j} - f_{2j}, \quad \text{with} \quad j = 0..3 \tag{8-350}$$

$$g_{2j} = (f_{1j} >> 1) - f_{3j}, \quad \text{with} \quad j = 0..3 \tag{8-351}$$

$$g_{3j} = f_{1j} + (f_{3j} >> 1), \quad \text{with} \quad j = 0..3 \tag{8-352}$$

The bitstream shall not contain data that results in any element g_{ij} of g with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{bitDepth})}$ to $2^{(7 + \text{bitDepth})} - 1$, inclusive.

Then, the transformed result is computed from these intermediate values as follows.

$$h_{0j} = g_{0j} + g_{3j}, \quad \text{with} \quad j = 0..3 \tag{8-353}$$

$$h_{1j} = g_{1j} + g_{2j}, \quad \text{with} \quad j = 0..3 \tag{8-354}$$

$$h_{2j} = g_{1j} - g_{2j}, \quad \text{with} \quad j = 0..3 \tag{8-355}$$

$$h_{3j} = g_{0j} - g_{3j}, \quad \text{with} \quad j = 0..3 \tag{8-356}$$

The bitstream shall not contain data that results in any element h_{ij} of h with i, j = 0..3 that exceeds the range of integer values from $-2^{(7 + \text{bitDepth})}$ to $2^{(7 + \text{bitDepth})} - 33$, inclusive.

After performing both the one-dimensional horizontal and the one-dimensional vertical inverse transforms to produce an array of transformed samples, the final constructed residual sample values is derived as

$$r_{ij} = (h_{ij} + 2^5) >> 6 \quad \text{with} \quad i, j = 0..3 \tag{8-357}$$

8.5.11 Scaling and transformation process for residual 8x8 luma blocks

Input to this process is an 8x8 array c with elements c_{ij} which is an array relating to an 8x8 residual block of the luma component.

Outputs of this process are residual sample values as 8x8 array r with elements r_{ij}.

Depending on the values of qpprime_y_zero_transform_bypass_flag and QP'_Y, the following applies.

– If qpprime_y_zero_transform_bypass_flag is equal to 1 and QP'_Y is equal to 0, the output r is derived as

$$r_{ij} = c_{ij} \quad \text{with} \quad i, j = 0..7 \tag{8-358}$$

– Otherwise (qpprime_y_zero_transform_bypass_flag is equal to 0 or QP'_Y is not equal to 0), the following text of this process specifies the output.

The bitstream shall not contain data that results in any element c_{ij} of c with i, j = 0..7 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)} - 1$, inclusive.

The scaling process for 8x8 block transform coefficient levels c_{ij} proceeds as follows.

– If QP'_Y is greater than or equal to 36, the scaled result is derived as

$$d_{ij} = (c_{ij} * \text{LevelScale8x8}(QP'_Y \% 6, i, j)) << (QP'_Y / 6 - 6), \text{ with } i,j = 0..7 \tag{8-359}$$

– Otherwise (QP'_Y is less than 36), the scaled result is derived as

$$d_{ij} = (c_{ij} * \text{LevelScale8x8}(QP'_Y \% 6, i, j)) + 2^{5 - QP'_Y/6}) >> (6 - QP'_Y /6), \text{ with } i,j = 0..7 \tag{8-360}$$

The bitstream shall not contain data that results in any element d_{ij} of d with i, j = 0..7 that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)} - 1$, inclusive.

The transform process shall convert the block of scaled transform coefficients to a block of output samples in a manner mathematically equivalent to the following.

First, each (horizontal) row of scaled transform coefficients is transformed using a one-dimensional inverse transform as follows.

– A set of intermediate values e_{ij} is derived by

$$e_{i0} = d_{i0} + d_{i4}, \text{ with } i = 0..7 \quad (8\text{-}361)$$

$$e_{i1} = -d_{i3} + d_{i5} - d_{i7} - (d_{i7} >> 1), \text{ with } i = 0..7 \quad (8\text{-}362)$$

$$e_{i2} = d_{i0} - d_{i4}, \text{ with } i = 0..7 \quad (8\text{-}363)$$

$$e_{i3} = d_{i1} + d_{i7} - d_{i3} - (d_{i3} >> 1), \text{ with } i = 0..7 \quad (8\text{-}364)$$

$$e_{i4} = (d_{i2} >> 1) - d_{i6}, \text{ with } i = 0..7 \quad (8\text{-}365)$$

$$e_{i5} = -d_{i1} + d_{i7} + d_{i5} + (d_{i5} >> 1), \text{ with } i = 0..7 \quad (8\text{-}366)$$

$$e_{i6} = d_{i2} + (d_{i6} >> 1), \text{ with } i = 0..7 \quad (8\text{-}367)$$

$$e_{i7} = d_{i3} + d_{i5} + d_{i1} + (d_{i1} >> 1), \text{ with } i = 0..7 \quad (8\text{-}368)$$

– A second set of intermediate results f_{ij} is computed from the intermediate values e_{ij} as

$$f_{i0} = e_{i0} + e_{i6}, \text{ with } i = 0..7 \quad (8\text{-}369)$$

$$f_{i1} = e_{i1} + (e_{i7} >> 2), \text{ with } i = 0..7 \quad (8\text{-}370)$$

$$f_{i2} = e_{i2} + e_{i4}, \text{ with } i = 0..7 \quad (8\text{-}371)$$

$$f_{i3} = e_{i3} + (e_{i5} >> 2), \text{ with } i = 0..7 \quad (8\text{-}372)$$

$$f_{i4} = e_{i2} - e_{i4}, \text{ with } i = 0..7 \quad (8\text{-}373)$$

$$f_{i5} = (e_{i3} >> 2) - e_{i5}, \text{ with } i = 0..7 \quad (8\text{-}374)$$

$$f_{i6} = e_{i0} - e_{i6}, \text{ with } i = 0..7 \quad (8\text{-}375)$$

$$f_{i7} = e_{i7} - (e_{i1} >> 2), \text{ with } i = 0..7 \quad (8\text{-}376)$$

– Then, the transformed result g_{ij} is computed from these intermediate values f_{ij} as

$$g_{i0} = f_{i0} + f_{i7}, \text{ with } i = 0..7 \quad (8\text{-}377)$$

$$g_{i1} = f_{i2} + f_{i5}, \text{ with } i = 0..7 \quad (8\text{-}378)$$

$$g_{i2} = f_{i4} + f_{i3}, \text{ with } i = 0..7 \quad (8\text{-}379)$$

$$g_{i3} = f_{i6} + f_{i1}, \text{ with } i = 0..7 \quad (8\text{-}380)$$

$$g_{i4} = f_{i6} - f_{i1}, \text{ with } i = 0..7 \quad (8\text{-}381)$$

$$g_{i5} = f_{i4} - f_{i3}, \text{ with } i = 0..7 \quad (8\text{-}382)$$

$$g_{i6} = f_{i2} - f_{i5}, \text{ with } i = 0..7 \quad (8\text{-}383)$$

$$g_{i7} = f_{i0} - f_{i7}, \text{ with } i = 0..7 \quad (8\text{-}384)$$

Then, each (vertical) column of the resulting matrix is transformed using the same one-dimensional inverse transform as follows.

– A set of intermediate values h_{ij} is computed from the horizontally transformed value g_{ij} as

$$h_{0j} = g_{0j} + g_{4j}, \text{ with } j = 0..7 \quad (8\text{-}385)$$

$$h_{1j} = -g_{3j} + g_{5j} - g_{7j} - (g_{7j} >> 1), \text{ with } j = 0..7 \quad (8\text{-}386)$$

$$h_{2j} = g_{0j} - g_{4j}, \text{ with } j = 0..7 \quad (8\text{-}387)$$

$$h_{3j} = g_{1j} + g_{7j} - g_{3j} - (g_{3j} >> 1), \text{ with } j = 0..7 \quad (8\text{-}388)$$

$$h_{4j} = (g_{2j} >> 1) - g_{6j}, \text{ with } j = 0..7 \quad (8\text{-}389)$$

$$h_{5j} = -g_{1j} + g_{7j} + g_{5j} + (g_{5j} >> 1), \text{ with } j = 0..7 \quad (8\text{-}390)$$

$$h_{6j} = g_{2j} + (g_{6j} >> 1), \text{ with } j = 0..7 \quad (8\text{-}391)$$

$$h_{7j} = g_{3j} + g_{5j} + g_{1j} + (g_{1j} >> 1), \text{ with } j = 0..7 \quad (8\text{-}392)$$

– A second set of intermediate results k_{ij} is computed from the intermediate values h_{ij} as

$$k_{0j} = h_{0j} + h_{6j}, \text{ with } j = 0..7 \quad (8\text{-}393)$$

$$k_{1j} = h_{1j} + (h_{7j} >> 2), \text{ with } j = 0..7 \quad (8\text{-}394)$$

$$k_{2j} = h_{2j} + h_{4j}, \text{ with } j = 0..7 \quad (8\text{-}395)$$

$$k_{3j} = h_{3j} + (h_{5j} >> 2), \text{ with } j = 0..7 \quad (8\text{-}396)$$

$$k_{4j} = h_{2j} - h_{4j}, \text{ with } j = 0..7 \quad (8\text{-}397)$$

$$k_{5j} = (h_{3j} >> 2) - h_{5j}, \text{ with } j = 0..7 \quad (8\text{-}398)$$

$$k_{6j} = h_{0j} - h_{6j}, \text{ with } j = 0..7 \quad (8\text{-}399)$$

$$k_{7j} = h_{7j} - (h_{1j} >> 2), \text{ with } j = 0..7 \quad (8\text{-}400)$$

– Then, the transformed result m_{ij} is computed from these intermediate values k_{ij} as

$$m_{0j} = k_{0j} + k_{7j}, \text{ with } j = 0..7 \quad (8\text{-}401)$$

$$m_{1j} = k_{2j} + k_{5j}, \text{ with } j = 0..7 \quad (8\text{-}402)$$

$$m_{2j} = k_{4j} + k_{3j}, \text{ with } j = 0..7 \quad (8\text{-}403)$$

$$m_{3j} = k_{6j} + k_{1j}, \text{ with } j = 0..7 \quad (8\text{-}404)$$

$$m_{4j} = k_{6j} - k_{1j}, \text{ with } j = 0..7 \quad (8\text{-}405)$$

$$m_{5j} = k_{4j} - k_{3j}, \text{ with } j = 0..7 \quad (8\text{-}406)$$

$$m_{6j} = k_{2j} - k_{5j}, \text{ with } j = 0..7 \quad (8\text{-}407)$$

$$m_{7j} = k_{0j} - k_{7j}, \text{ with } j = 0..7 \quad (8\text{-}408)$$

The bitstream shall not contain data that results in any element e_{ij}, f_{ij}, g_{ij}, h_{ij}, or k_{ij} for i and j in the range of 0..7, inclusive, that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)} - 1$, inclusive.

The bitstream shall not contain data that results in any element m_{ij} for i and j in the range of 0..7, inclusive, that exceeds the range of integer values from $-2^{(7 + \text{BitDepth}_Y)}$ to $2^{(7 + \text{BitDepth}_Y)} - 33$, inclusive.

After performing both the one-dimensional horizontal and the one-dimensional vertical inverse transforms to produce an array of transformed samples, the final constructed residual sample values are derived as

$$r_{ij} = (m_{ij} + 2^5) >> 6 \text{ with } i, j = 0..7 \quad (8\text{-}409)$$

8.5.12 Picture construction process prior to deblocking filter process

Inputs to this process are

– luma4x4BlkIdx or chroma4x4BlkIdx or luma8x8BlkIdx

– a sample array u with elements u_{ij} which is either a 4x4 luma block or a 4x4 chroma block or an 8x8 luma block

The position of the upper-left luma sample of the current macroblock is derived by invoking the inverse macroblock scanning process in subclause 6.4.1 with CurrMbAddr as input and the output being assigned to (xP, yP).

When u is a luma block, for each sample u_{ij} of the luma block, the following applies.

– Depending on the size of the block u, the following applies.

  – If u is an 4x4 luma block, the position of the upper-left sample of the 4x4 luma block with index luma4x4BlkIdx inside the macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (xO, yO), and the variable nE is set equal to 4.

  – Otherwise (u is an 8x8 luma block), the position of the upper-left sample of the 8x8 luma block with index luma8x8BlkIdx inside the macroblock is derived by invoking the inverse 8x8 luma block scanning process in subclause 6.4.4 with luma8x8BlkIdx as the input and the output being assigned to (xO, yO), and the variable nE is set equal to 8.

– Depending on the variable MbaffFrameFlag and the current macroblock, the following applies.

  – If MbaffFrameFlag is equal to 1 and the current macroblock is a field macroblock

$$S'_L[xP + xO + j, yP + 2 * (yO + i)] = u_{ij} \quad \text{with } i, j = 0..nE - 1 \quad (8\text{-}410)$$

– Otherwise (MbaffFrameFlag is equal to 0 or the current macroblock is a frame macroblock),

$$S'_L[\ xP + xO + j, yP + yO + i\ ] = u_{ij} \quad \text{with } i, j = 0..nE - 1 \tag{8-411}$$

When u is a chroma block, for each sample u_{ij} of the 4x4 chroma block, the following applies.

– The subscript C in the variable S'_C is replaced with Cb for the Cb chroma component and with Cr for the Cr chroma component.

– Depending on the variable chroma_format_idc, the position of the upper-left sample of a 4x4 chroma block with index chroma4x4BlkIdx inside the macroblock is derived as follows.

– If chroma_format_idc is equal to 1 or 2, the following applies.

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0) \tag{8-412}$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1) \tag{8-413}$$

– Otherwise (chroma_format_idc is equal to 3), the following applies.

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0) \tag{8-414}$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1) \tag{8-415}$$

– Depending on the variable MbaffFrameFlag and the current macroblock, the following applies.

– If MbaffFrameFlag is equal to 1 and the current macroblock is a field macroblock

$$S'_C[\ (xP / \text{subWidthC}) + xO + j, ((yP + \text{SubHeightC} - 1) / \text{SubHeightC}) + 2 * (yO + i)\ ] = u_{ij} \quad \text{with } i, j = 0..3 \tag{8-416}$$

– Otherwise (MbaffFrameFlag is equal to 0 or the current macroblock is a frame macroblock),

$$S'_C[\ (xP/\text{subWidthC}) + xO + j, (yP / \text{SubHeightC}) + yO + i\ ] = u_{ij} \text{ with } i, j = 0..3 \tag{8-417}$$

8.5.13 Residual colour transform process

This process is invoked when residual_colour_transform_flag is equal to 1.

After invoking, this process is suspended until the derivation of $R_{Y,ij}$, $R_{Cb,ij}$, and $R_{Cr,ij}$ has been completed for i, j = 0..ijMax, where ijMax is specified as follows.

– If transform_size_8x8_flag is equal to 0, the variable ijMax is set equal to 3.

– Otherwise (transform_size_8x8_flag is equal to 1), the variable ijMax is set equal to 7.

At the resumption of this process, all values $R_{Y,ij}$, $R_{Cb,ij}$, and $R_{Cr,ij}$ with i, j = 0..ijMax shall be available through prior invocations of the relevant processes specified in subclauses 8.5.1, 8.5.2, 8.5.3, or 8.5.4

For each i, j = 0..ijMax, the residual colour transform is computed as

$$t = R_{Y,ij} - (R_{Cb,ij} >> 1) \tag{8-418}$$

$$R_{G,ij} = t + R_{Cb,ij} \tag{8-419}$$

$$R_{B,ij} = t - (R_{Cr,ij} >> 1) \tag{8-420}$$

$$R_{R,ij} = R_{B,ij} + R_{Cr,ij} \tag{8-421}$$

NOTE – The residual colour transform is similar to the YCgCo transformation specified in Equations E-30 through E-33. However, the residual colour transform operates on the decoded residual difference data within the decoding process rather than operating as a post-processing step that is outside the decoding process specified in this Recommendation | International Standard.

8.6 Decoding process for P macroblocks in SP slices or SI macroblocks

This process is invoked when decoding P macroblock types in an SP slice type or an SI macroblock type in SI slices.

Inputs to this process are the prediction residual transform coefficient levels and the predicted samples for the current macroblock.

Outputs of this process are the decoded samples of the current macroblock prior to the deblocking filter process.

This subclause specifies the transform coefficient decoding process and picture construction process for P macroblock types in SP slices and SI macroblock type in SI slices.

NOTE – SP slices make use of Inter predictive coding to exploit temporal redundancy in the sequence, in a similar manner to P slice coding. Unlike P slice coding, however, SP slice coding allows identical reconstruction of a slice even when different reference pictures are being used. SI slices make use of spatial prediction, in a similar manner to I slices. SI slice coding allows identical reconstruction to a corresponding SP slice. The properties of SP and SI slices aid in providing functionalities for bitstream switching, splicing, random access, fast-forward, fast reverse, and error resilience/recovery.

An SP slice consists of macroblocks coded either as I macroblock types or P macroblock types.

An SI slice consists of macroblocks coded either as I macroblock types or SI macroblock type.

The transform coefficient decoding process and picture construction process prior to deblocking filter process for I macroblock types in SI slices is invoked as specified in subclause 8.5. SI macroblock type is decoded as described below.

When the current macroblock is coded as P_Skip, all values of LumaLevel, ChromaDCLevel, ChromaACLevel are set equal to 0 for the current macroblock.

8.6.1 SP decoding process for non-switching pictures

This process is invoked, when decoding P macroblock types in SP slices in which sp_for_switch_flag is equal to 0.

Inputs to this process are Inter prediction samples for the current macroblock from subclause 8.4 and the prediction residual transform coefficient levels.

Outputs of this process are the decoded samples of the current macroblock prior to the deblocking filter process.

This subclause applies to all macroblocks in SP slices in which sp_for_switch_flag is equal to 0, except those with macroblock prediction mode equal to Intra_4x4 or Intra_16x16. It does not apply to SI slices.

8.6.1.1 Luma transform coefficient decoding process

Inputs to this process are Inter prediction luma samples for the current macroblock $pred_L$ from subclause 8.4 and the prediction residual transform coefficient levels, LumaLevel, and the index of the 4x4 luma block luma4x4BlkIdx.

The position of the upper-left sample of the 4x4 luma block with index luma4x4BlkIdx inside the current macroblock is derived by invoking the inverse 4x4 luma block scanning process in subclause 6.4.3 with luma4x4BlkIdx as the input and the output being assigned to (x, y).

Let the variable p be a 4x4 array of prediction samples with element p_{ij} being derived as follows.

$$p_{ij} = pred_L[\, x + j, y + i \,] \quad \text{with } i, j = 0..3 \tag{8-422}$$

The variable p is transformed producing transform coefficients c^p according to:

$$c^p = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \tag{8-423}$$

The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with LumaLevel[luma4x4BlkIdx] as the input and the two-dimensional array c^r with elements c_{ij}^r as the output.

The prediction residual transform coefficients c^r are scaled using quantisation parameter QP_Y, and added to the transform coefficients of the prediction block c^p with i, j = 0..3 as follows.

$$c_{ij}^s = c_{ij}^p + (((c_{ij}^r * \text{LevelScale}(QP_Y \% 6, i, j) * A_{ij}) << (QP_Y / 6)) >> 10) \quad (8\text{-}424)$$

where LevelScale(m, i, j) is specified in Equation 8-312, and where A_{ij} is specified as:

$$A_{ij} = \begin{cases} 16 & \text{for} \quad (i,j) \in \{(0,0),(0,2),(2,0),(2,2)\}, \\ 25 & \text{for} \quad (i,j) \in \{(1,1),(1,3),(3,1),(3,3)\}, \\ 20 & \text{otherwise;} \end{cases} \quad (8\text{-}425)$$

The function LevelScale2(m, i, j), used in the formulas below, is specified as:

$$\text{LevelScale2}(m,i,j) = \begin{cases} w_{m0} & \text{for} \quad (i,j) \in \{(0,0),(0,2),(2,0),(2,2)\}, \\ w_{m1} & \text{for} \quad (i,j) \in \{(1,1),(1,3),(3,1),(3,3)\}, \\ w_{m2} & \text{otherwise;} \end{cases} \quad (8\text{-}426)$$

where the first and second subscripts of w are row and column indices, respectively, of the matrix specified as:

$$w = \begin{bmatrix} 13107 & 5243 & 8066 \\ 11916 & 4660 & 7490 \\ 10082 & 4194 & 6554 \\ 9362 & 3647 & 5825 \\ 8192 & 3355 & 5243 \\ 7282 & 2893 & 4559 \end{bmatrix} \quad (8\text{-}427)$$

The resulting sum, c^s, is quantised with a quantisation parameter QS_Y and with i, j = 0..3 as follows.

$$c_{ij} = \text{Sign}(c_{ij}^s) * ((\text{Abs}(c_{ij}^s) * \text{LevelScale2}(QS_Y \% 6, i, j) + (1 << (14 + QS_Y / 6))) >> (15 + QS_Y / 6)) \quad (8\text{-}428)$$

The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c as the input and r as the output.

The 4x4 array u with elements u_{ij} is derived as follows.

$$u_{ij} = \text{Clip1}_Y(r_{ij}) \text{ with } i, j = 0..3 \quad (8\text{-}429)$$

The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with luma4x4BlkIdx and u as the inputs.

8.6.1.2 Chroma transform coefficient decoding process

Inputs to this process are Inter prediction chroma samples for the current macroblock from subclause 8.4 and the prediction residual transform coefficient levels, ChromaDCLevel and ChromaACLevel.

This process is invoked twice: once for the Cb component and once for the Cr component. The component is referred to by replacing C with Cb for the Cb component and C with Cr for the Cr component. Let iCbCr select the current chroma component.

For each 4x4 block of the current chroma component indexed using chroma4x4BlkIdx with chroma4x4BlkIdx equal to 0..3, the following applies.

– The position of the upper-left sample of a 4x4 chroma block with index chroma4x4BlkIdx inside the macroblock is derived as follows.

x = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 0) (8-430)

$$y = InverseRasterScan(chroma4x4BlkIdx, 4, 4, 8, 1) \quad (8\text{-}431)$$

– Let p be a 4x4 array of prediction samples with elements p_{ij} being derived as follows.

$$p_{ij} = pred_C[x + j, y + i] \quad \text{with } i, j = 0..3 \quad (8\text{-}432)$$

– The 4x4 array p is transformed producing transform coefficients c^p(chroma4x4BlkIdx) using Equation 8-423.

– The variable chromaList, which is a list of 16 entries, is derived. chromaList[0] is set equal to 0. chromaList[k] with index k = 1..15 are specified as follows.

$$chromaList[k] = ChromaACLevel[iCbCr][chroma4x4BlkIdx][k - 1] \quad (8\text{-}433)$$

– The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with chromaList as the input and the 4x4 array c^r as the output.

– The prediction residual transform coefficients c^r are scaled using quantisation parameter QPC, and added to the transform coefficients of the prediction block c^p with i, j = 0..3 except for the combination i = 0, j = 0 as follows.

$$c_{ij}^s = c_{ij}^p(chroma4x4BlkIdx) + (((c_{ij}^r * LevelScale(QP_C \% 6, i, j) * A_{ij}) << (QP_C / 6)) >> 10) \quad (8\text{-}434)$$

– The resulting sum, c^s, is quantised with a quantisation parameter QS_C and with i, j = 0..3 except for the combination i = 0, j = 0 as follows. The derivation of c_{00}(chroma4x4BlkIdx) is described below in this subclause.

$$c_{ij}(chroma4x4BlkIdx) = (Sign(c_{ij}^s) * (Abs(c_{ij}^s) * LevelScale2(QS_C \% 6, i, j) + (1 << (14 + QS_C / 6)))) >> (15 + QS_C / 6) \quad (8\text{-}435)$$

– The scaling and transformation process for residual 4x4 blocks as specified in 8.5.10 is invoked with c(chroma4x4BlkIdx) as the input and r as the output.

– The 4x4 array u with elements u_{ij} is derived as follows.

$$u_{ij} = Clip1_C(r_{ij}) \text{ with } i, j = 0..3 \quad (8\text{-}436)$$

– The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with chroma4x4BlkIdx and u as the inputs.

The derivation of the DC transform coefficient level c_{00}(chroma4x4BlkIdx) is specified as follows. The DC transform coefficients of the 4 prediction chroma 4x4 blocks of the current component of the macroblock are assembled into a 2x2 matrix with elements c_{00}^p(chroma4x4BlkIdx) and a 2x2 transform is applied to the DC transform coefficients as follows

$$dc^p = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00}^p(0) & c_{00}^p(1) \\ c_{00}^p(2) & c_{00}^p(3) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (8\text{-}437)$$

The chroma DC prediction residual transform coefficient levels, ChromaDCLevel[iCbCr][k] with k = 0..3 are scaled using quantisation parameter QP, and added to the prediction DC transform coefficients as follows.

$$dc_{ij}^s = dc_{ij}^p + (((ChromaDCLevel[iCbCr][j * 2 + i] * LevelScale(QP_C \% 6, 0, 0) * A_{00}) << (QP_C / 6)) >> 9) \quad \text{with } i, j = 0, 1 \quad (8\text{-}438)$$

The 2x2 array dc^s, is quantised using the quantisation parameter QS_C as follows.

$$dc_{ij}^r = (Sign(dc_{ij}^s) * (Abs(dc_{ij}^s) * LevelScale2(QS_C \% 6, 0, 0) + (1 << (15 + QS_C / 6)))) >> (16 + QS_C / 6) \quad \text{with } i, j = 0, 1 \quad (8\text{-}439)$$

The 2x2 array f with elements f_{ij} and i, j = 0..1 is derived as follows.

$$f = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} dc^{r}_{00} & dc^{r}_{01} \\ dc^{r}_{10} & dc^{r}_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8\text{-}440)$$

Scaling of the elements f_{ij} of f is performed as follows.

$$c_{00}(j * 2 + i) = ((f_{ij} * LevelScale(QS_C \% 6, 0, 0)) << (QS_C / 6)) >> 5 \text{ with i, j = 0, 1} \quad (8\text{-}441)$$

8.6.2 SP and SI slice decoding process for switching pictures

This process is invoked, when decoding P macroblock types in SP slices in which sp_for_switch_flag is equal to 1 and when decoding SI macroblock type in SI slices.

Inputs to this process are the prediction residual transform coefficient levels and the prediction sample arrays $pred_L$, $pred_{Cb}$ and $pred_{Cr}$ for the current macroblock.

8.6.2.1 Luma transform coefficient decoding process

Inputs to this process are prediction luma samples $pred_L$ and the luma prediction residual transform coefficient levels, LumaLevel.

The 4x4 array p with elements p_{ij} with i, j = 0..3 is derived as in subclause 8.6.1.1, is transformed according to Equation 8-423 to produce transform coefficients c^p. These transform coefficients are then quantised with the quantisation parameter QS_Y, as follows:

$$c_{ij}^{s} = Sign(c_{ij}^{p}) * ((Abs(c_{ij}^{p}) * LevelScale2(QS_Y \% 6, i, j) + (1 << (14 + QS_Y / 6))) >> (15 + QS_Y / 6)) \text{ with i, j = 0..3} \quad (8\text{-}442)$$

The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with LumaLevel[luma4x4BlkIdx] as the input and the two-dimensional array c^r with elements c_{ij}^r as the output.

The 4x4 array c with elements c_{ij} with i, j = 0..3 is derived as follows.

$$c_{ij} = c_{ij}^{r} + c_{ij}^{s} \text{ with i, j = 0..3} \quad (8\text{-}443)$$

The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c as the input and r as the output.

The 4x4 array u with elements u_{ij} is derived as follows.

$$u_{ij} = Clip1_Y(r_{ij}) \text{ with i, j = 0..3} \quad (8\text{-}444)$$

The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with luma4x4BlkIdx and u as the inputs.

8.6.2.2 Chroma transform coefficient decoding process

Inputs to this process are predicted chroma samples for the current macroblock from subclause 8.4 and the prediction residual transform coefficient levels, ChromaDCLevel and ChromaACLevel.

This process is invoked twice: once for the Cb component and once for the Cr component. The component is referred to by replacing C with Cb for the Cb component and C with Cr for the Cr component. Let iCbCr select the current chroma component.

For each 4x4 block of the current chroma component indexed using chroma4x4BlkIdx with chroma4x4BlkIdx equal to 0..3, the following applies.

1. The 4x4 array p with elements p_{ij} with i, j = 0..3 is derived as in subclause 8.6.1.2, is transformed according to Equation 8-423 to produce transform coefficients c^p(chroma4x4BlkIdx). These transform coefficients are then quantised with the quantisation parameter QS_C, with i, j = 0..3 except for the combination i = 0, j = 0 as follows. The processing of c_{00}^p(chroma4x4BlkIdx) is described below in this subclause.

$$c_{ij}^{s} = (Sign(c_{ij}^{p}(chroma4x4BlkIdx)) * (Abs(c_{ij}^{p}(chroma4x4BlkIdx)) * LevelScale2(QS_C \% 6, i, j) + (1 << (14 + QS_C / 6)))) >> (15 + QS_C / 6) \quad (8\text{-}445)$$

– The variable chromaList, which is a list of 16 entries, is derived. chromaList[0] is set equal to 0. chromaList[k] with index k = 1..15 are specified as follows.

$$chromaList[k] = ChromaACLevel[iCbCr][chroma4x4BlkIdx][k - 1] \quad (8\text{-}446)$$

– The inverse transform coefficient scanning process as described in subclause 8.5.5 is invoked with chromaList as the input and the two-dimensional array c^{r}(chroma4x4BlkIdx) with elements c_{ij}^{r}(chroma4x4BlkIdx) as the output.

– The 4x4 array c(chroma4x4BlkIdx) with elements c_{ij}(chroma4x4BlkIdx) with i, j = 0..3 except for the combination i = 0, j = 0 is derived as follows. The derivation of c_{00}(chroma4x4BlkIdx) is described below.

$$c_{ij}(chroma4x4BlkIdx) = c_{ij}^{r}(chroma4x4BlkIdx) + c_{ij}^{s} \quad (8\text{-}447)$$

– The scaling and transformation process for residual 4x4 blocks as specified in subclause 8.5.10 is invoked with c(chroma4x4BlkIdx) as the input and r as the output.

– The 4x4 array u with elements u_{ij} is derived as follows.

$$u_{ij} = Clip1_C(r_{ij}) \text{ with } i, j = 0..3 \quad (8\text{-}448)$$

– The picture construction process prior to deblocking filter process in subclause 8.5.12 is invoked with chroma4x4BlkIdx and u as the inputs.

The derivation of the DC transform coefficient level c_{00}(chroma4x4BlkIdx) is specified as follows. The DC transform coefficients of the 4 prediction 4x4 chroma blocks of the current component of the macroblock, c_{00}^{p}(chroma4x4BlkIdx), are assembled into a 2x2 matrix, and a 2x2 transform is applied to the DC transform coefficients of these blocks according to Equation 8-437 resulting in DC transform coefficients dc_{ij}^{p}.

These DC transform coefficients are then quantised with the quantisation parameter QS_C, as given by:

$$dc_{ij}^{s} = (Sign(dc_{ij}^{p}) * (Abs(dc_{ij}^{p}) * LevelScale2(QS_C \% 6, 0, 0) + (1 << (15 + QS_C / 6)))) >> (16 + QS_C / 6) \text{ with } i, j = 0, 1 \quad (8\text{-}449)$$

The parsed chroma DC prediction residual transform coefficients, ChromaDCLevel[iCbCr][k] with k = 0..3 are added to these quantised DC transform coefficients of the prediction block, as given by:

$$dc_{ij}^{r} = dc_{ij}^{s} + ChromaDCLevel[iCbCr][j * 2 + i] \text{ with } i, j = 0, 1 \quad (8\text{-}450)$$

The 2x2 array f with elements f_{ij} and i, j = 0..1 is derived using Equation 8-440.

The 2x2 array f with elements f_{ij} and i, j = 0..1 is copied as follows.

$$c_{00}(j * 2 + i) = f_{ij} \text{ with } i, j = 0, 1 \quad (8\text{-}451)$$

8.7 Deblocking filter process

A conditional filtering process is applied to all NxN (where N = 4 or N = 8 for luma, and N = 4 for chroma) block edges of a picture, except edges at the boundary of the picture and any edges for which the deblocking filter process is disabled by disable_deblocking_filter_idc, as specified below. This filtering process is performed on a macroblock basis after the completion of the picture construction process prior to deblocking filter process (as specified in subclauses 8.5 and 8.6) for the entire decoded picture, with all macroblocks in a picture processed in order of increasing macroblock addresses.

NOTE 1 – Prior to the operation of the deblocking filter process for each macroblock, the deblocked samples of the macroblock or macroblock pair above (if any) and the macroblock or macroblock pair to the left (if any) of the current macroblock are always available because the deblocking filter process is performed after the completion of the picture construction process prior to deblocking filter process for the entire decoded picture. However, for purposes of determining which edges are to be filtered

when disable_deblocking_filter_idc is equal to 2, macroblocks in different slices are considered not available during specified steps of the operation of the deblocking filter process.

The deblocking filter process is invoked for the luma and chroma components separately. For each macroblock and each component, vertical edges are filtered first, starting with the edge on the left-hand side of the macroblock proceeding through the edges towards the right-hand side of the macroblock in their geometrical order, and then horizontal edges are filtered, starting with the edge on the top of the macroblock proceeding through the edges towards the bottom of the macroblock in their geometrical order. Figure 8-10 shows edges of a macroblock which can be interpreted as luma or chroma edges.

When interpreting the edges in Figure 8-10 as luma edges, depending on the transform_size_8x8_flag, the following applies.

– If transform_size_8x8_flag is equal to 0, both types, the solid bold and dashed bold luma edges are filtered.

– Otherwise (transform_size_8x8_flag is equal to 1), only the solid bold luma edges are filtered.

When interpreting the edges in Figure 8-10 as chroma edges, depending on chroma_format_idc, the following applies.

– If chroma_format_idc is equal to 1 (4:2:0 format), only the solid bold chroma edges are filtered.

– Otherwise, if chroma_format_idc is equal to 2 (4:2:2 format), the solid bold vertical chroma edges are filtered and both types, the solid bold and dashed bold horizontal chroma edges are filtered.

– Otherwise, if chroma_format_idc is equal to 3 (4:4:4 format), both types, the solid bold and dashed bold chroma edges are filtered.

– Otherwise (chroma_format_idc is equal to 0 (monochrome)), no chroma edges are filtered.

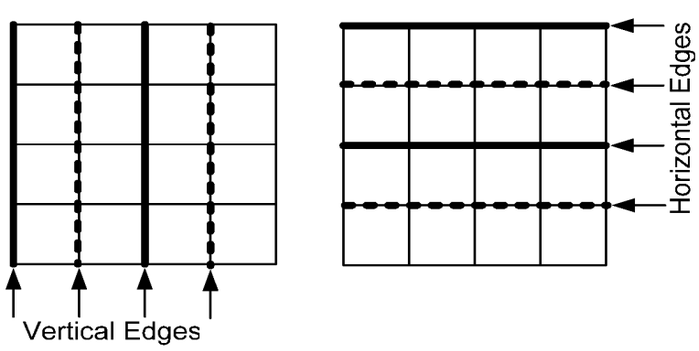


Figure 8-10 – Boundaries in a macroblock to be filtered

For the current macroblock address CurrMbAddr proceeding over values 0..PicSizeInMbs – 1, the following applies.

1. The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.
2. The variables fieldModeMbFlag, filterInternalEdgesFlag, filterLeftMbEdgeFlag and filterTopMbEdgeFlag are derived as follows.
   - The variable fieldModeMbFlag is derived as follows.
     - If any of the following conditions is true, fieldModeMbFlag is set equal to 1.
       - field_pic_flag is equal to 1
       - MbaffFrameFlag is equal to 1 and the macroblock CurrMbAddr is a field macroblock
     - Otherwise, fieldModeMbFlag is set equal to 0.
   - The variable filterInternalEdgesFlag is derived as follows.

- If disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is equal to 1, the variable filterInternalEdgesFlag is set equal to 0;
- Otherwise (disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is not equal to 1), the variable filterInternalEdgesFlag is set equal to 1.

- The variable filterLeftMbEdgeFlag is derived as follows.
  - If any of the following conditions is true, the variable filterLeftMbEdgeFlag is set equal to 0.
    - MbaffFrameFlag is equal to 0 and CurrMbAddr % PicWidthInMbs is equal to 0.
    - MbaffFrameFlag is equal to 1 and (CurrMbAddr >> 1) % PicWidthInMbs is equal to 0
    - disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is equal to 1
    - disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is equal to 2 and the macroblock mbAddrA is not available
  - Otherwise, the variable filterLeftMbEdgeFlag is set equal to 1.
- The variable filterTopMbEdgeFlag is derived as follows.
  - If any of the following conditions is true, the variable filterTopMbEdgeFlag is set equal to 0.
    - MbaffFrameFlag is equal to 0 and CurrMbAddr is less than PicWidthInMbs.
    - MbaffFrameFlag is equal to 1, (CurrMbAddr >> 1) is less than PicWidthInMbs, and the macroblock CurrMbAddr is a field macroblock.
    - MbaffFrameFlag is equal to 1, (CurrMbAddr >> 1) is less than PicWidthInMbs, the macroblock CurrMbAddr is a frame macroblock, and CurrMbAddr % 2 is equal to 0.
    - disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is equal to 1
    - disable_deblocking_filter_idc for the slice that contains the macroblock CurrMbAddr is equal to 2 and the macroblock mbAddrB is not available
  - Otherwise, the variable filterTopMbEdgeFlag is set equal to 1.

3. Given the variables fieldModeMbFlag, filterInternalEdgesFlag, filterLeftMbEdgeFlag and filterTopMbEdgeFlag the deblocking filtering is controlled as follows.
   - When filterLeftMbEdgeFlag is equal to 1, the filtering of the left vertical luma edge is specified as follows.
     - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (0, k)$ with k = 0..15 as input and S'_L as output.
   - When filterInternalEdgesFlag is equal to 1, the filtering of the internal vertical luma edges is specified as follows.
     - When transform_size_8x8_flag is equal to 0, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (4, k)$ with k = 0..15 as input and S'_L as output.
     - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (8, k)$ with k = 0..15 as input and S'_L as output.
     - When transform_size_8x8_flag is equal to 0, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (12, k)$ with k = 0..15 as input and S'_L as output.
   - When filterTopMbEdgeFlag is equal to 1, the filtering of the top horizontal luma edge is specified as follows.
     - If MbaffFrameFlag is equal to 1, (CurrMbAddr % 2) is equal to 0, CurrMbAddr is greater than or equal to 2 * PicWidthInMbs, the macroblock CurrMbAddr is a frame macroblock, and the macroblock (CurrMbAddr - 2 * PicWidthInMbs + 1) is a field macroblock, the following applies.
       - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = 1, and $(xE_k, yE_k) = (k, 0)$ with k = 0..15 as input and S'_L as output.

- The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = 1, and $(xE_k, yE_k) = (k, 1)$ with k = 0..15 as input and S'_L as output.

- Otherwise, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (k, 0)$ with k = 0..15 as input and S'_L as output.

- When filterInternalEdgesFlag is equal to 1, the filtering of the internal horizontal luma edges is specified as follows.

  - When transform_size_8x8_flag is equal to 0, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (k, 4)$ with k = 0..15 as input and S'_L as output.

  - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (k, 8)$ with k = 0..15 as input and S'_L as output.

  - When transform_size_8x8_flag is equal to 0, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 0, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (k, 12)$ with k = 0..15 as input and S'_L as output.

- For the filtering of both chroma components with iCbCr = 0 for Cb and iCbCr = 1 for Cr, the following applies.

  - When filterLeftMbEdgeFlag is equal to 1, the filtering of the left vertical chroma edge is specified as follows.

  - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (0, k)$ with k = 0..MbHeightC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

  - When filterInternalEdgesFlag is equal to 1, the filtering of the internal vertical chroma edge is specified as follows.

    - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (4, k)$ with k = 0..MbHeightC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

    - When chroma_format_idc is equal to 3, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (8, k)$ with k = 0..MbHeightC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

    - When chroma_format_idc is equal to 3, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 1, fieldModeFilteringFlag = fieldModeMbFlag, and $(xE_k, yE_k) = (12, k)$ with k = 0..MbHeightC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

  - When filterTopMbEdgeFlag is equal to 1, the filtering of the top horizontal chroma edge is specified as follows.

    - If MbaffFrameFlag is equal to 1, (CurrMbAddr % 2) is equal to 0, CurrMbAddr is greater than or equal to 2 * PicWidthInMbs, the macroblock CurrMbAddr is a frame macroblock, and the macroblock (CurrMbAddr – 2 * PicWidthInMbs + 1) is a field macroblock, the following applies.

      - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = 1, and $(xE_k, yE_k) = (k, 0)$ with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

      - The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = 1, and $(xE_k, yE_k) = (k, 1)$ with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

– Otherwise, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and (xE_k, yE_k) = (k, 0) with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

– When filterInternalEdgesFlag is equal to 1, the filtering of the internal horizontal chroma edge is specified as follows.

  – The process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and (xE_k, yE_k) = (k, 4) with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

  – When chroma_format_idc is not equal to 1, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and (xE_k, yE_k) = (k, 8) with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

  – When chroma_format_idc is not equal to 1, the process specified in subclause 8.7.1 is invoked with chromaEdgeFlag = 1, iCbCr, verticalEdgeFlag = 0, fieldModeFilteringFlag = fieldModeMbFlag, and (xE_k, yE_k) = (k, 12) with k = 0..MbWidthC - 1 as input and S'_C with C being replaced by Cb for iCbCr = 0 and C being replaced by Cr for iCbCr = 1 as output.

  NOTE 2 – When field mode filtering (fieldModeFilteringFlag is equal to 1) is applied across the top horizontal edges of a frame macroblock, this vertical filtering across the top or bottom macroblock boundary may involve some samples that extend across an internal block edge that is also filtered internally in frame mode.

  NOTE 3 – For example, in 4:2:0 chroma format when transform_size_8x8_flag is equal to 0, the following applies. 3 horizontal luma edges, 1 horizontal chroma edge for Cb, and 1 horizontal chroma edge for Cr are filtered that are internal to a macroblock. When field mode filtering (fieldModeFilteringFlag is equal to 1) is applied to the top edges of a frame macroblock, 2 horizontal luma, 2 horizontal chroma edges for Cb, and 2 horizontal chroma edges for Cr between the frame macroblock and the above macroblock pair are filtered using field mode filtering, for a total of up to 5 horizontal luma edges, 3 horizontal chroma edges for Cb, and 3 horizontal chroma edges for Cr filtered that are considered to be controlled by the frame macroblock. In all other cases, at most 4 horizontal luma, 2 horizontal chroma edges for Cb, and 2 horizontal chroma edges for Cr are filtered that are considered to be controlled by a particular macroblock.

Finally, the arrays S'_L, S'_{Cb}, S'_{Cr} are assigned to the arrays S_L, S_{Cb}, S_{Cr} (which represent the decoded picture), respectively.

8.7.1 Filtering process for block edges

Inputs to this process are chromaEdgeFlag, the chroma component index iCbCr (when chromaEdgeFlag is equal to 1), verticalEdgeFlag, fieldModeFilteringFlag, and a set of nE sample locations (xE_k, yE_k), with k = 0..nE - 1, expressed relative to the upper left corner of the macroblock CurrMbAddr. The set of sample locations (xE_k, yE_k) represent the sample locations immediately to the right of a vertical edge (when verticalEdgeFlag is equal to 1) or immediately below a horizontal edge (when verticalEdgeFlag is equal to 0).

The variable nE is derived as follows.

– If chromaEdgeFlag is equal to 0, nE is set equal to 16.

– Otherwise (chromaEdgeFlag is equal to 1), nE is set equal to (verticalEdgeFlag = = 1) ? MbHeightC : MbWidthC.

Let s' be a variable specifying a luma or chroma sample array, be derived as follows.

– If chromaEdgeFlag is equal to 0, s' represents the luma sample array S'_L of the current picture.

– Otherwise, if chromaEdgeFlag is equal to 1 and iCbCr is equal to 0, s' represents the chroma sample array S'_{Cb} of the chroma component Cb of the current picture.

– Otherwise (chromaEdgeFlag is equal to 1 and iCbCr is equal to 1), s' represents the chroma sample array S'_{Cr} of the chroma component Cr of the current picture.

The variable dy is derived as follows.

– If fieldModeFilteringFlag is equal to 1 and MbaffFrameFlag is equal to 1, dy is set equal to 2.

– Otherwise (fieldModeFilteringFlag is equal to 0 or MbaffFrameFlag is equal to 0), dy is set equal to 1.

The position of the upper-left luma sample of the macroblock CurrMbAddr is derived by invoking the inverse macroblock scanning process in subclause 6.4.1 with mbAddr = CurrMbAddr as input and the output being assigned to (xI, yI).

The variables xP and yP are derived as follows.

– If chromaEdgeFlag is equal to 0, xP is set equal to xI and yP is set equal to yI.

– Otherwise (chromaEdgeFlag is equal to 1), xP is set equal to xI / SubWidthC and yP is set equal to (yI + SubHeightC – 1) / SubHeightC.

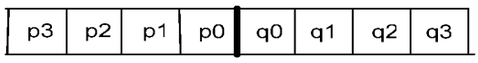


Figure 8-11 – Convention for describing samples across a 4x4 block horizontal or vertical boundary

For each sample location (xE_k, yE_k), k = 0 .. nE - 1, the following applies.

– The filtering process is applied to a set of eight samples across a 4x4 block horizontal or vertical edge denoted as p_i and q_i with i = 0..3 as shown in Figure 8-11 with the edge lying between p_0 and q_0. p_i and q_i with i = 0..3 are specified as follows.

  – If verticalEdgeFlag is equal to 1,

$$q_i = s'[\ xP + xE_k + i, yP + dy * yE_k\ ] \quad (8\text{-}452)$$

$$p_i = s'[\ xP + xE_k - i - 1, yP + dy * yE_k\ ] \quad (8\text{-}453)$$

  – Otherwise (verticalEdgeFlag is equal to 0),

$$q_i = s'[\ xP + xE_k, yP + dy * (\ yE_k + i\ ) - (yE_k \% 2\ )\ ] \quad (8\text{-}454)$$

$$p_i = s'[\ xP + xE_k, yP + dy * (\ yE_k - i - 1\ ) - (yE_k \% 2\ )\ ] \quad (8\text{-}455)$$

– The process specified in subclause 8.7.2 is invoked with the sample values p_i and q_i (i = 0..3), chromaEdgeFlag, verticalEdgeFlag, and fieldModeFilteringFlag as input, and the output is assigned to the filtered result sample values p'_i and q'_i with i = 0..2.

– The input sample values p_i and q_i with i = 0..2 are replaced by the corresponding filtered result sample values p'_i and q'_i with i = 0..2 inside the sample array s' as follows.

  – If verticalEdgeFlag is equal to 1,

$$s'[\ xP + xE_k + i, yP + dy * yE_k\ ] = q'_i \quad (8\text{-}456)$$

$$s'[\ xP + xE_k - i - 1, yP + dy * yE_k\ ] = p'_i \quad (8\text{-}457)$$

  – Otherwise (verticalEdgeFlag is equal to 0),

$$s'[\ xP + xE_k, yP + dy * (\ yE_k + i\ ) - (\ yE_k \% 2\ )\ ] = q'_i \quad (8\text{-}458)$$

$$s'[\ xP + xE_k, yP + dy * (\ yE_k - i - 1\ ) - (\ yE_k \% 2\ )\ ] = p'_i \quad (8\text{-}459)$$

8.7.2 Filtering process for a set of samples across a horizontal or vertical block edge

Inputs to this process are the input sample values p_i and q_i with i in the range of 0..3 of a single set of samples across an edge that is to be filtered, chromaEdgeFlag, verticalEdgeFlag, and fieldModeFilteringFlag.

Outputs of this process are the filtered result sample values p'_i and q'_i with i in the range of 0..2.

The content dependent boundary filtering strength variable bS is derived as follows.

– If chromaEdgeFlag is equal to 0, the derivation process for the content dependent boundary filtering strength specified in subclause 8.7.2.1 is invoked with p_0, q_0, and verticalEdgeFlag as input, and the output is assigned to bS.

– Otherwise (chromaEdgeFlag is equal to 1), the bS used for filtering a set of samples of a horizontal or vertical chroma edge is set equal to the value of bS for filtering the set of samples of a horizontal or vertical luma edge, respectively, that contains the luma sample at location (SubWidthC * x, SubHeightC * y) inside the luma array of the same field, where (x, y) is the location of the chroma sample q_0 inside the chroma array for that field.

The process specified in subclause 8.7.2.2 is invoked with p_0, q_0, p_1, q_1, chromaEdgeFlag, and bS as input, and the output is assigned to filterSamplesFlag, indexA, α, and β.

Depending on the variable filterSamplesFlag, the following applies.

– If filterSamplesFlag is equal to 1, the following applies.

  – If bS is less than 4, the process specified in subclause 8.7.2.3 is invoked with p_i and q_i (i = 0..2), chromaEdgeFlag, bS, β, and indexA given as input, and the output is assigned to p'_i and q'_i (i = 0..2).

  – Otherwise (bS is equal to 4), the process specified in subclause 8.7.2.4 is invoked with p_i and q_i (i = 0..3), chromaEdgeFlag, α, and β given as input, and the output is assigned to p'_i and q'_i (i = 0..2).

– Otherwise (filterSamplesFlag is equal to 0), the filtered result samples p'_i and q'_i (i = 0..2) are replaced by the corresponding input samples p_i and q_i:

$$\text{for } i = 0..2, \quad p'_i = p_i \tag{8-460}$$

$$\text{for } i = 0..2, \quad q'_i = q_i \tag{8-461}$$

8.7.2.1 Derivation process for the luma content dependent boundary filtering strength

Inputs to this process are the input sample values p_0 and q_0 of a single set of samples across an edge that is to be filtered and verticalEdgeFlag.

Output of this process is the variable bS.

Let the variable mixedModeEdgeFlag be derived as follows.

– If MbaffFrameFlag is equal to 1 and the samples p_0 and q_0 are in different macroblock pairs, one of which is a field macroblock pair and the other is a frame macroblock pair, mixedModeEdgeFlag is set equal to 1

– Otherwise, mixedModeEdgeFlag is set equal to 0.

The variable bS is derived as follows.

– If the block edge is also a macroblock edge and any of the following conditions are true, a value of bS equal to 4 is the output:

  – the samples p_0 and q_0 are both in frame macroblocks and either or both of the samples p_0 or q_0 is in a macroblock coded using an Intra macroblock prediction mode

  – the samples p_0 and q_0 are both in frame macroblocks and either or both of the samples p_0 or q_0 is in a macroblock that is in a slice with slice_type equal to SP or SI

  – MbaffFrameFlag is equal to 1 or field_pic_flag is equal to 1, and verticalEdgeFlag is equal to 1, and either or both of the samples p_0 or q_0 is in a macroblock coded using an Intra macroblock prediction mode

  – MbaffFrameFlag is equal to 1 or field_pic_flag is equal to 1, and verticalEdgeFlag is equal to 1, and either or both of the samples p_0 or q_0 is in a macroblock that is in a slice with slice_type equal to SP or SI

- Otherwise, if any of the following conditions are true, a value of bS equal to 3 is the output:
  - mixedModeEdgeFlag is equal to 0 and either or both of the samples p_0 or q_0 is in a macroblock coded using an Intra macroblock prediction mode
  - mixedModeEdgeFlag is equal to 0 and either or both of the samples p_0 or q_0 is in a macroblock that is in a slice with slice_type equal to SP or SI
  - mixedModeEdgeFlag is equal to 1, verticalEdgeFlag is equal to 0, and either or both of the samples p_0 or q_0 is in a macroblock coded using an Intra macroblock prediction mode
  - mixedModeEdgeFlag is equal to 1, verticalEdgeFlag is equal to 0, and either or both of the samples p_0 or q_0 is in a macroblock that is in a slice with slice_type equal to SP or SI
- Otherwise, if the following condition is true, a value of bS equal to 2 is the output:
  - the luma block containing sample p_0 or the luma block containing sample q_0 contains non-zero transform coefficient levels
- Otherwise, if any of the following conditions are true, a value of bS equal to 1 is the output:
  - mixedModeEdgeFlag is equal to 1
  - mixedModeEdgeFlag is equal to 0 and for the prediction of the macroblock/sub-macroblock partition containing the sample p_0 different reference pictures or a different number of motion vectors are used than for the prediction of the macroblock/sub-macroblock partition containing the sample q_0.

    NOTE 1 – The determination of whether the reference pictures used for the two macroblock/sub-macroblock partitions are the same or different is based only on which pictures are referenced, without regard to whether a prediction is formed using an index into reference picture list 0 or an index into reference picture list 1, and also without regard to whether or not the index position within a reference picture list is different or not.
  - mixedModeEdgeFlag is equal to 0 and one motion vector is used to predict the macroblock/sub-macroblock partition containing the sample p_0 and one motion vector is used to predict the macroblock/sub-macroblock partition containing the sample q_0 and the absolute difference between the horizontal or vertical component of the motion vectors used is greater than or equal to 4 in units of quarter luma frame samples.
  - mixedModeEdgeFlag is equal to 0 and two motion vectors and two different reference pictures are used to predict the macroblock/sub-macroblock partition containing the sample p_0 and two motion vectors for the same two reference pictures are used to predict the macroblock/sub-macroblock partition containing the sample q_0 and the absolute difference between the horizontal or vertical component of the two motion vectors used in the prediction of the two macroblock/sub-macroblock partitions for the same reference picture is greater than or equal to 4 in units of quarter luma frame samples.
  - mixedModeEdgeFlag is equal to 0 and two motion vectors for the same reference picture are used to predict the macroblock/sub-macroblock partition containing the sample p_0 and two motion vectors for the same reference picture are used to predict the macroblock/sub-macroblock partition containing the sample q_0 and both of the following conditions are true:
    - The absolute difference between the horizontal or vertical component of list 0 motion vectors used in the prediction of the two macroblock/sub-macroblock partitions is greater than or equal to 4 in quarter luma frame samples or the absolute difference between the horizontal or vertical component of the list 1 motion vectors used in the prediction of the two macroblock/sub-macroblock partitions is greater than or equal to 4 in units of quarter luma frame samples.
    - The absolute difference between the horizontal or vertical component of list 0 motion vector used in the prediction of the macroblock/sub-macroblock partition containing the sample p_0 and the list 1 motion vector used in the prediction of the macroblock/sub-macroblock partition containing the sample q_0 is greater than or equal to 4 in units of quarter luma frame samples or the absolute difference between the horizontal or vertical component of the list 1 motion vector used in the prediction of the macroblock/sub-macroblock partition containing the sample p_0 and list 0 motion vector used in the prediction of the macroblock/sub-macroblock partition containing the sample q_0 is greater than or equal to 4 in units of quarter luma frame samples.

      NOTE 2 – A vertical difference of 4 in units of quarter luma frame samples is a difference of 2 in units of quarter luma field samples

– Otherwise, a value of bS equal to 0 is the output.

8.7.2.2 Derivation process for the thresholds for each block edge

Inputs to this process are the input sample values p_0, q_0, p_1 and q_1 of a single set of samples across an edge that is to be filtered, chromaEdgeFlag, and bS, for the set of input samples, as specified in 8.7.2.

Outputs of this process are the variable filterSamplesFlag, which indicates whether the input samples are filtered, the value of indexA, and the values of the threshold variables α and β.

Let qP_p and qP_q be variables specifying quantisation parameter values for the macroblocks containing the samples p_0 and q_0, respectively. The variables qP_z (with z being replaced by p or q) are derived as follows.

– If chromaEdgeFlag is equal to 0, the following applies.
  – If the macroblock containing the sample z_0 is an I_PCM macroblock, qP_z is set to 0.
  – Otherwise (the macroblock containing the sample z_0 is not an I_PCM macroblock), qP_z is set to the value of QP_Y of the macroblock containing the sample z_0.
– Otherwise (chromaEdgeFlag is equal to 1), the following applies.
  – If the macroblock containing the sample z_0 is an I_PCM macroblock, qP_z is set to the value of QP_C that corresponds to a value of 0 for QP_Y as specified in subclause 8.5.7.
  – Otherwise (the macroblock containing the sample z_0 is not an I_PCM macroblock), qP_z is set to the value of QP_C that corresponds to the value QP_Y of the macroblock containing the sample z_0 as specified in subclause 8.5.7.

Let qP_{av} be a variable specifying an average quantisation parameter. It is derived as follows.

$$qP_{av} = (qP_p + qP_q + 1) >> 1 \quad (8\text{-}462)$$

NOTE – In SP and SI slices, qP_{av} is derived in the same way as in other slice types. QS_Y from Equation 7-28 is not used in the deblocking filter.

Let indexA be a variable that is used to access the α table (Table 8-16) as well as the t_{C0} table (Table 8-17), which is used in filtering of edges with bS less than 4 as specified in subclause 8.7.2.3, and let indexB be a variable that is used to access the β table (Table 8-16). The variables indexA and indexB are derived as follows, where the values of FilterOffsetA and FilterOffsetB are the values of those variables specified in subclause 7.4.3 for the slice that contains the macroblock containing sample q_0.

$$indexA = Clip3(0, 51, qP_{av} + FilterOffsetA) \quad (8\text{-}463)$$

$$indexB = Clip3(0, 51, qP_{av} + FilterOffsetB) \quad (8\text{-}464)$$

The variables α' and β' depending on the values of indexA and indexB are specified in Table 8-16. Depending on chromaEdgeFlag, the corresponding threshold variables α and β are derived as follows.

– If chromaEdgeFlag is equal to 0,

$$\alpha = \alpha' * (1 << (BitDepth_Y – 8)) \quad (8\text{-}465)$$

$$\beta = \beta' * (1 << (BitDepth_Y – 8)) \quad (8\text{-}466)$$

– Otherwise (chromaEdgeFlag is equal to 1),

$$\alpha = \alpha' * (1 << (BitDepth_C – 8)) \quad (8\text{-}467)$$

$$\beta = \beta' * (1 << (BitDepth_C – 8)) \quad (8\text{-}468)$$

The variable filterSamplesFlag is derived by

$$filterSamplesFlag = (bS != 0 \ \&\& \ Abs(p_0 – q_0) < \alpha \ \&\& \ Abs(p_1 – p_0) < \beta \ \&\& \ Abs(q_1 – q_0) < \beta) \quad (8\text{-}469)$$

Table 8-16 – Derivation of offset dependent threshold variables α' and β' from indexA and indexB

	indexA (for α') or indexB (for β')																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4

Table 8-16 (concluded) – Derivation of indexA and indexB from offset dependent threshold variables α' and β'

	indexA (for α') or indexB (for β')																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
α'	15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255
β'	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18

8.7.2.3 Filtering process for edges with bS less than 4

Inputs to this process are the input sample values p_i and q_i (i = 0..2) of a single set of samples across an edge that is to be filtered, chromaEdgeFlag, bS, β, and indexA, for the set of input samples, as specified in 8.7.2.

Outputs of this process are the filtered result sample values p'_i and q'_i (i = 0..2) for the set of input sample values.

The filtered result samples p'_0 and q'_0 are derived by

$$\Delta = \text{Clip3}(-t_C, t_C, ((((q_0 - p_0) << 2) + (p_1 - q_1) + 4) >> 3)) \quad (8\text{-}470)$$

$$p'_0 = \text{Clip1}(p_0 + \Delta) \quad (8\text{-}471)$$

$$q'_0 = \text{Clip1}(q_0 - \Delta) \quad (8\text{-}472)$$

where the threshold t_C is determined as follows.

– If chromaEdgeFlag is equal to 0,

$$t_C = t_{C0} + ((a_p < \beta) \ ? \ 1 : 0) + ((a_q < \beta) \ ? \ 1 : 0) \quad (8\text{-}473)$$

– Otherwise (chromaEdgeFlag is equal to 1),

$$t_C = t_{C0} + 1 \quad (8\text{-}474)$$

Depending on the values of indexA and bS the variable t'_{C0} is specified in Table 8-17. Depending on chromaEdgeFlag, the corresponding threshold variable t_{C0} is derived as follows.

– If chromaEdgeFlag is equal to 0,

$$t_{C0} = t'_{C0} * (1 << (\text{BitDepth}_Y - 8)) \quad (8\text{-}475)$$

– Otherwise (chromaEdgeFlag is equal to 1),

$$t_{C0} = t'_{C0} * (1 << (\text{BitDepth}_C - 8)) \quad (8\text{-}476)$$

Let a_p and a_q be two threshold variables specified by

$$a_p = Abs(p_2 - p_0) \quad (8\text{-}477)$$

$$a_q = Abs(q_2 - q_0) \quad (8\text{-}478)$$

The filtered result sample p'_1 is derived as follows

– If chromaEdgeFlag is equal to 0 and a_p is less than β,

$$p'_1 = p_1 + Clip3(-t_{C0}, t_{C0}, (p_2 + ((p_0 + q_0 + 1) >> 1) - (p_1 << 1)) >> 1) \quad (8\text{-}479)$$

– Otherwise (chromaEdgeFlag is equal to 1 or a_p is greater than or equal to β),

$$p'_1 = p_1 \quad (8\text{-}480)$$

The filtered result sample q'_1 is derived as follows

– If chromaEdgeFlag is equal to 0 and a_q is less than β,

$$q'_1 = q_1 + Clip3(-t_{C0}, t_{C0}, (q_2 + ((p_0 + q_0 + 1) >> 1) - (q_1 << 1)) >> 1) \quad (8\text{-}481)$$

– Otherwise (chromaEdgeFlag is equal to 1 or a_q is greater than or equal to β),

$$q'_1 = q_1 \quad (8\text{-}482)$$

The filtered result samples p'_2 and q'_2 are always set equal to the input samples p_2 and q_2:

$$p'_2 = p_2 \quad (8\text{-}483)$$

$$q'_2 = q_2 \quad (8\text{-}484)$$

Table 8-17 – Value of variable t'_{C0} as a function of indexA and bS

	indexA																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
bS = 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
bS = 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
bS = 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Table 8-17 (concluded) – Value of variable t'_{C0} as a function of indexA and bS

	indexA																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
bS = 1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13
bS = 2	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5	5	6	7	8	8	10	11	12	13	15	17
bS = 3	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13	14	16	18	20	23	25

8.7.2.4 Filtering process for edges for bS equal to 4

Inputs to this process are the input sample values p_i and q_i (i = 0..3) of a single set of samples across an edge that is to be filtered, the variable chromaEdgeFlag, and the values of the threshold variables α and β for the set of samples, as specified in subclause 8.7.2.

Outputs of this process are the filtered result sample values p'_i and q'_i (i = 0..2) for the set of input sample values.

Let a_p and a_q be two threshold variables as specified in Equations 8-477 and 8-478, respectively, in subclause 8.7.2.3.

The filtered result samples p'_i (i = 0..2) are derived as follows.

– If chromaEdgeFlag is equal to 0 and the following condition holds,

$$a_p < \beta \;\&\&\; Abs(p_0 - q_0) < ((\alpha >> 2) + 2) \tag{8-485}$$

then the variables p'_0, p'_1, and p'_2 are derived by

$$p'_0 = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3 \tag{8-486}$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) >> 2 \tag{8-487}$$

$$p'_2 = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3 \tag{8-488}$$

– Otherwise (chromaEdgeFlag is equal to 1 or the condition in Equation 8-485 does not hold), the variables p'_0, p'_1, and p'_2 are derived by

$$p'_0 = (2*p_1 + p_0 + q_1 + 2) >> 2 \tag{8-489}$$

$$p'_1 = p_1 \tag{8-490}$$

$$p'_2 = p_2 \tag{8-491}$$

The filtered result samples q'_i (i = 0..2) are derived as follows.

– If chromaEdgeFlag is equal to 0 and the following condition holds,

$$a_q < \beta \;\&\&\; Abs(p_0 - q_0) < ((\alpha >> 2) + 2) \tag{8-492}$$

then the variables q'_0, q'_1, and q'_2 are derived by

$$q'_0 = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3 \tag{8-493}$$

$$q'_1 = (p_0 + q_0 + q_1 + q_2 + 2) >> 2 \tag{8-494}$$

$$q'_2 = (2*q_3 + 3*q_2 + q_1 + q_0 + p_0 + 4) >> 3 \tag{8-495}$$

– Otherwise (chromaEdgeFlag is equal to 1 or the condition in Equation 8-492 does not hold), the variables q'_0, q'_1, and q'_2 are derived by

$$q'_0 = (2*q_1 + q_0 + p_1 + 2) >> 2 \tag{8-496}$$

$$q'_1 = q_1 \tag{8-497}$$

$$q'_2 = q_2 \tag{8-498}$$

9 Parsing process

Inputs to this process are bits from the RBSP.

Outputs of this process are syntax element values.

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ue(v), me(v), se(v), te(v) (see subclause 9.1), ce(v) (see subclause 9.2), or ae(v) (see subclause 9.3).

9.1 Parsing process for Exp-Golomb codes

This process is invoked when the descriptor of a syntax element in the syntax tables in subclause 7.3 is equal to ue(v), me(v), se(v), or te(v). For syntax elements in subclauses 7.3.4 and 7.3.5, this process is invoked only when entropy_coding_mode_flag is equal to 0.

Inputs to this process are bits from the RBSP.

Outputs of this process are syntax element values.

Syntax elements coded as ue(v), me(v), or se(v) are Exp-Golomb-coded. Syntax elements coded as te(v) are truncated Exp-Golomb-coded. The parsing process for these syntax elements begins with reading the bits starting at the current location in the bitstream up to and including the first non-zero bit, and counting the number of leading bits that are equal to 0. This process is specified as follows:

```
leadingZeroBits = -1;
for( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 )
```

The variable codeNum is then assigned as follows:

$$codeNum = 2^{leadingZeroBits} - 1 + read_bits(leadingZeroBits)$$

where the value returned from read_bits(leadingZeroBits) is interpreted as a binary representation of an unsigned integer with most significant bit written first.

Table 9-1 illustrates the structure of the Exp-Golomb code by separating the bit string into "prefix" and "suffix" bits. The "prefix" bits are those bits that are parsed in the above pseudo-code for the computation of leadingZeroBits, and are shown as either 0 or 1 in the bit string column of Table 9-1. The "suffix" bits are those bits that are parsed in the computation of codeNum and are shown as x_i in Table 9-1, with i being in the range 0 to leadingZeroBits - 1, inclusive. Each x_i can take on values 0 or 1.

Table 9-1 – Bit strings with "prefix" and "suffix" bits and assignment to codeNum ranges (informative)

Bit string form	Range of codeNum
1	0
0 1 x_0	1-2
0 0 1 x_1 x_0	3-6
0 0 0 1 x_2 x_1 x_0	7-14
0 0 0 0 1 x_3 x_2 x_1 x_0	15-30
0 0 0 0 0 1 x_4 x_3 x_2 x_1 x_0	31-62
...	...

Table 9-2 illustrates explicitly the assignment of bit strings to codeNum values.

Table 9-2 – Exp-Golomb bit strings and codeNum in explicit form and used as ue(v) (informative)

Bit string	codeNum
1	0
0 1 0	1
0 1 1	2
0 0 1 0 0	3
0 0 1 0 1	4
0 0 1 1 0	5
0 0 1 1 1	6
0 0 0 1 0 0 0	7
0 0 0 1 0 0 1	8
0 0 0 1 0 1 0	9
...	…

Depending on the descriptor, the value of a syntax element is derived as follows.

– If the syntax element is coded as ue(v), the value of the syntax element is equal to codeNum.

– Otherwise, if the syntax element is coded as se(v), the value of the syntax element is derived by invoking the mapping process for signed Exp-Golomb codes as specified in subclause 9.1.1 with codeNum as the input.

– Otherwise, if the syntax element is coded as me(v), the value of the syntax element is derived by invoking the mapping process for coded block pattern as specified in subclause 9.1.2 with codeNum as the input.

– Otherwise (the syntax element is coded as te(v)), the range of possible values for the syntax element is determined first. The range of this syntax element may be between 0 and x, with x being greater than or equal to 1 and the range is used in the derivation of the value of the syntax element value as follows

  – If x is greater than 1, codeNum and the value of the syntax element is derived in the same way as for syntax elements coded as ue(v)

  – Otherwise (x is equal to 1), the parsing process for codeNum which is equal to the value of the syntax element is given by a process equivalent to:

```
b = read_bits( 1 )
codeNum = !b
```

9.1.1 Mapping process for signed Exp-Golomb codes

Input to this process is codeNum as specified in subclause 9.1.

Output of this process is a value of a syntax element coded as se(v).

The syntax element is assigned to the codeNum by ordering the syntax element by its absolute value in increasing order and representing the positive value for a given absolute value with the lower codeNum. Table 9-3 provides the assignment rule.

Table 9-3 – Assignment of syntax element to codeNum for signed Exp-Golomb coded syntax elements se(v)

codeNum	syntax element value
0	0
1	1
2	–1
3	2
4	–2
5	3
6	–3
k	$(-1)^{k+1}$ Ceil(k÷2)

9.1.2 Mapping process for coded block pattern

Input to this process is codeNum as specified in subclause 9.1.

Output of this process is a value of the syntax element coded_block_pattern coded as me(v).

Table 9-4 shows the assignment of coded_block_pattern to codeNum depending on whether the macroblock prediction mode is equal to Intra_4x4, Intra_8x8 or Inter.

Table 9-4 – Assignment of codeNum to values of coded_block_pattern for macroblock prediction modes

(a) chroma_format_idc is not equal to 0

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
0	47	0
1	31	16
2	15	1
3	0	2
4	23	4
5	27	8
6	29	32
7	30	3
8	7	5
9	11	10
10	13	12
11	14	15
12	39	47
13	43	7
14	45	11

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
15	46	13
16	16	14
17	3	6
18	5	9
19	10	31
20	12	35
21	19	37
22	21	42
23	26	44
24	28	33
25	35	34
26	37	36
27	42	40
28	44	39
29	1	43
30	2	45
31	4	46
32	8	17
33	17	18
34	18	20
35	20	24
36	24	19
37	6	21
38	9	26
39	22	28
40	25	23
41	32	27
42	33	29
43	34	30
44	36	22
45	40	25
46	38	38
47	41	41

(b) chroma_format_idc is equal to 0

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
0	15	0
1	0	1
2	7	2
3	11	4
4	13	8
5	14	3
6	3	5
7	5	10
8	10	12
9	12	15
10	1	7
11	2	11
12	4	13
13	8	14
14	6	6
15	9	9

9.2 CAVLC parsing process for transform coefficient levels

This process is invoked when parsing syntax elements with descriptor equal to ce(v) in subclause 7.3.5.3.1 and when entropy_coding_mode_flag is equal to 0.

Inputs to this process are bits from slice data, a maximum number of non-zero transform coefficient levels maxNumCoeff, the luma block index luma4x4BlkIdx or the chroma block index chroma4x4BlkIdx of the current block of transform coefficient levels.

Output of this process is the list coeffLevel containing transform coefficient levels of the luma block with block index luma4x4BlkIdx or the chroma block with block index chroma4x4BlkIdx.

The process is specified in the following ordered steps:

1. All transform coefficient levels, with indices from 0 to maxNumCoeff - 1, in the list coeffLevel are set equal to 0.
2. The total number of non-zero transform coefficient levels TotalCoeff(coeff_token) and the number of trailing one transform coefficient levels TrailingOnes(coeff_token) are derived by parsing coeff_token (see subclause 9.2.1) as follows.
   - If the number of non-zero transform coefficient levels TotalCoeff(coeff_token) is equal to 0, the list coeffLevel containing 0 values is returned and no further step is carried out.
   - Otherwise, the following steps are carried out.
     a. The non-zero transform coefficient levels are derived by parsing trailing_ones_sign_flag, level_prefix, and level_suffix (see subclause 9.2.2).
     b. The runs of zero transform coefficient levels before each non-zero transform coefficient level are derived by parsing total_zeros and run_before (see subclause 9.2.3).

c. The level and run information are combined into the list coeffLevel (see subclause 9.2.4).

9.2.1 Parsing process for total number of transform coefficient levels and trailing ones

Inputs to this process are bits from slice data, a maximum number of non-zero transform coefficient levels maxNumCoeff, the luma block index luma4x4BlkIdx or the chroma block index chroma4x4BlkIdx of the current block of transform.

Outputs of this process are TotalCoeff(coeff_token) and TrailingOnes(coeff_token).

The syntax element coeff_token is decoded using one of the six VLCs specified in the six right-most columns of Table 9-5. Each VLC specifies both TotalCoeff(coeff_token) and TrailingOnes(coeff_token) for a given codeword coeff_token. VLC selection is dependent upon a variable nC that is derived as follows.

- If the CAVLC parsing process is invoked for ChromaDCLevel, nC is derived as follows.
  - If chroma_format_idc is equal to 1, nC is set equal to -1,
  - Otherwise, if chroma_format_idc is equal to 2, nC is set equal to -2,
  - Otherwise (chroma_format_idc is equal to 3), nC is set equal to 0.
- Otherwise, the following applies.
  - When the CAVLC parsing process is invoked for Intra16x16DCLevel, luma4x4BlkIdx is set equal to 0.
  - The variables blkA and blkB are derived as follows.
    - If the CAVLC parsing process is invoked for Intra16x16DCLevel, Intra16x16ACLevel, or LumaLevel, the process specified in subclause 6.4.8.3 is invoked with luma4x4BlkIdx as the input, and the output is assigned to mbAddrA, mbAddrB, luma4x4BlkIdxA, and luma4x4BlkIdxB. The 4x4 luma block specified by mbAddrA\luma4x4BlkIdxA is assigned to blkA, and the 4x4 luma block specified by mbAddrB\luma4x4BlkIdxB is assigned to blkB.
    - Otherwise (the CAVLC parsing process is invoked for ChromaACLevel), the process specified in subclause 6.4.8.4 is invoked with chroma4x4BlkIdx as input, and the output is assigned to mbAddrA, mbAddrB, chroma4x4BlkIdxA, and chroma4x4BlkIdxB. The 4x4 chroma block specified by mbAddrA\iCbCr\chroma4x4BlkIdxA is assigned to blkA, and the 4x4 chroma block specified by mbAddrB\iCbCr\chroma4x4BlkIdxB is assigned to blkB.
  - Let nA and nB be the number of non-zero transform coefficient levels (given by TotalCoeff(coeff_token)) in the block of transform coefficient levels blkA located to the left of the current block and the block of transform coefficient levels blkB located above the current block, respectively.
  - With N replaced by A and B, in mbAddrN, blkN, and nN the following applies.
    - If any of the following conditions is true, nN is set equal to 0.
      - mbAddrN is not available
      - The current macroblock is coded using an Intra prediction mode, constrained_intra_pred_flag is equal to 1 and mbAddrN is coded using Inter prediction and slice data partitioning is in use (nal_unit_type is in the range of 2 to 4, inclusive).
      - The macroblock mbAddrN has mb_type equal to P_Skip or B_Skip
      - All AC residual transform coefficient levels of the neighbouring block blkN are equal to 0 due to the corresponding bit of CodedBlockPatternLuma or CodedBlockPatternChroma being equal to 0
    - Otherwise, if mbAddrN is an I_PCM macroblock, nN is set equal to 16.
    - Otherwise, nN is set equal to the value TotalCoeff(coeff_token) of the neighbouring block blkN.

      NOTE 1 – The values nA and nB that are derived using TotalCoeff(coeff_token) do not include the DC transform coefficient levels in Intra_16x16 macroblocks or DC transform coefficient levels in chroma blocks, because these transform coefficient levels are decoded separately. When the block above or to the left belongs to an Intra_16x16 macroblock, or is a chroma block, nA and nB is the number of decoded non-zero AC transform coefficient levels.

      NOTE 2 – When parsing for Intra16x16DCLevel, the values nA and nB are based on the number of non-zero transform coefficient levels in adjacent 4x4 blocks and not on the number of non-zero DC transform coefficient levels in adjacent 16x16 blocks.
  - Given the values of nA and nB, the variable nC is derived as follows.

– If both mbAddrA and mbAddrB are available, the variable nC is set equal to (nA + nB + 1) >> 1.

– Otherwise (mbAddrA is not available or mbAddrB is not available), the variable nC is set equal to nA + nB.

The value of TotalCoeff(coeff_token) resulting from decoding coeff_token shall be in the range of 0 to maxNumCoeff, inclusive.

Table 9-5 – coeff_token mapping to TotalCoeff(coeff_token) and TrailingOnes(coeff_token)

TrailingOnes (coeff_token)	TotalCoeff (coeff_token)	0 <= nC < 2	2 <= nC < 4	4 <= nC < 8	8 <= nC	nC = = -1	nC = = -2
0	0	1	11	1111	0000 11	01	1
0	1	0001 01	0010 11	0011 11	0000 00	0001 11	0001 111
1	1	01	10	1110	0000 01	1	01
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00	0001 110
1	2	0001 00	0011 1	0111 1	0001 01	0001 10	0001 101
2	2	001	011	1101	0001 10	001	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11	0000 0011 1
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011	0001 100
2	3	0000 101	0010 01	0111 0	0010 10	0000 010	0001 011
3	3	0001 1	0101	1100	0010 11	0001 01	0000 1
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10	0000 0011 0
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 0011	0000 0010 1
2	4	0000 0101	0001 01	0101 1	0011 10	0000 0010	0001 010
3	4	0000 11	0100	1011	0011 11	0000 000	0000 01
0	5	0000 0000 111	0000 0100	0001 011	0100 00	-	0000 0001 11
1	5	0000 0001 10	0000 110	0100 0	0100 01	-	0000 0001 10
2	5	0000 0010 1	0000 101	0100 1	0100 10	-	0000 0010 0
3	5	0000 100	0011 0	1010	0100 11	-	0001 001
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	-	0000 0000 111
1	6	0000 0000 110	0000 0110	0011 10	0101 01	-	0000 0000 110
2	6	0000 0001 01	0000 0101	0011 01	0101 10	-	0000 0001 01
3	6	0000 0100	0010 00	1001	0101 11	-	0001 000
0	7	0000 0000 0101 1	0000 0001 111	0001 000	0110 00	-	0000 0000 0111
1	7	0000 0000 0111 0	0000 0011 0	0010 10	0110 01	-	0000 0000 0110
2	7	0000 0000 101	0000 0010 1	0010 01	0110 10	-	0000 0000 101
3	7	0000 0010 0	0001 00	1000	0110 11	-	0000 0001 00
0	8	0000 0000 0100 0	0000 0001 011	0000 1111	0111 00	-	0000 0000 0011 1
1	8	0000 0000 0101 0	0000 0001 110	0001 110	0111 01	-	0000 0000 0101
2	8	0000 0000 0110 1	0000 0001 101	0001 101	0111 10	-	0000 0000 0100
3	8	0000 0001 00	0000 100	0110 1	0111 11	-	0000 0000 100
0	9	0000 0000 0011 11	0000 0000 1111	0000 1011	1000 00	-	

TrailingOnes (coeff_token)	TotalCoeff (coeff_token)	0 <= nC < 2	2 <= nC < 4	4 <= nC < 8	8 <= nC	nC = = -1	nC = = -2
1	9	0000 0000 0011 10	0000 0001 010	0000 1110	1000 01	-	
2	9	0000 0000 0100 1	0000 0001 001	0001 010	1000 10	-	
3	9	0000 0000 100	0000 0010 0	0011 00	1000 11	-	
0	10	0000 0000 0010 11	0000 0000 1011	0000 0111 1	1001 00	-	
1	10	0000 0000 0010 10	0000 0000 1110	0000 1010	1001 01	-	
2	10	0000 0000 0011 01	0000 0000 1101	0000 1101	1001 10	-	
3	10	0000 0000 0110 0	0000 0001 100	0001 100	1001 11	-	
0	11	0000 0000 0001 111	0000 0000 1000	0000 0101 1	1010 00	-	
1	11	0000 0000 0001 110	0000 0000 1010	0000 0111 0	1010 01	-	
2	11	0000 0000 0010 01	0000 0000 1001	0000 1001	1010 10	-	
3	11	0000 0000 0011 00	0000 0001 000	0000 1100	1010 11	-	
0	12	0000 0000 0001 011	0000 0000 0111 1	0000 0100 0	1011 00	-	
1	12	0000 0000 0001 010	0000 0000 0111 0	0000 0101 0	1011 01	-	
2	12	0000 0000 0001 101	0000 0000 0110 1	0000 0110 1	1011 10	-	
3	12	0000 0000 0010 00	0000 0000 1100	0000 1000	1011 11	-	
0	13	0000 0000 0000 1111	0000 0000 0101 1	0000 0011 01	1100 00	-	
1	13	0000 0000 0000 001	0000 0000 0101 0	0000 0011 1	1100 01	-	
2	13	0000 0000 0001 001	0000 0000 0100 1	0000 0100 1	1100 10	-	
3	13	0000 0000 0001 100	0000 0000 0110 0	0000 0110 0	1100 11	-	
0	14	0000 0000 0000 1011	0000 0000 0011 1	0000 0010 01	1101 00	-	
1	14	0000 0000 0000 1110	0000 0000 0010 11	0000 0011 00	1101 01	-	
2	14	0000 0000 0000 1101	0000 0000 0011 0	0000 0010 11	1101 10	-	
3	14	0000 0000 0001 000	0000 0000 0100 0	0000 0010 10	1101 11	-	
0	15	0000 0000 0000 0111	0000 0000 0010 01	0000 0001 01	1110 00	-	
1	15	0000 0000 0000 1010	0000 0000 0010 00	0000 0010 00	1110 01	-	
2	15	0000 0000 0000 1001	0000 0000 0010 10	0000 0001 11	1110 10	-	
3	15	0000 0000 0000 1100	0000 0000 0000 1	0000 0001 10	1110 11	-	
0	16	0000 0000 0000 0100	0000 0000 0001 11	0000 0000 01	1111 00	-	
1	16	0000 0000 0000 0110	0000 0000 0001 10	0000 0001 00	1111 01	-	
2	16	0000 0000 0000 0101	0000 0000 0001 01	0000 0000 11	1111 10	-	
3	16	0000 0000 0000 1000	0000 0000 0001 00	0000 0000 10	1111 11	-	

9.2.2 Parsing process for level information

Inputs to this process are bits from slice data, the number of non-zero transform coefficient levels TotalCoeff(coeff_token), and the number of trailing one transform coefficient levels TrailingOnes(coeff_token).

Output of this process is a list with name level containing transform coefficient levels.

Initially an index i is set equal to 0. Then the following procedure is iteratively applied TrailingOnes(coeff_token) times to decode the trailing one transform coefficient levels (if any):

– A 1-bit syntax element trailing_ones_sign_flag is decoded and evaluated as follows.
  – If trailing_ones_sign_flag is equal to 0, the value +1 is assigned to level[i].
  – Otherwise (trailing_ones_sign_flag is equal to 1), the value -1 is assigned to level[i].
– The index i is incremented by 1.

Following the decoding of the trailing one transform coefficient levels, a variable suffixLength is initialised as follows.

– If TotalCoeff(coeff_token) is greater than 10 and TrailingOnes(coeff_token) is less than 3, suffixLength is set equal to 1.
– Otherwise (TotalCoeff(coeff_token) is less than or equal to 10 or TrailingOnes(coeff_token) is equal to 3), suffixLength is set equal to 0.

The following procedure is then applied iteratively (TotalCoeff(coeff_token) – TrailingOnes(coeff_token)) times to decode the remaining levels (if any):

– The syntax element level_prefix is decoded as specified in subclause 9.2.2.1.
– The variable levelSuffixSize is set equal to the variable suffixLength with the exception of the following two cases.
– When level_prefix is equal to 14 and suffixLength is equal to 0, levelSuffixSize is set equal to 4.
– When level_prefix is greater than or equal to 15, levelSuffixSize is set equal to level_prefix - 3.
– The syntax element level_suffix is decoded as follows.
  – If levelSuffixSize is greater than 0, the syntax element level_suffix is decoded as unsigned integer representation u(v) with levelSuffixSize bits.
  – Otherwise (levelSuffixSize is equal to 0), the syntax element level_suffix is inferred to be equal to 0.
– A variable levelCode is set equal to (Min(15, level_prefix) << suffixLength) + level_suffix.
– When level_prefix is greater than or equal to 15 and suffixLength is equal to 0, levelCode is incremented by 15.
– When level_prefix is greater than or equal to 16, levelCode is incremented by (1<<(level_prefix – 3)) – 4096.
– When the index i is equal to TrailingOnes(coeff_token) and TrailingOnes(coeff_token) is less than 3, levelCode is incremented by 2.
– The variable level[i] is derived as follows.
  – If levelCode is an even number, the value (levelCode + 2) >> 1 is assigned to level[i].
  – Otherwise (levelCode is an odd number), the value (-levelCode – 1) >> 1 is assigned to level[i].
– When suffixLength is equal to 0, suffixLength is set equal to 1.
– When the absolute value of level[i] is greater than (3 << (suffixLength – 1)) and suffixLength is less than 6, suffixLength is incremented by 1.
– The index i is incremented by 1.

9.2.2.1 Parsing process for level_prefix

Inputs to this process are bits from slice data.

Output of this process is level_prefix.

The parsing process for this syntax element consists in reading the bits starting at the current location in the bitstream up to and including the first non-zero bit, and counting the number of leading bits that are equal to 0. This process is specified as follows:

```
leadingZeroBits = -1
for( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 )
level_prefix = leadingZeroBits
```

Table 9-6 illustrates the codeword table for level_prefix.

Table 9-6 – Codeword table for level_prefix (informative)

level_prefix	bit string
0	1
1	01
2	001
3	0001
4	0000 1
5	0000 01
6	0000 001
7	0000 0001
8	0000 0000 1
9	0000 0000 01
10	0000 0000 001
11	0000 0000 0001
12	0000 0000 0000 1
13	0000 0000 0000 01
14	0000 0000 0000 001
15	0000 0000 0000 0001
…	…

9.2.3 Parsing process for run information

Inputs to this process are bits from slice data, the number of non-zero transform coefficient levels TotalCoeff(coeff_token), and the maximum number of non-zero transform coefficient levels maxNumCoeff.

Output of this process is a list of runs of zero transform coefficient levels preceding non-zero transform coefficient levels called run.

Initially, an index i is set equal to 0.

The variable zerosLeft is derived as follows.

– If the number of non-zero transform coefficient levels TotalCoeff(coeff_token) is equal to the maximum number of non-zero transform coefficient levels maxNumCoeff, a variable zerosLeft is set equal to 0.

– Otherwise (the number of non-zero transform coefficient levels TotalCoeff(coeff_token) is less than the maximum number of non-zero transform coefficient levels maxNumCoeff), total_zeros is decoded and zerosLeft is set equal to its value.

The VLC used to decode total_zeros is derived as follows:

If maxNumCoeff is equal to 4, one of the VLCs specified in Table 9-9 (a) is used.

– Otherwise, if maxNumCoeff is equal to 8, one of the VLCs specified in Table 9-9 (b) is used.

– Otherwise (maxNumCoeff is not equal to 4 and not equal to 8), VLCs from Tables 9-7 and 9-8 are used.

The following procedure is then applied iteratively (TotalCoeff(coeff_token) – 1) times:

– The variable run[i] is derived as follows.
  – If zerosLeft is greater than zero, a value run_before is decoded based on Table 9-10 and zerosLeft. run[i] is set equal to run_before.
  – Otherwise (zerosLeft is equal to 0), run[i] is set equal to 0.
– The value of run[i] is subtracted from zerosLeft and the result assigned to zerosLeft. The result of the subtraction shall be greater than or equal to 0.
– The index i is incremented by 1.

Finally the value of zerosLeft is assigned to run[i].

Table 9-7 – total_zeros tables for 4x4 blocks with TotalCoeff(coeff_token) 1 to 7

total_zeros	TotalCoeff(coeff_token)						
	1	2	3	4	5	6	7
0	1	111	0101	0001 1	0101	0000 01	0000 01
1	011	110	111	111	0100	0000 1	0000 1
2	010	101	110	0101	0011	111	101
3	0011	100	101	0100	111	110	100
4	0010	011	0100	110	110	101	011
5	0001 1	0101	0011	101	101	100	11
6	0001 0	0100	100	100	100	011	010
7	0000 11	0011	011	0011	011	010	0001
8	0000 10	0010	0010	011	0010	0001	001
9	0000 011	0001 1	0001 1	0010	0000 1	001	0000 00
10	0000 010	0001 0	0001 0	0001 0	0001	0000 00	
11	0000 0011	0000 11	0000 01	0000 1	0000 0		
12	0000 0010	0000 10	0000 1	0000 0			
13	0000 0001 1	0000 01	0000 00				
14	0000 0001 0	0000 00					
15	0000 0000 1						

Table 9-8 – total_zeros tables for 4x4 blocks with TotalCoeff(coeff_token) 8 to 15

total_zeros	TotalCoeff(coeff_token)							
	8	9	10	11	12	13	14	15
0	0000 01	0000 01	0000 1	0000	0000	000	00	0
1	0001	0000 00	0000 0	0001	0001	001	01	1
2	0000 1	0001	001	001	01	1	1	
3	011	11	11	010	1	01		
4	11	10	10	1	001			
5	10	001	01	011				
6	010	01	0001					
7	001	0000 1						
8	0000 00							

Table 9-9 – total_zeros tables for chroma DC 2x2 and 2x4 blocks

(a) Chroma DC 2x2 block (4:2:0 chroma sampling)

total_zeros	TotalCoeff(coeff_token)		
	1	2	3
0	1	1	1
1	01	01	0
2	001	00	
3	000		

(b) Chroma DC 2x4 block (4:2:2 chroma sampling)

total_zeros	TotalCoeff(coeff_token)						
	1	2	3	4	5	6	7
0	1	000	000	110	00	00	0
1	010	01	001	00	01	01	1
2	011	001	01	01	10	1	
3	0010	100	10	10	11		
4	0011	101	110	111			
5	0001	110	111				
6	0000 1	111					
7	0000 0						

Table 9-10 – Tables for run_before

run_before	zerosLeft						
	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	-	00	01	01	011	001	101
3	-	-	00	001	010	011	100
4	-	-	-	000	001	010	011
5	-	-	-	-	000	101	010
6	-	-	-	-	-	100	001
7	-	-	-	-	-	-	0001
8		-	-	-	-	-	00001
9	-	-	-	-	-	-	000001
10	-	-	-	-	-	-	0000001
11	-	-	-	-	-	-	00000001
12	-	-	-	-	-	-	000000001
13	-	-	-	-	-	-	0000000001
14	-	-	-	-	-	-	00000000001

9.2.4 Combining level and run information

Input to this process are a list of transform coefficient levels called level, a list of runs called run, and the number of non-zero transform coefficient levels TotalCoeff(coeff_token).

Output of this process is an list coeffLevel of transform coefficient levels.

A variable coeffNum is set equal to -1 and an index i is set equal to (TotalCoeff(coeff_token) – 1). The following procedure is iteratively applied TotalCoeff(coeff_token) times:

– coeffNum is incremented by run[i] + 1.

– coeffLevel[coeffNum] is set equal to level[i].

– The index i is decremented by 1.

9.3 CABAC parsing process for slice data

This process is invoked when parsing syntax elements with descriptor ae(v) in subclauses 7.3.4 and 7.3.5 when entropy_coding_mode_flag is equal to 1.

Inputs to this process are a request for a value of a syntax element and values of prior parsed syntax elements.

Output of this process is the value of the syntax element.

When starting the parsing of the slice data of a slice in subclause 7.3.4, the initialisation process of the CABAC parsing process is invoked as specified in subclause 9.3.1.

The parsing of syntax elements proceeds as follows:

For each requested value of a syntax element a binarization is derived as described in subclause 9.3.2.

The binarization for the syntax element and the sequence of parsed bins determines the decoding process flow as described in subclause 9.3.3.

For each bin of the binarization of the syntax element, which is indexed by the variable binIdx, a context index ctxIdx is derived as specified in subclause 9.3.3.1.

For each ctxIdx the arithmetic decoding process is invoked as specified in subclause 9.3.3.2.

The resulting sequence (b_0 .. b_{binIdx}) of parsed bins is compared to the set of bin strings given by the binarization process after decoding of each bin. When the sequence matches a bin string in the given set, the corresponding value is assigned to the syntax element.

In case the request for a value of a syntax element is processed for the syntax element mb_type and the decoded value of mb_type is equal to I_PCM, the decoding engine is initialised after the decoding of any pcm_alignment_zero_bit and all pcm_sample_luma and pcm_sample_chroma data as specified in subclause 9.3.1.2.

The whole CABAC parsing process is illustrated in the flowchart of Figure 9-1 with the abbreviation SE for syntax element.

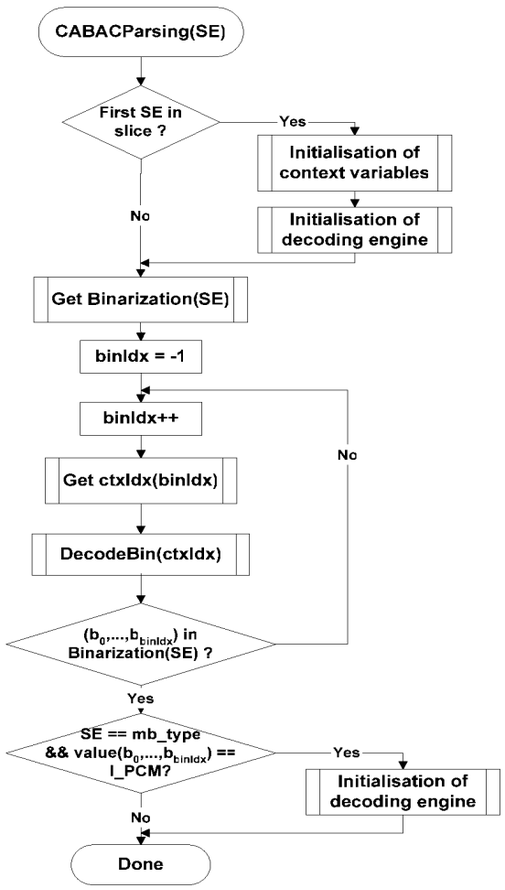


Figure 9-1 – Illustration of CABAC parsing process for a syntax element SE (informative)

9.3.1 Initialisation process

Outputs of this process are initialised CABAC internal variables.

The processes in subclauses 9.3.1.1 and 9.3.1.2 are invoked when starting the parsing of the slice data of a slice in subclause 7.3.4.

The process in subclause 9.3.1.2 is also invoked after decoding any pcm_alignment_zero_bit and all pcm_sample_luma and pcm_sample_chroma data for a macroblock of type I_PCM.

9.3.1.1 Initialisation process for context variables

Outputs of this process are the initialised CABAC context variables indexed by ctxIdx.

Table 9-12 to Table 9-23 contain the values of the variables n and m used in the initialisation of context variables that are assigned to all syntax elements in subclauses 7.3.4 and 7.3.5 except for the end-of-slice flag.

For each context variable, the two variables pStateIdx and valMPS are initialised.

NOTE 1 – The variable pStateIdx corresponds to a probability state index and the variable valMPS corresponds to the value of the most probable symbol as further described in subclause 9.3.3.2.

The two values assigned to pStateIdx and valMPS for the initialisation are derived from $SliceQP_Y$, which is derived in Equation 7-27. Given the two table entries (m, n),

1. preCtxState = Clip3(1, 126, ((m * Clip3(0, 51, $SliceQP_Y$)) >> 4) + n)
2. if(preCtxState <= 63) {

        pStateIdx = 63 - preCtxState

        valMPS = 0

    } else {

        pStateIdx = preCtxState - 64

        valMPS = 1

    }

In Table 9-11, the ctxIdx for which initialisation is needed for each of the slice types are listed. Also listed is the table number that includes the values of m and n needed for the initialisation. For P, SP and B slice type, the initialisation depends also on the value of the cabac_init_idc syntax element. Note that the syntax element names do not affect the initialisation process.

Table 9-11 – Association of ctxIdx and syntax elements for each slice type in the initialisation process

	Syntax element	Table	Slice type			
			SI	I	P, SP	B
slice_data( )	mb_skip_flag	Table 9-13 Table 9-14			11-13	24-26
	mb_field_decoding_flag	Table 9-18	70-72	70-72	70-72	70-72
macroblock_layer( )	mb_type	Table 9-12 Table 9-13 Table 9-14	0-10	3-10	14-20	27-35
	transform_size_8x8_flag	Table 9-16	na	399-401	399-401	399-401
	coded_block_pattern (luma)	Table 9-18	73-76	73-76	73-76	73-76
	coded_block_pattern (chroma)	Table 9-18	77-84	77-84	77-84	77-84
	mb_qp_delta	Table 9-17	60-63	60-63	60-63	60-63
mb_pred( )	prev_intra4x4_pred_mode_flag	Table 9-17	68	68	68	68
	rem_intra4x4_pred_mode	Table 9-17	69	69	69	69
	prev_intra8x8_pred_mode_flag	Table 9-17	na	68	68	68
	rem_intra8x8_pred_mode	Table 9-17	na	69	69	69
	intra_chroma_pred_mode	Table 9-17	64-67	64-67	64-67	64-67

Table 9-11 – Association of ctxIdx and syntax elements for each slice type in the initialisation process

	Syntax element	Table	Slice type			
			SI	I	P, SP	B
mb_pred( ) and sub_mb_pred( )	ref_idx_l0	Table 9-16			54-59	54-59
	ref_idx_l1	Table 9-16				54-59
	mvd_l0[ ][ ][0]	Table 9-15			40-46	40-46
	mvd_l1[ ][ ][0]	Table 9-15				40-46
	mvd_l0[ ][ ][1]	Table 9-15			47-53	47-53
	mvd_l1[ ][ ][1]	Table 9-15				47-53
sub_mb_pred( )	sub_mb_type	Table 9-13 Table 9-14			21-23	36-39
residual_block_cabac( )	coded_block_flag	Table 9-18	85-104	85-104	85-104	85-104
	significant_coeff_flag[ ]	Table 9-19 Table 9-22 Table 9-24 Table 9-24	105-165 277-337	105-165 277-337 402-416 436-450	105-165 277-337 402-416 436-450	105-165 277-337 402-416 436-450
	last_significant_coeff_flag[ ]	Table 9-20 Table 9-23 Table 9-24 Table 9-24	166-226 338-398	166-226 338-398 417-425 451-459	166-226 338-398 417-425 451-459	166-226 338-398 417-425 451-459
	coeff_abs_level_minus1[ ]	Table 9-21 Table 9-24	227-275	227-275 426-435	227-275 426-435	227-275 426-435

NOTE 2 – ctxIdx equal to 276 is associated with the end_of_slice_flag and the bin of mb_type, which specifies the I_PCM macroblock type. The decoding process specified in subclause 9.3.3.2.4 applies to ctxIdx equal to 276. This decoding process, however, may also be implemented by using the decoding process specified in subclause 9.3.3.2.1. In this case, the initial values associated with ctxIdx equal to 276 are specified to be pStateIdx = 63 and valMPS = 0, where pStateIdx = 63 represents a non-adapting probability state.

Table 9-12 – Values of variables m and n for ctxIdx from 0 to 10

Initialisation variables	ctxIdx										
	0	1	2	3	4	5	6	7	8	9	10
m	20	2	3	20	2	3	-28	-23	-6	-1	7
n	-15	54	74	-15	54	74	127	104	53	54	51

Table 9-13 – Values of variables m and n for ctxIdx from 11 to 23

Value of cabac_init_idc	Initialisation variables	ctxIdx												
		11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
0	**m**	23	23	21	1	0	-37	5	-13	-11	1	12	-4	17
	n	33	2	0	9	49	118	57	78	65	62	49	73	50
1	**m**	22	34	16	-2	4	-29	2	-6	-13	5	9	-3	10
	n	25	0	0	9	41	118	65	71	79	52	50	70	54
2	**m**	29	25	14	-10	-3	-27	26	-4	-24	5	6	-17	14
	n	16	0	0	51	62	99	16	85	102	57	57	73	57

Table 9-14 – Values of variables m and n for ctxIdx from 24 to 39

Value of cabac_init_idc	Initialisation variables	ctxIdx															
		24	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
0	**m**	18	9	29	26	16	9	-46	-20	1	-13	-11	1	-6	-17	-6	9
	n	64	43	0	67	90	104	127	104	67	78	65	62	86	95	61	45
1	**m**	26	19	40	57	41	26	-45	-15	-4	-6	-13	5	6	-13	0	8
	n	34	22	0	2	36	69	127	101	76	71	79	52	69	90	52	43
2	**m**	20	20	29	54	37	12	-32	-22	-2	-4	-24	5	-6	-14	-6	4
	n	40	10	0	0	42	97	127	117	74	85	102	57	93	88	44	55

Table 9-15 – Values of variables m and n for ctxIdx from 40 to 53

Value of cabac_init_idc	Initialisation variables	ctxIdx													
		40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**
0	**m**	-3	-6	-11	6	7	-5	2	0	-3	-10	5	4	-3	0
	n	69	81	96	55	67	86	88	58	76	94	54	69	81	88
1	**m**	-2	-5	-10	2	2	-3	-3	1	-3	-6	0	-3	-7	-5
	n	69	82	96	59	75	87	100	56	74	85	59	81	86	95
2	**m**	-11	-15	-21	19	20	4	6	1	-5	-13	5	6	-3	-1
	n	89	103	116	57	58	84	96	63	85	106	63	75	90	101

Table 9-16 – Values of variables m and n for ctxIdx from 54 to 59, and 399 to 401

Value of cabac_init_idc	Initialisation variables	ctxIdx								
		54	**55**	**56**	**57**	**58**	**59**	**399**	**400**	**401**
I slices	**m**	na	na	na	na	na	na	31	31	25
	n	na	na	na	na	na	na	21	31	50
0	**m**	-7	-5	-4	-5	-7	1	12	11	14
	n	67	74	74	80	72	58	40	51	59
1	**m**	-1	-1	1	-2	-5	0	25	21	21
	n	66	77	70	86	72	61	32	49	54
2	**m**	3	-4	-2	-12	-7	1	21	19	17
	n	55	79	75	97	50	60	33	50	61

Table 9-17 – Values of variables m and n for ctxIdx from 60 to 69

Initialisation variables	ctxIdx									
	60	**61**	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**
m	0	0	0	0	-9	4	0	-7	13	3
n	41	63	63	63	83	86	97	72	41	62

Table 9-18 – Values of variables m and n for ctxIdx from 70 to 104

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
70	0	11	0	45	13	15	7	34	**88**	-11	115	-13	108	-4	92	5	78
71	1	55	-4	78	7	51	-9	88	**89**	-12	63	-3	46	0	39	-6	55
72	0	69	-3	96	2	80	-20	127	**90**	-2	68	-1	65	0	65	4	61
73	-17	127	-27	126	-39	127	-36	127	**91**	-15	84	-1	57	-15	84	-14	83
74	-13	102	-28	98	-18	91	-17	91	**92**	-13	104	-9	93	-35	127	-37	127
75	0	82	-25	101	-17	96	-14	95	**93**	-3	70	-3	74	-2	73	-5	79
76	-7	74	-23	67	-26	81	-25	84	**94**	-8	93	-9	92	-12	104	-11	104
77	-21	107	-28	82	-35	98	-25	86	**95**	-10	90	-8	87	-9	91	-11	91
78	-27	127	-20	94	-24	102	-12	89	**96**	-30	127	-23	126	-31	127	-30	127
79	-31	127	-16	83	-23	97	-17	91	**97**	-1	74	5	54	3	55	0	65
80	-24	127	-22	110	-27	119	-31	127	**98**	-6	97	6	60	7	56	-2	79
81	-18	95	-21	91	-24	99	-14	76	**99**	-7	91	6	59	7	55	0	72
82	-27	127	-18	102	-21	110	-18	103	**100**	-20	127	6	69	8	61	-4	92
83	-21	114	-13	93	-18	102	-13	90	**101**	-4	56	-1	48	-3	53	-6	56
84	-30	127	-29	127	-36	127	-37	127	**102**	-5	82	0	68	0	68	3	68
85	-17	123	-7	92	0	80	11	80	**103**	-7	76	-4	69	-7	74	-8	71
86	-12	115	-5	89	-5	89	5	76	**104**	-22	125	-8	88	-9	88	-13	98
87	-16	122	-7	96	-7	94	2	84									

Table 9-19 – Values of variables m and n for ctxIdx from 105 to 165

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
105	-7	93	-2	85	-13	103	-4	86	**136**	-13	101	5	53	0	58	-5	75
106	-11	87	-6	78	-13	91	-12	88	**137**	-13	91	-2	61	-1	60	-8	80
107	-3	77	-1	75	-9	89	-5	82	**138**	-12	94	0	56	-3	61	-21	83
108	-5	71	-7	77	-14	92	-3	72	**139**	-10	88	0	56	-8	67	-21	64
109	-4	63	2	54	-8	76	-4	67	**140**	-16	84	-13	63	-25	84	-13	31
110	-4	68	5	50	-12	87	-8	72	**141**	-10	86	-5	60	-14	74	-25	64
111	-12	84	-3	68	-23	110	-16	89	**142**	-7	83	-1	62	-5	65	-29	94
112	-7	62	1	50	-24	105	-9	69	**143**	-13	87	4	57	5	52	9	75
113	-7	65	6	42	-10	78	-1	59	**144**	-19	94	-6	69	2	57	17	63
114	8	61	-4	81	-20	112	5	66	**145**	1	70	4	57	0	61	-8	74
115	5	56	1	63	-17	99	4	57	**146**	0	72	14	39	-9	69	-5	35
116	-2	66	-4	70	-78	127	-4	71	**147**	-5	74	4	51	-11	70	-2	27
117	1	64	0	67	-70	127	-2	71	**148**	18	59	13	68	18	55	13	91
118	0	61	2	57	-50	127	2	58	**149**	-8	102	3	64	-4	71	3	65
119	-2	78	-2	76	-46	127	-1	74	**150**	-15	100	1	61	0	58	-7	69
120	1	50	11	35	-4	66	-4	44	**151**	0	95	9	63	7	61	8	77
121	7	52	4	64	-5	78	-1	69	**152**	-4	75	7	50	9	41	-10	66
122	10	35	1	61	-4	71	0	62	**153**	2	72	16	39	18	25	3	62
123	0	44	11	35	-8	72	-7	51	**154**	-11	75	5	44	9	32	-3	68
124	11	38	18	25	2	59	-4	47	**155**	-3	71	4	52	5	43	-20	81
125	1	45	12	24	-1	55	-6	42	**156**	15	46	11	48	9	47	0	30
126	0	46	13	29	-7	70	-3	41	**157**	-13	69	-5	60	0	44	1	7
127	5	44	13	36	-6	75	-6	53	**158**	0	62	-1	59	0	51	-3	23
128	31	17	-10	93	-8	89	8	76	**159**	0	65	0	59	2	46	-21	74
129	1	51	-7	73	-34	119	-9	78	**160**	21	37	22	33	19	38	16	66
130	7	50	-2	73	-3	75	-11	83	**161**	-15	72	5	44	-4	66	-23	124
131	28	19	13	46	32	20	9	52	**162**	9	57	14	43	15	38	17	37
132	16	33	9	49	30	22	0	67	**163**	16	54	-1	78	12	42	44	-18
133	14	62	-7	100	-44	127	-5	90	**164**	0	62	0	60	9	34	50	-34
134	-13	108	9	53	0	54	1	67	**165**	12	72	9	69	0	89	-22	127
135	-15	100	2	53	-5	61	-15	72									

Table 9-20 – Values of variables m and n for ctxIdx from 166 to 226

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
166	24	0	11	28	4	45	4	39	**197**	26	-17	28	3	36	-28	28	-3
167	15	9	2	40	10	28	0	42	**198**	30	-25	28	4	38	-28	24	10
168	8	25	3	44	10	31	7	34	**199**	28	-20	32	0	38	-27	27	0
169	13	18	0	49	33	-11	11	29	**200**	33	-23	34	-1	34	-18	34	-14
170	15	9	0	46	52	-43	8	31	**201**	37	-27	30	6	35	-16	52	-44
171	13	19	2	44	18	15	6	37	**202**	33	-23	30	6	34	-14	39	-24
172	10	37	2	51	28	0	7	42	**203**	40	-28	32	9	32	-8	19	17
173	12	18	0	47	35	-22	3	40	**204**	38	-17	31	19	37	-6	31	25
174	6	29	4	39	38	-25	8	33	**205**	33	-11	26	27	35	0	36	29
175	20	33	2	62	34	0	13	43	**206**	40	-15	26	30	30	10	24	33
176	15	30	6	46	39	-18	13	36	**207**	41	-6	37	20	28	18	34	15
177	4	45	0	54	32	-12	4	47	**208**	38	1	28	34	26	25	30	20
178	1	58	3	54	102	-94	3	55	**209**	41	17	17	70	29	41	22	73
179	0	62	2	58	0	0	2	58	**210**	30	-6	1	67	0	75	20	34
180	7	61	4	63	56	-15	6	60	**211**	27	3	5	59	2	72	19	31
181	12	38	6	51	33	-4	8	44	**212**	26	22	9	67	8	77	27	44
182	11	45	6	57	29	10	11	44	**213**	37	-16	16	30	14	35	19	16
183	15	39	7	53	37	-5	14	42	**214**	35	-4	18	32	18	31	15	36
184	11	42	6	52	51	-29	7	48	**215**	38	-8	18	35	17	35	15	36
185	13	44	6	55	39	-9	4	56	**216**	38	-3	22	29	21	30	21	28
186	16	45	11	45	52	-34	4	52	**217**	37	3	24	31	17	45	25	21
187	12	41	14	36	69	-58	13	37	**218**	38	5	23	38	20	42	30	20
188	10	49	8	53	67	-63	9	49	**219**	42	0	18	43	18	45	31	12
189	30	34	-1	82	44	-5	19	58	**220**	35	16	20	41	27	26	27	16
190	18	42	7	55	32	7	10	48	**221**	39	22	11	63	16	54	24	42
191	10	55	-3	78	55	-29	12	45	**222**	14	48	9	59	7	66	0	93
192	17	51	15	46	32	1	0	69	**223**	27	37	9	64	16	56	14	56
193	17	46	22	31	0	0	20	33	**224**	21	60	-1	94	11	73	15	57
194	0	89	-1	84	27	36	8	63	**225**	12	68	-2	89	10	67	26	38
195	26	-19	25	7	33	-25	35	-18	**226**	2	97	-9	108	-10	116	-24	127
196	22	-17	30	-7	34	-30	33	-25									

Table 9-21 – Values of variables m and n for ctxIdx from 227 to 275

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
227	-3	71	-6	76	-23	112	-24	115	**252**	-12	73	-6	55	-16	72	-14	75
228	-6	42	-2	44	-15	71	-22	82	**253**	-8	76	0	58	-7	69	-10	79
229	-5	50	0	45	-7	61	-9	62	**254**	-7	80	0	64	-4	69	-9	83
230	-3	54	0	52	0	53	0	53	**255**	-9	88	-3	74	-5	74	-12	92
231	-2	62	-3	64	-5	66	0	59	**256**	-17	110	-10	90	-9	86	-18	108
232	0	58	-2	59	-11	77	-14	85	**257**	-11	97	0	70	2	66	-4	79
233	1	63	-4	70	-9	80	-13	89	**258**	-20	84	-4	29	-9	34	-22	69
234	-2	72	-4	75	-9	84	-13	94	**259**	-11	79	5	31	1	32	-16	75
235	-1	74	-8	82	-10	87	-11	92	**260**	-6	73	7	42	11	31	-2	58
236	-9	91	-17	102	-34	127	-29	127	**261**	-4	74	1	59	5	52	1	58
237	-5	67	-9	77	-21	101	-21	100	**262**	-13	86	-2	58	-2	55	-13	78
238	-5	27	3	24	-3	39	-14	57	**263**	-13	96	-3	72	-2	67	-9	83
239	-3	39	0	42	-5	53	-12	67	**264**	-11	97	-3	81	0	73	-4	81
240	-2	44	0	48	-7	61	-11	71	**265**	-19	117	-11	97	-8	89	-13	99
241	0	46	0	55	-11	75	-10	77	**266**	-8	78	0	58	3	52	-13	81
242	-16	64	-6	59	-15	77	-21	85	**267**	-5	33	8	5	7	4	-6	38
243	-8	68	-7	71	-17	91	-16	88	**268**	-4	48	10	14	10	8	-13	62
244	-10	78	-12	83	-25	107	-23	104	**269**	-2	53	14	18	17	8	-6	58
245	-6	77	-11	87	-25	111	-15	98	**270**	-3	62	13	27	16	19	-2	59
246	-10	86	-30	119	-28	122	-37	127	**271**	-13	71	2	40	3	37	-16	73
247	-12	92	1	58	-11	76	-10	82	**272**	-10	79	0	58	-1	61	-10	76
248	-15	55	-3	29	-10	44	-8	48	**273**	-12	86	-3	70	-5	73	-13	86
249	-10	60	-1	36	-10	52	-8	61	**274**	-13	90	-6	79	-1	70	-9	83
250	-6	62	1	38	-10	57	-8	66	**275**	-14	97	-8	85	-4	78	-10	87
251	-4	65	2	43	-9	58	-7	70									

Table 9-22 – Values of variables m and n for ctxIdx from 277 to 337

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
277	-6	93	-13	106	-21	126	-22	127	**308**	-16	96	-1	51	-16	77	-10	67
278	-6	84	-16	106	-23	124	-25	127	**309**	-7	88	7	49	-2	64	1	68
279	-8	79	-10	87	-20	110	-25	120	**310**	-8	85	8	52	2	61	0	77
280	0	66	-21	114	-26	126	-27	127	**311**	-7	85	9	41	-6	67	2	64
281	-1	71	-18	110	-25	124	-19	114	**312**	-9	85	6	47	-3	64	0	68
282	0	62	-14	98	-17	105	-23	117	**313**	-13	88	2	55	2	57	-5	78
283	-2	60	-22	110	-27	121	-25	118	**314**	4	66	13	41	-3	65	7	55
284	-2	59	-21	106	-27	117	-26	117	**315**	-3	77	10	44	-3	66	5	59
285	-5	75	-18	103	-17	102	-24	113	**316**	-3	76	6	50	0	62	2	65
286	-3	62	-21	107	-26	117	-28	118	**317**	-6	76	5	53	9	51	14	54
287	-4	58	-23	108	-27	116	-31	120	**318**	10	58	13	49	-1	66	15	44
288	-9	66	-26	112	-33	122	-37	124	**319**	-1	76	4	63	-2	71	5	60
289	-1	79	-10	96	-10	95	-10	94	**320**	-1	83	6	64	-2	75	2	70
290	0	71	-12	95	-14	100	-15	102	**321**	-7	99	-2	69	-1	70	-2	76
291	3	68	-5	91	-8	95	-10	99	**322**	-14	95	-2	59	-9	72	-18	86
292	10	44	-9	93	-17	111	-13	106	**323**	2	95	6	70	14	60	12	70
293	-7	62	-22	94	-28	114	-50	127	**324**	0	76	10	44	16	37	5	64
294	15	36	-5	86	-6	89	-5	92	**325**	-5	74	9	31	0	47	-12	70
295	14	40	9	67	-2	80	17	57	**326**	0	70	12	43	18	35	11	55
296	16	27	-4	80	-4	82	-5	86	**327**	-11	75	3	53	11	37	5	56
297	12	29	-10	85	-9	85	-13	94	**328**	1	68	14	34	12	41	0	69
298	1	44	-1	70	-8	81	-12	91	**329**	0	65	10	38	10	41	2	65
299	20	36	7	60	-1	72	-2	77	**330**	-14	73	-3	52	2	48	-6	74
300	18	32	9	58	5	64	0	71	**331**	3	62	13	40	12	41	5	54
301	5	42	5	61	1	67	-1	73	**332**	4	62	17	32	13	41	7	54
302	1	48	12	50	9	56	4	64	**333**	-1	68	7	44	0	59	-6	76
303	10	62	15	50	0	69	-7	81	**334**	-13	75	7	38	3	50	-11	82
304	17	46	18	49	1	69	5	64	**335**	11	55	13	50	19	40	-2	77
305	9	64	17	54	7	69	15	57	**336**	5	64	10	57	3	66	-2	77
306	-12	104	10	41	-7	69	1	67	**337**	12	70	26	43	18	50	25	42
307	-11	97	7	46	-6	67	0	68									

Table 9-23 – Values of variables m and n for ctxIdx from 338 to 398

ctxIdx	I and SI slices		Value of cabac_init_idc						ctxIdx	I and SI slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
338	15	6	14	11	19	-6	17	-13	**369**	32	-26	31	-4	40	-37	37	-17
339	6	19	11	14	18	-6	16	-9	**370**	37	-30	27	6	38	-30	32	1
340	7	16	9	11	14	0	17	-12	**371**	44	-32	34	8	46	-33	34	15
341	12	14	18	11	26	-12	27	-21	**372**	34	-18	30	10	42	-30	29	15
342	18	13	21	9	31	-16	37	-30	**373**	34	-15	24	22	40	-24	24	25
343	13	11	23	-2	33	-25	41	-40	**374**	40	-15	33	19	49	-29	34	22
344	13	15	32	-15	33	-22	42	-41	**375**	33	-7	22	32	38	-12	31	16
345	15	16	32	-15	37	-28	48	-47	**376**	35	-5	26	31	40	-10	35	18
346	12	23	34	-21	39	-30	39	-32	**377**	33	0	21	41	38	-3	31	28
347	13	23	39	-23	42	-30	46	-40	**378**	38	2	26	44	46	-5	33	41
348	15	20	42	-33	47	-42	52	-51	**379**	33	13	23	47	31	20	36	28
349	14	26	41	-31	45	-36	46	-41	**380**	23	35	16	65	29	30	27	47
350	14	44	46	-28	49	-34	52	-39	**381**	13	58	14	71	25	44	21	62
351	17	40	38	-12	41	-17	43	-19	**382**	29	-3	8	60	12	48	18	31
352	17	47	21	29	32	9	32	11	**383**	26	0	6	63	11	49	19	26
353	24	17	45	-24	69	-71	61	-55	**384**	22	30	17	65	26	45	36	24
354	21	21	53	-45	63	-63	56	-46	**385**	31	-7	21	24	22	22	24	23
355	25	22	48	-26	66	-64	62	-50	**386**	35	-15	23	20	23	22	27	16
356	31	27	65	-43	77	-74	81	-67	**387**	34	-3	26	23	27	21	24	30
357	22	29	43	-19	54	-39	45	-20	**388**	34	3	27	32	33	20	31	29
358	19	35	39	-10	52	-35	35	-2	**389**	36	-1	28	23	26	28	22	41
359	14	50	30	9	41	-10	28	15	**390**	34	5	28	24	30	24	22	42
360	10	57	18	26	36	0	34	1	**391**	32	11	23	40	27	34	16	60
361	7	63	20	27	40	-1	39	1	**392**	35	5	24	32	18	42	15	52
362	-2	77	0	57	30	14	30	17	**393**	34	12	28	29	25	39	14	60
363	-4	82	-14	82	28	26	20	38	**394**	39	11	23	42	18	50	3	78
364	-3	94	-5	75	23	37	18	45	**395**	30	29	19	57	12	70	-16	123
365	9	69	-19	97	12	55	15	54	**396**	34	26	22	53	21	54	21	53
366	-12	109	-35	125	11	65	0	79	**397**	29	39	22	61	14	71	22	56
367	36	-35	27	0	37	-33	36	-16	**398**	19	66	11	86	11	83	25	61
368	36	-34	28	0	39	-36	37	-14									

Table 9-24 – Values of variables m and n for ctxIdx from 402 to 459

ctxIdx	I slices		Value of cabac_init_idc						ctxIdx	I slices		Value of cabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n	m	n
402	-17	120	-4	79	-5	85	-3	78	**431**	-2	55	-12	56	-9	57	-12	59
403	-20	112	-7	71	-6	81	-8	74	**432**	0	61	-6	60	-6	63	-8	63
404	-18	114	-5	69	-10	77	-9	72	**433**	1	64	-5	62	-4	65	-9	67
405	-11	85	-9	70	-7	81	-10	72	**434**	0	68	-8	66	-4	67	-6	68
406	-15	92	-8	66	-17	80	-18	75	**435**	-9	92	-8	76	-7	82	-10	79
407	-14	89	-10	68	-18	73	-12	71	**436**	-14	106	-5	85	-3	81	-3	78
408	-26	71	-19	73	-4	74	-11	63	**437**	-13	97	-6	81	-3	76	-8	74
409	-15	81	-12	69	-10	83	-5	70	**438**	-15	90	-10	77	-7	72	-9	72
410	-14	80	-16	70	-9	71	-17	75	**439**	-12	90	-7	81	-6	78	-10	72
411	0	68	-15	67	-9	67	-14	72	**440**	-18	88	-17	80	-12	72	-18	75
412	-14	70	-20	62	-1	61	-16	67	**441**	-10	73	-18	73	-14	68	-12	71
413	-24	56	-19	70	-8	66	-8	53	**442**	-9	79	-4	74	-3	70	-11	63
414	-23	68	-16	66	-14	66	-14	59	**443**	-14	86	-10	83	-6	76	-5	70
415	-24	50	-22	65	0	59	-9	52	**444**	-10	73	-9	71	-5	66	-17	75
416	-11	74	-20	63	2	59	-11	68	**445**	-10	70	-9	67	-5	62	-14	72
417	23	-13	9	-2	17	-10	9	-2	**446**	-10	69	-1	61	0	57	-16	67
418	26	-13	26	-9	32	-13	30	-10	**447**	-5	66	-8	66	-4	61	-8	53
419	40	-15	33	-9	42	-9	31	-4	**448**	-9	64	-14	66	-9	60	-14	59
420	49	-14	39	-7	49	-5	33	-1	**449**	-5	58	0	59	1	54	-9	52
421	44	3	41	-2	53	0	33	7	**450**	2	59	2	59	2	58	-11	68
422	45	6	45	3	64	3	31	12	**451**	21	-10	21	-13	17	-10	9	-2
423	44	34	49	9	68	10	37	23	**452**	24	-11	33	-14	32	-13	30	-10
424	33	54	45	27	66	27	31	38	**453**	28	-8	39	-7	42	-9	31	-4
425	19	82	36	59	47	57	20	64	**454**	28	-1	46	-2	49	-5	33	-1
426	-3	75	-6	66	-5	71	-9	71	**455**	29	3	51	2	53	0	33	7
427	-1	23	-7	35	0	24	-7	37	**456**	29	9	60	6	64	3	31	12
428	1	34	-7	42	-1	36	-8	44	**457**	35	20	61	17	68	10	37	23
429	1	43	-8	45	-2	42	-11	49	**458**	29	36	55	34	66	27	31	38
430	0	54	-5	48	-2	52	-10	56	**459**	14	67	42	62	47	57	20	64

9.3.1.2 Initialisation process for the arithmetic decoding engine

This process is invoked before decoding the first macroblock of a slice or after the decoding of any pcm_alignment_zero_bit and all pcm_sample_luma and pcm_sample_chroma data for a macroblock of type I_PCM.

Outputs of this process are the initialised decoding engine registers codIRange and codIOffset both in 16 bit register precision.

The status of the arithmetic decoding engine is represented by the variables codIRange and codIOffset. In the initialisation procedure of the arithmetic decoding process, codIRange is set equal to 0x01FE and codIOffset is set equal

to the value returned from read_bits(9) interpreted as a 9 bit binary representation of an unsigned integer with most significant bit written first.

The bitstream shall not contain data that results in a value of codIOffset being equal to 0x01FE or 0x01FF.

NOTE – The description of the arithmetic decoding engine in this Recommendation | International Standard utilizes 16 bit register precision. However, the minimum register precision for the variables codIRange and codIOffset is 9 bits.

9.3.2 Binarization process

Input to this process is a request for a syntax element.

Output of this process is the binarization of the syntax element, maxBinIdxCtx, ctxIdxOffset, and bypassFlag.

Table 9-25 specifies the type of binarization process, maxBinIdxCtx, and ctxIdxOffset associated with each syntax element.

The specification of the unary (U) binarization process, the truncated unary (TU) binarization process, the concatenated unary / k-th order Exp-Golomb (UEGk) binarization process, and the fixed-length (FL) binarization process are given in subclauses 9.3.2.1 to 9.3.2.4, respectively. Other binarizations are specified in subclauses 9.3.2.5 to 9.3.2.7.

Except for I slices, the binarizations for the syntax element mb_type as specified in subclause 9.3.2.5 consist of bin strings given by a concatenation of prefix and suffix bit strings. The UEGk binarization as specified in 9.3.2.3, which is used for the binarization of the syntax elements mvd_lX (X = 0, 1) and coeff_abs_level_minus1, and the binarization of the coded_block_pattern also consist of a concatenation of prefix and suffix bit strings. For these binarization processes, the prefix and the suffix bit string are separately indexed using the binIdx variable as specified further in subclause 9.3.3. The two sets of prefix bit strings and suffix bit strings are referred to as the binarization prefix part and the binarization suffix part, respectively.

Associated with each binarization or binarization part of a syntax element is a specific value of the context index offset (ctxIdxOffset) variable and a specific value of the maxBinIdxCtx variable as given in Table 9-25. When two values for each of these variables are specified for one syntax element in Table 9-25, the value in the upper row is related to the prefix part while the value in the lower row is related to the suffix part of the binarization of the corresponding syntax element.

The use of the DecodeBypass process and the variable bypassFlag is derived as follows.

– If no value is assigned to ctxIdxOffset for the corresponding binarization or binarization part in Table 9-25 labelled as "na", all bins of the bit strings of the corresponding binarization or of the binarization prefix/suffix part are decoded by invoking the DecodeBypass process as specified in subclause 9.3.3.2.3. In such a case, bypassFlag is set equal to 1, where bypassFlag is used to indicate that for parsing the value of the bin from the bitstream the DecodeBypass process is applied.

– Otherwise, for each possible value of binIdx up to the specified value of maxBinIdxCtx given in Table 9-25, a specific value of the variable ctxIdx is further specified in subclause 9.3.3. bypassFlag is set equal to 0.

The possible values of the context index ctxIdx are in the range 0 to 459, inclusive. The value assigned to ctxIdxOffset specifies the lower value of the range of ctxIdx assigned to the corresponding binarization or binarization part of a syntax element.

ctxIdx = ctxIdxOffset = 276 is assigned to the syntax element end_of_slice_flag and the bin of mb_type, which specifies the I_PCM macroblock type as further specified in subclause 9.3.3.1. For parsing the value of the corresponding bin from the bitstream, the arithmetic decoding process for decisions before termination (DecodeTerminate) as specified in subclause 9.3.3.2.4 is applied.

NOTE – The bins of mb_type in I slices and the bins of the suffix for mb_type in SI slices that correspond to the same value of binIdx share the same ctxIdx. The last bin of the prefix of mb_type and the first bin of the suffix of mb_type in P, SP, and B slices may share the same ctxIdx.

Table 9-25 – Syntax elements and associated types of binarization, maxBinIdxCtx, and ctxIdxOffset

Syntax element	Type of binarization	maxBinIdxCtx	ctxIdxOffset
mb_type (SI slices only)	prefix and suffix as specified in subclause 9.3.2.5	prefix: 0 suffix: 6	prefix: 0 suffix: 3
mb_type (I slices only)	as specified in subclause 9.3.2.5	6	3
mb_skip_flag (P, SP slices only)	FL, cMax=1	0	11
mb_type (P, SP slices only)	prefix and suffix as specified in subclause 9.3.2.5	prefix: 2 suffix: 5	prefix: 14 suffix: 17
sub_mb_type (P, SP slices only)	as specified in subclause 9.3.2.5	2	21
mb_skip_flag (B slices only)	FL, cMax=1	0	24
mb_type (B slices only)	prefix and suffix as specified in subclause 9.3.2.5	prefix: 3 suffix: 5	prefix: 27 suffix: 32
sub_mb_type (B slices only)	as specified in subclause 9.3.2.5	3	36
mvd_l0[ ][ ][0], mvd_l1[ ][ ][0]	prefix and suffix as given by UEG3 with signedValFlag=1, uCoff=9	prefix: 4 suffix: na	prefix: 40 suffix: na (uses DecodeBypass)
mvd_l0[ ][ ][1], mvd_l1[ ][ ][1]		prefix: 4 suffix: na	prefix: 47 suffix: na (uses DecodeBypass)
ref_idx_l0, ref_idx_l1	U	2	54
mb_qp_delta	as specified in subclause 9.3.2.7	2	60
intra_chroma_pred_mode	TU, cMax=3	1	64
prev_intra4x4_pred_mode_flag, prev_intra8x8_pred_mode_flag	FL, cMax=1	0	68
rem_intra4x4_pred_mode, rem_intra8x8_pred_mode	FL, cMax=7	0	69
mb_field_decoding_flag	FL, cMax=1	0	70
coded_block_pattern	prefix and suffix as specified in subclause 9.3.2.6	prefix: 3 suffix: 1	prefix: 73 suffix: 77
coded_block_flag	FL, cMax=1	0	85
significant_coeff_flag (frame coded blocks with ctxBlockCat < 5)	FL, cMax=1	0	105
last_significant_coeff_flag (frame coded blocks with ctxBlockCat < 5)	FL, cMax=1	0	166
coeff_abs_level_minus1 (blocks with ctxBlockCat < 5)	prefix and suffix as given by UEG0 with signedValFlag=0, uCoff=14	prefix: 1 suffix: na	prefix: 227 suffix: na, (uses DecodeBypass)
coeff_sign_flag	FL, cMax=1	0	na, (uses DecodeBypass)
end_of_slice_flag	FL, cMax=1	0	276
significant_coeff_flag (field coded blocks with ctxBlockCat < 5)	FL, cMax=1	0	277

Syntax element	Type of binarization	maxBinIdxCtx	ctxIdxOffset
last_significant_coeff_flag (field coded blocks with ctxBlockCat < 5)	FL, cMax=1	0	338
transform_size_8x8_flag	FL, cMax=1	0	399
significant_coeff_flag (frame coded blocks with ctxBlockCat == 5)	FL, cMax=1	0	402
last_significant_coeff_flag (frame coded blocks with ctxBlockCat == 5)	FL, cMax=1	0	417
coeff_abs_level_minus1 (blocks with ctxBlockCat == 5)	prefix and suffix as given by UEG0 with signedValFlag=0, uCoff=14	prefix: 1 suffix: na	prefix: 426 suffix: na, (uses DecodeBypass)
significant_coeff_flag (field coded blocks with ctxBlockCat == 5)	FL, cMax=1	0	436
last_significant_coeff_flag (field coded blocks with ctxBlockCat == 5)	FL, cMax=1	0	451

9.3.2.1 Unary (U) binarization process

Input to this process is a request for a U binarization for a syntax element.

Output of this process is the U binarization of the syntax element.

The bin string of a syntax element having (unsigned integer) value synElVal is a bit string of length synElVal + 1 indexed by BinIdx. The bins for binIdx less than synElVal are equal to 1. The bin with binIdx equal to synElVal is equal to 0.

Table 9-26 illustrates the bin strings of the unary binarization for a syntax element.

Table 9-26 – Bin string of the unary binarization (informative)

Value of syntax element	Bin string					
0 (I_NxN)	0					
1	1	0				
2	1	1	0			
3	1	1	1	0		
4	1	1	1	1	0	
5	1	1	1	1	1	0
…						
binIdx	0	1	2	3	4	5

9.3.2.2 Truncated unary (TU) binarization process

Input to this process is a request for a TU binarization for a syntax element and cMax.

Output of this process is the TU binarization of the syntax element.

For syntax element (unsigned integer) values less than cMax, the U binarization process as specified in subclause 9.3.2.1 is invoked. For the syntax element value equal to cMax the bin string is a bit string of length cMax with all bins being equal to 1.

NOTE – TU binarization is always invoked with a cMax value equal to the largest possible value of the syntax element being decoded.

9.3.2.3 Concatenated unary/ k-th order Exp-Golomb (UEGk) binarization process

Input to this process is a request for a UEGk binarization for a syntax element, signedValFlag and uCoff.

Output of this process is the UEGk binarization of the syntax element.

A UEGk bin string is a concatenation of a prefix bit string and a suffix bit string. The prefix of the binarization is specified by invoking the TU binarization process for the prefix part Min(uCoff, Abs(synElVal)) of a syntax element value synElVal as specified in subclause 9.3.2.2 with cMax = uCoff, where uCoff > 0.

The UEGk bin string is derived as follows.

- If one of the following is true, the bin string of a syntax element having value synElVal consists only of a prefix bit string,
    - signedValFlag is equal to 0 and the prefix bit string is not equal to the bit string of length uCoff with all bits equal to 1.
    - signedValFlag is equal to 1 and the prefix bit string is equal to the bit string that consists of a single bit with value equal to 0.
- Otherwise, the bin string of the UEGk suffix part of a syntax element value synElVal is specified by a process equivalent to the following pseudo-code:

```
if( Abs( synElVal ) >= uCoff ) {
    sufS = Abs( synElVal ) − uCoff
    stopLoop = 0
    do {
        if( sufS >= ( 1 << k ) ) {
            put( 1 )
            sufS = sufS − ( 1<<k )
            k++
        } else {
            put( 0 )
            while( k− − )
                put( ( sufS >> k ) & 0x01 )
            stopLoop = 1
        }
    } while( !stopLoop )
}
if( signedValFlag && synElVal != 0)
    if( synElVal > 0 )
        put( 0 )
    else
        put( 1 )
```

NOTE – The specification for the k-th order Exp-Golomb (EGk) code uses 1's and 0's in reverse meaning for the unary part of the Exp-Golomb code of 0-th order as specified in subclause 9.1.

9.3.2.4 Fixed-length (FL) binarization process

Input to this process is a request for a FL binarization for a syntax element and cMax.

Output of this process is the FL binarization of the syntax element.

FL binarization is constructed by using an fixedLength-bit unsigned integer bin string of the syntax element value, where fixedLength = Ceil(Log2(cMax + 1)). The indexing of bins for the FL binarization is such that the binIdx = 0 relates to the least significant bit with increasing values of binIdx towards the most significant bit.

9.3.2.5 Binarization process for macroblock type and sub-macroblock type

Input to this process is a request for a binarization for syntax elements mb_type or sub_mb_type.

Output of this process is the binarization of the syntax element.

The binarization scheme for decoding of macroblock type in I slices is specified in Table 9-27.

For macroblock types in SI slices, the binarization consists of bin strings specified as a concatenation of a prefix and a suffix bit string as follows.

The prefix bit string consists of a single bit, which is specified by b_0 = ((mb_type == SI) ? 0 : 1). For the syntax element value for which b_0 is equal to 0, the bin string only consists of the prefix bit string. For the syntax element value for which b_0 is equal to 1, the binarization is given by concatenating the prefix b_0 and the suffix bit string as specified in Table 9-27 for macroblock type in I slices indexed by subtracting 1 from the value of mb_type in SI slices.

Table 9-27 – Binarization for macroblock types in I slices

Value (name) of mb_type	Bin string						
0 (I_4x4)	0						
1 (I_16x16_0_0_0)	1	0	0	0	0	0	
2 (I_16x16_1_0_0)	1	0	0	0	0	1	
3 (I_16x16_2_0_0)	1	0	0	0	1	0	
4 (I_16x16_3_0_0)	1	0	0	0	1	1	
5 (I_16x16_0_1_0)	1	0	0	1	0	0	0
6 (I_16x16_1_1_0)	1	0	0	1	0	0	1
7 (I_16x16_2_1_0)	1	0	0	1	0	1	0
8 (I_16x16_3_1_0)	1	0	0	1	0	1	1
9 (I_16x16_0_2_0)	1	0	0	1	1	0	0
10 (I_16x16_1_2_0)	1	0	0	1	1	0	1
11 (I_16x16_2_2_0)	1	0	0	1	1	1	0
12 (I_16x16_3_2_0)	1	0	0	1	1	1	1
13 (I_16x16_0_0_1)	1	0	1	0	0	0	
14 (I_16x16_1_0_1)	1	0	1	0	0	1	
15 (I_16x16_2_0_1)	1	0	1	0	1	0	
16 (I_16x16_3_0_1)	1	0	1	0	1	1	
17 (I_16x16_0_1_1)	1	0	1	1	0	0	0
18 (I_16x16_1_1_1)	1	0	1	1	0	0	1
19 (I_16x16_2_1_1)	1	0	1	1	0	1	0
20 (I_16x16_3_1_1)	1	0	1	1	0	1	1
21 (I_16x16_0_2_1)	1	0	1	1	1	0	0
22 (I_16x16_1_2_1)	1	0	1	1	1	0	1
23 (I_16x16_2_2_1)	1	0	1	1	1	1	0
24 (I_16x16_3_2_1)	1	0	1	1	1	1	1
25 (I_PCM)	1	1					
binIdx	0	1	2	3	4	5	6

The binarization schemes for P macroblock types in P and SP slices and for B macroblocks in B slices are specified in Table 9-28.

The bin string for I macroblock types in P and SP slices corresponding to mb_type values 5 to 30 consists of a concatenation of a prefix, which consists of a single bit with value equal to 1 as specified in Table 9-28 and a suffix as specified in Table 9-27, indexed by subtracting 5 from the value of mb_type.

mb_type equal to 4 (P_8x8ref0) is not allowed.

For I macroblock types in B slices (mb_type values 23 to 48) the binarization consists of bin strings specified as a concatenation of a prefix bit string as specified in Table 9-28 and suffix bit strings as specified in Table 9-27, indexed by subtracting 23 from the value of mb_type.

Table 9-28 – Binarization for macroblock types in P, SP, and B slices

Slice type	Value (name) of mb_type	Bin string						
P, SP slice	0 (P_L0_16x16)	0	0	0				
	1 (P_L0_L0_16x8)	0	1	1				
	2 (P_L0_L0_8x16)	0	1	0				
	3 (P_8x8)	0	0	1				
	4 (P_8x8ref0)	na						
	5 to 30 (Intra, prefix only)	1						
B slice	0 (B_Direct_16x16)	0						
	1 (B_L0_16x16)	1	0	0				
	2 (B_L1_16x16)	1	0	1				
	3 (B_Bi_16x16)	1	1	0	0	0	0	
	4 (B_L0_L0_16x8)	1	1	0	0	0	1	
	5 (B_L0_L0_8x16)	1	1	0	0	1	0	
	6 (B_L1_L1_16x8)	1	1	0	0	1	1	
	7 (B_L1_L1_8x16)	1	1	0	1	0	0	
	8 (B_L0_L1_16x8)	1	1	0	1	0	1	
	9 (B_L0_L1_8x16)	1	1	0	1	1	0	
	10 (B_L1_L0_16x8)	1	1	0	1	1	1	
	11 (B_L1_L0_8x16)	1	1	1	1	1	0	
	12 (B_L0_Bi_16x8)	1	1	1	0	0	0	0
	13 (B_L0_Bi_8x16)	1	1	1	0	0	0	1
	14 (B_L1_Bi_16x8)	1	1	1	0	0	1	0
	15 (B_L1_Bi_8x16)	1	1	1	0	0	1	1
	16 (B_Bi_L0_16x8)	1	1	1	0	1	0	0
	17 (B_Bi_L0_8x16)	1	1	1	0	1	0	1
	18 (B_Bi_L1_16x8)	1	1	1	0	1	1	0
	19 (B_Bi_L1_8x16)	1	1	1	0	1	1	1
	20 (B_Bi_Bi_16x8)	1	1	1	1	0	0	0
	21 (B_Bi_Bi_8x16)	1	1	1	1	0	0	1
	22 (B_8x8)	1	1	1	1	1	1	
	23 to 48 (Intra, prefix only)	1	1	1	1	0	1	
binIdx		0	1	2	3	4	5	6

For P, SP, and B slices the specification of the binarization for sub_mb_type is given in Table 9-29.

Table 9-29 – Binarization for sub-macroblock types in P, SP, and B slices

Slice type	Value (name) of sub_mb_type	Bin string					
P, SP slice	0 (P_L0_8x8)	1					
	1 (P_L0_8x4)	0	0				
	2 (P_L0_4x8)	0	1	1			
	3 (P_L0_4x4)	0	1	0			
B slice	0 (B_Direct_8x8)	0					
	1 (B_L0_8x8)	1	0	0			
	2 (B_L1_8x8)	1	0	1			
	3 (B_Bi_8x8)	1	1	0	0	0	
	4 (B_L0_8x4)	1	1	0	0	1	
	5 (B_L0_4x8)	1	1	0	1	0	
	6 (B_L1_8x4)	1	1	0	1	1	
	7 (B_L1_4x8)	1	1	1	0	0	0
	8 (B_Bi_8x4)	1	1	1	0	0	1
	9 (B_Bi_4x8)	1	1	1	0	1	0
	10 (B_L0_4x4)	1	1	1	0	1	1
	11 (B_L1_4x4)	1	1	1	1	0	
	12 (B_Bi_4x4)	1	1	1	1	1	
binIdx		0	1	2	3	4	5

9.3.2.6 Binarization process for coded block pattern

Input to this process is a request for a binarization for the syntax element coded_block_pattern.

Output of this process is the binarization of the syntax element.

The binarization of coded_block_pattern consists of a prefix part and (when present) a suffix part. The prefix part of the binarization is given by the FL binarization of CodedBlockPatternLuma with cMax = 15. When chroma_format_idc is not equal to 0 (monochrome), the suffix part is present and consists of the TU binarization of CodedBlockPatternChroma with cMax = 2. The relationship between the value of the syntax element coded_block_pattern and the values of CodedBlockPatternLuma and CodedBlockPatternChroma is given as specified in subclause 7.4.5.

9.3.2.7 Binarization process for mb_qp_delta

Input to this process is a request for a binarization for the syntax element mb_qp_delta.

Output of this process is the binarization of the syntax element.

The bin string of mb_qp_delta is derived by the U binarization of the mapped value of the syntax element mb_qp_delta, where the assignment rule between the signed value of mb_qp_delta and its mapped value is given as specified in Table 9-3.

9.3.3 Decoding process flow

Input to this process is a binarization of the requested syntax element, maxBinIdxCtx, bypassFlag and ctxIdxOffset as specified in subclause 9.3.2.

Output of this process is the value of the syntax element.

This process specifies how each bit of a bit string is parsed for each syntax element.

After parsing each bit, the resulting bit string is compared to all bin strings of the binarization of the syntax element and the following applies.

– If the bit string is equal to one of the bin strings, the corresponding value of the syntax element is the output.

– Otherwise (the bit string is not equal to one of the bin strings), the next bit is parsed.

While parsing each bin, the variable binIdx is incremented by 1 starting with binIdx being set equal to 0 for the first bin.

When the binarization of the corresponding syntax element consists of a prefix and a suffix binarization part,, the variable binIdx is set equal to 0 for the first bin of each part of the bin string (prefix part or suffix part). In this case, after parsing the prefix bit string, the parsing process of the suffix bit string related to the binarizations specified in subclauses 9.3.2.3 and 9.3.2.5 is invoked depending on the resulting prefix bit string as specified in subclauses 9.3.2.3 and 9.3.2.5. Note that for the binarization of the syntax element coded_block_pattern, the suffix bit string is present regardless of the prefix bit string of length 4 as specified in subclause 9.3.2.6.

Depending on the variable bypassFlag, the following applies.

– If bypassFlag is equal to 1, the bypass decoding process as specified in subclause 9.3.3.2.3 is applied for parsing the value of the bins from the bitstream.

– Otherwise (bypassFlag is equal to 0), the parsing of each bin is specified by the following two ordered steps:

1. Given binIdx, maxBinIdxCtx and ctxIdxOffset, ctxIdx is derived as specified in subclause 9.3.3.1.
2. Given ctxIdx, the value of the bin from the bitstream as specified in subclause 9.3.3.2 is decoded.

9.3.3.1 Derivation process for ctxIdx

Inputs to this process are binIdx, maxBinIdxCtx and ctxIdxOffset.

Output of this process is ctxIdx.

Table 9-30 shows the assignment of ctxIdx increments (ctxIdxInc) to binIdx for all ctxIdxOffset values except those related to the syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1.

The ctxIdx to be used with a specific binIdx is specified by first determining the ctxIdxOffset associated with the given bin string or part thereof. The ctxIdx is determined as follows.

– If the ctxIdxOffset is listed in Table 9-30, the ctxIdx for a binIdx is the sum of ctxIdxOffset and ctxIdxInc, which is found in Table 9-30. When more than one value is listed in Table 9-30 for a binIdx, the assignment process for ctxIdxInc for that binIdx is further specified in the subclauses given in parenthesis of the corresponding table entry.

– Otherwise (ctxIdxOffset is not listed in Table 9-30), the ctxIdx is specified to be the sum of the following terms: ctxIdxOffset and ctxIdxBlockCatOffset(ctxBlockCat) as specified in Table 9-31 and ctxIdxInc(ctxBlockCat). Subclause 9.3.3.1.3 specifies which ctxBlockCat is used. Subclause 9.3.3.1.1.9 specifies the assignment of ctxIdxInc(ctxBlockCat) for coded_block_flag, and subclause 9.3.3.1.3 specifies the assignment of ctxIdxInc(ctxBlockCat) for significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1.

All bins with binIdx greater than maxBinIdxCtx are parsed using the value of ctxIdx being assigned to binIdx equal to maxBinIdxCtx.

All entries in Table 9-30 labelled with "na" correspond to values of binIdx that do not occur for the corresponding ctxIdxOffset.

ctxIdx = 276 is assigned to the binIdx of mb_type indicating the I_PCM mode. For parsing the value of the corresponding bins from the bitstream, the arithmetic decoding process for decisions before termination as specified in subclause 9.3.3.2.4 is applied.

Table 9-30 – Assignment of ctxIdxInc to binIdx for all ctxIdxOffset values except those related to the syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1

ctxIdxOffset	binIdx						
	0	1	2	3	4	5	>= 6
0	0,1,2 (subclause 9.3.3.1.1.3)	na	na	na	na	na	na
3	0,1,2 (subclause 9.3.3.1.1.3)	ctxIdx=276	3	4	5,6 (subclause 9.3.3.1.2)	6,7 (subclause 9.3.3.1.2)	7
11	0,1,2 (subclause 9.3.3.1.1.1)	na	na	na	na	na	na
14	0	1	2,3 (subclause 9.3.3.1.2)	na	na	na	na
17	0	ctxIdx=276	1	2	2,3 (subclause 9.3.3.1.2)	3	3
21	0	1	2	na	na	na	na
24	0,1,2 (subclause 9.3.3.1.1.1)	na	na	na	na	na	na
27	0,1,2 (subclause 9.3.3.1.1.3)	3	4,5 (subclause 9.3.3.1.2)	5	5	5	5
32	0	ctxIdx=276	1	2	2,3 (subclause 9.3.3.1.2)	3	3
36	0	1	2,3 (subclause 9.3.3.1.2)	3	3	3	na
40	0,1,2 (subclause 9.3.3.1.1.7)	3	4	5	6	6	6
47	0,1,2 (subclause 9.3.3.1.1.7)	3	4	5	6	6	6
54	0,1,2,3 (subclause 9.3.3.1.1.6)	4	5	5	5	5	5
60	0,1 (subclause 9.3.3.1.1.5)	2	3	3	3	3	3
64	0,1,2 (subclause 9.3.3.1.1.8)	3	3	na	na	na	na
68	0	na	na	na	na	na	na
69	0	0	0	na	na	na	na
70	0,1,2 (subclause 9.3.3.1.1.2)	na	na	na	na	na	na
73	0,1,2,3 (subclause 9.3.3.1.1.4)	0,1,2,3 (subclause 9.3.3.1.1.4)	0,1,2,3 (subclause 9.3.3.1.1.4)	0,1,2,3 (subclause 9.3.3.1.1.4)	na	na	na
77	0,1,2,3 (subclause 9.3.3.1.1.4)	4,5,6,7 (subclause 9.3.3.1.1.4)	na	na	na	na	na
276	0	na	na	na	na	na	na
399	0,1,2 (subclause 9.3.3.1.1.10)	na	na	na	na	na	na

Table 9-31 shows the values of ctxIdxBlockCatOffset depending on ctxBlockCat for the syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1. The specification of ctxBlockCat is given in Table 9-33.

Table 9-31 – Assignment of ctxIdxBlockCatOffset to ctxBlockCat for syntax elements coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1

Syntax element	ctxBlockCat (as specified in Table 9-33)					
	0	**1**	**2**	**3**	**4**	**5**
coded_block_flag	0	4	8	12	16	na
significant_coeff_flag	0	15	29	44	47	0
last_significant_coeff_flag	0	15	29	44	47	0
coeff_abs_level_minus1	0	10	20	30	39	0

9.3.3.1.1 Assignment process of ctxIdxInc using neighbouring syntax elements

Subclause 9.3.3.1.1.1 specifies the derivation process of ctxIdxInc for the syntax element mb_skip_flag.

Subclause 9.3.3.1.1.2 specifies the derivation process of ctxIdxInc for the syntax element mb_field_decoding_flag.

Subclause 9.3.3.1.1.3 specifies the derivation process of ctxIdxInc for the syntax element mb_type.

Subclause 9.3.3.1.1.4 specifies the derivation process of ctxIdxInc for the syntax element coded_block_pattern.

Subclause 9.3.3.1.1.5 specifies the derivation process of ctxIdxInc for the syntax element mb_qp_delta.

Subclause 9.3.3.1.1.6 specifies the derivation process of ctxIdxInc for the syntax elements ref_idx_l0 and ref_idx_l1.

Subclause 9.3.3.1.1.7 specifies the derivation process of ctxIdxInc for the syntax elements mvd_l0 and mvd_l1.

Subclause 9.3.3.1.1.8 specifies the derivation process of ctxIdxInc for the syntax element intra_chroma_pred_mode.

Subclause 9.3.3.1.1.9 specifies the derivation process of ctxIdxInc for the syntax element coded_block_flag.

Subclause 9.3.3.1.1.10 specifies the derivation process of ctxIdxInc for the syntax element transform_size_8x8_flag.

9.3.3.1.1.1 Derivation process of ctxIdxInc for the syntax element mb_skip_flag

Output of this process is ctxIdxInc.

When MbaffFrameFlag is equal to 1 and mb_field_decoding_flag has not been decoded (yet) for the current macroblock pair with top macroblock address 2 * (CurrMbAddr / 2), the inference rule for the syntax element mb_field_decoding_flag as specified in subclause 7.4.4 is applied.

The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If mbAddrN is not available or mb_skip_flag for the macroblock mbAddrN is equal to 1, condTermFlagN is set equal to 0.

– Otherwise (mbAddrN is available and mb_skip_flag for the macroblock mbAddrN is equal to 0), condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived by

$$ctxIdxInc = condTermFlagA + condTermFlagB \tag{9-1}$$

9.3.3.1.1.2 Derivation process of ctxIdxInc for the syntax element mb_field_decoding_flag

Output of this process is ctxIdxInc.

The derivation process for neighbouring macroblock addresses and their availability in MBAFF frames as specified in subclause 6.4.7 is invoked and the output is assigned to mbAddrA and mbAddrB.

When both macroblocks mbAddrN and mbAddrN + 1 have mb_type equal to P_Skip or B_Skip, the inference rule for the syntax element mb_field_decoding_flag as specified in subclause 7.4.4 is applied for the macroblock mbAddrN.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If any of the following conditions is true, condTermFlagN is set equal to 0,
  – mbAddrN is not available
  – the macroblock mbAddrN is a frame macroblock.
– Otherwise, condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived by

$$ctxIdxInc = condTermFlagA + condTermFlagB \quad (9\text{-}2)$$

9.3.3.1.1.3 Derivation process of ctxIdxInc for the syntax element mb_type

Input to this process is ctxIdxOffset.

Output of this process is ctxIdxInc.

The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If any of the following conditions is true, condTermFlagN is set equal to 0
  – mbAddrN is not available
  – ctxIdxOffset is equal to 0 and mb_type for the macroblock mbAddrN is equal to SI
  – ctxIdxOffset is equal to 3 and mb_type for the macroblock mbAddrN is equal to I_NxN
  – ctxIdxOffset is equal to 27 and mb_type for the macroblock mbAddrN is equal to P_Skip, B_Skip, or B_Direct_16x16
– Otherwise, condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived as

$$ctxIdxInc = condTermFlagA + condTermFlagB \quad (9\text{-}3)$$

9.3.3.1.1.4 Derivation process of ctxIdxInc for the syntax element coded_block_pattern

Inputs to this process are ctxIdxOffset and binIdx.

Output of this process is ctxIdxInc.

Depending on the value of the variable ctxIdxOffset, the following applies.

– If ctxIdxOffset is equal to 73, the following applies
  – The derivation process for neighbouring 8x8 luma blocks specified in subclause 6.4.8.2 is invoked with luma8x8BlkIdx = binIdx as input and the output is assigned to mbAddrA, mbAddrB, luma8x8BlkIdxA, and luma8x8BlkIdxB.
  – Let the variable condTermFlagN (with N being either A or B) be derived as follows.
    – If any of the following conditions are true, condTermFlagN is set equal to 0
      – mbAddrN is not available
      – mb_type for the macroblock mbAddrN is equal to I_PCM
      – the macroblock mbAddrN is not the current macroblock CurrMbAddr and the macroblock mbAddrN does not have mb_type equal to P_Skip or B_Skip, and ((CodedBlockPatternLuma >> luma8x8BlkIdxN) & 1) is not equal to 0 for the value of CodedBlockPatternLuma for the macroblock mbAddrN

– the macroblock mbAddrN is the current macroblock CurrMbAddr and the prior decoded bin value b_k of coded_block_pattern with k = luma8x8BlkIdxN is not equal to 0.

– Otherwise, condTermFlagN is set equal to 1.

– The variable ctxIdxInc is derived as

$$ctxIdxInc = condTermFlagA + 2 * condTermFlagB \quad (9\text{-}4)$$

– Otherwise (ctxIdxOffset is equal to 77), the following applies.

– The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

– Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If mbAddrN is available and mb_type for the macroblock mbAddrN is equal to I_PCM, condTermFlagN is set equal to 1

– Otherwise, if any of the following conditions is true, condTermFlagN is set equal to 0

– mbAddrN is not available or the macroblock mbAddrN has mb_type equal to P_Skip or B_Skip

– binIdx is equal to 0 and CodedBlockPatternChroma for the macroblock mbAddrN is equal to 0

– binIdx is equal to 1 and CodedBlockPatternChroma for the macroblock mbAddrN is not equal to 2

– Otherwise, condTermFlagN is set equal to 1.

– The variable ctxIdxInc is derived as

$$ctxIdxInc = condTermFlagA + 2 * condTermFlagB + ((binIdx == 1) ? 4 : 0) \quad (9\text{-}5)$$

NOTE – When a macroblock uses an Intra_16x16 prediction mode, the values of CodedBlockPatternLuma and CodedBlockPatternChroma for the macroblock are derived from mb_type as specified in Table 7-11.

9.3.3.1.1.5 Derivation process of ctxIdxInc for the syntax element mb_qp_delta

Output of this process is ctxIdxInc.

Let prevMbAddr be the macroblock address of the macroblock that precedes the current macroblock in decoding order. When the current macroblock is the first macroblock of a slice, prevMbAddr is marked as not available.

Let the variable ctxIdxInc be derived as follows.

– If any of the following conditions is true, ctxIdxInc is set equal to 0

– prevMbAddr is not available or the macroblock prevMbAddr has mb_type equal to P_Skip or B_Skip

– mb_type of the macroblock prevMbAddr is equal to I_PCM

– The macroblock prevMbAddr is not coded in Intra_16x16 prediction mode and both CodedBlockPatternLuma and CodedBlockPatternChroma for the macroblock prevMbAddr are equal to 0

– mb_qp_delta for the macroblock prevMbAddr is equal to 0

– Otherwise, ctxIdxInc is set equal to 1.

9.3.3.1.1.6 Derivation process of ctxIdxInc for the syntax elements ref_idx_l0 and ref_idx_l1

Input to this process is mbPartIdx.

Output of this process is ctxIdxInc.

The interpretation of ref_idx_lX and Pred_LX within this subclause is specified as follows.

– If this process is invoked for the derivation of ref_idx_l0, ref_idx_lX is interpreted as ref_idx_l0 and Pred_LX is interpreted as Pred_L0.

– Otherwise (this process is invoked for the derivation of ref_idx_l1), ref_idx_lX is interpreted as ref_idx_l1 and Pred_LX is interpreted as Pred_L1.

Let currSubMbType be set equal to sub_mb_type[mbPartIdx].

The derivation process for neighbouring partitions specified in subclause 6.4.8.5 is invoked with mbPartIdx, currSubMbType, and subMbPartIdx = 0 as input and the output is assigned to mbAddrA\mbPartIdxA and mbAddrB\mbPartIdxB.

With ref_idx_lX[mbPartIdxN] (with N being either A or B) specifying the syntax element for the macroblock mbAddrN, let the variable refIdxZeroFlagN be derived as follows.

– If MbaffFrameFlag is equal to 1, the current macroblock is a frame macroblock, and the macroblock mbAddrN is a field macroblock

$$refIdxZeroFlagN = ((ref_idx_lX[mbPartIdxN] > 1) ? 0 : 1) \quad (9\text{-}6)$$

– Otherwise,

$$refIdxZeroFlagN = ((ref_idx_lX[mbPartIdxN] > 0) ? 0 : 1) \quad (9\text{-}7)$$

Let the variable predModeEqualFlagN be specified as follows.

– If the macroblock mbAddrN has mb_type equal to P_8x8 or B_8x8, the following applies.
  – If SubMbPredMode(sub_mb_type[mbPartIdxN]) is not equal to Pred_LX and not equal to BiPred, predModeEqualFlagN is set equal to 0, where sub_mb_type specifies the syntax element for the macroblock mbAddrN.
  – Otherwise, predModeEqualFlagN is set equal to 1.
– Otherwise, the following applies.
  – If MbPartPredMode(mb_type, mbPartIdxN) is not equal to Pred_LX and not equal to BiPred, predModeEqualFlagN is set equal to 0, where mb_type specifies the syntax element for the macroblock mbAddrN.
  – Otherwise, predModeEqualFlagN is set equal to 1.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If any of the following conditions is true, condTermFlagN is set equal to 0
  – mbAddrN is not available
  – the macroblock mbAddrN has mb_type equal to P_Skip or B_Skip
  – The macroblock mbAddrN is coded in Intra prediction mode
  – predModeEqualFlagN is equal to 0
  – refIdxZeroFlagN is equal to 1
– Otherwise, condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived as

$$ctxIdxInc = condTermFlagA + 2 * condTermFlagB \quad (9\text{-}8)$$

9.3.3.1.1.7 Derivation process of ctxIdxInc for the syntax elements mvd_l0 and mvd_l1

Inputs to this process are mbPartIdx, subMbPartIdx, and ctxIdxOffset.

Output of this process is ctxIdxInc.

The interpretation of mvd_lX and Pred_LX within this subclause is specified as follows.

– If this process is invoked for the derivation of mvd_l0, mvd_lX is interpreted as mvd_l0 and Pred_LX is interpreted as Pred_L0.
– Otherwise (this process is invoked for the derivation of mvd_l1), mvd_lX is interpreted as mvd_l1 and Pred_LX is interpreted as Pred_L1.

Let currSubMbType be set equal to sub_mb_type[mbPartIdx].

The derivation process for neighbouring partitions specified in subclause 6.4.8.5 is invoked with mbPartIdx, currSubMbType, and subMbPartIdx as input and the output is assigned to mbAddrA\mbPartIdxA\subMbPartIdxA and mbAddrB\mbPartIdxB\subMbPartIdxB.

Let the variable compIdx be derived as follows.

– If ctxIdxOffset is equal to 40, compIdx is set equal to 0.

– Otherwise (ctxIdxOffset is equal to 47), compIdx is set equal to 1.

Let the variable predModeEqualFlagN be specified as follows.

– If the macroblock mbAddrN has mb_type equal to P_8x8 or B_8x8, the following applies.

  – If SubMbPredMode(sub_mb_type[mbPartIdxN]) is not equal to Pred_LX and not equal to BiPred, predModeEqualFlagN is set equal to 0, where sub_mb_type specifies the syntax element for the macroblock mbAddrN.

  – Otherwise, predModeEqualFlagN is set equal to 1.

– Otherwise, the following applies.

  – If MbPartPredMode(mb_type, mbPartIdxN) is not equal to Pred_LX and not equal to BiPred, predModeEqualFlagN is set equal to 0, where mb_type specifies the syntax element for the macroblock mbAddrN.

  – Otherwise, predModeEqualFlagN is set equal to 1.

Let the variable absMvdCompN (with N being either A or B) be derived as follows.

– If any of the following conditions is true, absMvdCompN is set equal to 0

  – mbAddrN is not available

  – the macroblock mbAddrN has mb_type equal to P_Skip or B_Skip

  – The macroblock mbAddrN is coded in an Intra prediction mode

  – predModeEqualFlagN is equal to 0

– Otherwise, the following applies

  – If compIdx is equal to 1, MbaffFrameFlag is equal to 1, the current macroblock is a frame macroblock, and the macroblock mbAddrN is a field macroblock

$$absMvdCompN = Abs(mvd_lX[mbPartIdxN][subMbPartIdxN][compIdx]) * 2 \quad (9\text{-}9)$$

  – Otherwise, if compIdx is equal to 1, MbaffFrameFlag is equal to 1, the current macroblock is a field macroblock, and the macroblock mbAddrN is a frame macroblock

$$absMvdCompN = Abs(mvd_lX[mbPartIdxN][subMbPartIdxN][compIdx]) / 2 \quad (9\text{-}10)$$

  – Otherwise,

$$absMvdCompN = Abs(mvd_lX[mbPartIdxN][subMbPartIdxN][compIdx]) \quad (9\text{-}11)$$

The variable ctxIdxInc is derived as follows

– If (absMvdCompA + absMvdCompB) is less than 3, ctxIdxInc is set equal to 0.

– Otherwise, if (absMvdCompA + absMvdCompB) is greater than 32, ctxIdxInc is set equal to 2.

– Otherwise ((absMvdCompA + absMvdCompB) is in the range of 3 to 32, inclusive), ctxIdxInc is set equal to 1.

9.3.3.1.1.8 Derivation process of ctxIdxInc for the syntax element intra_chroma_pred_mode

Output of this process is ctxIdxInc.

The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

Let the variable condTermFlagN (with N being replaced by either A or B) be derived as follows.

– If any of the following conditions is true, condTermFlagN is set equal to 0
  – mbAddrN is not available
  – The macroblock mbAddrN is coded in Inter prediction mode
  – mb_type for the macroblock mbAddrN is equal to I_PCM
  – intra_chroma_pred_mode for the macroblock mbAddrN is equal to 0
– Otherwise, condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived by

$$ctxIdxInc = condTermFlagA + condTermFlagB \quad (9\text{-}12)$$

9.3.3.1.1.9 Derivation process of ctxIdxInc for the syntax element coded_block_flag

Input to this process is ctxBlockCat and additional input is specified as follows.

- If ctxBlockCat is equal to 0, no additional input
- Otherwise, if ctxBlockCat is equal to 1 or 2, luma4x4BlkIdx
- Otherwise, if ctxBlockCat is equal to 3, the chroma component index iCbCr
- Otherwise (ctxBlockCat is equal to 4), chroma4x4BlkIdx and the chroma component index iCbCr

Output of this process is ctxIdxInc(ctxBlockCat).

Let the variable transBlockN (with N being either A or B) be derived as follows.

- If ctxBlockCat is equal to 0, the following applies.
  - The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrN (with N being either A or B).
  - The variable transBlockN is derived as follows.
    - If mbAddrN is available and the macroblock mbAddrN is coded in Intra_16x16 prediction mode, the luma DC block of macroblock mbAddrN is assigned to transBlockN
    - Otherwise, transBlockN is marked as not available.
- Otherwise, if ctxBlockCat is equal to 1 or 2, the following applies.
  - The derivation process for neighbouring 4x4 luma blocks specified in subclause 6.4.8.3 is invoked with luma4x4BlkIdx as input and the output is assigned to mbAddrN, luma4x4BlkIdxN (with N being either A or B).
  - The variable transBlockN is derived as follows.
    - If mbAddrN is available, the macroblock mbAddrN does not have mb_type equal to P_Skip, B_Skip, or I_PCM, ((CodedBlockPatternLuma >> (luma4x4BlkIdxN >>2)) & 1) is not equal to 0 for the macroblock mbAddrN, and transform_size_8x8_flag is equal to 0 for the macroblock mbAddrN, the 4x4 luma block with index luma4x4BlkIdxN of macroblock mbAddrN is assigned to transBlockN.
    - Otherwise, if mbAddrN is available, the macroblock mbAddrN does not have mb_type equal to P_Skip or B_Skip, ((CodedBlockPatternLuma >> (luma4x4BlkIdxN >>2)) & 1) is not equal to 0 for the macroblock mbAddrN, and transform_size_8x8_flag is equal to 1 for the macroblock mbAddrN, the 8x8 luma block with index (luma4x4BlkIdxN >> 2) of macroblock mbAddrN is assigned to transBlockN.
    - Otherwise, transBlockN is marked as not available.
- Otherwise, if ctxBlockCat is equal to 3, the following applies.
  - The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrN (with N being either A or B).
  - The variable transBlockN is derived as follows.

- If mbAddrN is available, the macroblock mbAddrN does not have mb_type equal to P_Skip, B_Skip, or I_PCM, and CodedBlockPatternChroma is not equal to 0 for the macroblock mbAddrN, the chroma DC block of chroma component iCbCr of macroblock mbAddrN is assigned to transBlockN.
- Otherwise, transBlockN is marked as not available.

- Otherwise (ctxBlockCat is equal to 4), the following applies.
  - The derivation process for neighbouring 4x4 chroma blocks specified in subclause 6.4.8.4 is invoked with chroma4x4BlkIdx as input and the output is assigned to mbAddrN, chroma4x4BlkIdxN (with N being either A or B).
  - The variable transBlockN is derived as follows.
    - If mbAddrN is available, the macroblock mbAddrN does not have mb_type equal to P_Skip, B_Skip, or I_PCM, and CodedBlockPatternChroma is equal to 2 for the macroblock mbAddrN, the 4x4 chroma block with chroma4x4BlkIdxN of the chroma component iCbCr of macroblock mbAddrN is assigned to transBlockN.
    - Otherwise, transBlockN is marked as not available.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

- If any of the following conditions is true, condTermFlagN is set equal to 0
  - mbAddrN is not available and the current macroblock is coded in Inter prediction mode
  - mbAddrN is available and transBlockN is not available and mb_type for the macroblock mbAddrN is not equal to I_PCM
  - The current macroblock is coded in Intra prediction mode, constrained_intra_pred_flag is equal to 1, the macroblock mbAddrN is available and coded in Inter prediction mode, and slice data partitioning is in use (nal_unit_type is in the range of 2 through 4, inclusive).
- Otherwise, if any of the following conditions is true, condTermFlagN is set equal to 1
  - mbAddrN is not available and the current macroblock is coded in Intra prediction mode
  - mb_type for the macroblock mbAddrN is equal to I_PCM
- Otherwise, condTermFlagN is set equal to the value of the coded_block_flag of the transform block transBlockN that was decoded for the macroblock mbAddrN.

The variable ctxIdxInc(ctxBlockCat) is derived by

$$ctxIdxInc(ctxBlockCat) = condTermFlagA + 2 * condTermFlagB \quad (9\text{-}13)$$

9.3.3.1.1.10 Derivation process of ctxIdxInc for the syntax element transform_size_8x8_flag

Output of this process is ctxIdxInc.

The derivation process for neighbouring macroblocks specified in subclause 6.4.8.1 is invoked and the output is assigned to mbAddrA and mbAddrB.

Let the variable condTermFlagN (with N being either A or B) be derived as follows.

– If any of the following conditions is true, condTermFlagN is set equal to 0.
  – mbAddrN is not available
  – transform_size_8x8_flag for the macroblock mbAddrN is equal to 0
– Otherwise, condTermFlagN is set equal to 1.

The variable ctxIdxInc is derived by

$$ctxIdxInc = condTermFlagA + condTermFlagB \quad (9\text{-}14)$$

9.3.3.1.2 Assignment process of ctxIdxInc using prior decoded bin values

Inputs to this process are ctxIdxOffset and binIdx.

Output of this process is ctxIdxInc.

Table 9-32 contains the specification of ctxIdxInc for the given values of ctxIdxOffset and binIdx.

For each value of ctxIdxOffset and binIdx, ctxIdxInc is derived by using some of the values of prior decoded bin values (b_0, b_1, b_2,…, b_k), where the value of the index k is less than the value of binIdx.

Table 9-32 – Specification of ctxIdxInc for specific values of ctxIdxOffset and binIdx

Value (name) of ctxIdxOffset	binIdx	ctxIdxInc
3	4	(b_3 != 0) ? 5: 6
	5	(b_3 != 0) ? 6: 7
14	2	(b_1 != 1) ? 2: 3
17	4	(b_3 != 0) ? 2: 3
27	2	(b_1 != 0) ? 4: 5
32	4	(b_3 != 0) ? 2: 3
36	2	(b_1 != 0) ? 2: 3

9.3.3.1.3 Assignment process of ctxIdxInc for syntax elements significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1

Inputs to this process are ctxIdxOffset and binIdx.

Output of this process is ctxIdxInc.

The assignment process of ctxIdxInc for syntax elements significant_coeff_flag, last_significant_coeff_flag, and coeff_abs_level_minus1 as well as for coded_block_flag depends on categories of different blocks denoted by the variable ctxBlockCat. The specification of these block categories is given in Table 9-33.

Table 9-33 – Specification of ctxBlockCat for the different blocks

Block description	maxNumCoeff	ctxBlockCat
block of luma DC transform coefficient levels (i.e., list Intra16x16DCLevel as described in subclause 7.4.5.3)	16	0
block of luma AC transform coefficient levels (i.e., list Intra16x16ACLevel[i] as described in subclause 7.4.5.3)	15	1
block of 16 luma transform coefficient levels (i.e., list LumaLevel[i] as described in subclause 7.4.5.3)	16	2
block of chroma DC transform coefficient levels	4 * NumC8x8	3
block of chroma AC transform coefficient levels	15	4
block of 64 luma transform coefficient levels (i.e., list LumaLevel8x8[i] as described in subclause 7.4.5.3)	64	5

Let the variable levelListIdx be set equal to the index of the list of transform coefficient levels as specified in subclause 7.4.5.3.

For the syntax elements significant_coeff_flag and last_significant_coeff_flag in blocks with ctxBlockCat < 5 and ctxBlockCat != 3, the variable ctxIdxInc is derived by

ctxIdxInc = levelListIdx (9-15)

where levelListIdx ranges from 0 to maxNumCoeff – 2, inclusive.

For the syntax elements significant_coeff_flag and last_significant_coeff_flag in blocks with ctxBlockCat == 3, the variable ctxIdxInc is derived by

ctxIdxInc = Min(levelListIdx / NumC8x8, 2) (9-16)

where levelListIdx ranges from 0 to 4 * NumC8x8 − 2, inclusive.

For the syntax elements significant_coeff_flag and last_significant_coeff_flag in 8x8 luma blocks with ctxBlockCat == 5, Table 9-34 contains the specification of ctxIdxInc for the given values of levelListIdx, where levelListIdx ranges from 0 to 62, inclusive.

Table 9-34 – Mapping of scanning position to ctxIdxInc for ctxBlockCat == 5

levelListIdx	ctxIdxInc for significant_coeff_flag (frame coded macroblocks)	ctxIdxInc for significant_coeff_flag (field coded macroblocks)	ctxIdxInc for last_significant_coeff_flag	levelListIdx	ctxIdxInc for significant_coeff_flag (frame coded macroblocks)	ctxIdxInc for significant_coeff_flag (field coded macroblocks)	ctxIdxInc for last_significant_coeff_flag
0	0	0	0	**32**	7	9	3
1	1	1	1	**33**	6	9	3
2	2	1	1	**34**	11	10	3
3	3	2	1	**35**	12	10	3
4	4	2	1	**36**	13	8	3
5	5	3	1	**37**	11	11	3
6	5	3	1	**38**	6	12	3
7	4	4	1	**39**	7	11	3
8	4	5	1	**40**	8	9	4
9	3	6	1	**41**	9	9	4
10	3	7	1	**42**	14	10	4
11	4	7	1	**43**	10	10	4
12	4	7	1	**44**	9	8	4
13	4	8	1	**45**	8	13	4
14	5	4	1	**46**	6	13	4
15	5	5	1	**47**	11	9	4
16	4	6	2	**48**	12	9	5
17	4	9	2	**49**	13	10	5
18	4	10	2	**50**	11	10	5
19	4	10	2	**51**	6	8	5
20	3	8	2	**52**	9	13	6
21	3	11	2	**53**	14	13	6
22	6	12	2	**54**	10	9	6
23	7	11	2	**55**	9	9	6
24	7	9	2	**56**	11	10	7

levelListIdx	ctxIdxInc for significant_coeff_flag (frame coded macroblocks)	ctxIdxInc for significant_coeff_flag (field coded macroblocks)	ctxIdxInc for last_significant_coeff_flag	levelListIdx	ctxIdxInc for significant_coeff_flag (frame coded macroblocks)	ctxIdxInc for significant_coeff_flag (field coded macroblocks)	ctxIdxInc for last_significant_coeff_flag
25	7	9	2	**57**	12	10	7
26	8	10	2	**58**	13	14	7
27	9	10	2	**59**	11	14	7
28	10	8	2	**60**	14	14	8
29	9	11	2	**61**	10	14	8
30	8	12	2	**62**	12	14	8
31	7	11	2				

Let numDecodAbsLevelEq1 denotes the accumulated number of decoded transform coefficient levels with absolute value equal to 1, and let numDecodAbsLevelGt1 denotes the accumulated number of decoded transform coefficient levels with absolute value greater than 1. Both numbers are related to the same transform coefficient block, where the current decoding process takes place. Then, for decoding of coeff_abs_level_minus1, ctxIdxInc for coeff_abs_level_minus1 is specified depending on binIdx as follows.

– If binIdx is equal to 0, ctxIdxInc is derived by

ctxIdxInc = ((numDecodAbsLevelGt1 != 0) ? 0: Min(4, 1 + numDecodAbsLevelEq1)) (9-17)

– Otherwise (binIdx is greater than 0), ctxIdxInc is derived by

ctxIdxInc = 5 + Min(4 – (ctxBlockCat == 3), numDecodAbsLevelGt1) (9-18)

9.3.3.2 Arithmetic decoding process

Inputs to this process are the bypassFlag, ctxIdx as derived in subclause 9.3.3.1, and the state variables codIRange and codIOffset of the arithmetic decoding engine.

Output of this process is the value of the bin.

Figure 9-2 illustrates the whole arithmetic decoding process for a single bin. For decoding the value of a bin, the context index ctxIdx is passed to the arithmetic decoding process DecodeBin(ctxIdx), which is specified as follows.

– If bypassFlag is equal to 1, DecodeBypass( ) as specified in subclause 9.3.3.2.3 is invoked.

– Otherwise, if bypassFlag is equal to 0 and ctxIdx is equal to 276, DecodeTerminate( ) as specified in subclause 9.3.3.2.4 is invoked.

– Otherwise (bypassFlag is equal to 0 and ctxIdx is not equal to 276), DecodeDecision( ) as specified in subclause 9.3.3.2.1 is applied.

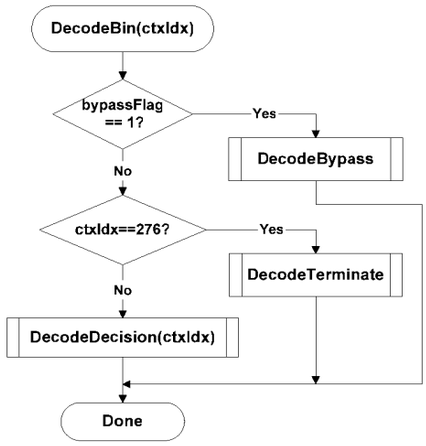


Figure 9-2 – Overview of the arithmetic decoding process for a single bin (informative)

NOTE – Arithmetic coding is based on the principle of recursive interval subdivision. Given a probability estimation p(0) and p(1) = 1 – p(0) of a binary decision (0, 1), an initially given code sub-interval with the range codIRange will be subdivided into two sub-intervals having range p(0) * codIRange and codIRange – p(0) * codIRange, respectively. Depending on the decision, which has been observed, the corresponding sub-interval will be chosen as the new code interval, and a binary code string pointing into that interval will represent the sequence of observed binary decisions. It is useful to distinguish between the most probable symbol (MPS) and the least probable symbol (LPS), so that binary decisions have to be identified as either MPS or LPS, rather than 0 or 1. Given this terminology, each context is specified by the probability p_{LPS} of the LPS and the value of MPS (valMPS), which is either 0 or 1.

The arithmetic core engine in this Recommendation | International Standard has three distinct properties:

– The probability estimation is performed by means of a finite-state machine with a table-based transition process between 64 different representative probability states { p_{LPS}(pStateIdx) | 0 <= pStateIdx < 64 } for the LPS probability p_{LPS}. The numbering of the states is arranged in such a way that the probability state with index pStateIdx = 0 corresponds to an LPS probability value of 0.5, with decreasing LPS probability towards higher state indices.

– The range codIRange representing the state of the coding engine is quantised to a small set {Q_1,…,Q_4} of pre-set quantisation values prior to the calculation of the new interval range. Storing a table containing all 64x4 pre-computed product values of Q_i * p_{LPS}(pStateIdx) allows a multiplication-free approximation of the product codIRange * p_{LPS}(pStateIdx).

– For syntax elements or parts thereof for which an approximately uniform probability distribution is assumed to be given a separate simplified encoding and decoding bypass process is used.

9.3.3.2.1 Arithmetic decoding process for a binary decision

Inputs to this process are ctxIdx, codIRange, and codIOffset.

Outputs of this process are the decoded value binVal, and the updated variables codIRange and codIOffset.

Figure 9-3 shows the flowchart for decoding a single decision (DecodeDecision).

1. The value of the variable codIRangeLPS is derived as follows.

– Given the current value of codIRange, the variable qCodIRangeIdx is derived by

qCodIRangeIdx =(codIRange >> 6) & 0x03 (9-19)

– Given qCodIRangeIdx and pStateIdx associated with ctxIdx, the value of the variable rangeTabLPS as specified in Table 9-35 is assigned to codIRangeLPS:

codIRangeLPS = rangeTabLPS[pStateIdx][qCodIRangeIdx] (9-20)

2. The variable codIRange is set equal to codIRange – codIRangeLPS and the following applies.

– If codIOffset is greater than or equal to codIRange, the variable binVal is set equal to 1 - valMPS, codIOffset is decremented by codIRange, and codIRange is set equal to codIRangeLPS.

– Otherwise, the variable binVal is set equal to valMPS.

Given the value of binVal, the state transition is performed as specified in subclause 9.3.3.2.1.1. Depending on the current value of codIRange, renormalization is performed as specified in subclause 9.3.3.2.2.

9.3.3.2.1.1 State transition process

Inputs to this process are the current pStateIdx, the decoded value binVal and valMPS values of the context variable associated with ctxIdx.

Outputs of this process are the updated pStateIdx and valMPS of the context variable associated with ctxIdx.

Depending on the decoded value binVal, the update of the two variables pStateIdx and valMPS associated with ctxIdx is derived as follows:

```
if( binVal  = =  valMPS )
    pStateIdx = transIdxMPS( pStateIdx )
else {                                                    (9-21)
    if( pStateIdx  = =  0 )
        valMPS = 1 – valMPS
    pStateIdx = transIdxLPS( pStateIdx )
}
```

Table 9-36 specifies the transition rules transIdxMPS( ) and transIdxLPS( ) after decoding the value of valMPS and 1 – valMPS, respectively.

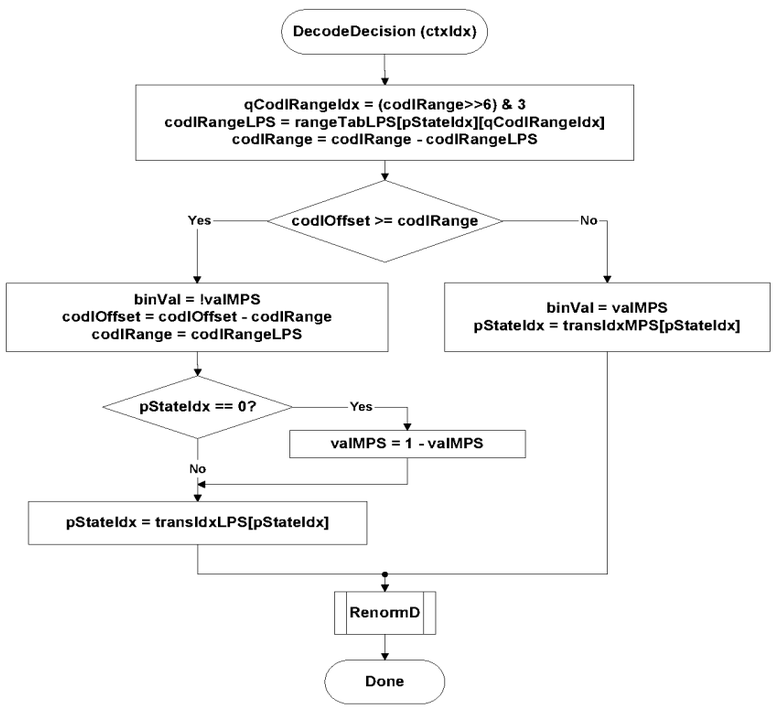


Figure 9-3 – Flowchart for decoding a decision

Table 9-35 – Specification of rangeTabLPS depending on pStateIdx and qCodIRangeIdx

pStateIdx	qCodIRangeIdx				pStateIdx	qCodIRangeIdx			
	0	**1**	**2**	**3**		**0**	**1**	**2**	**3**
0	128	176	208	240	**32**	27	33	39	45
1	128	167	197	227	**33**	26	31	37	43
2	128	158	187	216	**34**	24	30	35	41
3	123	150	178	205	**35**	23	28	33	39
4	116	142	169	195	**36**	22	27	32	37
5	111	135	160	185	**37**	21	26	30	35
6	105	128	152	175	**38**	20	24	29	33
7	100	122	144	166	**39**	19	23	27	31
8	95	116	137	158	**40**	18	22	26	30
9	90	110	130	150	**41**	17	21	25	28
10	85	104	123	142	**42**	16	20	23	27
11	81	99	117	135	**43**	15	19	22	25
12	77	94	111	128	**44**	14	18	21	24
13	73	89	105	122	**45**	14	17	20	23
14	69	85	100	116	**46**	13	16	19	22
15	66	80	95	110	**47**	12	15	18	21
16	62	76	90	104	**48**	12	14	17	20
17	59	72	86	99	**49**	11	14	16	19
18	56	69	81	94	**50**	11	13	15	18
19	53	65	77	89	**51**	10	12	15	17
20	51	62	73	85	**52**	10	12	14	16
21	48	59	69	80	**53**	9	11	13	15
22	46	56	66	76	**54**	9	11	12	14
23	43	53	63	72	**55**	8	10	12	14
24	41	50	59	69	**56**	8	9	11	13
25	39	48	56	65	**57**	7	9	11	12
26	37	45	54	62	**58**	7	9	10	12
27	35	43	51	59	**59**	7	8	10	11
28	33	41	48	56	**60**	6	8	9	11
29	32	39	46	53	**61**	6	7	9	10
30	30	37	43	50	**62**	6	7	8	9
31	29	35	41	48	**63**	2	2	2	2

Table 9-36 – State transition table

pStateIdx	**0**	**1**	**2**	**3**	**4**	**5**	**6**	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
transIdxLPS	0	0	1	2	2	4	4	5	6	7	8	9	9	11	11	12
transIdxMPS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
pStateIdx	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
transIdxLPS	13	13	15	15	16	16	18	18	19	19	21	21	22	22	23	24
transIdxMPS	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
pStateIdx	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
transIdxLPS	24	25	26	26	27	27	28	29	29	30	30	30	31	32	32	33
transIdxMPS	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
pStateIdx	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	57	**58**	**59**	**60**	**61**	**62**	**63**
transIdxLPS	33	33	34	34	35	35	35	36	36	36	37	37	37	38	38	63
transIdxMPS	49	50	51	52	53	54	55	56	57	58	59	60	61	62	62	63

9.3.3.2.2 Renormalization process in the arithmetic decoding engine

Inputs to this process are bits from slice data and the variables codIRange and codIOffset.

Outputs of this process are the updated variables codIRange and codIOffset.

A flowchart of the renormalization is shown in Figure 9-4. The current value of codIRange is first compared to 0x0100 and further steps are specified as follows.

– If codIRange is greater than or equal to 0x0100, no renormalization is needed and the RenormD process is finished;

– Otherwise (codIRange is less than 0x0100), the renormalization loop is entered. Within this loop, the value of codIRange is doubled, i.e., left-shifted by 1 and a single bit is shifted into codIOffset by using read_bits(1).

The bitstream shall not contain data that results in a value of codIOffset being greater than or equal to codIRange upon completion of this process.

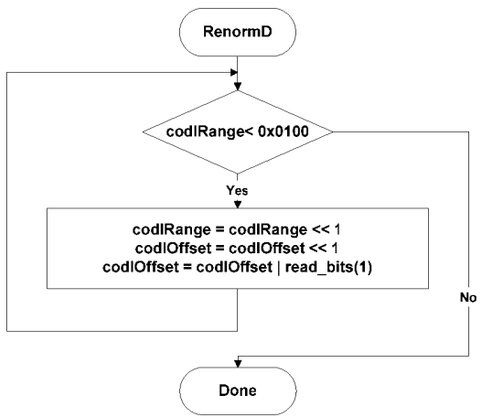


Figure 9-4 – Flowchart of renormalization

9.3.3.2.3 Bypass decoding process for binary decisions

Inputs to this process are bits from slice data and the variables codIRange and codIOffset.

Outputs of this process are the updated variable codIOffset and the decoded value binVal.

The bypass decoding process is invoked when bypassFlag is equal to 1. Figure 9-5 shows a flowchart of the corresponding process.

First, the value of codIOffset is doubled, i.e., left-shifted by 1 and a single bit is shifted into codIOffset by using read_bits(1). Then, the value of codIOffset is compared to the value of codIRange and further steps are specified as follows.

– If codIOffset is greater than or equal to codIRange, the variable binVal is set equal to 1 and codIOffset is decremented by codIRange.

– Otherwise (codIOffset is less than codIRange), the variable binVal is set equal to 0.

The bitstream shall not contain data that results in a value of codIOffset being greater than or equal to codIRange upon completion of this process.

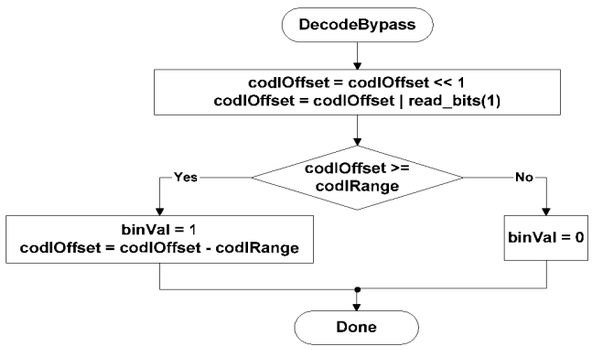


Figure 9-5 – Flowchart of bypass decoding process

9.3.3.2.4 Decoding process for binary decisions before termination

Inputs to this process are bits from slice data and the variables codIRange and codIOffset.

Outputs of this process are the updated variables codIRange and codIOffset, and the decoded value binVal.

This special decoding routine applies to decoding of end_of_slice_flag and of the bin indicating the I_PCM mode corresponding to ctxIdx equal to 276. Figure 9-6 shows the flowchart of the corresponding decoding process, which is specified as follows.

First, the value of codIRange is decremented by 2. Then, the value of codIOffset is compared to the value of codIRange and further steps are specified as follows.

– If codIOffset is greater than or equal to codIRange, the variable binVal is set equal to 1, no renormalization is carried out, and CABAC decoding is terminated. The last bit inserted in register codIOffset is equal to 1. When decoding end_of_slice_flag, this last bit inserted in register codIOffset is interpreted as rbsp_stop_one_bit.

– Otherwise (codIOffset is less than codIRange), the variable binVal is set equal to 0 and renormalization is performed as specified in subclause 9.3.3.2.2.

NOTE – This procedure may also be implemented using DecodeDecision(ctxIdx) with ctxIdx = 276. In the case where the decoded value is equal to 1, seven more bits would be read by DecodeDecision(ctxIdx) and a decoding process would have to adjust its bitstream pointer accordingly to properly decode following syntax elements.

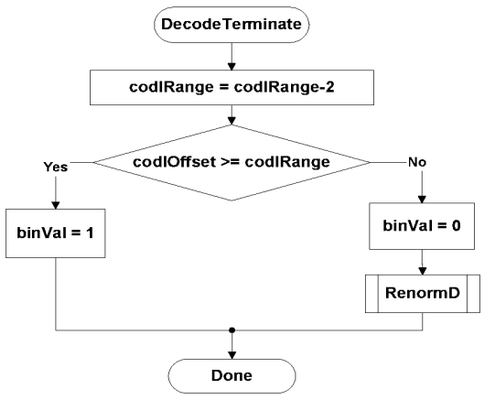


Figure 9-6 – Flowchart of decoding a decision before termination

9.3.4 Arithmetic encoding process (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Inputs to this process are decisions that are to be encoded and written.

Outputs of this process are bits that are written to the RBSP.

This informative subclause describes an arithmetic encoding engine that matches the arithmetic decoding engine described in subclause 9.3.3.2. The encoding engine is essentially symmetric with the decoding engine, i.e., procedures are called in the same order. The following procedures are described in this section: InitEncoder, EncodeDecision, EncodeBypass, EncodeTerminate, which correspond to InitDecoder, DecodeDecision, DecodeBypass, and DecodeTerminate, respectively. The state of the arithmetic encoding engine is represented by a value of the variable codILow pointing to the lower end of a sub-interval and a value of the variable codIRange specifying the corresponding range of that sub-interval.

9.3.4.1 Initialisation process for the arithmetic encoding engine (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

This process is invoked before encoding the first macroblock of a slice, and after encoding any pcm_alignment_zero_bit and all pcm_sample_luma and pcm_sample_chroma data for a macroblock of type I_PCM.

Outputs of this process are the values codILow, codIRange, firstBitFlag, bitsOutstanding, and symCnt of the arithmetic encoding engine.

In the initialisation procedure of the encoder, codILow is set equal to 0, and codIRange is set equal to 0x01FE. Furthermore, a firstBitFlag is set equal to 1, and bitsOutstanding and symCnt counters are set equal to 0.

NOTE – The minimum register precision required for codILow is 10 bits and for CodIRange is 9 bits. The precision required for the counters bitsOutstanding and symCnt should be sufficiently large to prevent overflow of the related registers. When MaxBinCountInSlice denotes the maximum total number of binary decisions to encode in one slice, the minimum register precision required for the variables bitsOutstanding and symCnt is given by Ceil(Log2(MaxBinCountInSlice + 1)).

9.3.4.2 Encoding process for a binary decision (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Inputs to this process are the context index ctxIdx, the value of binVal to be encoded, and the variables codIRange, codILow and symCnt.

Outputs of this process are the variables codIRange, codILow, and symCnt.

Figure 9-7 shows the flowchart for encoding a single decision. In a first step, the variable codIRangeLPS is derived as follows.

Given the current value of codIRange, codIRange is mapped to the index qCodIRangeIdx of a quantised value of codIRange by using Equation 9-19. The value of qCodIRangeIdx and the value of pStateIdx associated with ctxIdx are

used to determine the value of the variable rangeTabLPS as specified in Table 9-35, which is assigned to codIRangeLPS. The value of codIRange – codIRangeLPS is assigned to codIRange.

In a second step, the value of binVal is compared to valMPS associated with ctxIdx. When binVal is different from valMPS, codIRange is added to codILow and codIRange is set equal to the value codIRangeLPS. Given the encoded decision, the state transition is performed as specified in subclause 9.3.3.2.1.1. Depending on the current value of codIRange, renormalization is performed as specified in subclause 9.3.4.3. Finally, the variable symCnt is incremented by 1.

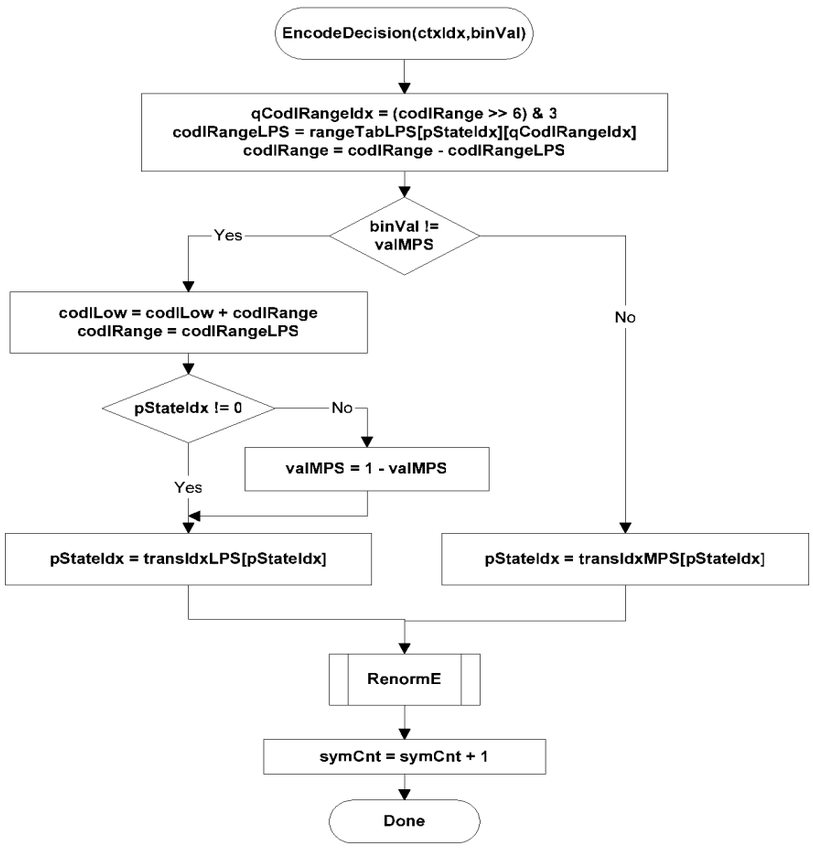


Figure 9-7 – Flowchart for encoding a decision

9.3.4.3 Renormalization process in the arithmetic encoding engine (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Inputs to this process are the variables codIRange, codILow, firstBitFlag, and bitsOutstanding.

Outputs of this process are zero or more bits written to the RBSP and the updated variables codIRange, codILow, firstBitFlag, and bitsOutstanding.

Renormalization is illustrated in Figure 9-8.

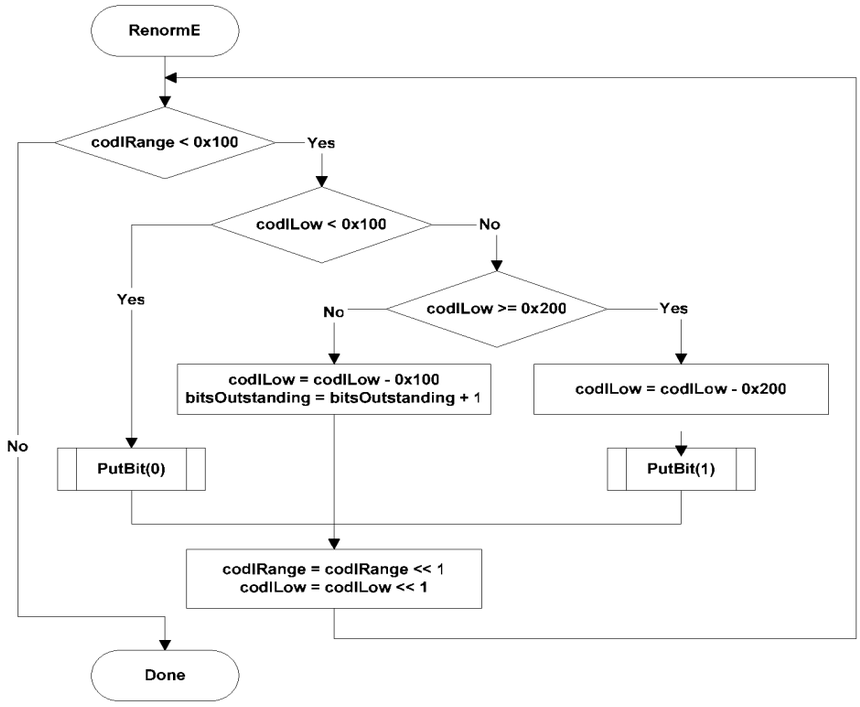


Figure 9-8 – Flowchart of renormalization in the encoder

The PutBit( ) procedure described in Figure 9-9 provides carry over control. It uses the function WriteBits(B, N) that writes N bits with value B to the bitstream and advances the bitstream pointer by N bit positions. This function assumes the existence of a bitstream pointer with an indication of the position of the next bit to be written to the bitstream by the encoding process.

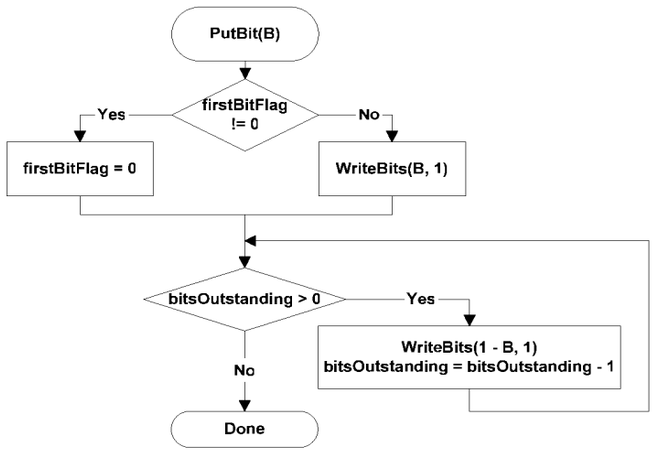


Figure 9-9 – Flowchart of PutBit(B)

9.3.4.4 Bypass encoding process for binary decisions (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Inputs to this process are the variables binVal, codILow, codIRange, bitsOutstanding, and symCnt.

Output of this process is a bit written to the RBSP and the updated variables codILow, bitsOutstanding, and symCnt.

This encoding process applies to all binary decisions with bypassFlag equal to 1. Renormalization is included in the specification of this process as given in Figure 9-10.

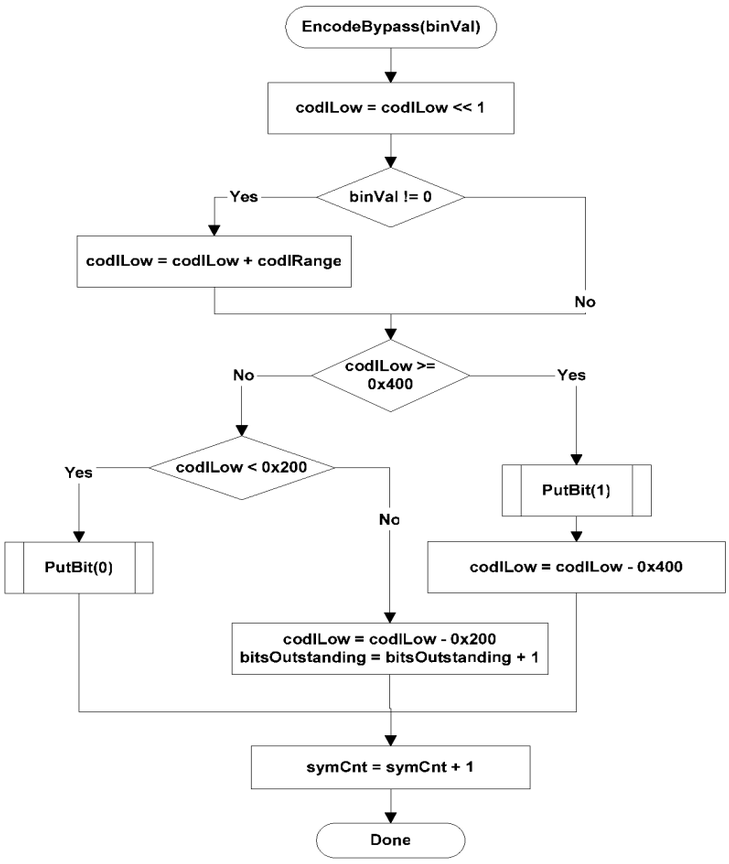


Figure 9-10 – Flowchart of encoding bypass

9.3.4.5 Encoding process for a binary decision before termination (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Inputs to this process are the variables binVal, codIRange, codILow, bitsOutstanding, and symCnt.

Outputs of this process are zero or more bits written to the RBSP and the updated variables codILow, codIRange, bitsOutstanding, and symCnt.

This encoding routine shown in Figure 9-11 applies to encoding of the end_of_slice_flag and of the bin indicating the I_PCM mb_type both associated with ctxIdx equal to 276.

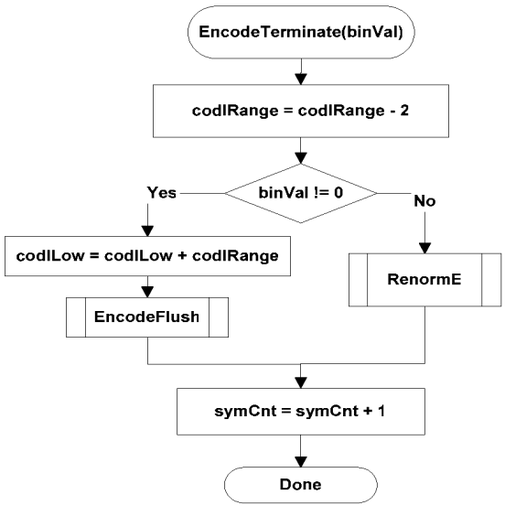


Figure 9-11 – Flowchart of encoding a decision before termination

When the value of binVal to encode is equal to 1, CABAC encoding is terminated and the flushing procedure shown in Figure 9-12 is applied. In this flushing procedure, the last bit written by WriteBits(B, N) is equal to 1. When encoding end_of_slice_flag, this last bit is interpreted as the rbsp_stop_one_bit.

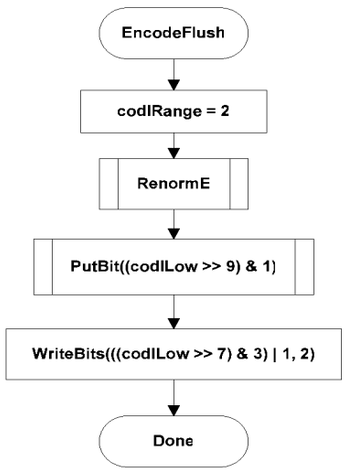


Figure 9-12 – Flowchart of flushing at termination

9.3.4.6 Byte stuffing process (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

This process is invoked after encoding the last macroblock of the last slice of a picture and after encapsulation.

Inputs to this process are the number of bytes NumBytesInVclNALunits of all VCL NAL units of a picture, the number of macroblocks PicSizeInMbs in the picture, and the number of binary symbols BinCountsInNALunits resulting from encoding the contents of all VCL NAL units of the picture.

Outputs of this process are zero or more bytes appended to the NAL unit.

Let the variable k be set equal to Ceil((Ceil(3 * (32 * BinCountsInNALunits – RawMbBits * PicSizeInMbs) ÷ 1024) – NumBytesInVclNALunits) ÷ 3). Depending on the variable k the following applies.

– If k is less than or equal to 0, no cabac_zero_word is appended to the NAL unit.

– Otherwise (k is greater than 0), the 3-byte sequence 0x000003 is appended k times to the NAL unit after encapsulation, where the first two bytes 0x0000 represent a cabac_zero_word and the third byte 0x03 represents an emulation_prevention_three_byte.

Annex A
Profiles and levels

(This annex forms an integral part of this Recommendation | International Standard)

Profiles and levels specify restrictions on bitstreams and hence limits on the capabilities needed to decode the bitstreams. Profiles and levels may also be used to indicate interoperability points between individual decoder implementations.

NOTE 1 – This Recommendation | International Standard does not include individually selectable "options" at the decoder, as this would increase interoperability difficulties.

Each profile specifies a subset of algorithmic features and limits that shall be supported by all decoders conforming to that profile.

NOTE 2 – Encoders are not required to make use of any particular subset of features supported in a profile.

Each level specifies a set of limits on the values that may be taken by the syntax elements of this Recommendation | International Standard. The same set of level definitions is used with all profiles, but individual implementations may support a different level for each supported profile. For any given profile, levels generally correspond to decoder processing load and memory capability.

A.1 Requirements on video decoder capability

Capabilities of video decoders conforming to this Recommendation | International Standard are specified in terms of the ability to decode video streams conforming to the constraints of profiles and levels specified in this Annex. For each such profile, the level supported for that profile shall also be expressed.

Specific values are specified in this annex for the syntax elements profile_idc and level_idc. All other values of profile_idc and level_idc are reserved for future use by ITU-T | ISO/IEC.

NOTE – Decoders should not infer that when a reserved value of profile_idc or level_idc falls between the values specified in this Recommendation | International Standard that this indicates intermediate capabilities between the specified profiles or levels, as there are no restrictions on the method to be chosen by ITU-T | ISO/IEC for the use of such future reserved values.

A.2 Profiles

A.2.1 Baseline profile

Bitstreams conforming to the Baseline profile shall obey the following constraints:

– Only I and P slice types may be present.

– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.

– Sequence parameter sets shall have frame_mbs_only_flag equal to 1.

– The syntax elements chroma_format_idc, bit_depth_luma_minus8, bit_depth_chroma_minus8, qpprime_y_zero_transform_bypass_flag, and seq_scaling_matrix_present_flag shall not be present in sequence parameter sets.

– Picture parameter sets shall have weighted_pred_flag and weighted_bipred_idc both equal to 0.

– Picture parameter sets shall have entropy_coding_mode_flag equal to 0.

– Picture parameter sets shall have num_slice_groups_minus1 in the range of 0 to 7, inclusive.

– The syntax elements transform_8x8_mode_flag, pic_scaling_matrix_present_flag, and second_chroma_qp_index_offset shall not be present in picture parameter sets.

– The syntax element level_prefix shall not be greater than 15.

– The level constraints specified for the Baseline profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the Baseline profile is specified by profile_idc being equal to 66.

Decoders conforming to the Baseline profile at a specific level shall be capable of decoding all bitstreams in which profile_idc is equal to 66 or constraint_set0_flag is equal to 1 and in which level_idc and constraint_set3_flag represent a level less than or equal to the specified level.

A.2.2 Main profile

Bitstreams conforming to the Main profile shall obey the following constraints:

– Only I, P, and B slice types may be present.

– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.

– Arbitrary slice order is not allowed.

– The syntax elements chroma_format_idc, bit_depth_luma_minus8, bit_depth_chroma_minus8, qpprime_y_zero_transform_bypass_flag, and seq_scaling_matrix_present_flag shall not be present in sequence parameter sets.

– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.

– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.

– The syntax elements transform_8x8_mode_flag, pic_scaling_matrix_present_flag, and second_chroma_qp_index_offset shall not be present in picture parameter sets.

– The syntax element level_prefix shall not be greater than 15 (when present).

– The level constraints specified for the Main profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the Main profile is specified by profile_idc being equal to 77.

Decoders conforming to the Main profile at a specified level shall be capable of decoding all bitstreams in which profile_idc is equal to 77 or constraint_set1_flag is equal to 1 and in which level_idc and constraint_set3_flag represent a level less than or equal to the specified level.

A.2.3 Extended profile

Bitstreams conforming to the Extended profile shall obey the following constraints:

– Sequence parameter sets shall have direct_8x8_inference_flag equal to 1.

– The syntax elements chroma_format_idc, bit_depth_luma_minus8, bit_depth_chroma_minus8, qpprime_y_zero_transform_bypass_flag, and seq_scaling_matrix_present_flag shall not be present in sequence parameter sets.

– Picture parameter sets shall have entropy_coding_mode_flag equal to 0.

– Picture parameter sets shall have num_slice_groups_minus1 in the range of 0 to 7, inclusive.

– The syntax elements transform_8x8_mode_flag, pic_scaling_matrix_present_flag, and second_chroma_qp_index_offset shall not be present in picture parameter sets.

– The syntax element level_prefix shall not be greater than 15 (when present).

– The level constraints specified for the Extended profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the Extended profile is specified by profile_idc being equal to 88.

Decoders conforming to the Extended profile at a specified level shall be capable of decoding all bitstreams in which profile_idc is equal to 88 or constraint_set2_flag is equal to 1 and in which level_idc represents a level less than or equal to specified level.

Decoders conforming to the Extended profile at a specified level shall also be capable of decoding all bitstreams in which profile_idc is equal to 66 or constraint_set0_flag is equal to 1, in which level_idc and constraint_set3_flag represent a level less than or equal to the specified level.

A.2.4 High profile

Bitstreams conforming to the High profile shall obey the following constraints:

– Only I, P, and B slice types may be present.

– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.

– Arbitrary slice order is not allowed.

– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.

– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.

– Sequence parameter sets shall have chroma_format_idc in the range of 0 to 1 inclusive.

– Sequence parameter sets shall have bit_depth_luma_minus8 equal to 0 only.

– Sequence parameter sets shall have bit_depth_chroma_minus8 equal to 0 only.

– Sequence parameter sets shall have qpprime_y_zero_transform_bypass_flag equal to 0 only.
– The level constraints specified for the High profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the High profile is specified by profile_idc being equal to 100. Decoders conforming to the High profile at a specific level shall be capable of decoding all bitstreams in which level_idc and constraint_set3_flag represents a level less than or equal to the specified level and either or both of the following conditions are true:

– profile_idc is equal to 77 or 100, or
– constraint_set1_flag is equal to 1.

A.2.5 High 10 profile

Bitstreams conforming to the High 10 profile shall obey the following constraints:

– Only I, P, and B slice types may be present.
– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.
– Arbitrary slice order is not allowed.
– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.
– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.
– Sequence parameter sets shall have chroma_format_idc in the range of 0 to 1 inclusive.
– Sequence parameter sets shall have bit_depth_luma_minus8 in the range of 0 to 2 inclusive.
– Sequence parameter sets shall have bit_depth_chroma_minus8 in the range of 0 to 2 inclusive.
– Sequence parameter sets shall have qpprime_y_zero_transform_bypass_flag equal to 0 only.
– The level constraints specified for the High 10 profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the High 10 profile is specified by profile_idc being equal to 110. Decoders conforming to the High 10 profile at a specific level shall be capable of decoding all bitstreams in which level_idc and constraint_set3_flag represent a level less than or equal to the specified level and either or both of the following conditions are true:

– profile_idc is equal to 77, 100, or 110, or
– constraint_set1_flag is equal to 1.

A.2.6 High 4:2:2 profile

Bitstreams conforming to the High 4:2:2 profile shall obey the following constraints:

– Only I, P, and B slice types may be present.
– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.
– Arbitrary slice order is not allowed.
– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.
– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.
– Sequence parameter sets shall have chroma_format_idc in the range of 0 to 2 inclusive
– Sequence parameter sets shall have bit_depth_luma_minus8 in the range of 0 to 2 inclusive.
– Sequence parameter sets shall have bit_depth_chroma_minus8 in the range of 0 to 2 inclusive.
– Sequence parameter sets shall have qpprime_y_zero_transform_bypass_flag equal to 0 only.
– The level constraints specified for the High 4:2:2 profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the High 4:2:2 profile is specified by profile_idc being equal to 122. Decoders conforming to the High 4:2:2 profile at a specific level shall be capable of decoding all bitstreams in which level_idc and constraint_set3_flag represents a level less than or equal to the specified level and either or both of the following conditions are true:

– profile_idc is equal to 77, 100, 110, or 122, or
– constraint_set1_flag is equal to 1.

A.2.7 High 4:4:4 profile

Bitstreams conforming to the High 4:4:4 profile shall obey the following constraints:

– Only I, P, and B slice types may be present.

– NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive.

– Arbitrary slice order is not allowed.

– Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only.

– Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only.

– Sequence parameter sets shall have bit_depth_luma_minus8 in the range of 0 to 4 inclusive.

– Sequence parameter sets shall have bit_depth_chroma_minus8 in the range of 0 to 4 inclusive.

– The level constraints specified for the High 4:4:4 profile in subclause A.3 shall be fulfilled.

Conformance of a bitstream to the High 4:4:4 profile is specified by profile_idc being equal to 144. Decoders conforming to the High 4:4:4 profile at a specific level shall be capable of decoding all bitstreams in which level_idc and constraint_set3_flag represent a level less than or equal to the specified level and either or both of the following conditions are true:

– profile_idc is equal to 77, 100, 110, 122, or 144, or

– constraint_set1_flag is equal to 1.

A.3 Levels

The following is specified for expressing the constraints in this Annex.

– Let access unit n be the n-th access unit in decoding order with the first access unit being access unit 0.

– Let picture n be the primary coded picture or the corresponding decoded picture of access unit n.

A.3.1 Level limits common to the Baseline, Main, and Extended profiles

Let the variable fR be derived as follows.

– If picture n is a frame, fR is set equal to 1 ÷ 172.

– Otherwise (picture n is a field), fR is set equal to 1 ÷ (172 * 2).

Bitstreams conforming to the Baseline, Main, or Extended profiles at a specified level shall obey the following constraints:

a) The nominal removal time of access unit n (with n > 0) from the CPB as specified in subclause C.1.2, satisfies the constraint that $t_{r,n}(n) - t_r(n - 1)$ is greater than or equal to Max(PicSizeInMbs ÷ MaxMBPS, fR), where MaxMBPS is the value specified in Table A-1 that applies to picture n – 1, and PicSizeInMbs is the number of macroblocks in picture n – 1.

b) The difference between consecutive output times of pictures from the DPB as specified in subclause C.2.2, satisfies the constraint that $\Delta t_{o,dpb}(n) >= Max(PicSizeInMbs \div MaxMBPS, fR)$, where MaxMBPS is the value specified in Table A-1 for picture n, and PicSizeInMbs is the number of macroblocks of picture n, provided that picture n is a picture that is output and is not the last picture of the bitstream that is output.

c) The sum of the NumBytesInNALunit variables for access unit 0 is less than or equal to $384 * (PicSizeInMbs + MaxMBPS * (t_r(0) - t_{r,n}(0))) \div MinCR$, where MaxMBPS and MinCR are the values specified in Table A-1 that apply to picture 0 and PicSizeInMbs is the number of macroblocks in picture 0.

d) The sum of the NumBytesInNALunit variables for access unit n (with n > 0) is less than or equal to $384 * MaxMBPS * (t_r(n) - t_r(n - 1)) \div MinCR$, where MaxMBPS and MinCR are the values specified in Table A-1 that apply to picture n.

e) PicWidthInMbs * FrameHeightInMbs <= MaxFS, where MaxFS is specified in Table A-1

f) PicWidthInMbs <= Sqrt(MaxFS * 8)

g) FrameHeightInMbs <= Sqrt(MaxFS * 8)

h) max_dec_frame_buffering <= MaxDpbSize, where MaxDpbSize is equal to Min(1024 * MaxDPB / (PicWidthInMbs * FrameHeightInMbs * 384), 16) and MaxDPB is given in Table A-1 in units of 1024 bytes.

i) For the VCL HRD parameters, BitRate[SchedSelIdx] <= 1000 * MaxBR and CpbSize[SchedSelIdx] <= 1000 * MaxCPB for at least one value of SchedSelIdx, where BitRate[SchedSelIdx] is given by Equation E-37 and CpbSize[SchedSelIdx] is given by Equation E-38 when vcl_hrd_parameters_present_flag is equal to 1. MaxBR and MaxCPB are specified in Table A-1 in units of 1000 bits/s and 1000 bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1, inclusive.

j) For the NAL HRD parameters, BitRate[SchedSelIdx] <= 1200 * MaxBR and CpbSize[SchedSelIdx] <= 1200 * MaxCPB for at least one value of SchedSelIdx, where BitRate[SchedSelIdx] is given by Equation E-37 and CpbSize[SchedSelIdx] is given by Equation E-38 when nal_hrd_parameters_present_flag equal to 1. MaxBR and MaxCPB are specified in Table A-1 in units of 1200 bits/s and 1200 bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1.

k) Vertical motion vector component range luma motion vectors does not exceed MaxVmvR in units of luma frame samples, where MaxVmvR is specified in Table A-1

NOTE 1 – When chroma_format_idc is equal to 1 and the current macroblock is a field macroblock, the motion vector component range for chroma motion vectors may exceed MaxVmvR in units of luma frame samples, due to the method of deriving chroma motion vectors as specified in subclause 8.4.1.4.

l) Horizontal motion vector range does not exceed the range of -2048 to 2047.75, inclusive, in units of luma samples

m) Number of motion vectors per two consecutive macroblocks in decoding order (also applying to the total from the last macroblock of a slice and the first macroblock of the next slice in decoding order, and in particular also applying to the total from the last macroblock of the last slice of a picture and the first macroblock of the first slice of the next picture in decoding order) does not exceed MaxMvsPer2Mb, where MaxMvsPer2Mb is specified in Table A-1. The number of motion vectors for each macroblock is the value of the variable MvCnt after the completion of the intra or inter prediction process for the macroblock.

n) Number of bits of macroblock_layer( ) data for any macroblock is not greater than 3200. Depending on entropy_coding_mode_flag, the bits of macroblock_layer( ) data are counted as follows.

– If entropy_coding_mode_flag is equal to 0, the number of bits of macroblock_layer( ) data is given by the number of bits in the macroblock_layer( ) syntax structure for a macroblock.

– Otherwise (entropy_coding_mode_flag is equal to 1), the number of bits of macroblock_layer( ) data for a macroblock is given by the number of times read_bits(1) is called in subclauses 9.3.3.2.2 and 9.3.3.2.3 when parsing the macroblock_layer( ) associated with the macroblock.

Table A-1 specifies the limits for each level. Entries marked "-" in Table A-1 denote the absence of a corresponding limit. For purposes of comparison of level capabilities, a level shall be considered to be a lower (higher) level than some other level if the level appears nearer to the top (bottom) row of Table A-1 than the other level.

A level to which the bitstream conforms shall be indicated by the syntax elements level_idc and constraint_set3_flag as follows.

– If level_idc is equal to 11 and constraint_set3_flag is equal to 1, the indicated level is level 1b.

– Otherwise (level_idc is not equal to 11 or constraint_set3_flag is not equal to 1), level_idc shall be set equal to a value of ten times the level number specified in Table A-1 and constraint_set3_flag shall be set equal to 0.

Table A-1 – Level limits

Level number	Max macroblock processing rate MaxMBPS (MB/s)	Max frame size MaxFS (MBs)	Max decoded picture buffer size MaxDPB (1024 bytes for 4:2:0)	Max video bit rate MaxBR (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s, or cpbBrNalFactor bits/s)	Max CPB size MaxCPB (1000 bits, 1200 bits, cpbBrVclFactor bits, or cpbBrNalFactor bits)	Vertical MV component range MaxVmvR (luma frame samples)	Min compression ratio MinCR	Max number of motion vectors per two consecutive MBs MaxMvsPer2Mb
1	1 485	99	148.5	64	175	[-64,+63.75]	2	-
1b	1 485	99	148.5	128	350	[-64,+63.75]	2	-
1.1	3 000	396	337.5	192	500	[-128,+127.75]	2	-
1.2	6 000	396	891.0	384	1 000	[-128,+127.75]	2	-
1.3	11 880	396	891.0	768	2 000	[-128,+127.75]	2	-
2	11 880	396	891.0	2 000	2 000	[-128,+127.75]	2	-
2.1	19 800	792	1 782.0	4 000	4 000	[-256,+255.75]	2	-
2.2	20 250	1 620	3 037.5	4 000	4 000	[-256,+255.75]	2	-
3	40 500	1 620	3 037.5	10 000	10 000	[-256,+255.75]	2	32
3.1	108 000	3 600	6 750.0	14 000	14 000	[-512,+511.75]	4	16
3.2	216 000	5 120	7 680.0	20 000	20 000	[-512,+511.75]	4	16
4	245 760	8 192	12 288.0	20 000	25 000	[-512,+511.75]	4	16
4.1	245 760	8 192	12 288.0	50 000	62 500	[-512,+511.75]	2	16
4.2	522 240	8 704	13 056.0	50 000	62 500	[-512,+511.75]	2	16
5	589 824	22 080	41 400.0	135 000	135 000	[-512,+511.75]	2	16
5.1	983 040	36 864	69 120.0	240 000	240 000	[-512,+511.75]	2	16

Levels with non-integer level numbers in Table A-1 are referred to as "intermediate levels".

NOTE 2 – All levels have the same status, but some applications may choose to use only the integer-numbered levels.

Informative subclause A.3.4 shows the effect of these limits on frame rates for several example picture formats.

A.3.2 Level limits common to the High, High 10, High 4:2:2, and High 4:4:4 profiles

Let the variable fR be derived as follows.

– If picture n is a frame, fR is set equal to 1 ÷ 172.

– Otherwise (picture n is a field), fR is set equal to 1 ÷ (172 * 2).

Bitstreams conforming to the High, High 10, High 4:2:2, or High 4:4:4 profiles at a specified level shall obey the following constraints:

a) The nominal removal time of access unit n (with n > 0) from the CPB as specified in subclause C.1.2, satisfies the constraint that $t_{r,n}(n) - t_r(n - 1)$ is greater than or equal to Max(PicSizeInMbs ÷ MaxMBPS, fR), where MaxMBPS is the value specified in Table A-1 that applies to picture n – 1, and PicSizeInMbs is the number of macroblocks in picture n – 1.

b) The difference between consecutive output times of pictures from the DPB as specified in subclause C.2.2, satisfies the constraint that $\Delta t_{o,dpb}(n)$ >= Max(PicSizeInMbs ÷ MaxMBPS, fR), where MaxMBPS is the value specified in Table A-1 for picture n, and PicSizeInMbs is the number of macroblocks of picture n, provided that picture n is a picture that is output and is not the last picture of the bitstream that is output.

c) PicWidthInMbs * FrameHeightInMbs <= MaxFS, where MaxFS is specified in Table A-1

d) PicWidthInMbs <= Sqrt(MaxFS * 8)

e) FrameHeightInMbs <= Sqrt(MaxFS * 8)

f) max_dec_frame_buffering <= MaxDpbSize, where MaxDpbSize is equal to Min(1024 * MaxDPB / (PicWidthInMbs * FrameHeightInMbs * 384), 16) and MaxDPB is specified in Table A-1.

g) Vertical motion vector component range does not exceed MaxVmvR in units of luma frame samples, where MaxVmvR is specified in Table A-1.

h) Horizontal motion vector range does not exceed the range of -2048 to 2047.75, inclusive, in units of luma samples.

i) Number of motion vectors per two consecutive macroblocks in decoding order (also applying to the total from the last macroblock of a slice and the first macroblock of the next slice in decoding order) does not exceed MaxMvsPer2Mb, where MaxMvsPer2Mb is specified in Table A-1. The number of motion vectors for each macroblock is value of the variable MvCnt after the completion of the intra or inter prediction process for the macroblock.

j) Number of bits of macroblock_layer( ) data for any macroblock is not greater than 128 + RawMbBits. Depending on entropy_coding_mode_flag, the bits of macroblock_layer( ) data are counted as follows.

   – If entropy_coding_mode_flag is equal to 0, the number of bits of macroblock_layer( ) data is given by the number of bits in the macroblock_layer( ) syntax structure for a macroblock.

   – Otherwise (entropy_coding_mode_flag is equal to 1), the number of bits of macroblock_layer( ) data for a macroblock is given by the number of times read_bits(1) is called in subclauses 9.3.3.2.2 and 9.3.3.2.3 when parsing the macroblock_layer( ) associated with the macroblock.

Table A-1 specifies the limits for each level. Entries marked "-" in Table A-1 denote the absence of a corresponding limit.

A level to which the bitstream conforms shall be indicated by the syntax element level_idc as follows.

– If level_idc is equal to 9, the indicated level is level 1b.

– Otherwise (level_idc is not equal to 9), level_idc shall be set equal to a value of ten times the level number specified in Table A-1.

A.3.3 Profile-specific level limits

a) In bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 profiles, the removal time of access unit 0 shall satisfy the constraint that the number of slices in picture 0 is less than or equal to (PicSizeInMbs + MaxMBPS * (t_r(0) - $t_{r,n}$(0))) ÷ SliceRate, where SliceRate is the value specified in Table A-4 that applies to picture 0.

b) In bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 profiles, the difference between consecutive removal time of access units n and n - 1 (with n > 0) shall satisfy the constraint that the number of slices in picture n is less than or equal to MaxMBPS * (t_r(n) - t_r(n - 1)) ÷ SliceRate, where SliceRate is the value specified in Table A-4 that applies to picture n.

c) In bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 profiles, sequence parameter sets shall have direct_8x8_inference_flag equal to 1 for the levels specified in Table A-4.

   NOTE 1 – direct_8x8_inference_flag is not relevant to the Baseline profile as it does not allow B slice types (specified in subclause A.2.1), and direct_8x8_inference_flag is equal to 1 for all levels of the Extended profile (specified in subclause A.2.3).

d) In bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 or Extended profiles, sequence parameter sets shall have frame_mbs_only_flag equal to 1 for the levels specified in Table A-4 for the Main, High, High 10, High 4:2:2, and High 4:4:4 profiles and in Table A-5 for the Extended profile.

   NOTE 2 – frame_mbs_only_flag is equal to 1 for all levels of the Baseline profile (specified in subclause A.2.1).

e) In bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 or Extended profiles, the value of sub_mb_type in B macroblocks shall not be equal to B_Bi_8x4, B_Bi_4x8, or B_Bi_4x4 for the levels in which MinLumaBiPredSize is shown as 8x8 in Table A-4 for the Main, High, High 10, High 4:2:2, and High 4:4:4 profiles and in Table A-5 for the Extended profile.

f) In bitstreams conforming to the Baseline and Extended profiles, ($xInt_{max}$ – $xInt_{min}$ + 6) * ($yInt_{max}$ – $yInt_{min}$ + 6) <= MaxSubMbRectSize in macroblocks coded with mb_type equal to P_8x8, P_8x8ref0 or B_8x8 for all

invocations of the process specified in subclause 8.4.2.2.1 used to generate the predicted luma sample array for a single reference picture list (reference picture list 0 or reference picture list 1) for each 8x8 sub-macroblock, where NumSubMbPart(sub_mb_type) > 1, where MaxSubMbRectSize is specified in Table A-3 for the Baseline profile and in Table A-5 for the Extended profile and

- $xInt_{min}$ is the minimum value of $xInt_L$ among all luma sample predictions for the sub-macroblock
- $xInt_{max}$ is the maximum value of $xInt_L$ among all luma sample predictions for the sub-macroblock
- $yInt_{min}$ is the minimum value of $yInt_L$ among all luma sample predictions for the sub-macroblock
- $yInt_{max}$ is the maximum value of $yInt_L$ among all luma sample predictions for the sub-macroblock

g) In bitstreams conforming to the High, High 10, High 4:2:2, or High 4:4:4 profile, for the VCL HRD parameters, BitRate[SchedSelIdx] <= cpbBrVclFactor * MaxBR and CpbSize[SchedSelIdx] <= cpbBrVclFactor * MaxCPB for at least one value of SchedSelIdx, where cpbBrVclFactor is specified in Table A-2, BitRate[SchedSelIdx] is specified by Equation E-37 and CpbSize[SchedSelIdx] is specified by Equation E-38 when vcl_hrd_parameters_present_flag is equal to 1. MaxBR and MaxCPB are specified in Table A-1 in units of cpbBrVclFactor bits/s and cpbBrVclFactor bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1, inclusive.

h) In bitstreams conforming to the High, High 10, High 4:2:2, or High 4:4:4 profile, for the NAL HRD parameters, BitRate[SchedSelIdx] <= cpbBrNalFactor * MaxBR and CpbSize[SchedSelIdx] <= cpbBrNalFactor * MaxCPB for at least one value of SchedSelIdx, where cpbBrNalFactor is specified in Table A-2, BitRate[SchedSelIdx] is specified by Equation E-37 and CpbSize[SchedSelIdx] is specified by Equation E-38 when nal_hrd_parameters_present_flag equal to 1. MaxBR and MaxCPB are specified in Table A-1 in units of cpbBrNalFactor bits/s and cpbBrNalFactor bits, respectively. The bitstream shall satisfy these conditions for at least one value of SchedSelIdx in the range 0 to cpb_cnt_minus1.

Table A-2 – Specification of cpbBrVclFactor and cpbBrNalFactor

Profile	cpbBrVclFactor	cpbBrNalFactor
High	1 250	1 500
High 10	3 000	3 600
High 4:2:2	4 000	4 800
High 4:4:4	4 000	4 800

A.3.3.1 Baseline profile limits

Table A-3 specifies limits for each level that are specific to bitstreams conforming to the Baseline profile. Entries marked "-" in Table A-3 denote the absence of a corresponding limit.

Table A-3 – Baseline profile level limits

Level number	MaxSubMbRectSize
1	576
1b	576
1.1	576
1.2	576
1.3	576
2	576
2.1	576
2.2	576
3	576
3.1	-
3.2	-
4	-
4.1	-
4.2	-
5	-
5.1	-

A.3.3.2 Main, High, High 10, High 4:2:2, or High 4:4:4 profile limits

Table A-4 specifies limits for each level that are specific to bitstreams conforming to the Main, High, High 10, High 4:2:2, or High 4:4:4 profiles. Entries marked "-" in Table A-4 denote the absence of a corresponding limit.

Table A-4 – Main, High, High 10, High 4:2:2, or High 4:4:4 profile level limits

Level number	SliceRate	MinLumaBiPredSize	direct_8x8_inference_flag	frame_mbs_only_flag
1	-	-	-	1
1b	-	-	-	1
1.1	-	-	-	1
1.2	-	-	-	1
1.3	-	-	-	1
2	-	-	-	1
2.1	-	-	-	-
2.2	-	-	-	-
3	22	-	1	-
3.1	60	8x8	1	-
3.2	60	8x8	1	-
4	60	8x8	1	-
4.1	24	8x8	1	-
4.2	24	8x8	1	1
5	24	8x8	1	1
5.1	24	8x8	1	1

A.3.3.3 Extended Profile Limits

Table A-5 specifies limits for each level that are specific to bitstreams conforming to the Extended profile. Entries marked "-" in Table A-5 denote the absence of a corresponding limit.

Table A-5 – Extended profile level limits

Level number	MaxSubMbRectSize	MinLumaBiPredSize	frame_mbs_only_flag
1	576	-	1
1b	576	-	1
1.1	576	-	1
1.2	576	-	1
1.3	576	-	1
2	576	-	1
2.1	576	-	-
2.2	576	-	-
3	576	-	-
3.1	-	8x8	-
3.2	-	8x8	-
4	-	8x8	-
4.1	-	8x8	-
4.2	-	8x8	1
5	-	8x8	1
5.1	-	8x8	1

A.3.4 Effect of level limits on frame rate (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Table A-6 – Maximum frame rates (frames per second) for some example frame sizes

Level:					1	1b	1.1	1.2	1.3	2	2.1
Max frame size (macroblocks):					99	99	396	396	396	396	792
Max macroblocks/second:					1 485	1 485	3 000	6 000	11 880	11 880	19 800
Max frame size (samples):					25 344	25 344	101 376	101 376	101 376	101 376	202 752
Max samples/second:					380 160	380 160	768 000	1 536 000	3 041 280	3 041 280	5 068 800
Format	Luma Width	Luma Height	MBs Total	Luma Samples							
SQCIF	128	96	48	12 288	30.9	30.9	62.5	125.0	172.0	172.0	172.0
QCIF	176	144	99	25 344	15.0	15.0	30.3	60.6	120.0	120.0	172.0
QVGA	320	240	300	76 800	-	-	10.0	20.0	39.6	39.6	66.0
525 SIF	352	240	330	84 480	-	-	9.1	18.2	36.0	36.0	60.0
CIF	352	288	396	101 376	-	-	7.6	15.2	30.0	30.0	50.0
525 HHR	352	480	660	168 960	-	-	-	-	-	-	30.0
625 HHR	352	576	792	202 752	-	-	-	-	-	-	25.0
VGA	640	480	1 200	307 200	-	-	-	-	-	-	-
525 4SIF	704	480	1 320	337 920	-	-	-	-	-	-	-
525 SD	720	480	1 350	345 600	-	-	-	-	-	-	-
4CIF	704	576	1 584	405 504	-	-	-	-	-	-	-
625 SD	720	576	1 620	414 720	-	-	-	-	-	-	-
SVGA	800	600	1 900	486 400	-	-	-	-	-	-	-
XGA	1024	768	3 072	786 432	-	-	-	-	-	-	-
720p HD	1280	720	3 600	921 600	-	-	-	-	-	-	-
4VGA	1280	960	4 800	1 228 800	-	-	-	-	-	-	-
SXGA	1280	1024	5 120	1 310 720	-	-	-	-	-	-	-
525 16SIF	1408	960	5 280	1 351 680	-	-	-	-	-	-	-
16CIF	1408	1152	6 336	1 622 016	-	-	-	-	-	-	-
4SVGA	1600	1200	7 500	1 920 000	-	-	-	-	-	-	-
1080 HD	1920	1088	8 160	2 088 960	-	-	-	-	-	-	-
2Kx1K	2048	1024	8 192	2 097 152	-	-	-	-	-	-	-
2Kx1080	2048	1088	8 704	2 228 224	-	-	-	-	-	-	-
4XGA	2048	1536	12 288	3 145 728	-	-	-	-	-	-	-
16VGA	2560	1920	19 200	4 915 200	-	-	-	-	-	-	-
3616x1536 (2.35:1)	3616	1536	21 696	5 554 176	-	-	-	-	-	-	-
3672x1536 (2.39:1)	3680	1536	22 080	5 652 480	-	-	-	-	-	-	-
4Kx2K	4096	2048	32 768	8 388 608	-	-	-	-	-	-	-
4096x2304 (16:9)	4096	2304	36 864	9 437 184	-	-	-	-	-	-	-

Table A-6 (continued) – Maximum frame rates (frames per second) for some example frame sizes

Level:					2.2	3	3.1	3.2	4	4.1	4.2
Max frame size (macroblocks):					1 620	1 620	3 600	5 120	8 192	8 192	8 704
Max macroblocks/second:					20 250	40 500	108 000	216 000	245 760	245 760	522 240
Max frame size (samples):					414 720	414 720	921 600	1 310 720	2 097 152	2 097 152	2 228 224
Max samples/second:					5 184 000	10 368 000	27 648 000	55 296 000	62 914 560	62 914 560	133 693 440
Format	Luma Width	Luma Height	MBs Total	Luma Samples							
SQCIF	128	96	48	12 288	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QCIF	176	144	99	25 344	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QVGA	320	240	300	76 800	67.5	135.0	172.0	172.0	172.0	172.0	172.0
525 SIF	352	240	330	84 480	61.4	122.7	172.0	172.0	172.0	172.0	172.0
CIF	352	288	396	101 376	51.1	102.3	172.0	172.0	172.0	172.0	172.0
525 HHR	352	480	660	168 960	30.7	61.4	163.6	172.0	172.0	172.0	172.0
625 HHR	352	576	792	202 752	25.6	51.1	136.4	172.0	172.0	172.0	172.0
VGA	640	480	1 200	307 200	16.9	33.8	90.0	172.0	172.0	172.0	172.0
525 4SIF	704	480	1 320	337 920	15.3	30.7	81.8	163.6	172.0	172.0	172.0
525 SD	720	480	1 350	345 600	15.0	30.0	80.0	160.0	172.0	172.0	172.0
4CIF	704	576	1 584	405 504	12.8	25.6	68.2	136.4	155.2	155.2	172.0
625 SD	720	576	1 620	414 720	12.5	25.0	66.7	133.3	151.7	151.7	172.0
SVGA	800	600	1 900	486 400	-	-	56.8	113.7	129.3	129.3	172.0
XGA	1024	768	3 072	786 432	-	-	35.2	70.3	80.0	80.0	172.0
720p HD	1280	720	3 600	921 600	-	-	30.0	60.0	68.3	68.3	145.1
4VGA	1280	960	4 800	1 228 800	-	-	-	45.0	51.2	51.2	108.8
SXGA	1280	1024	5 120	1 310 720	-	-	-	42.2	48.0	48.0	102.0
525 16SIF	1408	960	5 280	1 351 680	-	-	-	-	46.5	46.5	98.9
16CIF	1408	1152	6 336	1 622 016	-	-	-	-	38.8	38.8	82.4
4SVGA	1600	1200	7 500	1 920 000	-	-	-	-	32.8	32.8	69.6
1080 HD	1920	1088	8 160	2 088 960	-	-	-	-	30.1	30.1	64.0
2Kx1K	2048	1024	8 192	2 097 152	-	-	-	-	30.0	30.0	63.8
2Kx1080	2048	1088	8 704	2 228 224	-	-	-	-	-	-	60.0
4XGA	2048	1536	12 288	3 145 728	-	-	-	-	-	-	-
16VGA	2560	1920	19 200	4 915 200	-	-	-	-	-	-	-
3616x1536 (2.35:1)	3616	1536	21 696	5 554 176	-	-	-	-	-	-	-
3672x1536 (2.39:1)	3680	1536	22 080	5 652 480	-	-	-	-	-	-	-
4Kx2K	4096	2048	32 768	8 388 608	-	-	-	-	-	-	-
4096x2304 (16:9)	4096	2304	36 864	9 437 184	-	-	-	-	-	-	-

Table A-6 (concluded) – Maximum frame rates (frames per second) for some example frame sizes

Level:					5	5.1
Max frame size (macroblocks):					22 080	36 864
Max macroblocks/second:					589 824	983 040
Max frame size (samples):					5 652 480	9 437 184
Max samples/second:					150 994 944	251 658 240
Format	Luma Width	Luma Height	MBs Total	Luma Samples		
SQCIF	128	96	48	12 288	172.0	172.0
QCIF	176	144	99	25 344	172.0	172.0
QVGA	320	240	300	76 800	172.0	172.0
525 SIF	352	240	330	84 480	172.0	172.0
CIF	352	288	396	101 376	172.0	172.0
525 HHR	352	480	660	168 960	172.0	172.0
625 HHR	352	576	792	202 752	172.0	172.0
VGA	640	480	1 200	307 200	172.0	172.0
525 4SIF	704	480	1 320	337 920	172.0	172.0
525 SD	720	480	1 350	345 600	172.0	172.0
4CIF	704	576	1 584	405 504	172.0	172.0
625 SD	720	576	1 620	414 720	172.0	172.0
SVGA	800	600	1 900	486 400	172.0	172.0
XGA	1024	768	3 072	786 432	172.0	172.0
720p HD	1280	720	3 600	921 600	163.8	172.0
4VGA	1280	960	4 800	1 228 800	122.9	172.0
SXGA	1280	1024	5 120	1 310 720	115.2	172.0
525 16SIF	1408	960	5 280	1 351 680	111.7	172.0
16CIF	1408	1152	6 336	1 622 016	93.1	155.2
4SVGA	1600	1200	7 500	1 920 000	78.6	131.1
1080 HD	1920	1088	8 160	2 088 960	72.3	120.5
2Kx1K	2048	1024	8 192	2 097 152	72.0	120.0
2Kx1080	2048	1088	8 704	2 228 224	67.8	112.9
4XGA	2048	1536	12 288	3 145 728	48.0	80.0
16VGA	2560	1920	19 200	4 915 200	30.7	51.2
3616x1536 (2.35:1)	3616	1536	21 696	5 554 176	27.2	45.3
3672x1536 (2.39:1)	3680	1536	22 080	5 652 480	26.7	44.5
4Kx2K	4096	2048	32 768	8 388 608	-	30.0
4096x2304 (16:9)	4096	2304	36 864	9 437 184	-	26.7

The following should be noted.

– This Recommendation | International Standard is a variable-frame-size specification. The specific frame sizes in Table A-6 are illustrative examples only.

– As used in Table A-6, "525" refers to typical use for environments using 525 analogue scan lines (of which approximately 480 lines contain the visible picture region), and "625" refers to environments using 625 analogue scan lines (of which approximately 576 lines contain the visible picture region).

– XGA is also known as (aka) XVGA, 4SVGA aka UXGA, 16XGA aka 4Kx3K, CIF aka 625 SIF, 625 HHR aka 2CIF aka half 625 D-1, aka half 625 ITU-R BT.601, 525 SD aka 525 D-1 aka 525 ITU-R BT.601, 625 SD aka 625 D-1 aka 625 ITU-R BT.601.

– Frame rates given are correct for progressive scan modes. The frame rates are also correct for interlaced video coding for the cases of frame height divisible by 32.

Annex B
Byte stream format

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies syntax and semantics of a byte stream format specified for use by applications that deliver some or all of the NAL unit stream as an ordered stream of bytes or bits within which the locations of NAL unit boundaries need to be identifiable from patterns in the data, such as ITU-T Rec. H.222.0 | ISO/IEC 13818-1 systems or ITU-T Rec. H.320 systems. For bit-oriented delivery, the bit order for the byte stream format is specified to start with the MSB of the first byte, proceed to the LSB of the first byte, followed by the MSB of the second byte, etc.

The byte stream format consists of a sequence of byte stream NAL unit syntax structures. Each byte stream NAL unit syntax structure contains one start code prefix followed by one nal_unit(NumBytesInNALunit) syntax structure. It may (and under some circumstances, it shall) also contain an additional zero_byte syntax element. It may also contain one or more additional trailing_zero_8bits syntax elements. When it is the first byte stream NAL unit in the bitstream, it may also contain one or more additional leading_zero_8bits syntax elements.

B.1 Byte stream NAL unit syntax and semantics

B.1.1 Byte stream NAL unit syntax

byte_stream_nal_unit(NumBytesInNALunit) {	C	Descriptor
while(next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
leading_zero_8bits /* equal to 0x00 */		f(8)
if(next_bits(24) != 0x000001)		
zero_byte /* equal to 0x00 */		f(8)
start_code_prefix_one_3bytes /* equal to 0x000001 */		f(24)
nal_unit(NumBytesInNALunit)		
while(more_data_in_byte_stream( ) && next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
trailing_zero_8bits /* equal to 0x00 */		f(8)
}		

B.1.2 Byte stream NAL unit semantics

The order of byte stream NAL units in the byte stream shall follow the decoding order of the NAL units contained in the byte stream NAL units (see subclause 7.4.1.2). The content of each byte stream NAL unit is associated with the same access unit as the NAL unit contained in the byte stream NAL unit (see subclause 7.4.1.2.3).

leading_zero_8bits is a byte equal to 0x00.

NOTE – The leading_zero_8bits syntax element can only be present in the first byte stream NAL unit of the bitstream, because (as shown in the syntax diagram of subclause B.1.1) any bytes equal to 0x00 that follow a NAL unit syntax structure and precede the four-byte sequence 0x00000001 (which is to be interpreted as a zero_byte followed by a start_code_prefix_one_3bytes) will be considered to be trailing_zero_8bits syntax elements that are part of the preceding byte stream NAL unit.

zero_byte is a single byte equal to 0x00.

When any of the following conditions are fulfilled, the zero_byte syntax element shall be present.

– the nal_unit_type within the nal_unit( ) is equal to 7 (sequence parameter set) or 8 (picture parameter set)

– the byte stream NAL unit syntax structure contains the first NAL unit of an access unit in decoding order, as specified by subclause 7.4.1.2.3.

start_code_prefix_one_3bytes is a fixed-value sequence of 3 bytes equal to 0x000001. This syntax element is called a start code prefix.

trailing_zero_8bits is a byte equal to 0x00.

B.2 Byte stream NAL unit decoding process

Input to this process consists of an ordered stream of bytes consisting of a sequence of byte stream NAL unit syntax structures.

Output of this process consists of a sequence of NAL unit syntax structures.

At the beginning of the decoding process, the decoder initialises its current position in the byte stream to the beginning of the byte stream. It then extracts and discards each leading_zero_8bits syntax element (if present), moving the current position in the byte stream forward one byte at a time, until the current position in the byte stream is such that the next four bytes in the bitstream form the four-byte sequence 0x00000001.

The decoder then performs the following step-wise process repeatedly to extract and decode each NAL unit syntax structure in the byte stream until the end of the byte stream has been encountered (as determined by unspecified means) and the last NAL unit in the byte stream has been decoded:

1. When the next four bytes in the bitstream form the four-byte sequence 0x00000001, the next byte in the byte stream (which is a zero_byte syntax element) is extracted and discarded and the current position in the byte stream is set equal to the position of the byte following this discarded byte.
2. The next three-byte sequence in the byte stream (which is a start_code_prefix_one_3bytes) is extracted and discarded and the current position in the byte stream is set equal to the position of the byte following this three-byte sequence.
3. NumBytesInNALunit is set equal to the number of bytes starting with the byte at the current position in the byte stream up to and including the last byte that precedes the location of any of the following conditions:
    a. A subsequent byte-aligned three-byte sequence equal to 0x000000, or
    b. A subsequent byte-aligned three-byte sequence equal to 0x000001, or
    c. The end of the byte stream, as determined by unspecified means.
4. NumBytesInNALunit bytes are removed from the bitstream and the current position in the byte stream is advanced by NumBytesInNALunit bytes. This sequence of bytes is nal_unit(NumBytesInNALunit) and is decoded using the NAL unit decoding process.
5. When the current position in the byte stream is not at the end of the byte stream (as determined by unspecified means) and the next bytes in the byte stream do not start with a three-byte sequence equal to 0x000001 and the next bytes in the byte stream do not start with a four byte sequence equal to 0x00000001, the decoder extracts and discards each trailing_zero_8bits syntax element, moving the current position in the byte stream forward one byte at a time, until the current position in the byte stream is such that the next bytes in the byte stream form the four-byte sequence 0x00000001 or the end of the byte stream has been encountered (as determined by unspecified means).

B.3 Decoder byte-alignment recovery (informative)

This subclause does not form an integral part of this Recommendation | International Standard.

Many applications provide data to a decoder in a manner that is inherently byte aligned, and thus have no need for the bit-oriented byte alignment detection procedure described in this subclause.

A decoder is said to have byte-alignment with a bitstream when the decoder is able to determine whether or not the positions of data in the bitstream are byte-aligned. When a decoder does not have byte alignment with the encoder's byte stream, the decoder may examine the incoming bitstream for the binary pattern '00000000 00000000 00000000 00000001' (31 consecutive bits equal to 0 followed by a bit equal to 1). The bit immediately following this pattern is the first bit of an aligned byte following a start code prefix. Upon detecting this pattern, the decoder will be byte aligned with the encoder and positioned at the start of a NAL unit in the byte stream.

Once byte aligned with the encoder, the decoder can examine the incoming byte stream for subsequent three-byte sequences 0x000001 and 0x000003.

When the three-byte sequence 0x000001 is detected, this is a start code prefix.

When the three-byte sequence 0x000003 is detected, the third byte (0x03) is an emulation_prevention_three_byte to be discarded as specified in subclause 7.4.1.

When an error in the bitstream syntax is detected (e.g., a non-zero value of the forbidden_zero_bit or one of the three-byte or four-byte sequences that are prohibited in subclause 7.4.1), the decoder may consider the detected condition as an indication that byte alignment may have been lost and may discard all bitstream data until the detection of byte alignment at a later position in the bitstream as described in this subclause.

Annex C
Hypothetical reference decoder

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies the hypothetical reference decoder (HRD) and its use to check bitstream and decoder conformance.

Two types of bitstreams are subject to HRD conformance checking for this Recommendation | International Standard. The first such type of bitstream, called Type I bitstream, is a NAL unit stream containing only the VCL NAL units and filler data NAL units for all access units in the bitstream. The second type of bitstream, called a Type II bitstream, contains, in addition to the VCL NAL units and filler data NAL units for all access units in the bitstream, at least one of the following.

- additional non-VCL NAL units other than filler data NAL units
- all leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, and trailing_zero_8bits syntax elements that form a byte stream from the NAL unit stream (as specified in Annex B)

Figure C-1 shows the types of bitstream conformance points checked by the HRD.

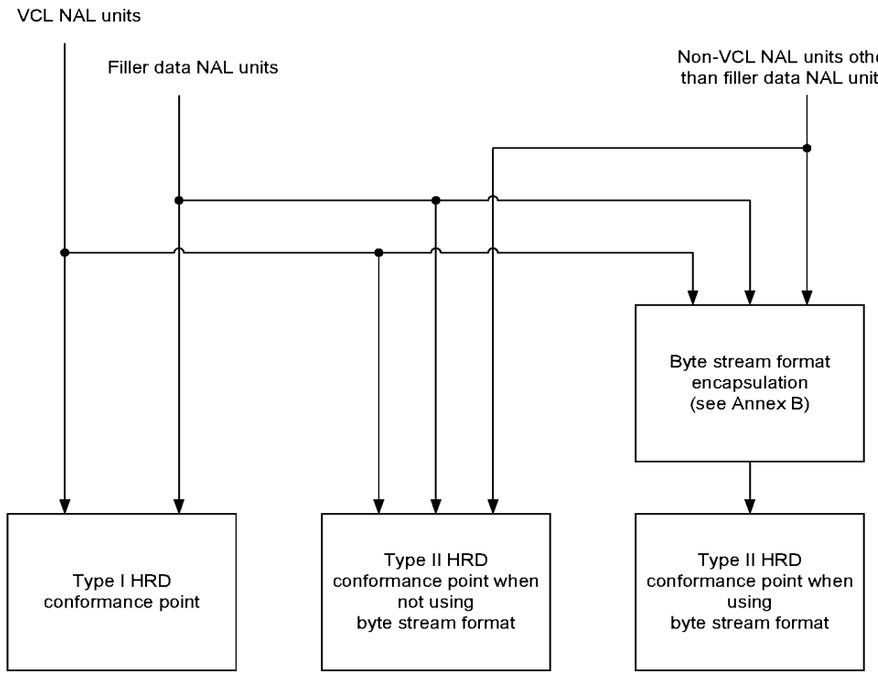


Figure C-1 – Structure of byte streams and NAL unit streams for HRD conformance checks

The syntax elements of non-VCL NAL units (or their default values for some of the syntax elements), required for the HRD, are specified in the semantic subclauses of clause 7 and Annexes D and E.

Two types of HRD parameter sets are used. The HRD parameter sets are signalled through video usability information as specified in subclauses E.1 and E.2, which is part of the sequence parameters set syntax structure.

In order to check conformance of a bitstream using the HRD, all sequence parameter sets and picture parameters sets referred to in the VCL NAL units, and corresponding buffering period and picture timing SEI messages shall be conveyed to the HRD, in a timely manner, either in the bitstream (by non-VCL NAL units), or by other means not specified in this Recommendation | International Standard.

In Annexes C, D and E, the specification for "presence" of non-VCL NAL units is also satisfied when those NAL units (or just some of them) are conveyed to decoders (or to the HRD) by other means not specified by this

Recommendation | International Standard. For the purpose of counting bits, only the appropriate bits that are actually present in the bitstream are counted.

NOTE 1 – As an example, synchronization of a non-VCL NAL unit, conveyed by means other than presence in the bitstream, with the NAL units that are present in the bitstream, can be achieved by indicating two points in the bitstream, between which the non-VCL NAL unit would have been present in the bitstream, had the encoder decided to convey it in the bitstream.

When the content of a non-VCL NAL unit is conveyed for the application by some means other than presence within the bitstream, the representation of the content of the non-VCL NAL unit is not required to use the same syntax specified in this annex.

NOTE 2 – When HRD information is contained within the bitstream, it is possible to verify the conformance of a bitstream to the requirements of this subclause based solely on information contained in the bitstream. When the HRD information is not present in the bitstream, as is the case for all "stand-alone" Type I bitstreams, conformance can only be verified when the HRD data is supplied by some other means not specified in this Recommendation | International Standard.

The HRD contains a coded picture buffer (CPB), an instantaneous decoding process, a decoded picture buffer (DPB), and output cropping as shown in Figure C-2.

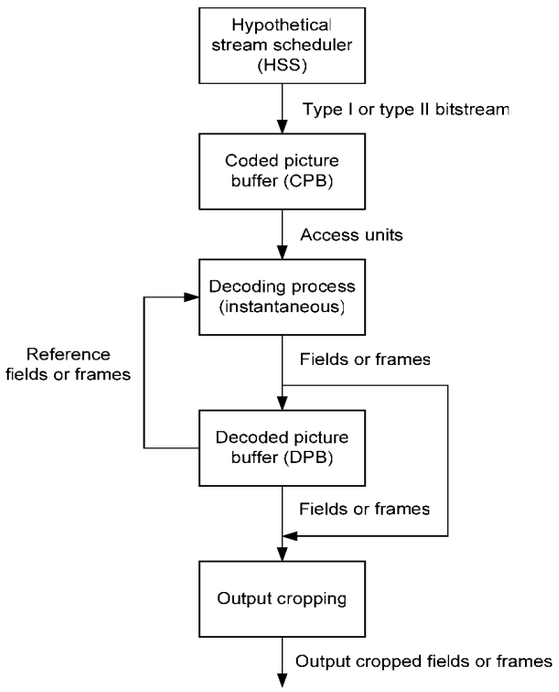


Figure C-2 – HRD buffer model

The CPB size (number of bits) is CpbSize[SchedSelIdx]. The DPB size (number of frame buffers) is Max(1, max_dec_frame_buffering).

The HRD operates as follows. Data associated with access units that flow into the CPB according to a specified arrival schedule are delivered by the HSS. The data associated with each access unit are removed and decoded instantaneously by the instantaneous decoding process at CPB removal times. Each decoded picture is placed in the DPB at its CPB removal time unless it is output at its CPB removal time and is a non-reference picture. When a picture is placed in the DPB it is removed from the DPB at the later of the DPB output time or the time that it is marked as "unused for reference".

The operation of the CPB is specified in subclause C.1. The instantaneous decoder operation is specified in clauses 8 and 9. The operation of the DPB is specified in subclause C.2. The output cropping is specified in subclause C.2.2.

HSS and HRD information concerning the number of enumerated delivery schedules and their associated bit rates and buffer sizes is specified in subclauses E.1.1, E.1.2, E.2.1 and E.2.2. The HRD is initialised as specified by the buffering period SEI message as specified in subclauses D.1.1 and D.2.1. The removal timing of access units from the CPB and output timing from the DPB are specified in the picture timing SEI message as specified in subclauses D.1.2 and D.2.2. All timing information relating to a specific access unit shall arrive prior to the CPB removal time of the access unit.

The HRD is used to check conformance of bitstreams and decoders as specified in subclauses C.3 and C.4, respectively.

NOTE 3 – While conformance is guaranteed under the assumption that all frame-rates and clocks used to generate the bitstream match exactly the values signalled in the bitstream, in a real system each of these may vary from the sigualled or specified value.

All the arithmetic in this annex is done with real values, so that no rounding errors can propagate. For example, the number of bits in a CPB just prior to or after removal of an access unit is not necessarily an integer.

The variable t_c is derived as follows and is called a clock tick.

$$t_c = num_units_in_tick \div time_scale \quad (C\text{-}1)$$

The following is specified for expressing the constraints in this Annex.

– Let access unit n be the n-th access unit in decoding order with the first access unit being access unit 0.

– Let picture n be the primary coded picture or the decoded primary picture of access unit n.

C.1 Operation of coded picture buffer (CPB)

The specifications in this subclause apply independently to each set of CPB parameters that is present and to both the Type I and Type II conformance points shown in Figure C-1.

C.1.1 Timing of bitstream arrival

The HRD may be initialised at any one of the buffering period SEI messages. Prior to initialisation, the CPB is empty.

NOTE – After initialisation, the HRD is not initialised again by subsequent buffering period SEI messages.

Each access unit is referred to as access unit n, where the number n identifies the particular access unit. The access unit that is associated with the buffering period SEI message that initializes the CPB is referred to as access unit 0. The value of n is incremented by 1 for each subsequent access unit in decoding order.

The time at which the first bit of access unit n begins to enter the CPB is referred to as the initial arrival time $t_{ai}(n)$.

The initial arrival time of access units is derived as follows.

– If the access unit is access unit 0, $t_{ai}(0) = 0$,

– Otherwise (the access unit is access unit n with $n > 0$), the following applies.

  – If cbr_flag[SchedSelIdx] is equal to 1, the initial arrival time for access unit n, is equal to the final arrival time (which is derived below) of access unit n - 1, i.e.

$$t_{ai}(n) = t_{af}(n - 1) \quad (C\text{-}2)$$

  – Otherwise (cbr_flag[SchedSelIdx] is equal to 0), the initial arrival time for access unit n is derived by

$$t_{ai}(n) = Max(t_{af}(n - 1), t_{ai,earliest}(n)) \quad (C\text{-}3)$$

where $t_{ai,earliest}(n)$ is derived as follows

– If access unit n is not the first access unit of a subsequent buffering period, $t_{ai,earliest}(n)$ is derived as

$$t_{ai,earliest}(n) = t_{r,n}(n) - (initial_cpb_removal_delay[SchedSelIdx] + initial_cpb_removal_delay_offset[SchedSelIdx]) \div 90000 \quad (C\text{-}4)$$

with $t_{r,n}(n)$ being the nominal removal time of access unit n from the CPB as specified in subclause C.1.2 and initial_cpb_removal_delay[SchedSelIdx] and initial_cpb_removal_delay_offset[SchedSelIdx] being specified in the previous buffering period SEI message.

– Otherwise (access unit n is the first access unit of a subsequent buffering period), $t_{ai,earliest}(n)$ is derived as

$$t_{ai,earliest}(n) = t_{r,n}(n) – (initial_cpb_removal_delay[SchedSelIdx] ÷ 90000) \quad (C-5)$$

with initial_cpb_removal_delay[SchedSelIdx] being specified in the buffering period SEI message associated with access unit n.

The final arrival time for access unit n is derived by

$$t_{af}(n) = t_{ai}(n) + b(n) ÷ BitRate[SchedSelIdx] \quad (C-6)$$

where b(n) is the size in bits of access unit n, counting the bits of the VCL NAL units and the filler data NAL units for the Type I conformance point or all bits of the Type II bitstream for the Type II conformance point, where the Type I and Type II conformance points are as shown in Figure C-1.

The values of SchedSelIdx, BitRate[SchedSelIdx], and CpbSize[SchedSelIdx] are constrained as follows.

– If access unit n and access unit n - 1 are part of different coded video sequences and the content of the active sequence parameter sets of the two coded video sequences differ, the HSS selects a value SchedSelIdx1 of SchedSelIdx from among the values of SchedSelIdx provided for the coded video sequence containing access unit n that results in a BitRate[SchedSelIdx1] or CpbSize[SchedSelIdx1] for the second of the two coded video sequences (which contains access unit n). The value of BitRate[SchedSelIdx1] or CpbSize[SchedSelIdx1] may differ from the value of BitRate[SchedSelIdx0] or CpbSize[SchedSelIdx0] for the value SchedSelIdx0 of SchedSelIdx that was in use for the coded video sequence containing access unit n - 1.

– Otherwise, the HSS continues to operate with the previous values of SchedSelIdx, BitRate[SchedSelIdx] and CpbSize[SchedSelIdx].

When the HSS selects values of BitRate[SchedSelIdx] or CpbSize[SchedSelIdx] that differ from those of the previous access unit, the following applies.

– the variable BitRate[SchedSelIdx] comes into effect at time $t_{ai}(n)$

– the variable CpbSize[SchedSelIdx] comes into effect as follows.

  – If the new value of CpbSize[SchedSelIdx] exceeds the old CPB size, it comes into effect at time $t_{ai}(n)$,

  – Otherwise, the new value of CpbSize[SchedSelIdx] comes into effect at the time $t_r(n)$.

C.1.2 Timing of coded picture removal

For access unit 0, the nominal removal time of the access unit from the CPB is specified by

$$t_{r,n}(0) = initial_cpb_removal_delay[SchedSelIdx] ÷ 90000 \quad (C-7)$$

For the first access unit of a buffering period that does not initialise the HRD, the nominal removal time of the access unit from the CPB is specified by

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * cpb_removal_delay(n) \quad (C-8)$$

where $t_{r,n}(n_b)$ is the nominal removal time of the first access unit of the previous buffering period and cpb_removal_delay(n) is the value of cpb_removal_delay specified in the picture timing SEI message associated with access unit n.

When an access unit n is the first access unit of a buffering period, n_b is set equal to n at the removal time of access unit n.

The nominal removal time $t_{r,n}(n)$ of an access unit n that is not the first access unit of a buffering period is given by

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * cpb_removal_delay(n) \quad (C-9)$$

where $t_{r,n}(n_b)$ is the nominal removal time of the first access unit of the current buffering period and cpb_removal_delay(n) is the value of cpb_removal_delay specified in the picture timing SEI message associated with access unit n.

The removal time of access unit n is specified as follows.

– If low_delay_hrd_flag is equal to 0 or $t_{r,n}(n) >= t_{af}(n)$, the removal time of access unit n is specified by

$$t_r(n) = t_{r,n}(n) \quad \text{(C-10)}$$

– Otherwise (low_delay_hrd_flag is equal to 1 and $t_{r,n}(n) < t_{af}(n)$), the removal time of access unit n is specified by

$$t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) \div t_c) \quad \text{(C-11)}$$

NOTE – The latter case indicates that the size of access unit n, b(n), is so large that it prevents removal at the nominal removal time.

C.2 Operation of the decoded picture buffer (DPB)

The decoded picture buffer contains frame buffers. Each of the frame buffers may contain a decoded frame, a decoded complementary field pair or a single (non-paired) decoded field that are marked as "used for reference" (reference pictures) or are held for future output (reordered or delayed pictures). Prior to initialisation, the DPB is empty (the DPB fullness is set to zero). The following steps of the subclauses of this subclause all happen instantaneously at $t_r(n)$ and in the sequence listed.

C.2.1 Decoding of gaps in frame_num and storage of "non-existing" frames

If applicable, gaps in frame_num are detected by the decoding process and the generated frames are marked and inserted into the DPB as specified below.

Gaps in frame_num are detected by the decoding process and the generated frames are marked as specified in subclause 8.2.5.2.

After the marking of each generated frame, each picture m marked by the “sliding window” process as “unused for reference” is removed from the DPB when it is also marked as "non-existing" or its DPB output time is less than or equal to the CPB removal time of the current picture n; i.e., $t_{o,dpb}(m) <= t_r(n)$. When a frame or the last field in a frame buffer is removed from the DPB, the DPB fullness is decremented by one. The “non-existing” generated frame is inserted into the DPB and the DPB fullness is incremented by one.

C.2.2 Picture decoding and output

Picture n is decoded and its DPB output time $t_{o,dpb}(n)$ is derived by

$$t_{o,dpb}(n) = t_r(n) + t_c * \text{dpb_output_delay}(n) \quad \text{(C-12)}$$

The output of the current picture is specified as follows.

– If $t_{o,dpb}(n) = t_r(n)$, the current picture is output.
NOTE – When the current picture is a reference picture it will be stored in the DPB.

– Otherwise ($t_{o,dpb}(n) > t_r(n)$), the current picture is output later and will be stored in the DPB (as specified in subclause C.2.4) and is output at time $t_{o,dpb}(n)$ unless indicated not to be output by the decoding or inference of no_output_of_prior_pics_flag equal to 1 at a time that precedes $t_{o,dpb}(n)$.

The output picture shall be cropped, using the cropping rectangle specified in the sequence parameter set for the sequence.

When picture n is a picture that is output and is not the last picture of the bitstream that is output, the value of $\Delta t_{o,dpb}(n)$ is defined as:

$$\Delta t_{o,dpb}(n) = t_{o,dpb}(n_n) - t_{o,dpb}(n) \quad \text{(C-13)}$$

where n_n indicates the picture that follows after picture n in output order.

The decoded picture is temporarily stored (not in the DPB).

C.2.3 Removal of pictures from the DPB before possible insertion of the current picture

The removal of pictures from the DPB before possible insertion of the current picture proceeds as follows.

– If the decoded picture is an IDR picture the following applies.

- All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5.1.
- When the IDR picture is not the first IDR picture decoded and the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the active sequence parameter set is different from the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the sequence parameter set that was active for the preceding sequence, respectively, no_output_of_prior_pics_flag is inferred to be equal to 1 by the HRD, regardless of the actual value of no_output_of_prior_pics_flag.

  NOTE – Decoder implementations should try to handle frame or DPB size changes more gracefully than the HRD in regard to changes in PicWidthInMbs or FrameHeightInMbs.
- When no_output_of_prior_pics_flag is equal to 1 or is inferred to be equal to 1, all frame buffers in the DPB are emptied without output of the pictures they contain, and DPB fullness is set to 0.

– Otherwise (the decoded picture is not an IDR picture), the following applies.
  - If the slice header of the current picture includes memory_management_control_operation equal to 5, all reference pictures in the DPB are marked as "unused for reference".
  - Otherwise (the slice header of the current picture does not include memory_management_control_operation equal to 5), the decoded reference picture marking process specified in subclause 8.2.5 is invoked.

All pictures m in the DPB, for which all of the following conditions are true, are removed from the DPB.

– picture m is marked as “unused for reference” or picture m is a non-reference picture. When a picture is a reference frame, it is considered to be marked as "unused for reference" only when both of its fields have been marked as "unused for reference".

– picture m is marked as "non-existing" or its DPB output time is less than or equal to the CPB removal time of the current picture n; i.e., $t_{o,dpb}(m) <= t_r(n)$

When a frame or the last field in a frame buffer is removed from the DPB, the DPB fullness is decremented by one.

C.2.4 Current decoded picture marking and storage

C.2.4.1 Marking and storage of a reference decoded picture into the DPB

When the current picture is a reference picture it is stored in the DPB as follows.

– If the current decoded picture is a second field (in decoding order) of a complementary reference field pair, and the first field of the pair is still in the DPB, the current decoded picture is stored in the same frame buffer as the first field of the pair.

– Otherwise, the current decoded picture is stored in an empty frame buffer, and the DPB fullness is incremented by one.

C.2.4.2 Storage of a non-reference picture into the DPB

When the current picture is a non-reference picture and current picture n has $t_{o,dpb}(n) > t_r(n)$, it is stored in the DPB as follows.

– If the current decoded picture is a second field (in decoding order) of a complementary non-reference field pair, and the first field of the pair is still in the DPB, the current decoded picture is stored in the same frame buffer as the first field of the pair.

– Otherwise, the current decoded picture is stored in an empty frame buffer, and the DPB fullness is incremented by one.

C.3 Bitstream conformance

A bitstream of coded data conforming to this Recommendation | International Standard fulfils the following requirements.

The bitstream is constructed according to the syntax, semantics, and constraints specified in this Recommendation | International Standard outside of this Annex.

The bitstream is tested by the HRD as specified below:

For Type I bitstreams, the number of tests carried out is equal to cpb_cnt_minus1 + 1 where cpb_cnt_minus1 is either the syntax element of hrd_parameters( ) following the vcl_hrd_parameters_present_flag or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for

each bit rate and CPB size combination specified by hrd_parameters( ) following the vcl_hrd_parameters_present_flag. Each of these tests is conducted at the Type I conformance point shown in Figure C-1.

For Type II bitstreams there are two sets of tests. The number of tests of the first set is equal to cpb_cnt_minus1 + 1 where cpb_cnt_minus1 is either the syntax element of hrd_parameters( ) following the vcl_hrd_parameters_present_flag or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for each bit rate and CPB size combination. Each of these tests is conducted at the Type I conformance point shown in Figure C-1. For these tests, only VCL and filler data NAL units are counted for the input bit rate and CPB storage.

The number of tests of the second set, for Type II bitstreams, is equal to cpb_cnt_minus1 + 1 where cpb_cnt_minus1 is either the syntax element of hrd_parameters( ) following the nal_hrd_parameters_present_flag or is determined by the application by other means not specified in this Recommendation | International Standard. One test is carried out for each bit rate and CPB size combination specified by hrd_parameters( ) following the nal_hrd_parameters_present_flag. Each of these tests is conducted at the Type II conformance point shown in Figure C-1. For these tests, all NAL units (of a Type II NAL unit stream) or all bytes (of a byte stream) are counted for the input bit rate and CPB storage.

NOTE 1 – NAL HRD parameters established by a value of SchedSelIdx for the Type II conformance point shown in Figure C-1 are sufficient to also establish VCL HRD conformance for the Type I conformance point shown in Figure C-1 for the same values of initial_cpb_removal_delay[SchedSelIdx], BitRate[SchedSelIdx], and CpbSize[SchedSelIdx] for the VBR case (cbr_flag[SchedSelIdx] equal to 0). This is because the data flow into the Type I conformance point is a subset of the data flow into the Type II conformance point and because, for the VBR case, the CPB is allowed to become empty and stay empty until the time a next picture is scheduled to begin to arrive. For example, when NAL HRD parameters are provided for the Type II conformance point that not only fall within the bounds set for NAL HRD parameters for profile conformance in item j of subclause A.3.1 or item i of subclause A.3.3 (depending on the profile in use) but also fall within the bounds set for VCL HRD parameters for profile conformance in item i of subclause A.3.1 or item h of subclause A.3.3 (depending on the profile in use), conformance of the VCL HRD for the Type I conformance point is also assured to fall within the bounds of item i of subclause A.3.1.

For conforming bitstreams, all of the following conditions shall be fulfilled for each of the tests.

– For each access unit n, with n>0, associated with a buffering period SEI message, with $\Delta t_{g,90}(n)$ specified by

$$\Delta t_{g,90}(n) = 90000 * (t_{r,n}(n) - t_{af}(n - 1)) \quad \text{(C-14)}$$

The value of initial_cpb_removal_delay[SchedSelIdx] shall be constrained as follows.

– If cbr_flag[SchedSelIdx] is equal to 0,

$$\text{initial_cpb_removal_delay[SchedSelIdx]} <= \text{Ceil}(\Delta t_{g,90}(n)) \quad \text{(C-15)}$$

– Otherwise (cbr_flag[SchedSelIdx] is equal to 1),

$$\text{Floor}(\Delta t_{g,90}(n)) <= \text{initial_cpb_removal_delay[SchedSelIdx]} <= \text{Ceil}(\Delta t_{g,90}(n)) \quad \text{(C-16)}$$

NOTE 2 – The exact number of bits in the CPB at the removal time of each picture may depend on which buffering period SEI message is selected to initialize the HRD. Encoders must take this into account to ensure that all specified constraints must be obeyed regardless of which buffering period SEI message is selected to initialize the HRD, as the HRD may be initialised at any one of the buffering period SEI messages.

– A CPB overflow is specified as the condition in which the total number of bits in the CPB is larger than the CPB size. The CPB shall never overflow.

– A CPB underflow is specified as the condition in which tr,n(n) is less than taf(n). When low_delay_hrd_flag is equal to 0, the CPB shall never underflow.

– The nominal removal times of pictures from the CPB (starting from the second picture in decoding order), shall satisfy the constraints on tr,n(n) and tr(n) expressed in subclauses A.3.1 through A.3.3 for the profile and level specified in the bitstream.

– Immediately after any decoded picture is added to the DPB, the fullness of the DPB shall be less than or equal to the DPB size as constrained by Annexes A, D, and E for the profile and level specified in the bitstream.

– All reference pictures shall be present in the DPB when needed for prediction. Each picture shall be present in the DPB at its DPB output time unless it is not stored in the DPB at all, or is removed from the DPB before its output time by one of the processes specified in subclause C.2.

– The value of $\Delta_{to,dpb}(n)$ as given by Equation C-13, which is the difference between the output time of a picture and that of the picture immediately following it in output order, shall satisfy the constraint expressed in subclause A.3.1 for the profile and level specified in the bitstream.

C.4 Decoder conformance

A decoder conforming to this Recommendation | International Standard fulfils the following requirements.

A decoder claiming conformance to a specific profile and level shall be able to decode successfully all conforming bitstreams specified for decoder conformance in subclause C.3, provided that all sequence parameter sets and picture parameters sets referred to in the VCL NAL units, and appropriate buffering period and picture timing SEI messages are conveyed to the decoder, in a timely manner, either in the bitstream (by non-VCL NAL units), or by external means not specified by this Recommendation | International Standard.

There are two types of conformance that can be claimed by a decoder: output timing conformance and output order conformance.

To check conformance of a decoder, test bitstreams conforming to the claimed profile and level, as specified by subclause C.3 are delivered by a hypothetical stream scheduler (HSS) both to the HRD and to the decoder under test (DUT). All pictures output by the HRD shall also be output by the DUT and, for each picture output by the HRD, the values of all samples that are output by the DUT for the corresponding picture shall be equal to the values of the samples output by the HRD.

For output timing decoder conformance, the HSS operates as described above, with delivery schedules selected only from the subset of values of SchedSelIdx for which the bit rate and CPB size are restricted as specified in Annex A for the specified profile and level, or with "interpolated" delivery schedules as specified below for which the bit rate and CPB size are restricted as specified in Annex A. The same delivery schedule is used for both the HRD and DUT.

When the HRD parameters and the buffering period SEI messages are present with cpb_cnt_minus1 greater than 0, the decoder shall be capable of decoding the bitstream as delivered from the HSS operating using an "interpolated" delivery schedule specified as having peak bit rate r, CPB size c(r), and initial CPB removal delay (f(r) ÷ r) as follows

$$\alpha = (r - BitRate[SchedSelIdx - 1]) \div (BitRate[SchedSelIdx] - BitRate[SchedSelIdx - 1]), \quad \text{(C-17)}$$

$$c(r) = \alpha * CpbSize[SchedSelIdx] + (1 - \alpha) * CpbSize[SchedSelIdx-1], \quad \text{(C-18)}$$

$$f(r) = \alpha * initial_cpb_removal_delay[SchedSelIdx] * BitRate[SchedSelIdx] + (1 - \alpha) * initial_cpb_removal_delay[SchedSelIdx - 1] * BitRate[SchedSelIdx - 1] \quad \text{(C-19)}$$

for any SchedSelIdx > 0 and r such that BitRate[SchedSelIdx - 1] <= r <= BitRate[SchedSelIdx] such that r and c(r) are within the limits as specified in Annex A for the maximum bit rate and buffer size for the specified profile and level.

NOTE 1 – initial_cpb_removal_delay[SchedSelIdx] can be different from one buffering period to another and have to be re-calculated.

For output timing decoder conformance, an HRD as described above is used and the timing (relative to the delivery time of the first bit) of picture output is the same for both HRD and the DUT up to a fixed delay.

For output order decoder conformance, the HSS delivers the bitstream to the DUT "by demand" from the DUT, meaning that the HSS delivers bits (in decoding order) only when the DUT requires more bits to proceed with its processing.

NOTE 2 – This means that for this test, the coded picture buffer of the DUT could be as small as the size of the largest access unit.

A modified HRD as described below is used, and the HSS delivers the bitstream to the HRD by one of the schedules specified in the bitstream such that the bit rate and CPB size are restricted as specified in Annex A. The order of pictures output shall be the same for both HRD and the DUT.

For output order decoder conformance, the HRD CPB size is equal to CpbSize[SchedSelIdx] for the selected schedule and the DPB size is equal to MaxDpbSize. Removal time from the CPB for the HRD is equal to final bit arrival time and decoding is immediate. The operation of the DPB of this HRD is described below.

C.4.1 Operation of the output order DPB

The decoded picture buffer contains frame buffers. Each of the frame buffers may contain a decoded frame, a decoded complementary field pair or a single (non-paired) decoded field that is marked as "used for reference" or is held for future output (reordered pictures). At HRD initialization, the DPB fullness, measured in frames, is set to 0. The following steps all happen instantaneously when an access unit is removed from the CPB, and in the order listed.

C.4.2 Decoding of gaps in frame_num and storage of "non-existing" pictures

When applicable, gaps in frame_num are detected by the decoding process and the necessary number of "non-existing" frames are inferred in the order specified by the generation of values of UnusedShortTermFrameNum in Equation 7-21 and are marked as specified in subclause 8.2.5.2. Frame buffers containing a frame or a complementary field pair or a non-paired field which are marked as "not needed for output" and "unused for reference" are emptied (without output), and the DPB fullness is decremented by the number of frame buffers emptied. Each "non-existing" frame is stored in the DPB as follows.

- When there is no empty frame buffer (i.e., DPB fullness is equal to DPB size), the "bumping" process specified in subclause C.4.5.3 is invoked repeatedly until there is an empty frame buffer in which to store the "non-existing" frame.
- The "non-existing" frame is stored in an empty frame buffer and is marked as "not needed for output", and the DPB fullness is incremented by one.

C.4.3 Picture decoding

Primary coded picture n is decoded and is temporarily stored (not in the DPB).

C.4.4 Removal of pictures from the DPB before possible insertion of the current picture

The removal of pictures from the DPB before possible insertion of the current picture proceeds as follows.

- If the decoded picture is an IDR picture the following applies.
  - All reference pictures in the DPB are marked as "unused for reference" as specified in subclause 8.2.5.
  - When the IDR picture is not the first IDR picture decoded and the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the active sequence parameter set is different from the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering derived from the sequence parameter set that was active for the preceding sequence, respectively, no_output_of_prior_pics_flag is inferred to be equal to 1 by the HRD, regardless of the actual value of no_output_of_prior_pics_flag.

    NOTE – Decoder implementations should try to handle changes in the value of PicWidthInMbs or FrameHeightInMbs or max_dec_frame_buffering more gracefully than the HRD.
  - When no_output_of_prior_pics_flag is equal to 1 or is inferred to be equal to 1, all frame buffers in the DPB are emptied without output of the pictures they contain, and DPB fullness is set to 0.
- Otherwise (the decoded picture is not an IDR picture), the decoded reference picture marking process is invoked as specified in subclause 8.2.5. Frame buffers containing a frame or a complementary field pair or a non-paired field which are marked as "not needed for output" and "unused for reference" are emptied (without output), and the DPB fullness is decremented by the number of frame buffers emptied.

When the current picture has a memory_management_control_operation equal to 5 or is an IDR picture for which no_output_of_prior_pics_flag is not equal to 1 and is not inferred to be equal to 1, the following two steps are performed.

1. Frame buffers containing a frame or a complementary field pair or a non-paired field which are marked as "not needed for output" and "unused for reference" are emptied (without output), and the DPB fullness is decremented by the number of frame buffers emptied.
2. All non-empty frame buffers in the DPB are emptied by repeatedly invoking the “bumping” process specified in subclause C.4.5.3, and the DPB fullness is set to 0.

C.4.5 Current decoded picture marking and storage

C.4.5.1 Storage and marking of a reference decoded picture into the DPB

When the current picture is a reference picture, it is stored in the DPB as follows.

- If the current decoded picture is the second field (in decoding order) of a complementary reference field pair, and the first field of the pair is still in the DPB, the current picture is stored in the same frame buffer as the first field of the pair and is marked as "needed for output".

- Otherwise, the following operations are performed:
  - When there is no empty frame buffer (i.e., DPB fullness is equal to DPB size), the "bumping" process specified in subclause C.4.5.3 is invoked repeatedly until there is an empty frame buffer in which to store the current decoded picture.
  - The current decoded picture is stored in an empty frame buffer and is marked as "needed for output", and the DPB fullness is incremented by one.

C.4.5.2 Storage and marking of a non-reference decoded picture into the DPB

When the current picture is a non-reference picture, the following operations are performed.

- If the current decoded picture is the second field (in decoding order) of a complementary non-reference field pair and the first field of the pair is still in the DPB, the current picture is stored in the same frame buffer as the first field of the pair and is marked as "needed for output".
- Otherwise, the following operations are performed repeatedly until the current decoded picture has been cropped and output or has been stored in the DPB:
  - If there is no empty frame buffer (i.e., DPB fullness is equal to DPB size), the following applies.
    - If the current picture does not have a lower value of PicOrderCnt( ) than all pictures in the DPB that are marked as "needed for output", the "bumping" process described in subclause C.4.5.3 is performed.
    - Otherwise (the current picture has a lower value of PicOrderCnt( ) than all pictures in the DPB that are marked as "needed for output"), the current picture is cropped, using the cropping rectangle specified in the sequence parameter set for the sequence and the cropped picture is output.
  - Otherwise (there is an empty frame buffer, i.e., DPB fullness is less than DPB size) the current decoded picture is stored in an empty frame buffer and is marked as "needed for output", and the DPB fullness is incremented by one.

C.4.5.3 "Bumping" process

The "bumping" process is invoked in the following cases.

- There is no empty frame buffer (i.e., DPB fullness is equal to DPB size) and a empty frame buffer is needed for storage of an inferred "non-existing" frame, as specified in subclause C.4.2.
- The current picture is an IDR picture and no_output_of_prior_pics_flag is not equal to 1 and is not inferred to be equal to 1, as specified in subclause C.4.4.
- The current picture has memory_management_control_operation equal to 5, as specified in subclause C.4.4.
- There is no empty frame buffer (i.e., DPB fullness is equal to DPB size) and an empty frame buffer is needed for storage of a decoded (non-IDR) reference picture, as specified in subclause C.4.5.1.
- There is no empty frame buffer (i.e., DPB fullness is equal to DPB size) and the current picture is a non-reference picture that is not the second field of a complementary non-reference field pair and there are pictures in the DPB that are marked as "needed for output" that precede the current non-reference picture in output order, as specified in subclause C.4.5.2, so an empty buffer is needed for storage of the current picture.

The "bumping" process consists of the following:

- The picture or complementary reference field pair that is first for output is selected as follows.
  - The frame buffer is selected that contains the picture having the smallest value of PicOrderCnt( ) of all pictures in the DPB marked as "needed for output".
  - If this frame buffer contains a complementary non-reference field pair with both fields marked as "needed for output" and both fields have the same PicOrderCnt( ), the first of these two fields in decoding order is considered first for output.
  - Otherwise, if this frame buffer contains a complementary reference field pair with both fields marked as "needed for output" and both fields have the same PicOrderCnt( ), the entire complementary reference field pair is considered first for output.
  - Otherwise, the picture in this frame buffer that has the smallest value of PicOrderCnt( ) is considered first for output.
- If a single picture is considered first for output, this picture is cropped, using the cropping rectangle specified in the sequence parameter set for the sequence, the cropped picture is output, and the picture is marked as "not needed for output".

- Otherwise (a complementary reference field pair is considered first for output), the two fields of the complementary reference field pair are both cropped, using the cropping rectangle specified in the sequence parameter set for the sequence, the two fields of the complementary reference field pair are output together, and both fields of the complementary reference field pair are marked as "not needed for output".
- The frame buffer that included the picture or complementary reference field pair that was cropped and output is checked, and when any of the following conditions is satisfied, the frame buffer is emptied and the DPB fullness is decremented by 1.
  - The frame buffer contains a non-reference non-paired field.
  - The frame buffer contains a non-reference frame.
  - The frame buffer contains a complementary non-reference field pair with both fields marked as "not needed for output".
  - The frame buffer contains a non-paired reference field marked as "unused for reference".
  - The frame buffer contains a reference frame with both fields marked as "unused for reference".
  - The frame buffer contains a complementary reference field pair with both fields marked as "unused for reference" and "not needed for output".

Annex D
Supplemental enhancement information

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies syntax and semantics for SEI message payloads.

SEI messages assist in processes related to decoding, display or other purposes. However, SEI messages are not required for constructing the luma or chroma samples by the decoding process. Conforming decoders are not required to process this information for output order conformance to this Recommendation | International Standard (see Annex C for the specification of conformance). Some SEI message information is required to check bitstream conformance and for output timing decoder conformance.

In Annex D, specification for presence of SEI messages are also satisfied when those messages (or some subset of them) are conveyed to decoders (or to the HRD) by other means not specified by this Recommendation | International Standard. When present in the bitstream, SEI messages shall obey the syntax and semantics specified in subclauses 7.3.2.3 and 7.4.2.3 and this annex. When the content of an SEI message is conveyed for the application by some means other than presence within the bitstream, the representation of the content of the SEI message is not required to use the same syntax specified in this annex. For the purpose of counting bits, only the appropriate bits that are actually present in the bitstream are counted.

D.1 SEI payload syntax

sei_payload(payloadType, payloadSize) {	C	Descriptor
if(payloadType == 0)		
buffering_period(payloadSize)	5	
else if(payloadType == 1)		
pic_timing(payloadSize)	5	
else if(payloadType == 2)		
pan_scan_rect(payloadSize)	5	
else if(payloadType == 3)		
filler_payload(payloadSize)	5	
else if(payloadType == 4)		
user_data_registered_itu_t_t35(payloadSize)	5	
else if(payloadType == 5)		
user_data_unregistered(payloadSize)	5	
else if(payloadType == 6)		
recovery_point(payloadSize)	5	
else if(payloadType == 7)		
dec_ref_pic_marking_repetition(payloadSize)	5	
else if(payloadType == 8)		
spare_pic(payloadSize)	5	
else if(payloadType == 9)		
scene_info(payloadSize)	5	
else if(payloadType == 10)		
sub_seq_info(payloadSize)	5	
else if(payloadType == 11)		
sub_seq_layer_characteristics(payloadSize)	5	
else if(payloadType == 12)		
sub_seq_characteristics(payloadSize)	5	
else if(payloadType == 13)		
full_frame_freeze(payloadSize)	5	
else if(payloadType == 14)		
full_frame_freeze_release(payloadSize)	5	
else if(payloadType == 15)		
full_frame_snapshot(payloadSize)	5	
else if(payloadType == 16)		
progressive_refinement_segment_start(payloadSize)	5	
else if(payloadType == 17)		
progressive_refinement_segment_end(payloadSize)	5	
else if(payloadType == 18)		
motion_constrained_slice_group_set(payloadSize)	5	
else if(payloadType == 19)		
film_grain_characteristics(payloadSize)	5	
else if(payloadType == 20)		
deblocking_filter_display_preference(payloadSize)	5	
else if(payloadType == 21)		
stereo_video_info(payloadSize)	5	
else		
reserved_sei_message(payloadSize)	5	

if(!byte_aligned( )) {		
bit_equal_to_one /* equal to 1 */	5	f(1)
while(!byte_aligned( ))		
bit_equal_to_zero /* equal to 0 */	5	f(1)
}		
}		

D.1.1 Buffering period SEI message syntax

buffering_period(payloadSize) {	C	Descriptor
seq_parameter_set_id	5	ue(v)
if(NalHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx **]**	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx **]**	5	u(v)
}		
}		
if(VclHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx **]**	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx **]**	5	u(v)
}		
}		
}		

D.1.2 Picture timing SEI message syntax

pic_timing(payloadSize) {	C	Descriptor
if(CpbDpbDelaysPresentFlag) {		
cpb_removal_delay	5	u(v)
dpb_output_delay	5	u(v)
}		
if(pic_struct_present_flag) {		
pic_struct	5	u(4)
for(i = 0; i < NumClockTS ; i++) {		
clock_timestamp_flag[i]	5	u(1)
if(clock_timestamp_flag[i]) {		
ct_type	5	u(2)
nuit_field_based_flag	5	u(1)
counting_type	5	u(5)
full_timestamp_flag	5	u(1)
discontinuity_flag	5	u(1)
cnt_dropped_flag	5	u(1)
n_frames	5	u(8)
if(full_timestamp_flag) {		
seconds_value /* 0..59 */	5	u(6)

minutes_value /* 0..59 */	5	u(6)
hours_value /* 0..23 */	5	u(5)
} else {		
seconds_flag	5	u(1)
if(seconds_flag) {		
seconds_value /* range 0..59 */	5	u(6)
minutes_flag	5	u(1)
if(minutes_flag) {		
minutes_value /* 0..59 */	5	u(6)
hours_flag	5	u(1)
if(hours_flag)		
hours_value /* 0..23 */	5	u(5)
}		
}		
}		
if(time_offset_length > 0)		
time_offset	5	i(v)
}		
}		
}		
}		

D.1.3 Pan-scan rectangle SEI message syntax

pan_scan_rect(payloadSize) {	C	Descriptor
pan_scan_rect_id	5	ue(v)
pan_scan_rect_cancel_flag	5	u(1)
if(!pan_scan_rect_cancel_flag) {		
pan_scan_cnt_minus1	5	ue(v)
for(i = 0; i <= pan_scan_cnt_minus1; i++) {		
pan_scan_rect_left_offset[i **]**	5	se(v)
pan_scan_rect_right_offset[i **]**	5	se(v)
pan_scan_rect_top_offset[i **]**	5	se(v)
pan_scan_rect_bottom_offset[i **]**	5	se(v)
}		
pan_scan_rect_repetition_period	5	ue(v)
}		
}		

D.1.4 Filler payload SEI message syntax

filler_payload(payloadSize) {	C	Descriptor
for(k = 0; k < payloadSize; k++)		
ff_byte /* equal to 0xFF */	5	f(8)
}		

D.1.5 User data registered by ITU-T Rec. T.35 SEI message syntax

user_data_registered_itu_t_t35(payloadSize) {	C	Descriptor
itu_t_t35_country_code	5	b(8)
if(itu_t_t35_country_code != 0xFF)		
i = 1		
else {		
itu_t_t35_country_code_extension_byte	5	b(8)
i = 2		
}		
do {		
itu_t_t35_payload_byte	5	b(8)
i++		
} while(i < payloadSize)		
}		

D.1.6 User data unregistered SEI message syntax

user_data_unregistered(payloadSize) {	C	Descriptor
uuid_iso_iec_11578	**5**	u(128)
for(i = 16; i < payloadSize; i++)		
user_data_payload_byte	5	b(8)
}		

D.1.7 Recovery point SEI message syntax

recovery_point(payloadSize) {	C	Descriptor
recovery_frame_cnt	5	ue(v)
exact_match_flag	5	u(1)
broken_link_flag	5	u(1)
changing_slice_group_idc	5	u(2)
}		

D.1.8 Decoded reference picture marking repetition SEI message syntax

dec_ref_pic_marking_repetition(payloadSize) {	C	Descriptor
original_idr_flag	5	u(1)
original_frame_num	5	ue(v)
if(!frame_mbs_only_flag) {		
original_field_pic_flag	5	u(1)
if(original_field_pic_flag)		
original_bottom_field_flag	5	u(1)
}		
dec_ref_pic_marking( )	5	
}		

D.1.9 Spare picture SEI message syntax

spare_pic(payloadSize) {	C	Descriptor
target_frame_num	5	ue(v)
spare_field_flag	5	u(1)
if(spare_field_flag)		
target_bottom_field_flag	5	u(1)
num_spare_pics_minus1	5	ue(v)
for(i = 0; i < num_spare_pics_minus1 + 1; i++) {		
delta_spare_frame_num[i **]**	5	ue(v)
if(spare_field_flag)		
spare_bottom_field_flag[i **]**	5	u(1)
spare_area_idc[i **]**	5	ue(v)
if(spare_area_idc[i] == 1)		
for(j = 0; j < PicSizeInMapUnits; j++)		
spare_unit_flag[i **][** j **]**	5	u(1)
else if(spare_area_idc[i] == 2) {		
mapUnitCnt = 0		
for(j=0; mapUnitCnt < PicSizeInMapUnits; j++) {		
zero_run_length[i **][** j **]**	5	ue(v)
mapUnitCnt += zero_run_length[i][j] + 1		
}		
}		
}		
}		

D.1.10 Scene information SEI message syntax

scene_info(payloadSize) {	C	Descriptor
scene_info_present_flag	5	u(1)
if(scene_info_present_flag) {		
scene_id	5	ue(v)
scene_transition_type	5	ue(v)
if(scene_transition_type > 3)		
second_scene_id	5	ue(v)
}		
}		

D.1.11 Sub-sequence information SEI message syntax

sub_seq_info(payloadSize) {	C	Descriptor
sub_seq_layer_num	5	ue(v)
sub_seq_id	5	ue(v)
first_ref_pic_flag	5	u(1)
leading_non_ref_pic_flag	5	u(1)
last_pic_flag	5	u(1)
sub_seq_frame_num_flag	5	u(1)
if(sub_seq_frame_num_flag)		
sub_seq_frame_num	5	ue(v)
}		

D.1.12 Sub-sequence layer characteristics SEI message syntax

sub_seq_layer_characteristics(payloadSize) {	C	Descriptor
num_sub_seq_layers_minus1	5	ue(v)
for(layer = 0; layer <= num_sub_seq_layers_minus1; layer++) {		
accurate_statistics_flag	5	u(1)
average_bit_rate	5	u(16)
average_frame_rate	5	u(16)
}		
}		

D.1.13 Sub-sequence characteristics SEI message syntax

sub_seq_characteristics(payloadSize) {	C	Descriptor
sub_seq_layer_num	5	ue(v)
sub_seq_id	5	ue(v)
duration_flag	5	u(1)
if(duration_flag)		
sub_seq_duration	5	u(32)
average_rate_flag	5	u(1)
if(average_rate_flag) {		
accurate_statistics_flag	5	u(1)
average_bit_rate	5	u(16)
average_frame_rate	5	u(16)
}		
num_referenced_subseqs	5	ue(v)
for(n = 0; n < num_referenced_subseqs; n++) {		
ref_sub_seq_layer_num	5	ue(v)
ref_sub_seq_id	5	ue(v)
ref_sub_seq_direction	5	u(1)
}		
}		

D.1.14 Full-frame freeze SEI message syntax

full_frame_freeze(payloadSize) {	C	Descriptor
full_frame_freeze_repetition_period	5	ue(v)
}		

D.1.15 Full-frame freeze release SEI message syntax

full_frame_freeze_release(payloadSize) {	C	Descriptor
}		

D.1.16 Full-frame snapshot SEI message syntax

full_frame_snapshot(payloadSize) {	C	Descriptor
snapshot_id	5	ue(v)
}		

D.1.17 Progressive refinement segment start SEI message syntax

progressive_refinement_segment_start(payloadSize) {	C	Descriptor
progressive_refinement_id	5	ue(v)
num_refinement_steps_minus1	5	ue(v)
}		

D.1.18 Progressive refinement segment end SEI message syntax

progressive_refinement_segment_end(payloadSize) {	C	Descriptor
progressive_refinement_id	5	ue(v)
}		

D.1.19 Motion-constrained slice group set SEI message syntax

motion_constrained_slice_group_set(payloadSize) {	C	Descriptor
num_slice_groups_in_set_minus1	5	ue(v)
for(i = 0; i <= num_slice_groups_in_set_minus1; i++)		
slice_group_id[i]	5	u(v)
exact_sample_value_match_flag	5	u(1)
pan_scan_rect_flag	5	u(1)
if(pan_scan_rect_flag)		
pan_scan_rect_id	5	ue(v)
}		

D.1.20 Film grain characteristics SEI message syntax

film_grain_characteristics(payloadSize) {	C	Descriptor
film_grain_characteristics_cancel_flag	5	u(1)
if(!film_grain_**characteristics**_cancel_flag) {		
model_id	5	u(2)
separate_colour_description_present_flag	5	u(1)
if(separate_colour_description_present_flag) {		
film_grain_bit_depth_luma_minus8	5	u(3)
film_grain_bit_depth_chroma_minus8	5	u(3)
film_grain_full_range_flag	5	u(1)
film_grain_colour_primaries	5	u(8)
film_grain_transfer_characteristics	5	u(8)
film_grain_matrix_coefficients	5	u(8)
}		
blending_mode_id	5	u(2)
log2_scale_factor	5	u(4)
for(c = 0; c < 3; c++)		
comp_model_present_flag[c]	5	u(1)
for(c = 0; c < 3; c++)		
if(comp_model_present_flag[c]) {		
num_intensity_intervals_minus1[c]	5	u(8)
num_model_values_minus1[c]	5	u(3)
for(i = 0; i <= num_intensity_intervals_minus1[c]; i++) {		
intensity_interval_lower_bound[c **][** i]	5	u(8)
intensity_interval_upper_bound[c **][** i]	5	u(8)
for(j = 0; j <= num_model_values_minus1[c]; j++)		
comp_model_value[c **][** i **][** j]	5	se(v)
}		
}		
film_grain_characteristics_repetition_period	5	ue(v)
}		
}		

D.1.21 Deblocking filter display preference SEI message syntax

deblocking_filter_display_preference(payloadSize) {	C	Descriptor
deblocking_display_preference_cancel_flag	5	u(1)
if(!deblocking_display_preference_cancel_flag) {		
display_prior_to_deblocking_preferred_flag	5	u(1)
dec_frame_buffering_constraint_flag	5	u(1)
deblocking_display_preference_repetition_period	5	ue(v)
}		
}		

D.1.22 Stereo video information SEI message syntax

stereo_video_info(payloadSize) {	C	Descriptor
field_views_flag	5	u(1)
if(field_views_flag)		
top_field_is_left_view_flag	5	u(1)
else {		
current_frame_is_left_view_flag	5	u(1)
next_frame_is_second_view_flag	5	u(1)
}		
left_view_self_contained_flag	5	u(1)
right_view_self_contained_flag	5	u(1)
}		

D.1.23 Reserved SEI message syntax

reserved_sei_message(payloadSize) {	C	Descriptor
for(i = 0; i < payloadSize; i++)		
reserved_sei_message_payload_byte	5	b(8)
}		

D.2 SEI payload semantics

D.2.1 Buffering period SEI message semantics

When NalHrdBpPresentFlag or VclHrdBpPresentFlag are equal to 1, a buffering period SEI message can be associated with any access unit in the bitstream, and a buffering period SEI message shall be associated with each IDR access unit and with each access unit associated with a recovery point SEI message.

NOTE – For some applications, the frequent presence of a buffering period SEI message may be desirable.

A buffering period is specified as the set of access units between two instances of the buffering period SEI message in decoding order.

seq_parameter_set_id specifies the sequence parameter set that contains the sequence HRD attributes. The value of seq_parameter_set_id shall be equal to the value of seq_parameter_set_id in the picture parameter set referenced by the primary coded picture associated with the buffering period SEI message. The value of seq_parameter_set_id shall be in the range of 0 to 31, inclusive.

initial_cpb_removal_delay[SchedSelIdx] specifies the delay for the SchedSelIdx-th CPB between the time of arrival in the CPB of the first bit of the coded data associated with the access unit associated with the buffering period SEI message and the time of removal from the CPB of the coded data associated with the same access unit, for the first buffering period after HRD initialisation. The syntax element has a length in bits given by initial_cpb_removal_delay_length_minus1 + 1. It is in units of a 90 kHz clock. initial_cpb_removal_delay[SchedSelIdx] shall not be equal to 0 and shall not exceed 90000 * (CpbSize[SchedSelIdx] ÷ BitRate[SchedSelIdx]), the time-equivalent of the CPB size in 90 kHz clock units.

initial_cpb_removal_delay_offset[SchedSelIdx] is used for the SchedSelIdx-th CPB in combination with the cpb_removal_delay to specify the initial delivery time of coded access units to the CPB. initial_cpb_removal_delay_offset[SchedSelIdx] is in units of a 90 kHz clock. The initial_cpb_removal_delay_offset[SchedSelIdx] syntax element is a fixed length code whose length in bits is given by initial_cpb_removal_delay_length_minus1 + 1. This syntax element is not used by decoders and is needed only for the delivery scheduler (HSS) specified in Annex C.

Over the entire coded video sequence, the sum of initial_cpb_removal_delay[SchedSelIdx] and initial_cpb_removal_delay_offset[SchedSelIdx] shall be constant for each value of SchedSelIdx.

D.2.2 Picture timing SEI message semantics

The presence of picture timing SEI message in the bitstream is specified as follows.

– If CpbDpbDelaysPresentFlag is equal to 1 or pic_struct_present_flag is equal to 1, one picture timing SEI message shall be present in every access unit of the coded video sequence.

– Otherwise (CpbDpbDelaysPresentFlag is equal to 0 and pic_struct_present_flag is equal to 0), no picture timing SEI messages shall be present in any access unit of the coded video sequence.

cpb_removal_delay specifies how many clock ticks (see subclause E.2.1) to wait after removal from the CPB of the access unit associated with the most recent buffering period SEI message before removing from the buffer the access unit data associated with the picture timing SEI message. This value is also used to calculate an earliest possible time of arrival of access unit data into the CPB for the HSS, as specified in Annex C. The syntax element is a fixed length code whose length in bits is given by cpb_removal_delay_length_minus1 + 1. The cpb_removal_delay is the remainder of a $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ counter.

The value of cpb_removal_delay for the first picture in the bitstream shall be equal to 0.

dpb_output_delay is used to compute the DPB output time of the picture. It specifies how many clock ticks to wait after removal of an access unit from the CPB before the decoded picture can be output from the DPB (see subclause C.2).

NOTE 1 – A picture is not removed from the DPB at its output time when it is still marked as "used for short-term reference" or "used for long-term reference".

NOTE 2 – Only one dpb_output_delay is specified for a decoded picture.

The size of the syntax element dpb_output_delay is given in bits by dpb_output_delay_length_minus1 + 1. When max_dec_frame_buffering is equal to 0, dpb_output_delay shall be equal to 0.

The output time derived from the dpb_output_delay of any picture that is output from an output timing conforming decoder as specified in subclause C.2 shall precede the output time derived from the dpb_output_delay of all pictures in any subsequent coded video sequence in decoding order.

The output time derived from the dpb_output_delay of the second field, in decoding order, of a complementary non-reference field pair shall exceed the output time derived from the dpb_output_delay of the first field of the same complementary non-reference field pair.

The picture output order established by the values of this syntax element shall be the same order as established by the values of PicOrderCnt( ) as specified by subclauses C.4.1 to C.4.5, except that when the two fields of a complementary reference field pair have the same value of PicOrderCnt( ), the two fields have different output times.

For pictures that are not output by the "bumping" process of subclause C.4.5 because they precede, in decoding order, an IDR picture with no_output_of_prior_pics_flag equal to 1 or inferred to be equal to 1, the output times derived from dpb_output_delay shall be increasing with increasing value of PicOrderCnt( ) relative to all pictures within the same coded video sequence subsequent to any picture having a memory_management_control_operation equal to 5.

pic_struct indicates whether a picture should be displayed as a frame or one or more fields, according to Table D-1. Frame doubling (pic_struct equal to 7) indicates that the frame should be displayed two times consecutively, and frame tripling (pic_struct equal to 8) indicates that the frame should be displayed three times consecutively.

NOTE 3 – Frame doubling can facilitate the display, for example, of 25p video on a 50p display and 29.97p video on a 59.94p display. Using frame doubling and frame tripling in combination on every other frame can facilitate the display of 23.98p video on a 59.94p display.

Table D-1 – Interpretation of pic_struct

Value	Indicated display of picture	Restrictions	NumClockTS
0	frame	field_pic_flag shall be 0	1
1	top field	field_pic_flag shall be 1, bottom_field_flag shall be 0	1
2	bottom field	field_pic_flag shall be 1, bottom_field_flag shall be 1	1
3	top field, bottom field, in that order	field_pic_flag shall be 0	2
4	bottom field, top field, in that order	field_pic_flag shall be 0	2
5	top field, bottom field, top field repeated, in that order	field_pic_flag shall be 0	3
6	bottom field, top field, bottom field repeated, in that order	field_pic_flag shall be 0	3
7	frame doubling	field_pic_flag shall be 0 fixed_frame_rate_flag shall be 1	2
8	frame tripling	field_pic_flag shall be 0 fixed_frame_rate_flag shall be 1	3
9..15	reserved		

NumClockTS is determined by pic_struct as specified in Table D-1. There are up to NumClockTS sets of clock timestamp information for a picture, as specified by clock_timestamp_flag[i] for each set. The sets of clock timestamp information apply to the field(s) or the frame(s) associated with the picture by pic_struct.

The contents of the clock timestamp syntax elements indicate a time of origin, capture, or alternative ideal display. This indicated time is computed as

$$\begin{aligned} \text{clockTimestamp} = {} & ((\text{hH} * 60 + \text{mM}) * 60 + \text{sS}) * \text{time_scale} + \\ & \text{nFrames} * (\text{num_units_in_tick} * (1 + \text{nuit_field_based_flag})) + \text{tOffset}, \end{aligned} \tag{D-1}$$

in units of clock ticks of a clock with clock frequency equal to time_scale Hz, relative to some unspecified point in time for which clockTimestamp is equal to 0. Output order and DPB output timing are not affected by the value of clockTimestamp. When two or more frames with pic_struct equal to 0 are consecutive in output order and have equal values of clockTimestamp, the indication is that the frames represent the same content and that the last such frame in output order is the preferred representation.

NOTE 4 – clockTimestamp time indications may aid display on devices with refresh rates other than those well-matched to DPB output times.

clock_timestamp_flag[i **]** equal to 1 indicates that a number of clock timestamp syntax elements are present and follow immediately. clock_timestamp_flag[i] equal to 0 indicates that the associated clock timestamp syntax elements are not present. When NumClockTS is greater than 1 and clock_timestamp_flag[i] is equal to 1 for more than one value of i, the value of clockTimestamp shall be non-decreasing with increasing value of i.

ct_type indicates the scan type (interlaced or progressive) of the source material as specified in Table D-2.

Two fields of a coded frame may have different values of ct_type.

When clockTimestamp is equal for two fields of opposite parity that are consecutive in output order, both with ct_type equal to 0 (progressive) or ct_type equal to 2 (unknown), the two fields are indicated to have come from the same original progressive frame. Two consecutive fields in output order shall have different values of clockTimestamp when the value of ct_type for either field is 1 (interlaced).

Table D-2 – Mapping of ct_type to source picture scan

Value	Original picture scan
0	progressive
1	interlaced
2	unknown
3	reserved

nuit_field_based_flag is used in calculating clockTimestamp, as specified in Equation D-1.

counting_type specifies the method of dropping values of the n_frames as specified in Table D-3.

Table D-3 – Definition of counting_type values

Value	Interpretation
0	no dropping of n_frames count values and no use of time_offset
1	no dropping of n_frames count values
2	dropping of individual zero values of n_frames count
3	dropping of individual MaxFPS-1 values of n_frames count
4	dropping of the two lowest (value 0 and 1) n_frames counts when seconds_value is equal to 0 and minutes_value is not an integer multiple of 10
5	dropping of unspecified individual n_frames count values
6	dropping of unspecified numbers of unspecified n_frames count values
7..31	reserved

full_timestamp_flag equal to 1 specifies that the n_frames syntax element is followed by seconds_value, minutes_value, and hours_value. full_timestamp_flag equal to 0 specifies that the n_frames syntax element is followed by seconds_flag.

discontinuity_flag equal to 0 indicates that the difference between the current value of clockTimestamp and the value of clockTimestamp computed from the previous clock timestamp in output order can be interpreted as the time difference between the times of origin or capture of the associated frames or fields. discontinuity_flag equal to 1 indicates that the difference between the current value of clockTimestamp and the value of clockTimestamp computed from the previous clock timestamp in output order should not be interpreted as the time difference between the times of origin or capture of the associated frames or fields. When discontinuity_flag is equal to 0, the value of clockTimestamp shall be greater than or equal to all values of clockTimestamp present for the preceding picture in DPB output order.

cnt_dropped_flag specifies the skipping of one or more values of n_frames using the counting method specified by counting_type.

n_frames specifies the value of nFrames used to compute clockTimestamp. n_frames shall be less than

$$MaxFPS = Ceil(time_scale \div num_units_in_tick) \quad \text{(D-2)}$$

NOTE 5 – n_frames is a frame-based counter. For field-specific timing indications, time_offset should be used to indicate a distinct clockTimestamp for each field.

When counting_type is equal to 2 and cnt_dropped_flag is equal to 1, n_frames shall be equal to 1 and the value of n_frames for the previous picture in output order shall not be equal to 0 unless discontinuity_flag is equal to 1.

NOTE 6 – When counting_type is equal to 2, the need for increasingly large magnitudes of tOffset in Equation D-1 when using fixed non-integer frame rates (e.g., 12.5 frames per second with time_scale equal to 25 and num_units_in_tick equal to 2 and nuit_field_based_flag equal to 0) can be avoided by occasionally skipping over the value n_frames equal to 0 when counting (e.g., counting n_frames from 0 to 12, then incrementing seconds_value and counting n_frames from 1 to 12, then incrementing seconds_value and counting n_frames from 0 to 12, etc.).

When counting_type is equal to 3 and cnt_dropped_flag is equal to 1, n_frames shall be equal to 0 and the value of n_frames for the previous picture in output order shall not be equal to MaxFPS – 1 unless discontinuity_flag is equal to 1.

NOTE 7 – When counting_type is equal to 3, the need for increasingly large magnitudes of tOffset in Equation D-1 when using fixed non-integer frame rates (e.g., 12.5 frames per second with time_scale equal to 25 and num_units_in_tick equal to 2 and nuit_field_based_flag equal to 0) can be avoided by occasionally skipping over the value n_frames equal to MaxFPS when counting (e.g., counting n_frames from 0 to 12, then incrementing seconds_value and counting n_frames from 0 to 11, then incrementing seconds_value and counting n_frames from 0 to 12, etc.).

When counting_type is equal to 4 and cnt_dropped_flag is equal to 1, n_frames shall be equal to 2 and the specified value of sS shall be zero and the specified value of mM shall not be an integer multiple of ten and n_frames for the previous picture in output order shall not be equal to 0 or 1 unless discontinuity_flag is equal to 1.

NOTE 8 – When counting_type is equal to 4, the need for increasingly large magnitudes of tOffset in Equation D-1 when using fixed non-integer frame rates (e.g., 30000÷1001 frames per second with time_scale equal to 60000 and num_units_in_tick equal to 1 001 and nuit_field_based_flag equal to 1) can be reduced by occasionally skipping over the value n_frames equal to MaxFPS when counting (e.g., counting n_frames from 0 to 29, then incrementing seconds_value and counting n_frames from 0 to 29, etc., until the seconds_value is zero and minutes_value is not an integer multiple of ten, then counting n_frames from 2 to 29, then incrementing seconds_value and counting n_frames from 0 to 29, etc.). This counting method is well known in industry and is often referred to as "NTSC drop-frame" counting.

When counting_type is equal to 5 or 6 and cnt_dropped_flag is equal to 1, n_frames shall not be equal to 1 plus the value of n_frames for the previous picture in output order modulo MaxFPS unless discontinuity_flag is equal to 1.

NOTE 9 – When counting_type is equal to 5 or 6, the need for increasingly large magnitudes of tOffset in Equation D-1 when using fixed non-integer frame rates can be avoided by occasionally skipping over some values of n_frames when counting. The specific values of n_frames that are skipped are not specified when counting_type is equal to 5 or 6.

seconds_flag equal to 1 specifies that seconds_value and minutes_flag are present when full_timestamp_flag is equal to 0. seconds_flag equal to 0 specifies that seconds_value and minutes_flag are not present.

seconds_value specifies the value of sS used to compute clockTimestamp. The value of seconds_value shall be in the range of 0 to 59, inclusive. When seconds_value is not present, the previous seconds_value in decoding order shall be used as sS to compute clockTimestamp.

minutes_flag equal to 1 specifies that minutes_value and hours_flag are present when full_timestamp_flag is equal to 0 and seconds_flag is equal to 1. minutes_flag equal to 0 specifies that minutes_value and hours_flag are not present.

minutes_value specifies the value of mM used to compute clockTimestamp. The value of minutes_value shall be in the range of 0 to 59, inclusive. When minutes_value is not present, the previous minutes_value in decoding order shall be used as mM to compute clockTimestamp.

hours_flag equal to 1 specifies that hours_value is present when full_timestamp_flag is equal to 0 and seconds_flag is equal to 1 and minutes_flag is equal to 1.

hours_value specifies the value of hH used to compute clockTimestamp. The value of hours_value shall be in the range of 0 to 23, inclusive. When hours_value is not present, the previous hours_value in decoding order shall be used as hH to compute clockTimestamp.

time_offset specifies the value of tOffset used to compute clockTimestamp. The number of bits used to represent time_offset shall be equal to time_offset_length. When time_offset is not present, the value 0 shall be used as tOffset to compute clockTimestamp.

D.2.3 Pan-scan rectangle SEI message semantics

The pan-scan rectangle SEI message syntax elements specify the coordinates of a rectangle relative to the cropping rectangle of the sequence parameter set. Each coordinate of this rectangle is specified in units of one-sixteenth sample spacing relative to the luma sampling grid.

pan_scan_rect_id contains an identifying number that may be used to identify the purpose of the pan-scan rectangle (for example, to identify the rectangle as the area to be shown on a particular display device or as the area that contains a particular actor in the scene). The value of pan_scan_rect_id shall be in the range of 0 to $2^{32} - 1$, inclusive.

Values of pan_scan_rect_id from 0 to 255 and from 512 to 2^{31}-1 may be used as determined by the application. Values of pan_scan_rect_id from 256 to 511 and from 2^{31} to 2^{32}-1 are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of pan_scan_rect_id in the range of 256 to 511 or in the range of 2^{31} to 2^{32} - 1 shall ignore (remove from the bitstream and discard) it.

pan_scan_rect_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous pan-scan rectangle SEI message in output order. pan_scan_rect_cancel_flag equal to 0 indicates that pan-scan rectangle information follows.

pan_scan_cnt_minus1 specifies the number of pan-scan rectangles that are present in the SEI message. pan_scan_cnt_minus1 shall be in the range of 0 to 2, inclusive. pan_scan_cnt_minus1 equal to 0 indicates that a single pan-scan rectangle is present that applies to all fields of the decoded picture. pan_scan_cnt_minus1 shall be equal to 0 when the current picture is a field. pan_scan_cnt_minus1 equal to 1 indicates that two pan-scan rectangles are present, the first of which applies to the first field of the picture in output order and the second of which applies to the second field of the picture in output order. pan_scan_cnt_minus1 equal to 2 indicates that three pan-scan rectangles are present, the first of which applies to the first field of the picture in output order, the second of which applies to the second field of the picture in output order, and the third of which applies to a repetition of the first field as a third field in output order.

pan_scan_rect_left_offset[i **]**, **pan_scan_rect_right_offset[** i **]**, **pan_scan_rect_top_offset[** i **]**, and **pan_scan_rect_bottom_offset[** i **]**, specify, as signed integer quantities in units of one-sixteenth sample spacing relative to the luma sampling grid, the location of the pan-scan rectangle. The values of each of these four syntax elements shall be in the range of -2^{31} to 2^{31} - 1, inclusive.

The pan-scan rectangle is specified, in units of one-sixteenth sample spacing relative to a luma frame sampling grid, as the region with frame horizontal coordinates from 16*CropUnitX * frame_crop_left_offset + pan_scan_rect_left_offset[i] to 16 * (16 * PicWidthInMbs – CropUnitX * frame_crop_right_offset) + pan_scan_rect_right_offset[i] – 1 and with vertical coordinates from 16 *CropUnitY * frame_crop_top_offset + pan_scan_rect_top_offset[i] to 16 * (16 * PicHeightInMbs – CropUnitY * frame_crop_bottom_offset) + pan_scan_rect_bottom_offset[i] – 1, inclusive. The value of 16 * CropUnitX * frame_crop_left_offset + pan_scan_rect_left_offset[i] shall be less than or equal to 16 * (16 * PicWidthInMbs – CropUnitX * frame_crop_right_offset) + pan_scan_rect_right_offset[i] – 1; and the value of 16 CropUnitY * frame_crop_top_offset + pan_scan_rect_top_offset[i] shall be less than or equal to 16 * (16 * PicHeightInMbs – CropUnitY * frame_crop_bottom_offset) + pan_scan_rect_bottom_offset[i] – 1.

When the pan-scan rectangular area includes samples outside of the cropping rectangle, the region outside of the cropping rectangle may be filled with synthesized content (such as black video content or neutral grey video content) for display.

pan_scan_rect_repetition_period specifies the persistence of the pan-scan rectangle SEI message and may specify a picture order count interval within which another pan-scan rectangle SEI message with the same value of pan_scan_rect_id or the end of the coded video sequence shall be present in the bitstream. The value of pan_scan_rect_repetition_period shall be in the range of 0 to 16 384, inclusive. When pan_scan_cnt_minus1 is greater than 0, pan_scan_rect_repetition_period shall not be greater than 1.

pan_scan_rect_repetition_period equal to 0 specifies that the pan-scan rectangle information applies to the current decoded picture only.

pan_scan_rect_repetition_period equal to 1 specifies that the pan-scan rectangle information persists in output order until any of the following conditions are true.

- A new coded video sequence begins
- A picture in an access unit containing a pan-scan rectangle SEI message with the same value of pan_scan_rect_id is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

pan_scan_rect_repetition_period equal to 0 or equal to 1 indicates that another pan-scan rectangle SEI message with the same value of pan_scan_rect_id may or may not be present.

pan_scan_rect_repetition_period greater than 1 specifies that the pan-scan rectangle information persists until any of the following conditions are true.

- A new coded video sequence begins
- A picture in an access unit containing a pan-scan rectangle SEI message with the same value of pan_scan_rect_id is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + pan_scan_rect_repetition_period.

pan_scan_rect_repetition_period greater than 1 indicates that another pan-scan rectangle SEI message with the same value of pan_scan_rect_id shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater

than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + pan_scan_rect_repetition_period; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

D.2.4 Filler payload SEI message semantics

This message contains a series of payloadSize bytes of value 0xFF, which can be discarded.

ff_byte shall be a byte having the value 0xFF.

D.2.5 User data registered by ITU-T Rec. T.35 SEI message semantics

This message contains user data registered as specified by ITU-T Rec. T.35, the contents of which are not specified by this Recommendation | International Standard.

itu_t_t35_country_code shall be a byte having a value specified as a country code by ITU-T Rec. T.35 Annex A.

itu_t_t35_country_code_extension_byte shall be a byte having a value specified as a country code by ITU-T Rec. T.35 Annex B.

itu_t_t35_payload_byte shall be a byte containing data registered as specified by ITU-T Rec. T.35.

The ITU-T T.35 terminal provider code and terminal provider oriented code shall be contained in the first one or more bytes of the itu_t_t35_payload_byte, in the format specified by the Administration that issued the terminal provider code. Any remaining itu_t_t35_payload_byte data shall be data having syntax and semantics as specified by the entity identified by the ITU-T T.35 country code and terminal provider code.

D.2.6 User data unregistered SEI message semantics

This message contains unregistered user data identified by a UUID, the contents of which are not specified by this Recommendation | International Standard.

uuid_iso_iec_11578 shall have a value specified as a UUID according to the procedures of ISO/IEC 11578:1996 Annex A.

user_data_payload_byte shall be a byte containing data having syntax and semantics as specified by the UUID generator.

D.2.7 Recovery point SEI message semantics

The recovery point SEI message assists a decoder in determining when the decoding process will produce acceptable pictures for display after the decoder initiates random access or after the encoder indicates a broken link in the sequence. When the decoding process is started with the access unit in decoding order associated with the recovery point SEI message, all decoded pictures at or subsequent to the recovery point in output order specified in this SEI message are indicated to be correct or approximately correct in content. Decoded pictures produced by random access at or before the picture associated with the recovery point SEI message need not be correct in content until the indicated recovery point, and the operation of the decoding process starting at the picture associated with the recovery point SEI message may contain references to pictures not available in the decoded picture buffer.

In addition, by use of the broken_link_flag, the recovery point SEI message can indicate to the decoder the location of some pictures in the bitstream that can result in serious visual artefacts when displayed, even when the decoding process was begun at the location of a previous IDR access unit in decoding order.

NOTE 1 – The broken_link_flag can be used by encoders to indicate the location of a point after which the decoding process for the decoding of some pictures may cause references to pictures that, though available for use in the decoding process, are not the pictures that were used for reference when the bitstream was originally encoded (e.g., due to a splicing operation performed during the generation of the bitstream).

The recovery point is specified as a count in units of frame_num increments subsequent to the frame_num of the current access unit at the position of the SEI message.

NOTE 2 – When HRD information is present in the bitstream, a buffering period SEI message should be associated with the access unit associated with the recovery point SEI message in order to establish initialisation of the HRD buffer model after a random access.

recovery_frame_cnt specifies the recovery point of output pictures in output order. All decoded pictures in output order are indicated to be correct or approximately correct in content starting at the output order position of the reference picture having the frame_num equal to the frame_num of the VCL NAL units for the current access unit incremented by recovery_frame_cnt in modulo MaxFrameNum arithmetic. recovery_frame_cnt shall be in the range of 0 to MaxFrameNum – 1, inclusive.

exact_match_flag indicates whether decoded pictures at and subsequent to the specified recovery point in output order derived by starting the decoding process at the access unit associated with the recovery point SEI message shall be an

exact match to the pictures that would be produced by starting the decoding process at the location of a previous IDR access unit in the NAL unit stream. The value 0 indicates that the match need not be exact and the value 1 indicates that the match shall be exact.

When decoding starts from the location of the recovery point SEI message, all references to not available reference pictures shall be inferred as references to pictures containing only macroblocks coded using Intra macroblock prediction modes and having sample values given by Y samples equal to 128, Cb samples equal to 128, and Cr samples equal to 128 (mid-level grey) for purposes of determining the conformance of the value of exact_match_flag.

NOTE 3 – When performing random access, decoders should infer all references to not available reference pictures as references to pictures containing only intra macroblocks and having sample values given by Y equal to 128, Cb equal to 128, and Cr equal to 128 (mid-level grey), regardless of the value of exact_match_flag.

When exact_match_flag is equal to 0, the quality of the approximation at the recovery point is chosen by the encoding process and is not specified by this Recommendation | International Standard.

broken_link_flag indicates the presence or absence of a broken link in the NAL unit stream at the location of the recovery point SEI message and is assigned further semantics as follows.

- If broken_link_flag is equal to 1, pictures produced by starting the decoding process at the location of a previous IDR access unit may contain undesirable visual artefacts to the extent that decoded pictures at and subsequent to the access unit associated with the recovery point SEI message in decoding order should not be displayed until the specified recovery point in output order.
- Otherwise (broken_link_flag is equal to 0), no indication is given regarding any potential presence of visual artefacts.

Regardless of the value of the broken_link_flag, pictures subsequent to the specified recovery point in output order are specified to be correct or approximately correct in content.

NOTE 4 – When a sub-sequence information SEI message is present in conjunction with a recovery point SEI message in which broken_link_flag is equal to 1 and when sub_seq_layer_num is equal to 0, sub_seq_id should be different from the latest sub_seq_id for sub_seq_layer_num equal to 0 that was decoded prior to the location of the recovery point SEI message. When broken_link_flag is equal to 0, the sub_seq_id in sub-sequence layer 0 should remain unchanged.

changing_slice_group_idc equal to 0 indicates that decoded pictures are correct or approximately correct in content at and subsequent to the recovery point in output order when all macroblocks of the primary coded pictures are decoded within the changing slice group period, i.e., the period between the access unit associated with the recovery point SEI message (inclusive) and the specified recovery point (inclusive) in decoding order. changing_slice_group_idc shall be equal to 0 when num_slice_groups_minus1 is equal to 0 in any primary coded picture within the changing slice group period.

When changing_slice_group_idc is equal to 1 or 2, num_slice_groups_minus1 shall be equal to 1 and the macroblock-to-slice-group map type 3, 4, or 5 shall be applied in each primary coded picture in the changing slice group period.

changing_slice_group_idc equal to 1 indicates that within the changing slice group period no sample values outside the decoded macroblocks covered by slice group 0 are used for inter prediction of any macroblock within slice group 0. In addition, changing_slice_group_idc equal to 1 indicates that when all macroblocks in slice group 0 within the changing slice group period are decoded, decoded pictures will be correct or approximately correct in content at and subsequent to the specified recovery point in output order regardless of whether any macroblock in slice group 1 within the changing slice group period is decoded.

changing_slice_group_idc equal to 2 indicates that within the changing slice group period no sample values outside the decoded macroblocks covered by slice group 1 are used for inter prediction of any macroblock within slice group 1. In addition, changing_slice_group_idc equal to 2 indicates that when all macroblocks in slice group 1 within the changing slice group period are decoded, decoded pictures will be correct or approximately correct in content at and subsequent to the specified recovery point in output order regardless of whether any macroblock in slice group 0 within the changing slice group period is decoded.

changing_slice_group_idc shall be in the range of 0 to 2, inclusive.

D.2.8 Decoded reference picture marking repetition SEI message semantics

The decoded reference picture marking repetition SEI message is used to repeat the decoded reference picture marking syntax structure that was located in the slice header of an earlier picture in the sequence in decoding order.

original_idr_flag shall be equal to 1 when the decoded reference picture marking syntax structure occurred originally in an IDR picture. original_idr_flag shall be equal to 0 when the repeated decoded reference picture marking syntax structure did not occur in an IDR picture originally.

original_frame_num shall be equal to the frame_num of the picture where the repeated decoded reference picture marking syntax structure originally occurred. The picture indicated by original_frame_num is the previous coded picture having the specified value of frame_num. The value of original_frame_num used to refer to a picture having a memory_management_control_operation equal to 5 shall be 0.

original_field_pic_flag shall be equal to the field_pic_flag of the picture where the repeated decoded reference picture marking syntax structure originally occurred.

original_bottom_field_flag shall be equal to the bottom_field_flag of the picture where the repeated decoded reference picture marking syntax structure originally occurred.

dec_ref_pic_marking( ) shall contain a copy of the decoded reference picture marking syntax structure of the picture whose frame_num was original_frame_num. The nal_unit_type used for specification of the repeated dec_ref_pic_marking( ) syntax structure shall be the nal_unit_type of the slice header(s) of the picture whose frame_num was original_frame_num (i.e., nal_unit_type as used in subclause 7.3.3.3 shall be considered equal to 5 when original_idr_flag is equal to 1 and shall not be considered equal to 5 when original_idr_flag is equal to 0).

D.2.9 Spare picture SEI message semantics

This SEI message indicates that certain slice group map units, called spare slice group map units, in one or more decoded reference pictures resemble the co-located slice group map units in a specified decoded picture called the target picture. A spare slice group map unit may be used to replace a co-located, incorrectly decoded slice group map unit, in the target picture. A decoded picture containing spare slice group map units is called a spare picture.

For all spare pictures identified in a spare picture SEI message, the value of frame_mbs_only_flag shall be equal to the value of frame_mbs_only_flag of the target picture in the same SEI message. The spare pictures in the SEI message are constrained as follows.

- If the target picture is a decoded field, all spare pictures identified in the same SEI message shall be decoded fields.
- Otherwise (the target picture is a decoded frame), all spare pictures identified in the same SEI message shall be decoded frames.

For all spare pictures identified in a spare picture SEI message, the values of pic_width_in_mbs_minus1 and pic_height_in_map_units_minus1 shall be equal to the values of pic_width_in_mbs_minus1 and pic_height_in_map_units_minus1, respectively, of the target picture in the same SEI message. The picture associated (as specified in subclause 7.4.1.2.3) with this message shall appear after the target picture, in decoding order.

target_frame_num indicates the frame_num of the target picture.

spare_field_flag equal to 0 indicates that the target picture and the spare pictures are decoded frames. spare_field_flag equal to 1 indicates that the target picture and the spare pictures are decoded fields.

target_bottom_field_flag equal to 0 indicates that the target picture is a top field. target_bottom_field_flag equal to 1 indicates that the target picture is a bottom field.

A target picture is a decoded reference picture whose corresponding primary coded picture precedes the current picture, in decoding order, and in which the values of frame_num, field_pic_flag (when present) and bottom_field_flag (when present) are equal to target_frame_num, spare_field_flag and target_bottom_field_flag, respectively.

num_spare_pics_minus1 indicates the number of spare pictures for the specified target picture. The number of spare pictures is equal to num_spare_pics_minus1 + 1. The value of num_spare_pics_minus1 shall be in the range of 0 to 15, inclusive.

delta_spare_frame_num[i **]** is used to identify the spare picture that contains the i-th set of spare slice group map units, hereafter called the i-th spare picture, as specified below. The value of delta_spare_frame_num[i] shall be in the range of 0 to MaxFrameNum - 1 - !spare_field_flag, inclusive.

The frame_num of the i-th spare picture, spareFrameNum[i], is derived as follows for all values of i from 0 to num_spare_pics_minus1, inclusive:

```
candidateSpareFrameNum = target_frame_num - !spare_field_flag
for ( i = 0; i <= num_spare_pics_minus1; i++ ) {
    if( candidateSpareFrameNum < 0 )
        candidateSpareFrameNum = MaxFrameNum – 1
    spareFrameNum[ i ] = candidateSpareFrameNum – delta_spare_frame_num[ i ]          (D-3)
    if( spareFrameNum[ i ] < 0 )
        spareFrameNum[ i ] = MaxFrameNum + spareFrameNum[ i ]
```

```
    candidateSpareFrameNum = spareFrameNum[ i ] - !spare_field_flag
}
```

spare_bottom_field_flag[i **]** equal to 0 indicates that the i-th spare picture is a top field. spare_bottom_field_flag[i] equal to 1 indicates that the i-th spare picture is a bottom field.

The 0-th spare picture is a decoded reference picture whose corresponding primary coded picture precedes the target picture, in decoding order, and in which the values of frame_num, field_pic_flag (when present) and bottom_field_flag (when present) are equal to spareFrameNum[0], spare_field_flag and spare_bottom_field_flag[0], respectively. The i-th spare picture is a decoded reference picture whose corresponding primary coded picture precedes the (i - 1)-th spare picture, in decoding order, and in which the values of frame_num, field_pic_flag (when present) and bottom_field_flag (when present) are equal to spareFrameNum[i], spare_field_flag and spare_bottom_field_flag[i], respectively.

spare_area_idc[i **]** indicates the method used to identify the spare slice group map units in the i-th spare picture. spare_area_idc[i] shall be in the range of 0 to 2, inclusive. spare_area_idc[i] equal to 0 indicates that all slice group map units in the i-th spare picture are spare units. spare_area_idc[i] equal to 1 indicates that the value of the syntax element spare_unit_flag[i][j] is used to identify the spare slice group map units. spare_area_idc[i] equal to 2 indicates that the zero_run_length[i][j] syntax element is used to derive the values of spareUnitFlagInBoxOutOrder[i][j], as described below.

spare_unit_flag[i **][** j **]** equal to 0 indicates that the j-th slice group map unit in raster scan order in the i-th spare picture is a spare unit. spare_unit_flag[i][j] equal to 1 indicates that the j-th slice group map unit in raster scan order in the i-th spare picture is not a spare unit.

zero_run_length[i **][** j **]** is used to derive the values of spareUnitFlagInBoxOutOrder[i][j] when spare_area_idc[i] is equal to 2. In this case, the spare slice group map units identified in spareUnitFlagInBoxOutOrder[i][j] appear in counter-clockwise box-out order, as specified in subclause 8.2.2.4, for each spare picture. spareUnitFlagInBoxOutOrder[i][j] equal to 0 indicates that the j-th slice group map unit in counter-clockwise box-out order in the i-th spare picture is a spare unit. spareUnitFlagInBoxOutOrder[i][j] equal to 1 indicates that the j-th slice group map unit in counter-clockwise box-out order in the i-th spare picture is not a spare unit.

When spare_area_idc[0] is equal to 2, spareUnitFlagInBoxOutOrder[0][j] is derived as follows:

```
for( j = 0, loop = 0; j < PicSizeInMapUnits; loop++ ) {
   for( k = 0; k < zero_run_length[ 0 ][ loop ]; k++ )
      spareUnitFlagInBoxOutOrder[ 0 ][ j++ ] = 0                                   (D-4)
   spareUnitFlagInBoxOutOrder[ 0 ][ j++ ] = 1
}
```

When spare_area_idc[i] is equal to 2 and the value of i is greater than 0, spareUnitFlagInBoxOutOrder[i][j] is derived as follows:

```
for( j = 0, loop = 0; j < PicSizeInMapUnits; loop++ ) {
   for( k = 0; k < zero_run_length[ i ][ loop ]; k++ )
      spareUnitFlagInBoxOutOrder[ i ][ j ] = spareUnitFlagInBoxOutOrder[ i - 1 ][ j++ ]   (D-5)
   spareUnitFlagInBoxOutOrder[ i ][ j ] = !spareUnitFlagInBoxOutOrder[ i - 1 ][ j++ ]
}
```

D.2.10 Scene information SEI message semantics

A scene and a scene transition are herein defined as a set of consecutive pictures in output order.

NOTE 1 – Decoded pictures within one scene generally have similar content. The scene information SEI message is used to label pictures with scene identifiers and to indicate scene changes. The message specifies how the source pictures for the labelled pictures were created. The decoder may use the information to select an appropriate algorithm to conceal transmission errors. For example, a specific algorithm may be used to conceal transmission errors that occurred in pictures belonging to a gradual scene transition. Furthermore, the scene information SEI message may be used in a manner determined by the application, such as for indexing the scenes of a coded sequence.

A scene information SEI message labels all pictures, in decoding order, from the primary coded picture to which the SEI message is associated (inclusive), as specified in subclause 7.4.1.2.3, to the primary coded picture to which the next scene information SEI message (if present) in decoding order is associated (exclusive) or (otherwise) to the last access unit in the bitstream (inclusive). These pictures are herein referred to as the target pictures.

scene_info_present_flag equal to 0 indicates that the scene or scene transition to which the target pictures belong is unspecified. scene_info_present_flag equal to 1 indicates that the target pictures belong to the same scene or scene transition.

scene_id identifies the scene to which the target pictures belong. When the value of scene_transition_type of the target pictures is less than 4, and the previous picture in output order is marked with a value of scene_transition_type less than 4, and the value of scene_id is the same as the value of scene_id of the previous picture in output order, this indicates that the source scene for the target pictures and the source scene for the previous picture (in output order) are considered by the encoder to have been the same scene. When the value of scene_transition_type of the target pictures is greater than 3, and the previous picture in output order is marked with a value of scene_transition_type less than 4, and the value of scene_id is the same as the value of scene_id of the previous picture in output order, this indicates that one of the source scenes for the target pictures and the source scene for the previous picture (in output order) are considered by the encoder to have been the same scene. When the value of scene_id is not equal to the value of scene_id of the previous picture in output order, this indicates that the target pictures and the previous picture (in output order) are considered by the encoder to have been from different source scenes.

The value of scene_id shall be in the range of 0 to 2^{32}-1, inclusive. Values of scene_id in the range of 0 to 255, inclusive, and in the range of 512 to $2^{31} - 1$, inclusive, may be used as determined by the application. Values of scene_id in the range of 256 to 511, inclusive, and in the range of 2^{31} to $2^{32} - 1$, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of scene_id in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 1, inclusive, shall ignore (remove from the bitstream and discard) it.

scene_transition_type specifies in which type of a scene transition (if any) the target pictures are involved. The valid values of scene_transition_type are specified in Table D-4.

Table D-4 – scene_transition_type values

Value	Description
0	No transition
1	Fade to black
2	Fade from black
3	Unspecified transition from or to constant colour
4	Dissolve
5	Wipe
6	Unspecified mixture of two scenes

When scene_transition_type is greater than 3, the target pictures include contents both from the scene labelled by its scene_id and the next scene, in output order, which is labelled by second_scene_id (see below). The term "the current scene" is used to indicate the scene labelled by scene_id. The term "the next scene" is used to indicate the scene labelled by second_scene_id. It is not required for any following picture, in output order, to be labelled with scene_id equal to second_scene_id of the current SEI message.

Scene transition types are specified as follows.

"No transition" specifies that the target pictures are not involved in a gradual scene transition.

NOTE 2 – When two consecutive pictures in output order have scene_transition_type equal to 0 and different values of scene_id, a scene cut occurred between the two pictures.

"Fade to black" indicates that the target pictures are part of a sequence of pictures, in output order, involved in a fade to black scene transition, i.e., the luma samples of the scene gradually approach zero and the chroma samples of the scene gradually approach 128.

NOTE 3 – When two pictures are labelled to belong to the same scene transition and their scene_transition_type is "Fade to black", the later one, in output order, is darker than the previous one.

"Fade from black" indicates that the target pictures are part of a sequence of pictures, in output order, involved in a fade from black scene transition, i.e., the luma samples of the scene gradually diverge from zero and the chroma samples of the scene may gradually diverge from 128.

NOTE 4 – When two pictures are labelled to belong to the same scene transition and their scene_transition_type is "Fade from black", the later one in output order is lighter than the previous one.

"Dissolve" indicates that the sample values of each target picture (before encoding) were generated by calculating a sum of co-located weighted sample values of a picture from the current scene and a picture from the next scene. The

weight of the current scene gradually decreases from full level to zero level, whereas the weight of the next scene gradually increases from zero level to full level. When two pictures are labelled to belong to the same scene transition and their scene_transition_type is "Dissolve", the weight of the current scene for the later one, in output order, is less than the weight of the current scene for the previous one, and the weight of the next scene for the later one, in output order, is greater than the weight of the next scene for the previous one.

"Wipe" indicates that some of the sample values of each target picture (before encoding) were generated by copying co-located sample values of a picture in the current scene and the remaining sample values of each target picture (before encoding) were generated by copying co-located sample values of a picture in the next scene. When two pictures are labelled to belong to the same scene transition and their scene_transition_type is "Wipe", the number of samples copied from the next scene to the later picture in output order is greater than the number of samples copied from the next scene to the previous picture.

second_scene_id identifies the next scene in the gradual scene transition in which the target pictures are involved. The value of second_scene_id shall not be equal to the value of scene_id. The value of second_scene_id shall not be equal to the value of scene_id in the previous picture in output order. When the next picture in output order is marked with a value of scene_transition_type less than 4, and the value of second_scene_id is the same as the value of scene_id of the next picture in output order, this indicates that the encoder considers one of the source scenes for the target pictures and the source scene for the next picture (in output order) to have been the same scene. When the value of second_scene_id is not equal to the value of scene_id or second_scene_id (if present) of the next picture in output order, this indicates that the encoder considers the target pictures and the next picture (in output order) to have been from different source scenes.

When the value of scene_id of a picture is equal to the value of scene_id of the following picture in output order and the value of scene_transition_type in both of these pictures is less than 4, this indicates that the encoder considers the two pictures to have been from the same source scene. When the values of scene_id, scene_transition_type and second_scene_id (if present) of a picture are equal to the values of scene_id, scene_transition_type and second_scene_id (respectively) of the following picture in output order and the value of scene_transition_type is greater than 0, this indicates that the encoder considers the two pictures to have been from the same source gradual scene transition.

The value of second_scene_id shall be in the range of 0 to 2^{32}-1, inclusive. Values of second_scene_id in the range of 0 to 255, inclusive, and in the range of 512 to 2^{31}-1, inclusive, may be used as determined by the application. Values of second_scene_id in the range of 256 to 511, inclusive, and in the range of 2^{31} to 2^{32}-1, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of second_scene_id in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32}-1, inclusive, shall ignore (remove from the bitstream and discard) it.

D.2.11 Sub-sequence information SEI message semantics

The sub-sequence information SEI message is used to indicate the position of a picture in data dependency hierarchy that consists of sub-sequence layers and sub-sequences.

A sub-sequence layer contains a subset of the coded pictures in a sequence. Sub-sequence layers are numbered with non-negative integers. A layer having a larger layer number is a higher layer than a layer having a smaller layer number. The layers are ordered hierarchically based on their dependency on each other so that any picture in a layer shall not be predicted from any picture on any higher layer.

NOTE 1 – In other words, any picture in layer 0 must not be predicted from any picture in layer 1 or above, pictures in layer 1 may be predicted from layer 0, pictures in layer 2 may be predicted from layers 0 and 1, etc.

NOTE 2 – The subjective quality is expected to increase along with the number of decoded layers.

A sub-sequence is a set of coded pictures within a sub-sequence layer. A picture shall reside in one sub-sequence layer and in one sub-sequence only. Any picture in a sub-sequence shall not be predicted from any picture in another sub-sequence in the same or in a higher sub-sequence layer. A sub-sequence in layer 0 can be decoded independently of any picture that does not belong to the sub-sequence.

The sub-sequence information SEI message concerns the current access unit. The primary coded picture in the access unit is herein referred to as the current picture.

The sub-sequence information SEI message shall not be present unless gaps_in_frame_num_value_allowed_flag in the sequence parameter set referenced by the picture associated with the sub-sequence SEI message is equal to 1.

sub_seq_layer_num specifies the sub-sequence layer number of the current picture. When sub_seq_layer_num is greater than 0, memory management control operations shall not be used in any slice header of the current picture. When the current picture resides in a sub-sequence whose first picture in decoding order is an IDR picture, the value of sub_seq_layer_num shall be equal to 0. For a non-paired reference field, the value of sub_seq_layer_num shall be equal to 0. sub_seq_layer_num shall be in the range of 0 to 255, inclusive.

sub_seq_id identifies the sub-sequence within a layer. When the current picture resides in a sub-sequence whose first picture in decoding order is an IDR picture, the value of sub_seq_id shall be the same as the value of idr_pic_id of the IDR picture. sub_seq_id shall be in the range of 0 to 65535, inclusive.

first_ref_pic_flag equal to 1 specifies that the current picture is the first reference picture of the sub-sequence in decoding order. When the current picture is not the first picture of the sub-sequence in decoding order, the first_ref_pic_flag shall be equal to 0.

leading_non_ref_pic_flag equal to 1 specifies that the current picture is a non-reference picture preceding any reference picture in decoding order within the sub-sequence or that the sub-sequence contains no reference pictures. When the current picture is a reference picture or the current picture is a non-reference picture succeeding at least one reference picture in decoding order within the sub-sequence, the leading_non_ref_pic_flag shall be equal to 0.

last_pic_flag equal to 1 indicates that the current picture is the last picture of the sub-sequence (in decoding order), including all reference and non-reference pictures of the sub-sequence. When the current picture is not the last picture of the sub-sequence (in decoding order), last_pic_flag shall be equal to 0.

The current picture is assigned to a sub-sequence as follows.

- If one or more of the following conditions is true, the current picture is the first picture of a sub-sequence in decoding order.
    - no earlier picture in decoding order is labelled with the same values of sub_seq_id and sub_seq_layer_num as the current picture
    - the value of leading_non_ref_pic_flag is equal to 1 and the value of leading_non_ref_pic_flag is equal to 0 in the previous picture in decoding order having the same values of sub_seq_id and sub_seq_layer_num as the current picture
    - the value of first_ref_pic_flag is equal to 1 and the value of leading_non_ref_pic_flag is equal to 0 in the previous picture in decoding order having the same values of sub_seq_id and sub_seq_layer_num as the current picture
    - the value of last_pic_flag is equal to 1 in the previous picture in decoding order having the same values of sub_seq_id and sub_seq_layer_num as the current picture
- Otherwise, the current picture belongs to the same sub-sequence as the previous picture in decoding order having the same values of sub_seq_id and sub_seq_layer_num as the current picture.

sub_seq_frame_num_flag equal to 0 specifies that sub_seq_frame_num is not present. sub_seq_frame_num_flag equal to 1 specifies that sub_seq_frame_num is present.

sub_seq_frame_num shall be equal to 0 for the first reference picture of the sub-sequence and for any non-reference picture preceding the first reference picture of the sub-sequence in decoding order. sub_seq_frame_num is further constrained as follows.

- If the current picture is not the second field of a complementary field pair, sub_seq_frame_num shall be incremented by 1, in modulo MaxFrameNum operation, relative to the previous reference picture, in decoding order, that belongs to the sub-sequence.
- Otherwise (the current picture is the second field of a complementary field pair), the value of sub_seq_frame_num shall be the same as the value of sub_seq_frame_num for the first field of the complementary field pair.

sub_seq_frame_num shall be in the range of 0 to MaxFrameNum – 1, inclusive.

When the current picture is an IDR picture, it shall start a new sub-sequence in sub-sequence layer 0. Thus, the sub_seq_layer_num shall be 0, the sub_seq_id shall be different from the previous sub-sequence in sub-sequence layer 0, first_ref_pic_flag shall be 1, and leading_non_ref_pic_flag shall be equal to 0.

When the sub-sequence information SEI message is present for both coded fields of a complementary field pair, the values of sub_seq_layer_num, sub_seq_id, leading_non_ref_pic_flag and sub_seq_frame_num, when present, shall be identical for both of these pictures.

When the sub-sequence information SEI message is present only for one coded field of a complementary field pair, the values of sub_seq_layer_num, sub_seq_id, leading_non_ref_pic_flag and sub_seq_frame_num, when present, are also applicable to the other coded field of the complementary field pair.

D.2.12 Sub-sequence layer characteristics SEI message semantics

The sub-sequence layer characteristics SEI message specifies the characteristics of sub-sequence layers.

num_sub_seq_layers_minus1 plus 1 specifies the number of sub-sequence layers in the sequence. num_sub_seq_layers_minus1 shall be in the range of 0 to 255, inclusive.

A pair of average_bit_rate and average_frame_rate characterizes each sub-sequence layer. The first pair of average_bit_rate and average_frame_rate specifies the characteristics of sub-sequence layer 0. When present, the second pair specifies the characteristics of sub-sequence layers 0 and 1 jointly. Each pair in decoding order specifies the characteristics for a range of sub-sequence layers from layer number 0 to the layer number specified by the layer loop counter. The values are in effect from the point they are decoded until an update of the values is decoded.

accurate_statistics_flag equal to 1 indicates that the values of average_bit_rate and average_frame_rate are rounded from statistically correct values. accurate_statistics_flag equal to 0 indicates that the average_bit_rate and the average_frame_rate are estimates and may deviate somewhat from the correct values.

When accurate_statistics_flag is equal to 0, the quality of the approximation used in the computation of the values of average_bit_rate and the average_frame_rate is chosen by the encoding process and is not specified by this Recommendation | International Standard.

average_bit_rate indicates the average bit rate in units of 1000 bits per second. All NAL units in the range of sub-sequence layers specified above are taken into account in the calculation. The average bit rate is derived according to the access unit removal time specified in Annex C of the Recommendation | International Standard. In the following, bTotal is the number of bits in all NAL units succeeding a sub-sequence layer characteristics SEI message (including the bits of the NAL units of the current access unit) and preceding the next access unit (in decoding order) including a sub-sequence layer characteristics SEI message (if present) or the end of the stream (otherwise). t_1 is the removal time (in seconds) of the current access unit, and t_2 is the removal time (in seconds) of the latest access unit (in decoding order) before the next sub-sequence layer characteristics SEI message (if present) or the end of the stream (otherwise).

When accurate_statistics_flag is equal to 1, the following conditions shall be fulfilled as follows.

– If t_1 is not equal to t_2, the following condition shall be true

$$\text{average_bit_rate} == \text{Round}(\text{bTotal} \div ((t_2 - t_1) * 1000))) \quad \text{(D-6)}$$

– Otherwise (t_1 is equal to t_2), the following condition shall be true

$$\text{average_bit_rate} == 0 \quad \text{(D-7)}$$

average_frame_rate indicates the average frame rate in units of frames/(256 seconds). All NAL units in the range of sub-sequence layers specified above are taken into account in the calculation. In the following, fTotal is the number of frames, complementary field pairs and non-paired fields between the current picture (inclusive) and the next sub-sequence layer characteristics SEI message (if present) or the end of the stream (otherwise). t_1 is the removal time (in seconds) of the current access unit, and t_2 is the removal time (in seconds) of the latest access unit (in decoding order) before the next sub-sequence layer characteristics SEI message (if present) or the end of the stream (otherwise).

When accurate_statistics_flag is equal to 1, the following conditions shall be fulfilled as follows.

– If t_1 is not equal to t_2, the following condition shall be true

$$\text{average_frame_rate} == \text{Round}(\text{fTotal} * 256 \div (t_2 - t_1)) \quad \text{(D-8)}$$

– Otherwise (t_1 is equal to t_2), the following condition shall be true

$$\text{average_frame_rate} == 0 \quad \text{(D-9)}$$

D.2.13 Sub-sequence characteristics SEI message semantics

The sub-sequence characteristics SEI message indicates the characteristics of a sub-sequence. It also indicates inter prediction dependencies between sub-sequences. This message shall be contained in the first access unit in decoding order of the sub-sequence to which the sub-sequence characteristics SEI message applies. This sub-sequence is herein called the target sub-sequence.

sub_seq_layer_num identifies the sub-sequence layer number of the target sub-sequence. sub_seq_layer_num shall be in the range of 0 to 255, inclusive.

sub_seq_id identifies the target sub-sequence. sub_seq_id shall be in the range of 0 to 65535, inclusive.

duration_flag equal to 0 indicates that the duration of the target sub-sequence is not specified.

sub_seq_duration specifies the duration of the target sub-sequence in clock ticks of a 90-kHz clock.

average_rate_flag equal to 0 indicates that the average bit rate and the average frame rate of the target sub-sequence are unspecified.

accurate_statistics_flag indicates how reliable the values of average_bit_rate and average_frame_rate are. accurate_statistics_flag equal to 1, indicates that the average_bit_rate and the average_frame_rate are rounded from statistically correct values. accurate_statistics_flag equal to 0 indicates that the average_bit_rate and the average_frame_rate are estimates and may deviate from the statistically correct values.

average_bit_rate indicates the average bit rate in (1000 bits)/second of the target sub-sequence. All NAL units of the target sub-sequence are taken into account in the calculation. The average bit rate is derived according to the access unit removal time specified in subclause C.1.2. In the following, nB is the number of bits in all NAL units in the sub-sequence. t_1 is the removal time (in seconds) of the first access unit of the sub-sequence (in decoding order), and t_2 is the removal time (in seconds) of the last access unit of the sub-sequence (in decoding order).

When accurate_statistics_flag is equal to 1, the following conditions shall be fulfilled as follows.

– If t_1 is not equal to t_2, the following condition shall be true

$$\text{average_bit_rate} == \text{Round}(nB \div ((t_2 - t_1) * 1000)) \quad \text{(D-10)}$$

– Otherwise (t_1 is equal to t_2), the following condition shall be true

$$\text{average_bit_rate} == 0 \quad \text{(D-11)}$$

average_frame_rate indicates the average frame rate in units of frames/(256 seconds) of the target sub-sequence. All NAL units of the target sub-sequence are taken into account in the calculation. The average frame rate is derived according to the access unit removal time specified in subclause C.1.2. In the following, fC is the number of frames, complementary field pairs and non-paired fields in the sub-sequence. t_1 is the removal time (in seconds) of the first access unit of the sub-sequence (in decoding order), and t_2 is the removal time (in seconds) of the last access unit of the sub-sequence (in decoding order).

When accurate_statistics_flag is equal to 1, the following conditions shall be fulfilled as follows.

– If t_1 is not equal to t_2, the following condition shall be true

$$\text{average_frame_rate} == \text{Round}(fC * 256 \div (t_2 - t_1)) \quad \text{(D-12)}$$

– Otherwise (t_1 is equal to t_2), the following condition shall be true

$$\text{average_frame_rate} == 0 \quad \text{(D-13)}$$

num_referenced_subseqs specifies the number of sub-sequences that contain pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence. num_referenced_subseqs shall be in the range of 0 to 255, inclusive.

ref_sub_seq_layer_num, ref_sub_seq_id, and **ref_sub_seq_direction** identify the sub-sequence that contains pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence. Depending on ref_sub_seq_direction, the following applies.

– If ref_sub_seq_direction is equal to 0, a set of candidate sub-sequences consists of the sub-sequences whose sub_seq_id is equal to ref_sub_seq_id, which reside in the sub-sequence layer having sub_seq_layer_num equal to ref_sub_seq_layer_num, and whose first picture in decoding order precedes the first picture of the target sub-sequence in decoding order.

– Otherwise (ref_sub_seq_direction is equal to 1), a set of candidate sub-sequences consists of the sub-sequences whose sub_seq_id is equal to ref_sub_seq_id, which reside in the sub-sequence layer having sub_seq_layer_num equal to ref_sub_seq_layer_num, and whose first picture in decoding order succeeds the first picture of the target sub-sequence in decoding order.

The sub-sequence used as a reference for the target sub-sequence is the sub-sequence among the set of candidate sub-sequences whose first picture is the closest to the first picture of the target sub-sequence in decoding order.

D.2.14 Full-frame freeze SEI message semantics

The full-frame freeze SEI message indicates that the current picture and any subsequent pictures in output order that meet specified conditions should not affect the content of the display. No more than one full-frame freeze SEI message shall be present in any access unit.

full_frame_freeze_repetition_period specifies the persistence of the full-frame freeze SEI message and may specify a picture order count interval within which another full-frame freeze SEI message or a full-frame freeze release SEI or the end of the coded video sequence shall be present in the bitstream. The value of full_frame_freeze_repetition_period shall be in the range of 0 to 16 384, inclusive.

full_frame_freeze_repetition_period equal to 0 specifies that the full-frame freeze SEI message applies to the current decoded picture only.

full_frame_freeze_repetition_period equal to 1 specifies that the full-frame freeze SEI message persists in output order until any of the following conditions are true.

- A new coded video sequence begins
- A picture in an access unit containing a full-frame freeze SEI message or a full-frame freeze release SEI message is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

full_frame_freeze_repetition_period greater than 1 specifies that the full-frame freeze SEI message persists until any of the following conditions are true.

- A new coded video sequence begins
- A picture in an access unit containing a full-frame freeze SEI message or a full-frame freeze release SEI message is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + full_frame_freeze_repetition_period.

full_frame_freeze_repetition_period greater than 1 indicates that another full-frame freeze SEI message or a full-frame freeze release SEI message shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + full_frame_freeze_repetition_period; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

D.2.15 Full-frame freeze release SEI message semantics

The full-frame freeze release SEI message cancels the effect of any full-frame freeze SEI message sent with pictures that precede the current picture in output order. The full-frame freeze release SEI message indicates that the current picture and subsequent pictures in output order should affect the contents of the display.

No more than one full-frame freeze release SEI message shall be present in any access unit. A full-frame freeze release SEI message shall not be present in an access unit containing a full-frame freeze SEI message. When a full-frame freeze SEI message is present in an access unit containing a field of a complementary field pair in which the values of PicOrderCnt(CurrPic) for the two fields of the complementary field pair are equal to each other, a full-frame freeze release SEI message shall not be present in either of the two access units.

D.2.16 Full-frame snapshot SEI message semantics

The full-frame snapshot SEI message indicates that the current frame is labelled for use as determined by the application as a still-image snapshot of the video content.

snapshot_id specifies a snapshot identification number. snapshot_id shall be in the range of 0 to 2^{32} - 1, inclusive.

Values of snapshot_id in the range of 0 to 255, inclusive, and in the range of 512 to 2^{31}-1, inclusive, may be used as determined by the application. Values of snapshot_id in the range of 256 to 511, inclusive, and in the range of 2^{31} to 2^{32}-1, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of snapshot_id in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32}-1, inclusive, shall ignore (remove from the bitstream and discard) it.

D.2.17 Progressive refinement segment start SEI message semantics

The progressive refinement segment start SEI message specifies the beginning of a set of consecutive coded pictures that is labelled as the current picture followed by a sequence of one or more pictures of refinement of the quality of the current picture, rather than as a representation of a continually moving scene.

The tagged set of consecutive coded pictures shall continue until one of the following conditions is true. When a condition below becomes true, the next slice to be decoded does not belong to the tagged set of consecutive coded pictures.

1. The next slice to be decoded belongs to an IDR picture.

2. num_refinement_steps_minus1 is greater than 0 and the frame_num of the next slice to be decoded is (currFrameNum + num_refinement_steps_minus1 + 1) % MaxFrameNum, where currFrameNum is the value of frame_num of the picture in the access unit containing the SEI message.

3. num_refinement_steps_minus1 is 0 and a progressive refinement segment end SEI message with the same progressive_refinement_id as the one in this SEI message is decoded.

The decoding order of picture within the tagged set of consecutive pictures should be the same as their output order. **progressive_refinement_id** specifies an identification number for the progressive refinement operation. progressive_refinement_id shall be in the range of 0 to 2^{32} - 1, inclusive.

Values of progressive_refinement_id in the range of 0 to 255, inclusive, and in the range of 512 to 2^{31} - 1, inclusive, may be used as determined by the application. Values of progressive_refinement_id in the range of 256 to 511, inclusive, and in the range of 2^{31} to 2^{32}-1, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of progressive_refinement_id in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 1, inclusive, shall ignore (remove from the bitstream and discard) it.

num_refinement_steps_minus1 specifies the number of reference frames in the tagged set of consecutive coded pictures as follows.

– If num_refinement_steps_minus1 is equal to 0, the number of reference frames in the tagged set of consecutive coded pictures is unknown.

– Otherwise, the number of reference frames in the tagged set of consecutive coded pictures is equal to num_refinement_steps_minus1 + 1.

num_refinement_steps_minus1 shall be in the range of 0 to MaxFrameNum - 1, inclusive.

D.2.18 Progressive refinement segment end SEI message semantics

The progressive refinement segment end SEI message specifies the end of a set of consecutive coded pictures that has been labelled by use of a progressive refinement segment start SEI message as an initial picture followed by a sequence of one or more pictures of the refinement of the quality of the initial picture, and ending with the current picture.

progressive_refinement_id specifies an identification number for the progressive refinement operation. progressive_refinement_id shall be in the range of 0 to 2^{32} - 1, inclusive.

The progressive refinement segment end SEI message specifies the end of any progressive refinement segment previously started using a progressive refinement segment start SEI message with the same value of progressive_refinement_id.

Values of progressive_refinement_id in the range of 0 to 255, inclusive, and in the range of 512 to 2^{31} - 1, inclusive, may be used as determined by the application. Values of progressive_refinement_id in the range of 256 to 511, inclusive, and in the range of 2^{31} to 2^{32} - 1, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders encountering a value of progressive_refinement_id in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 1, inclusive, shall ignore (remove from the bitstream and discard) it.

D.2.19 Motion-constrained slice group set SEI message semantics

This SEI message indicates that inter prediction over slice group boundaries is constrained as specified below. When present, the message shall only appear where it is associated, as specified in subclause 7.4.1.2.3, with an IDR access unit.

The target picture set for this SEI message contains all consecutive primary coded pictures in decoding order starting with the associated primary coded IDR picture (inclusive) and ending with the following primary coded IDR picture (exclusive) or with the very last primary coded picture in the bitstream (inclusive) in decoding order when there is no following primary coded IDR picture. The slice group set is a collection of one or more slice groups, identified by the slice_group_id[i] syntax element.

This SEI message indicates that, for each picture in the target picture set, the inter prediction process is constrained as follows: No sample value outside the slice group set, and no sample value at a fractional sample position that is derived using one or more sample values outside the slice group set is used to inter predict any sample within the slice group set.

num_slice_groups_in_set_minus1 + 1 specifies the number of slice groups in the slice group set. The allowed range of num_slice_groups_in_set_minus1 is 0 to num_slice_groups_minus1, inclusive. The allowed range of num_slice_groups_minus1 is specified in Annex A.

slice_group_id[i] identifies the slice group(s) contained within the slice group set. The allowed range is from 0 to num_slice_groups_in_set_minus1, inclusive. The size of the slice_group_id[i] syntax element is Ceil(Log2(num_slice_groups_minus1 + 1)) bits.

exact_sample_value_match_flag equal to 0 indicates that, within the target picture set, when the macroblocks that do not belong to the slice group set are not decoded, the value of each sample in the slice group set need not be exactly the same as the value of the same sample when all the macroblocks are decoded. exact_sample_value_match_flag equal to 1 indicates that, within the target picture set, when the macroblocks that do not belong to the slice group set are not decoded, the value of each sample in the slice group set shall be exactly the same as the value of the same sample when all the macroblocks in the target picture set are decoded.

NOTE 1 – When disable_deblocking_filter_idc is equal to 2 in all slices in the target picture set, exact_sample_value_match_flag should be 1.

pan_scan_rect_flag equal to 0 specifies that pan_scan_rect_id is not present. pan_scan_rect_flag equal to 1 specifies that pan_scan_rect_id is present.

pan_scan_rect_id indicates that the specified slice group set covers at least the pan-scan rectangle identified by pan_scan_rect_id within the target picture set.

NOTE 2 – Multiple motion_constrained_slice_group_set SEI messages may be associated with the same IDR picture. Consequently, more than one slice group set may be active within a target picture set.

NOTE 3 – The size, shape, and location of the slice groups in the slice group set may change within the target picture set.

D.2.20 Film grain characteristics SEI message semantics

This SEI message provides the decoder with a parameterised model for film grain synthesis. For example, an encoder may use the film grain characteristics SEI message to characterise film grain that was present in the original source video material and was removed by pre-processing filtering techniques. Synthesis of simulated film grain on the decoded images for the display process is optional and does not affect the decoding process specified in this Recommendation | International Standard. If synthesis of simulated film grain on the decoded images for the display process is performed, there is no requirement that the method by which the synthesis is performed be the same as the parameterised model for the film grain as provided in the film grain characteristics SEI message.

NOTE 1 – The display process is not specified in this Recommendation | International Standard.

film_grain_characteristics_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous film grain characteristics SEI message in output order. film_grain_characteristics_cancel_flag equal to 0 indicates that film grain modelling information follows.

model_id identifies the film grain simulation model as specified in Table D-5. The value of model_id shall be in the range of 0 to 1, inclusive.

Table D-5 – model_id values

Value	Description
0	frequency filtering
1	auto-regression
2	reserved
3	reserved

separate_colour_description_present_flag equal to 1 indicates that a distinct colour space description for the film grain characteristics specified in the SEI message is present in the film grain characteristics SEI message syntax. separate_colour_description_present_flag equal to 0 indicates that the colour description for the film grain characteristics specified in the SEI message is the same as for the coded video sequence as specified in subclause E.2.1.

NOTE 2 – When separate_colour_description_present_flag is equal to 1, the colour space specified for the film grain characteristics specified in the SEI message may differ from the colour space specified for the coded video as specified in subclause E.2.1.

film_grain_bit_depth_luma_minus8 plus 8 specifies the bit depth used for the luma component of the film grain characteristics specified in the SEI message. When film_grain_bit_depth_luma_minus8 is not present in the film grain characteristics SEI message, the value of film_grain_bit_depth_luma_minus8 shall be inferred to be equal to bit_depth_luma_minus8.

The value of filmGrainBitDepth[0] is derived as

$$filmGrainBitDepth[\ 0\ ] = film_grain_bit_depth_luma_minus8 + 8 \quad \text{(D-14)}$$

film_grain_bit_depth_chroma_minus8 plus 8 specifies the bit depth used for the Cb and Cr components of the film grain characteristics specified in the SEI message. When film_grain_bit_depth_chroma_minus8 is not present in the film grain characteristics SEI message, the value of film_grain_bit_depth_chroma_minus8 shall be inferred to be equal to bit_depth_chroma_minus8.

The value of filmGrainBitDepth[c] for c = 1 and 2 is derived as

$$filmGrainBitDepth[\ c\ ] = film_grain_bit_depth_chroma_minus8 + 8 \quad \text{with } c = 1, 2 \tag{D-15}$$

film_grain_full_range_flag has the same semantics as specified in subclause E.2.1 for the video_full_range_flag syntax element, except as follows.

– film_grain_full_range_flag specifies the colour space of the film grain characteristics specified in the SEI message, rather than the colour space used for the coded video sequence.

– When film_grain_full_range_flag is not present in the film grain characteristics SEI message, the value of film_grain_full_range_flag shall be inferred to be equal to video_full_range_flag.

film_grain_colour_primaries has the same semantics as specified in subclause E.2.1 for the colour_primaries syntax element, except as follows.

– film_grain_colour_primaries specifies the colour space of the film grain characteristics specified in the SEI message, rather than the colour space used for the coded video sequence.

– When film_grain_colour_primaries is not present in the film grain characteristics SEI message, the value of film_grain_colour_primaries shall be inferred to be equal to colour_primaries.

film_grain_transfer_characteristics has the same semantics as specified in subclause E.2.1 for the transfer_characteristics syntax element, except as follows.

– film_grain_transfer_characteristics specifies the colour space of the film grain characteristics specified in the SEI message, rather than the colour space used for the coded video sequence.

– When film_grain_transfer_characteristics is not present in the film grain characteristics SEI message, the value of film_grain_transfer_characteristics shall be inferred to be equal to transfer_characteristics.

film_grain_matrix_coefficients has the same semantics as specified in subclause E.2.1 for the matrix_coefficients syntax element, except as follows.

– film_grain_matrix_coefficients specifies the colour space of the film grain characteristics specified in the SEI message, rather than the colour space used for the coded video sequence.

– When film_grain_matrix_coefficients is not present in the film grain characteristics SEI message, the value of film_grain_matrix_coefficients shall be inferred to be equal to matrix_coefficients.

– The values allowed for film_grain_matrix_coefficients are not constrained by the value of chroma_format_idc.

The chroma_format_idc of the film grain characteristics specified in the film grain characteristics SEI message shall be inferred to be equal to 3 (4:4:4).

NOTE 3 – Because the use of a specific method is not required for performing film grain generation function used by the display process, a decoder may, if desired, down-convert the model information for chroma in order to simulate film grain for other chroma formats (4:2:0 or 4:2:2) rather than up-converting the decoded video (using a method not specified by this Recommendation | International Standard) before performing film grain generation.

blending_mode_id identifies the blending mode used to blend the simulated film grain with the decoded images as specified in Table D-6. blending_mode_id shall be in the range of 0 to 1, inclusive.

Table D-6 – blending_mode_id values

Value	Description
0	additive
1	multiplicative
2	reserved
3	reserved

Depending on blending_mode_id, the blending mode is specified as follows

– If blending_mode_id is equal to 0 the blending mode is additive as specified by

$$I_{grain}[x, y, c] = Clip3(0, (1 << filmGrainBitDepth[c]) - 1, I_{decoded}[x, y, c] + G[x, y, c]) \quad \text{(D-16)}$$

– Otherwise (blending_mode_id is equal to 1), the blending mode is multiplicative as specified by

$$I_{grain}[x, y, c] = Clip3(0, (1 << filmGrainBitDepth[c]) - 1, I_{decoded}[x, y, c] * (1 + G[x, y, c])) \quad \text{(D-17)}$$

where $I_{decoded}[x, y, c]$ represents the sample value at coordinates x, y of the colour component c of the decoded image $I_{decoded}$, $G[x, y, c]$ is the simulated film grain value at the same position and colour component, and filmGrainBitDepth[c] is the number of bits used for each sample in a fixed-length unsigned binary representation of the array $I_{grain}[x, y, c]$.

log2_scale_factor specifies a scale factor used in the film grain characterization equations.

comp_model_present_flag[c **]** equal to 0 indicates that film grain is not modelled on the c-th colour component, where c equal to 0 refers to the luma component, c equal to 1 refers to the Cb component, and c equal to 2 refers to the Cr component. comp_model_present_flag[c] equal to 1 indicates that syntax elements specifying modelling of film grain on colour component c are present in the SEI message.

num_intensity_intervals_minus1[c] plus 1 specifies the number of intensity intervals for which a specific set of model values has been estimated.

NOTE 4 – The intensity intervals may overlap in order to simulate multi-generational film grain.

num_model_values_minus1[c] plus 1 specifies the number of model values present for each intensity interval in which the film grain has been modelled. The value of num_model_values_minus1[c] shall be in the range of 0 to 5, inclusive.

intensity_interval_lower_bound[c][i] specifies the lower bound of the interval i of intensity levels for which the set of model values applies.

intensity_interval_upper_bound[c][i] specifies the upper bound of the interval i of intensity levels for which the set of model values applies.

Depending on model_id, the selection of the sets of model values is specified as follows.

– If model_id is equal to 0, the average value of each block b of 16x16 samples in $I_{decoded}$, referred as b_{avg}, is used to select the sets of model values with index s[j] that apply to all the samples in the block:

```
for( i = 0, j = 0; i <= num_intensity_intervals_minus1; i++ )
    if( b_avg >= intensity_interval_lower_bound[ c ][ i ] && b_avg <= intensity_interval_upper_bound[ c ][ i ] ) {
      s[ j ] = i                                                                                    (D-18)
      j++
    }
```

– Otherwise (model_id is equal to 1), the sets of model values used to generate the film grain are selected for each sample value in $I_{decoded}$ as follows:

```
for( i = 0, j = 0; i <= num_intensity_intervals_minus1; i++ )
    if( I_decoded[ x, y, c ] >= intensity_interval_lower_bound[ c ][ i ] &&
        I_decoded[ x, y, c ] <= intensity_interval_upper_bound[ c ][ i ] ) {                        (D-19)
      s[ j ] = i
      j++
    }
```

Samples that do not fall into any of the defined intervals are not modified by the grain generation function. Samples that fall into more than one interval will originate multi-generation grain. Multi-generation grain results from adding the grain computed independently for each intensity interval.

comp_model_value[c][i][j] represents each one of the model values present for the colour component c and the intensity interval i. The set of model values has different meaning depending on the value of model_id. The value of comp_model_value[c][i][j] shall be constrained as follows, and may be additionally constrained as specified elsewhere in this subclause.

– If model_id is equal to 0, comp_model_value[c][i][j] shall be in the range of 0 to $2^{filmGrainBitDepth[\,c\,]} - 1$, inclusive.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][j] shall be in the range of $-2^{(\,filmGrainBitDepth[\,c\,]\,-\,1\,)}$ to $2^{(\,filmGrainBitDepth[\,c\,]\,-\,1\,)} - 1$, inclusive.

Depending on model_id, the synthesis of the film grain is modelled as follows.

– If model_id is equal to 0, a frequency filtering model enables simulating the original film grain for c = 0..2, x = 0..PicWidthInSamples$_L$, and y = 0..PicHeightInSamples$_L$ as specified by:

$$G[\,x, y, c\,] = (\,comp_model_value[\,c\,][\,s\,][\,0\,] * Q[\,x, y, c\,] + comp_model_value[\,c\,][\,s\,][\,5\,] * G[\,x, y, c\text{-}1\,]\,) >> log2_scale_factor \quad \text{(D-20)}$$

where Q[c] is a two-dimensional random process generated by filtering 16x16 blocks gaussRv with random-value elements $gaussRv_{ij}$ generated with a normalized Gaussian distribution (independent and identically distributed Gaussian random variable samples with zero mean and unity variance) and where the value of an element G[x, y, c-1] used in the right-hand side of the equation is inferred to be equal to 0 when c-1 is less than 0.

NOTE 5 – A normalized Gaussian random value can be generated from two independent, uniformly distributed random values over the interval from 0 to 1 (and not equal to 0), denoted as uRv_0 and uRv_1, using the Box-Muller transformation specified by

$$gaussRv_{ij} = \sqrt{-2 * Ln(\,uRv_0\,)} * Cos(\,2 * \pi * uRv_1\,) \quad \text{(D-21)}$$

where Ln(x) is the natural logarithm of x (the base-e logarithm, where e is natural logarithm base constant 2.718 281 828...), Cos(x) is the trigonometric cosine function operating on an argument x in units of radians, and π is Archimedes' constant 3.141 592 653....

The band-pass filtering of blocks gaussRv may be performed in the discrete cosine transform (DCT) domain as follows:

```
for( y = 0; y < 16; y++ )
    for( x = 0; x < 16; x++ )
        if( ( x < comp_model_value[ c ][ s ][ 3 ]  &&  y < comp_model_value[ c ][ s ][ 4 ] )  ||
          x > comp_model_value[ c ][ s ][ 1 ]  ||  y > comp_model_value[ c ][ s ][ 2 ] )
            gaussRv[ x, y ] = 0
filteredRv = IDCT16x16( gaussRv )
```
(D-22)

where IDCT16x16(z) refers to a unitary inverse discrete cosine transformation (IDCT) operating on a 16x16 matrix argument z as specified by

$$IDCT16x16(\,z\,) = r * z * r^T \quad \text{(D-23)}$$

where the superscript T indicates a matrix transposition and r is the 16x16 matrix with elements r_{ij} specified by

$$r_{ij} = \frac{((\,i == 0\,)\,?\;1 : \sqrt{2}\,)}{4} Cos\left(\frac{i * (\,2 * j + 1\,) * \pi}{32}\right) \quad \text{(D-24)}$$

where Cos(x) is the trigonometric cosine function operating on an argument x in units of radians and π is Archimedes' constant 3.141 592 653.

Q[c] is formed by the frequency-filtered blocks filteredRv.

NOTE 6 – Coded model values are based on blocks of 16x16, but a decoder implementation may use other block sizes. For example, decoders implementing the IDCT on 8x8 blocks, should down-convert by a factor of two the set of coded model values comp_model_value[c][s][i] for i equal to 1..4.

NOTE 7 – To reduce the degree of visible blocks that can result from mosaicing the frequency-filtered blocks filteredRv, decoders may apply a low-pass filter to the boundaries between frequency-filtered blocks.

– Otherwise (model_id is equal to 1), an auto-regression model enables simulating the original film grain for c = 0..2, x = 0..PicWidthInSamples$_L$, and y = 0..PicHeightInSamples$_L$ as specified by

```
G[ x, y, c ] = ( comp_model_value[ c ][ s ][ 0 ] * n[ x, y, c ] +
    comp_model_value[ c ][ s ][ 1 ] * ( G[ x-1, y, c ] + ( ( comp_model_value[ c ][ s ][ 4 ] * G[ x, y-1, c ] ) >>
        log2_scale_factor ) ) +
    comp_model_value[ c ][ s ][ 3 ] * ( ( ( comp_model_value[ c ][ s ][ 4 ] * G[ x-1, y-1, c ] ) >>
        log2_scale_factor ) + G[ x+1, y-1, c ] ) +
    comp_model_value[ c ][ s ][ 5 ] * ( G[ x-2, y, c ] +
        ( ( comp_model_value[ c ][ s ][ 4 ] * comp_model_value[ c ][ s ][ 4 ] * G[ x, y-2, c ] ) >>
            ( 2 * log2_scale_factor ) ) ) +
    comp_model_value[ c ][ s ][ 2 ] * G[ x, y, c-1 ] ) >> log2_scale_factor                    (D-25)
```

where n[x, y, c] is a random value with normalized Gaussian distribution (independent and identically distributed Gaussian random variable samples with zero mean and unity variance for each value of x, y, and c) and where the value of an element G[x, y, c] used in the right-hand side of the equation is inferred to be equal to 0 when any of the following conditions are true:

– x is less than 0,

– y is less than 0,

– x is greater than or equal to PicWidthInSamples$_L$,

– c is less than 0.

comp_model_value[c][i][0] provides the first model value for the model as specified by model_id. comp_model_value[c][i][0] corresponds to the standard deviation of the Gaussian noise term in the generation functions specified in Equations D-20 through D-23.

comp_model_value[c][i][1] provides the second model value for the model as specified by model_id. comp_model_value[c][i][1] shall be greater than or equal to 0 and less than 16.

When not present in the film grain characteristics SEI message, comp_model_value[c][i][1] shall be inferred as follows.

– If model_id is equal to 0, comp_model_value[c][i][1] shall be inferred to be equal to 8.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][1] shall be inferred to be equal to 0.

comp_model_value[c][i][1] is interpreted as follows.

– If model_id is equal to 0, comp_model_value[c][i][1] indicates the horizontal high cut frequency to be used to filter the DCT of a block of 16x16 random values.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][1] indicates the first order spatial correlation for neighbouring samples (x-1, y) and (x, y-1).

comp_model_value[c][i][2] provides the third model value for the model as specified by model_id. comp_model_value[c][i][2] shall be greater than or equal to 0 and less than 16.

When not present in the film grain characteristics SEI message, comp_model_value[c][i][2] shall be inferred as follows.

– If model_id is equal to 0, comp_model_value[c][i][2] shall be inferred to be equal to comp_model_value[c][i][1]

– Otherwise (model_id is equal to 0), comp_model_value[c][i][2] shall be inferred to be equal to 0.

comp_model_value[c][i][2] is interpreted as follows.

– If model_id is equal to 0, comp_model_value[c][i][2] indicates the vertical high cut frequency to be used to filter the DCT of a block of 16x16 random values.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][2] indicates the colour correlation between consecutive colour components.

comp_model_value[c][i][3] provides the fourth model value for the model as specified by model_id. comp_model_value[c][i][3] shall be greater than or equal to 0 and less than or equal to comp_model_value[c][i][1].

When not present in the film grain characteristics SEI message, comp_model_value[c][i][3] shall be inferred to be equal to 0.

comp_model_value[c][i][3] is interpreted as follows.

– If model_id is equal to 0, comp_model_value[c][i][3] indicates the horizontal low cut frequency to be used to filter the DCT of a block of 16x16 random values.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][3] indicates the first order spatial correlation for neighbouring samples (x-1, y-1) and (x+1, y-1).

comp_model_value[c][i][4] provides the fifth model value for the model as specified by model_id. comp_model_value[c][i][4] shall be greater than or equal to 0 and less than or equal to comp_model_value[c][i][2].

When not present in the film grain characteristics SEI message, comp_model_value[c][i][4] shall be inferred to be equal to model_id.

comp_model_value[c][i][4] is interpreted as follows.

– If model_id is equal to 0, comp_model_value[c][i][4] indicates the vertical low cut frequency to be used to filter the DCT of a block of 16x16 random values.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][4] indicates the aspect ratio of the modelled grain.

comp_model_value[c][i][5] provides the sixth model value for the model as specified by model_id. When not present in the film grain characteristics SEI message, comp_model_value[c][i][5] shall be inferred to be equal to 0.

comp_model_value[c][i][5] is interpreted as follows.

– If model_id is equal to 0, comp_model_value[c][i][5] indicates the colour correlation between consecutive colour components.

– Otherwise (model_id is equal to 1), comp_model_value[c][i][5] indicates the second order spatial correlation for neighbouring samples (x, y-2) and (x-2, y).

film_grain_characteristics_repetition_period specifies the persistence of the film grain characteristics SEI message and may specify a picture order count interval within which another film grain characteristics SEI message or the end of the coded video sequence shall be present in the bitstream. The value of film_grain_characteristics_repetition_period shall be in the range 0 to 16 384, inclusive.

film_grain_characteristics_repetition_period equal to 0 specifies that the film grain characteristics SEI message applies to the current decoded picture only.

film_grain_characteristics_repetition_period equal to 1 specifies that the film grain characteristics SEI message persists in output order until any of the following conditions are true.

– A new coded video sequence begins, or

– A picture in an access unit containing a film grain characteristics SEI message that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

film_grain_characteristics_repetition_period greater than 1 specifies that the film grain characteristics SEI message persists until any of the following conditions are true.

– A new coded video sequence begins, or

– A picture in an access unit containing a film grain characteristics SEI message is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + film_grain_characteristics_repetition_period.

film_grain_characteristics_repetition_period greater than 1 indicates that another film grain characteristics SEI message shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + film_grain_characteristics_repetition_period; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

D.2.21 Deblocking filter display preference SEI message semantics

This SEI message provides the decoder with an indication of whether the display of the cropped result of the deblocking filter process specified in subclause 8.7 or of the cropped result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 is preferred by the encoder for the display of each decoded picture that is output.

NOTE 1 – The display process is not specified in this Recommendation | International Standard. The means by which an encoder determines what to indicate as its preference expressed in a deblocking filter display preference SEI message is also not specified in this Recommendation | International Standard, and the expression of an expressed preference in a deblocking filter display preference SEI message does not impose any requirement on the display process.

deblocking_display_preference_cancel_flag equal to 1 indicates that the SEI message cancels the persistence of any previous deblocking filter display preference SEI message in output order. deblocking_display_preference_cancel_flag equal to 0 indicates that a display_prior_to_deblocking_preferred_flag and deblocking_display_preference_repetition_period follow.

NOTE 2 – In the absence of the deblocking filter display preference SEI message, or after the receipt of a deblocking filter display preference SEI message in which deblocking_display_preference_cancel_flag is equal to 1, the decoder should infer that the display of the cropped result of the deblocking filter process specified in subclause 8.7 is preferred over the display of the cropped result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 for the display of each decoded picture that is output.

display_prior_to_deblocking_preferred_flag equal to 1 indicates that the encoder preference is for the display process (which is not specified in this Recommendation | International Standard) to display the cropped result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 rather than the cropped result of the deblocking filter process specified in subclause 8.7 for each picture that is cropped and output as specified in Annex C. display_prior_to_deblocking_preferred_flag equal to 0 indicates that the encoder preference is for the display process (which is not specified in this Recommendation | International Standard) to display the cropped result of the deblocking filter process specified in subclause 8.7 rather than the cropped result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 for each picture that is cropped and output as specified in Annex C.

NOTE 3 – The presence or absence of the deblocking filter display preference SEI message and the value of display_prior_to_deblocking_preferred_flag does not affect the requirements of the decoding process specified in this Recommendation | International Standard. Rather, it only provides an indication of when, in addition to fulfilling the requirements of this Recommendation | International Standard for the decoding process, enhanced visual quality might be obtained by performing the display process (which is not specified in this Recommendation | International Standard) in an alternative fashion. Encoders that use the deblocking filter display preference SEI message should be designed with an awareness that unless the encoder restricts its use of the DPB capacity specified in Annex A for the profile and level in use, some decoders may not have sufficient memory capacity for the storage of the result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 in addition to the storage of the result of the deblocking filter process specified in subclause 8.7 when reordering and delaying pictures for display, and such decoders would therefore not be able to benefit from the preference indication. By restricting its use of the DPB capacity, an encoder can be able to use at least half of the DPB capacity specified in Annex A while allowing the decoder to use the remaining capacity for storage of unfiltered pictures that have been indicated as preferable for display until the output time arrives for those pictures.

dec_frame_buffering_constraint_flag equal to 1 indicates that the use of the frame buffering capacity of the HRD decoded picture buffer (DPB) as specified by max_dec_frame_buffering has been constrained such that the coded video sequence will not require a decoded picture buffer with more than Max(1, max_dec_frame_buffering) frame buffers to enable the output of the decoded filtered or unfiltered pictures, as indicated by the deblocking filter display preference SEI messages, at the output times specified by the dpb_output_delay of the picture timing SEI messages. dec_frame_buffering_constraint_flag equal to 0 indicates that the use of the frame buffering capacity in the HRD may or may not be constrained in the manner that would be indicated by dec_frame_buffering_constraint_flag equal to 1.

For purposes of determining the constraint imposed when dec_frame_buffering_constraint_flag is equal to 1, the quantity of frame buffering capacity used at any given point in time by each frame buffer of the DPB that contains a picture shall be derived as follows:

- If both of the following criteria are satisfied for the frame buffer, the frame buffer is considered to use two frame buffers of capacity for its storage.
  - The frame buffer contains a frame or one or more fields that is marked as "used for reference", and
  - The frame buffer contains a picture for which both of the following criteria are fulfilled.
    - The HRD output time of the picture is greater than the given point in time, and
    - It has been indicated in a deblocking filter display preference SEI message that the encoder preference for the picture is for the display process to display the cropped result of the picture construction process prior to the deblocking filter process specified in subclause 8.5.12 rather than the cropped result of the deblocking filter process specified in subclause 8.7, and
- Otherwise, the frame buffer is considered to use one frame buffer of DPB capacity for its storage.

When dec_frame_buffering_constraint_flag is equal to 1, the frame buffering capacity used by all of the frame buffers in the DPB that contain pictures, as derived in this manner, shall not be greater than Max(1, max_dec_frame_buffering) during the operation of the HRD for the coded video sequence.

The value of dec_frame_buffering_constraint_flag shall be the same in all deblocking filter display preference SEI messages of the coded video sequence.

deblocking_display_preference_repetition_period specifies the persistence of the film grain characteristics SEI message and may specify a picture order count interval within which another film grain characteristics SEI message or the end of the coded video sequence shall be present in the bitstream. The value of deblocking_display_preference_repetition_period shall be in the range 0 to 16 384, inclusive.

deblocking_display_preference_repetition_period equal to 0 specifies that the deblocking filter display preference SEI message applies to the current decoded picture only.

deblocking_display_preference_repetition_period equal to 1 specifies that the deblocking filter display preference SEI message persists in output order until any of the following conditions are true.

– A new coded video sequence begins

– A picture in an access unit containing a deblocking filter display preference SEI message that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

deblocking_display_preference_repetition_period greater than 1 specifies that the deblocking filter display preference SEI message persists until any of the following conditions are true.

– A new coded video sequence begins

– A picture in an access unit containing a deblocking filter display preference SEI message is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + deblocking_display_preference_repetition_period.

deblocking_display_preference_repetition_period greater than 1 indicates that another deblocking filter display preference SEI message shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + deblocking_display_preference_repetition_period; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

D.2.22 Stereo video information SEI message semantics

This SEI message provides the decoder with an indication that the entire coded video sequence consists of pairs of pictures forming stereo-view content.

The stereo video information SEI message shall not be present in any access unit of a coded video sequence unless a stereo video information SEI message is present in the first access unit of the coded video sequence.

field_views_flag equal to 1 indicates that all pictures in the current coded video sequence are fields and all fields of a particular parity are considered a left view and all fields of the opposite parity are considered a right view for stereo-view content. field_views_flag equal to 0 indicates that all pictures in the current coded video sequence are frames and alternating frames in output order represent a view of a stereo view. The value of field_views_flag shall be the same in all stereo video information SEI messages within a coded video sequence.

When the stereo video information SEI message is present and field_views_flag is equal to 1, the left view and right view of a stereo video pair shall be coded as a complementary field pair, the display time of the first field of the field pair in output order should be delayed to coincide with the display time of the second field of the field pair in output order, and the spatial locations of the samples in each individual field should be interpreted for display purposes as representing complete pictures as shown in Figure 6-1 rather than as spatially-distinct fields within a frame as shown in Figure 6-2.

NOTE – The display process is not specified in this Recommendation | International Standard.

top_field_is_left_view_flag equal to 1 indicates that the top fields in the coded video sequence represent a left view and the bottom fields in the coded video sequence represent a right view. top_field_is_left_view_flag equal to 0 indicates that the bottom fields in the coded video sequence represent a left view and the top fields in the coded video sequence represent a right view. When present, the value of top_field_is_left_view_flag shall be the same in all stereo video information SEI messages within a coded video sequence.

current_frame_is_left_view_flag equal to 1 indicates that the current picture is the left view of a stereo-view pair. current_frame_is_left_view_flag equal to 0 indicates that the current picture is the right view of a stereo-view pair.

next_frame_is_second_view_flag equal to 1 indicates that the current picture and the next picture in output order form a stereo-view pair, and the display time of the current picture should be delayed to coincide with the display time of the next picture in output order. next_frame_is_second_view_flag equal to 0 indicates that the current picture and the previous picture in output order form a stereo-view pair, and the display time of the current picture should not be delayed for purposes of stereo-view pairing.

left_view_self_contained_flag equal to 1 indicates that no inter prediction operations within the decoding process for the left-view pictures of the coded video sequence refer to reference pictures that are right-view pictures. left_view_self_contained_flag equal to 0 indicates that some inter prediction operations within the decoding process for the left-view pictures of the coded video sequence may or may not refer to reference pictures that are right-view pictures. Within a coded video sequence, the value of left_view_self_contained_flag in all stereo video information SEI messages shall be the same.

right_view_self_contained_flag equal to 1 indicates that no inter prediction operations within the decoding process for the right-view pictures of the coded video sequence refer to reference pictures that are left-view pictures. right_view_self_contained_flag equal to 0 indicates that some inter prediction operations within the decoding process for the right-view pictures of the coded video sequence may or may not refer to reference pictures that are left-view pictures. Within a coded video sequence, the value of right_view_self_contained_flag in all stereo video information SEI messages shall be the same.

D.2.23 Reserved SEI message semantics

This message consists of data reserved for future backward-compatible use by ITU-T | ISO/IEC. Encoders conforming to this Recommendation | International Standard shall not send reserved SEI messages until and unless the use of such messages has been specified by ITU-T | ISO/IEC. Decoders conforming to this Recommendation | International Standard that encounter reserved SEI messages shall discard their content without effect on the decoding process, except as specified in future Recommendations | International Standards specified by ITU-T | ISO/IEC.

reserved_sei_message_payload_byte is a byte reserved for future use by ITU-T | ISO/IEC.

Annex E
Video usability information

(This annex forms an integral part of this Recommendation | International Standard)

This Annex specifies syntax and semantics of the VUI parameters of the sequence parameter sets.

VUI parameters are not required for constructing the luma or chroma samples by the decoding process. Conforming decoders are not required to process this information for output order conformance to this Recommendation | International Standard (see Annex C for the specification of conformance). Some VUI parameters are required to check bitstream conformance and for output timing decoder conformance.

In Annex E, specification for presence of VUI parameters is also satisfied when those parameters (or some subset of them) are conveyed to decoders (or to the HRD) by other means not specified by this Recommendation | International Standard. When present in the bitstream, VUI parameters shall follow the syntax and semantics specified in subclauses 7.3.2.1 and 7.4.2.1 and this annex. When the content of VUI parameters is conveyed for the application by some means other than presence within the bitstream, the representation of the content of the VUI parameters is not required to use the same syntax specified in this annex. For the purpose of counting bits, only the appropriate bits that are actually present in the bitstream are counted.

E.1 VUI syntax

E.1.1 VUI parameters syntax

vui_parameters( ) {	C	Descriptor
aspect_ratio_info_present_flag	0	u(1)
if(aspect_ratio_info_present_flag) {		
aspect_ratio_idc	0	u(8)
if(aspect_ratio_idc == Extended_SAR) {		
sar_width	0	u(16)
sar_height	0	u(16)
}		
}		
overscan_info_present_flag	0	u(1)
if(overscan_info_present_flag)		
overscan_appropriate_flag	0	u(1)
video_signal_type_present_flag	0	u(1)
if(video_signal_type_present_flag) {		
video_format	0	u(3)
video_full_range_flag	0	u(1)
colour_description_present_flag	0	u(1)
if(colour_description_present_flag) {		
colour_primaries	0	u(8)
transfer_characteristics	0	u(8)
matrix_coefficients	0	u(8)
}		
}		
chroma_loc_info_present_flag	0	u(1)
if(chroma_loc_info_present_flag) {		
chroma_sample_loc_type_top_field	0	ue(v)
chroma_sample_loc_type_bottom_field	0	ue(v)
}		
timing_info_present_flag	0	u(1)
if(timing_info_present_flag) {		
num_units_in_tick	0	u(32)
time_scale	0	u(32)
fixed_frame_rate_flag	0	u(1)
}		
nal_hrd_parameters_present_flag	0	u(1)
if(nal_hrd_parameters_present_flag)		
hrd_parameters( )		
vcl_hrd_parameters_present_flag	0	u(1)
if(vcl_hrd_parameters_present_flag)		
hrd_parameters( )		
if(nal_hrd_parameters_present_flag || vcl_hrd_parameters_present_flag)		
low_delay_hrd_flag	0	u(1)
pic_struct_present_flag	0	u(1)
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag) {		

motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
num_reorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

E.1.2 HRD parameters syntax

hrd_parameters( ) {	C	Descriptor
cpb_cnt_minus1	0	ue(v)
bit_rate_scale	0	u(4)
cpb_size_scale	0	u(4)
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
bit_rate_value_minus1[SchedSelIdx **]**	0	ue(v)
cpb_size_value_minus1[SchedSelIdx **]**	0	ue(v)
cbr_flag[SchedSelIdx **]**	0	u(1)
}		
initial_cpb_removal_delay_length_minus1	0	u(5)
cpb_removal_delay_length_minus1	0	u(5)
dpb_output_delay_length_minus1	0	u(5)
time_offset_length	0	u(5)
}		

E.2 VUI semantics

E.2.1 VUI parameters semantics

aspect_ratio_info_present_flag equal to 1 specifies that aspect_ratio_idc is present. aspect_ratio_info_present_flag equal to 0 specifies that aspect_ratio_idc is not present.

aspect_ratio_idc specifies the value of the sample aspect ratio of the luma samples. Table E-1 shows the meaning of the code. When aspect_ratio_idc indicates Extended_SAR, the sample aspect ratio is represented by sar_width and sar_height. When the aspect_ratio_idc syntax element is not present, aspect_ratio_idc value shall be inferred to be equal to 0.

Table E-1 – Meaning of sample aspect ratio indicator

aspect_ratio_idc	Sample aspect ratio	(informative) Examples of use
0	Unspecified	
1	1:1 ("square")	1280x720 16:9 frame without horizontal overscan 1920x1080 16:9 frame without horizontal overscan (cropped from 1920x1088) 640x480 4:3 frame without horizontal overscan
2	12:11	720x576 4:3 frame with horizontal overscan 352x288 4:3 frame without horizontal overscan
3	10:11	720x480 4:3 frame with horizontal overscan 352x240 4:3 frame without horizontal overscan
4	16:11	720x576 16:9 frame with horizontal overscan 528x576 4:3 frame without horizontal overscan
5	40:33	720x480 16:9 frame with horizontal overscan 528x480 4:3 frame without horizontal overscan
6	24:11	352x576 4:3 frame without horizontal overscan 480x576 16:9 frame with horizontal overscan
7	20:11	352x480 4:3 frame without horizontal overscan 480x480 16:9 frame with horizontal overscan
8	32:11	352x576 16:9 frame without horizontal overscan
9	80:33	352x480 16:9 frame without horizontal overscan
10	18:11	480x576 4:3 frame with horizontal overscan
11	15:11	480x480 4:3 frame with horizontal overscan
12	64:33	528x576 16:9 frame without horizontal overscan
13	160:99	528x480 16:9 frame without horizontal overscan
14	4:3	1440x1080 16:9 frame without horizontal overscan
15	3:2	1280x1080 16:9 frame without horizontal overscan
16	2:1	960x1080 16:9 frame without horizontal overscan
17..254	Reserved	
255	Extended_SAR	

sar_width indicates the horizontal size of the sample aspect ratio (in arbitrary units).

sar_height indicates the vertical size of the sample aspect ratio (in the same arbitrary units as sar_width).

sar_width and sar_height shall be relatively prime or equal to 0. When aspect_ratio_idc is equal to 0 or sar_width is equal to 0 or sar_height is equal to 0, the sample aspect ratio shall be considered unspecified by this Recommendation | International Standard.

overscan_info_present_flag equal to 1 specifies that the overscan_appropriate_flag is present. When overscan_info_present_flag is equal to 0 or is not present, the preferred display method for the video signal is unspecified.

overscan_appropriate_flag equal to 1 indicates that the cropped decoded pictures output are suitable for display using overscan. overscan_appropriate_flag equal to 0 indicates that the cropped decoded pictures output contain visually important information in the entire region out to the edges of the cropping rectangle of the picture, such that the cropped decoded pictures output should not be displayed using overscan. Instead, they should be displayed using either an exact match between the display area and the cropping rectangle, or using underscan.

NOTE 1 – For example, overscan_appropriate_flag equal to 1 might be used for entertainment television programming, or for a live view of people in a videoconference, and overscan_appropriate_flag equal to 0 might be used for computer screen capture or security camera content.

video_signal_type_present_flag equal to 1 specifies that video_format, video_full_range_flag and colour_description_present_flag are present. video_signal_type_present_flag equal to 0, specify that video_format, video_full_range_flag and colour_description_present_flag are not present.

video_format indicates the representation of the pictures as specified in Table E-2, before being coded in accordance with this Recommendation | International Standard. When the video_format syntax element is not present, video_format value shall be inferred to be equal to 5.

Table E-2 – Meaning of video_format

video_format	Meaning
0	Component
1	PAL
2	NTSC
3	SECAM
4	MAC
5	Unspecified video format
6	Reserved
7	Reserved

video_full_range_flag indicates the black level and range of the luma and chroma signals as derived from E'_Y, E'_{PB}, and E'_{PR} or E'_R, E'_G, and E'_B analogue component signals.

When the video_full_range_flag syntax element is not present, the value of video_full_range_flag shall be inferred to be equal to 0.

colour_description_present_flag equal to 1 specifies that colour_primaries, transfer_characteristics and matrix_coefficients are present. colour_description_present_flag equal to 0 specifies that colour_primaries, transfer_characteristics and matrix_coefficients are not present.

colour_primaries indicates the chromaticity coordinates of the source primaries as specified in Table E-3 in terms of the CIE 1931 definition of x and y as specified by ISO/CIE 10527.

When the colour_primaries syntax element is not present, the value of colour_primaries shall be inferred to be equal to 2 (the chromaticity is unspecified or is determined by the application).

Table E-3 – Colour primaries

Value	Primaries			Informative Remark
0	Reserved			For future use by ITU-T / ISO/IEC
1	primary green blue red white D65	x 0.300 0.150 0.640 0.3127	y 0.600 0.060 0.330 0.3290	ITU-R Recommendation BT.709-5
2	Unspecified			Image characteristics are unknown or are determined by the application.
3	Reserved			
4	primary green blue red white C	x 0.21 0.14 0.67 0.310	y 0.71 0.08 0.33 0.316	ITU-R Recommendation BT.470-6 System M
5	primary green blue red white D65	x 0.29 0.15 0.64 0.3127	y 0.60 0.06 0.33 0.3290	ITU-R Recommendation BT.470-6 System B, G
6	primary green blue red white D65	x 0.310 0.155 0.630 0.3127	y 0.595 0.070 0.340 0.3290	Society of Motion Picture and Television Engineers 170M (1999)
7	primary green blue red white D65	x 0.310 0.155 0.630 0.3127	y 0.595 0.070 0.340 0.3290	Society of Motion Picture and Television Engineers 240M (1999)
8	primary green blue red white C	x 0.243 0.145 0.681 0.310	y 0.692 (Wratten 58) 0.049 (Wratten 47) 0.319 (Wratten 25) 0.316	Generic film (colour filters using Illuminant C)
9-255	Reserved			For future use by ITU-T / ISO/IEC

transfer_characteristics indicates the opto-electronic transfer characteristic of the source picture as specified in Table E-4 as a function of a linear optical intensity input L_c with an analogue range of 0 to 1.

When the transfer_characteristics syntax element is not present, the value of transfer_characteristics shall be inferred to be equal to 2 (the transfer characteristics are unspecified or are determined by the application).

Table E-4 – Transfer characteristics

Value	Transfer Characteristic		Informative Remark
0	Reserved		For future use by ITU-T / ISO/IEC
1	$V = 1.099\ L_c^{0.45} - 0.099$ $V = 4.500\ L_c$	for $1 >= L_c >= 0.018$ for $0.018 > L_c >= 0$	ITU-R Recommendation BT.709-5
2	Unspecified		Image characteristics are unknown or are determined by the application.
3	Reserved		For future use by ITU-T / ISO/IEC
4	Assumed display gamma 2.2		ITU-R Recommendation BT.470-6 System M
5	Assumed display gamma 2.8		ITU-R Recommendation BT.470-6 System B, G
6	$V = 1.099\ L_c^{0.45} - 0.099$ $V = 4.500\ L_c$	for $1 >= L_c >= 0.018$ for $0.018 > L_c >= 0$	Society of Motion Picture and Television Engineers 170M (1999)
7	$V = 1.1115\ L_c^{0.45} - 0.1115$ $V = 4.0\ L_c$	for $1 >= L_c >= 0.0228$ for $0.0228 > L_c >= 0$	Society of Motion Picture and Television Engineers 240M (1999)
8	$V = L_C$	for $1 > L_c >= 0$	Linear transfer characteristics
9	$V = 1.0 - \text{Log10}(L_c) \div 2$ $V = 0.0$	for $1 >= L_c >= 0.01$ for $0.01 > L_c >= 0$	Logarithmic transfer characteristic (100:1 range)
10	$V = 1.0 - \text{Log10}(L_c) \div 2.5$ $V = 0.0$	for $1 >= L_c >= 0.0031622777$ for $0.0031622777 > L_c >= 0$	Logarithmic transfer characteristic (316.22777:1 range)
11..255	Reserved		For future use by ITU-T / ISO/IEC

matrix_coefficients describes the matrix coefficients used in deriving luma and chroma signals from the green, blue, and red primaries, as specified in Table E-5.

matrix_coefficients shall not be equal to 0 unless both of the following conditions are true

– BitDepth_C is equal to BitDepth_Y

– chroma_format_idc is equal to 3 (4:4:4)

The specification of the use of matrix_coefficients equal to 0 under all other conditions is reserved for future use by ITU-T | ISO/IEC.

matrix_coefficients shall not be equal to 8 unless one or both of the following conditions are true

– BitDepth_C is equal to BitDepth_Y

– BitDepth_C is equal to $\text{BitDepth}_Y + 1$ and chroma_format_idc is equal to 3 (4:4:4)

The specification of the use of matrix_coefficients equal to 8 under all other conditions is reserved for future use by ITU-T | ISO/IEC.

When the matrix_coefficients syntax element is not present, the value of matrix_coefficients shall be inferred to be equal to 2.

The interpretation of matrix_coefficients is defined as follows.

– E'_R, E'_G, and E'_B are analogue with values in the range of 0 to 1.

– White is specified as having E'_R equal to 1, E'_G equal to 1, and E'_B equal to 1.

– Black is specified as having E'_R equal to 0, E'_G equal to 0, and E'_B equal to 0.

– If video_full_range_flag is equal to 0, the following equations apply.

  – If matrix_coefficients is equal to 1, 4, 5, 6, or 7, the following equations apply.

$$Y = \text{Round}((1 << (\text{BitDepth}_Y - 8)) * (219 * E'_Y + 16)) \quad \text{(E-1)}$$

$$Cb = Round((1 << (BitDepth_C - 8)) * (224 * E'_{PB} + 128)) \quad (E\text{-}2)$$

$$Cr = Round((1 << (BitDepth_C - 8)) * (224 * E'_{PR} + 128)) \quad (E\text{-}3)$$

– Otherwise, if matrix_coefficients is equal to 0 or 8, the following equations apply.

$$R = (1 << (BitDepth_Y - 8)) * (219 * E'_R + 16) \quad (E\text{-}4)$$

$$G = (1 << (BitDepth_Y - 8)) * (219 * E'_G + 16) \quad (E\text{-}5)$$

$$B = (1 << (BitDepth_Y - 8)) * (219 * E'_B + 16) \quad (E\text{-}6)$$

– Otherwise, if matrix_coefficients is equal to 2, the interpretation of the matrix_coefficients syntax element is unknown or is determined by the application.

– Otherwise (matrix_coefficients is not equal to 0, 1, 2, 4, 5, 6, 7, or 8), the interpretation of the matrix_coefficients syntax element is reserved for future definition by ITU-T | ISO/IEC.

– Otherwise (video_full_range_flag is equal to 1), the following equations apply.

– If matrix_coefficients is equal to 1, 4, 5, 6, or 7, the following equations apply.

$$Y = Round(((1 << BitDepth_Y) - 1) * E'_Y) \quad (E\text{-}7)$$

$$Cb = Round(((1 << BitDepth_C) - 1) * E'_{PB} + (1 << (BitDepth_C - 1)) \quad (E\text{-}8)$$

$$Cr = Round(((1 << BitDepth_C) - 1) * E'_{PR} + (1 << (BitDepth_C - 1)) \quad (E\text{-}9)$$

– Otherwise, if matrix_coefficients is equal to 0 or 8, the following equations apply.

$$R = ((1 << BitDepth_Y) - 1) * E'_R \quad (E\text{-}10)$$

$$G = ((1 << BitDepth_Y) - 1) * E'_G \quad (E\text{-}11)$$

$$B = ((1 << BitDepth_Y) - 1) * E'_B \quad (E\text{-}12)$$

– Otherwise, if matrix_coefficients is equal to 2, the interpretation of the matrix_coefficients syntax element is unknown or is determined by the application.

– Otherwise (matrix_coefficients is not equal to 0, 1, 2, 4, 5, 6, 7, or 8), the interpretation of the matrix_coefficients syntax element is reserved for future definition by ITU-T | ISO/IEC.

– If matrix_coefficients is not equal to 0 or 8, the following equations apply.

$$E'_Y = K_R * E'_R + (1 - K_R - K_B) * E'_G + K_B * E'_B \quad (E\text{-}13)$$

$$E'_{PB} = 0.5 * (E'_B - E'_Y) \div (1 - K_B) \quad (E\text{-}14)$$

$$E'_{PR} = 0.5 * (E'_R - E'_Y) \div (1 - K_R) \quad (E\text{-}15)$$

NOTE 2 – Then E'_Y is analogue with values in the range of 0 to 1, E'_{PB} and E'_{PR} are analogue with values in the range of -0.5 to 0.5, and white is equivalently given by $E'_Y = 1$, $E'_{PB} = 0$, $E'_{PR} = 0$.

– Otherwise, if matrix_coefficients is equal to 0, the following equations apply.

$$Y = \text{Round}(G) \tag{E-16}$$

$$Cb = \text{Round}(B) \tag{E-17}$$

$$Cr = \text{Round}(R) \tag{E-18}$$

– Otherwise (matrix_coefficients is equal to 8), the following applies.
  – If BitDepth_C is equal to BitDepth_Y, the following equations apply.

$$Y = \text{Round}(0.5 * G + 0.25 * (R + B)) \tag{E-19}$$

$$Cb = \text{Round}(0.5 * G – 0.25 (R + B)) \tag{E-20}$$

$$Cr = \text{Round}(0.5 * (R – B)) \tag{E-21}$$

NOTE 3 – For purposes of the YCgCo nomenclature used in Table E-5, Cb and Cr of Equations E-20 and E-21 may be referred to as Cg and Co, respectively. The inverse conversion for the above four equations should be computed as.

$$t = Y – Cb \tag{E-22}$$

$$G = Y + Cb \tag{E-23}$$

$$B = t – Cr \tag{E-24}$$

$$R = t + Cr \tag{E-25}$$

  – Otherwise (BitDepth_C is not equal to BitDepth_Y), the following equations apply.

$$Cr = \text{Round}(R) – \text{Round}(B) \tag{E-26}$$

$$t = \text{Round}(B) + (Cr >> 1) \tag{E-27}$$

$$Cb = \text{Round}(G) – t \tag{E-28}$$

$$Y = t + (Cb >> 1) \tag{E-29}$$

NOTE 4 – For purposes of the YCgCo nomenclature used in Table E-5, Cb and Cr of Equations E-28 and E-26 may be referred to as Cg and Co, respectively. The inverse conversion for the above four equations should be computed as.

$$t = Y – (Cb >> 1) \tag{E-30}$$

$$G = t + Cb \tag{E-31}$$

$$B = t – (Cr >> 1) \tag{E-32}$$

$$R = B + Cr \tag{E-33}$$

Table E-5 – Matrix coefficients

Value	Matrix	Informative remark
0	GBR	Typically referred to as RGB; see Equations E-16 to E-18
1	$K_R = 0.2126$; $K_B = 0.0722$	ITU-R Recommendation BT.709-5 and Society of Motion Picture and Television Engineers RP177 (1993)
2	Unspecified	Image characteristics are unknown or are determined by the application.
3	Reserved	For future use by ITU-T / ISO/IEC
4	$K_R = 0.30$; $K_B = 0.11$	United States Federal Communications Commission Title 47 Code of Federal Regulations (2003) 73.682 (a) (20)
5	$K_R = 0.299$; $K_B = 0.114$	ITU-R Recommendation BT.470-6 System B, G
6	$K_R = 0.299$; $K_B = 0.114$	Society of Motion Picture and Television Engineers 170M (1999)
7	$K_R = 0.212$; $K_B = 0.087$	Society of Motion Picture and Television Engineers 240M (1999)
8	YCgCo	See Equations E-19 to E-33
9-255	Reserved	For future use by ITU-T / ISO/IEC

chroma_loc_info_present_flag equal to 1 specifies that chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field are present. chroma_loc_info_present_flag equal to 0 specifies that chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field are not present.

chroma_sample_loc_type_top_field and **chroma_sample_loc_type_bottom_field** specify the location of chroma samples for the top field and the bottom field as shown in Figure E-1. The value of chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field shall be in the range of 0 to 5, inclusive. When the chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field are not present, the values of chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field shall be inferred to be equal to 0.

NOTE 5 – When coding progressive source material, chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field should have the same value.

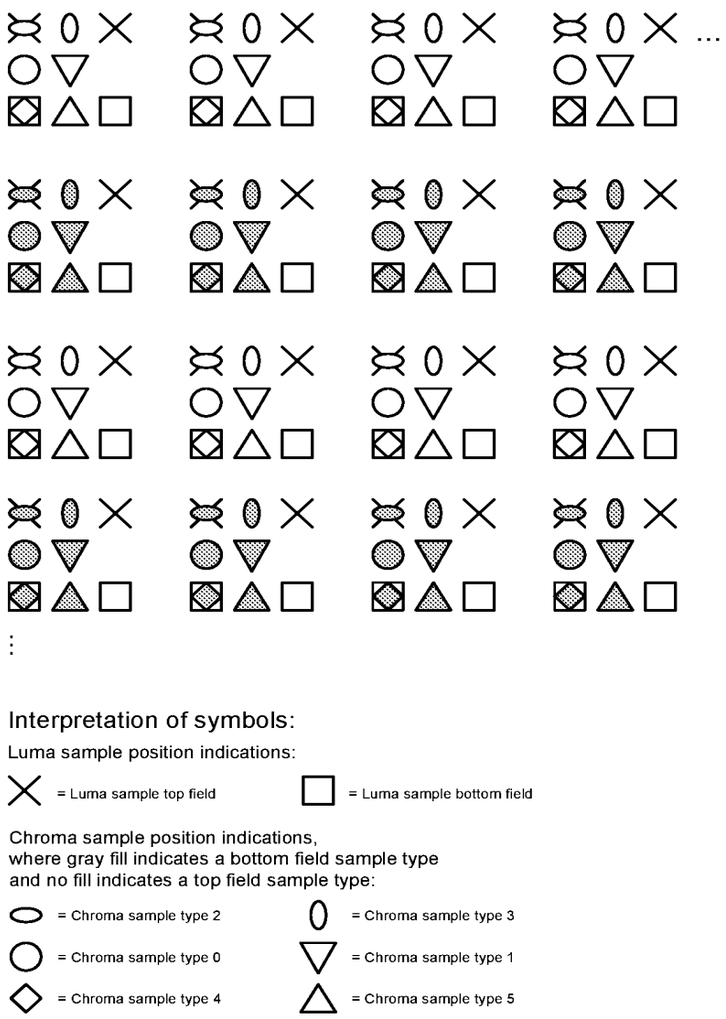


Figure E-1 – Location of chroma samples for top and bottom fields as a function of chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field

timing_info_present_flag equal to 1 specifies that num_units_in_tick, time_scale and fixed_frame_rate_flag are present in the bitstream. timing_info_present_flag equal to 0 specifies that num_units_in_tick, time_scale and fixed_frame_rate_flag are not present in the bitstream.

num_units_in_tick is the number of time units of a clock operating at the frequency time_scale Hz that corresponds to one increment (called a clock tick) of a clock tick counter. num_units_in_tick shall be greater than 0. A clock tick is the minimum interval of time that can be represented in the coded data. For example, when the clock frequency of a video signal is 60 000 ÷ 1001 Hz, time_scale may be equal to 60 000 and num_units_in_tick may be equal to 1001. See Equation C-1.

time_scale is the number of time units that pass in one second. For example, a time coordinate system that measures time using a 27 MHz clock has a time_scale of 27 000 000. time_scale shall be greater than 0.

fixed_frame_rate_flag equal to 1 indicates that the temporal distance between the HRD output times of any two consecutive pictures in output order is constrained as follows. fixed_frame_rate_flag equal to 0 indicates that no such constraints apply to the temporal distance between the HRD output times of any two consecutive pictures in output order.

For each picture n where n indicates the n-th picture (in output order) that is output and picture n is not the last picture in the bitstream (in output order) that is output, the value of $\Delta t_{fi,dpb}(n)$ is specified by

$$\Delta t_{fi,dpb}(n) = \Delta t_{o,dpb}(n) \div \text{DeltaTfiDivisor} \quad \text{(E-34)}$$

where $\Delta t_{o,dpb}(n)$ is specified in Equation C-13 and DeltaTfiDivisor is specified by Table E-6 based on the value of pic_struct_present_flag, field_pic_flag, and pic_struct for the coded video sequence containing picture n. Entries marked "-" in Table E-6 indicate a lack of dependence of DeltaTfiDivisor on the corresponding syntax element.

When fixed_frame_rate_flag is equal to 1 for a coded video sequence containing picture n, the value computed for $\Delta t_{fi,dpb}(n)$ shall be equal to t_c as specified in Equation C-1 (using the value of t_c for the coded video sequence containing picture n) when either or both of the following conditions are true for the following picture n_n that is specified for use in Equation C-13.

– picture n_n is in the same coded video sequence as picture n.

– picture n_n is in a different coded video sequence and fixed_frame_rate_flag is equal to 1 in the coded video sequence containing picture n_n and the value of num_units_in_tick ÷ time_scale is the same for both coded video sequences.

Table E-6 – Divisor for computation of $\Delta t_{fi,dpb}(n)$

pic_struct_present_flag	field_pic_flag	pic_struct	DeltaTfiDivisor
0	1	-	1
1	-	1	1
1	-	2	1
0	0	-	2
1	-	0	2
1	-	3	2
1	-	4	2
1	-	5	3
1	-	6	3
1	-	7	4
1	-	8	6

nal_hrd_parameters_present_flag equal to 1 specifies that NAL HRD parameters (pertaining to Type II bitstream conformance) are present. nal_hrd_parameters_present_flag equal to 0 specifies that NAL HRD parameters are not present.

NOTE 6 – When nal_hrd_parameters_present_flag is equal to 0, the conformance of the bitstream cannot be verified without provision of the NAL HRD parameters, including the NAL sequence HRD parameter information and all buffering period and picture timing SEI messages, by some means not specified in this Recommendation | International Standard.

When nal_hrd_parameters_present_flag is equal to 1, NAL HRD parameters (subclauses E.1.2 and E.2.2) immediately follow the flag.

The variable NalHrdBpPresentFlag is derived as follows.

– If any of the following is true, the value of NalHrdBpPresentFlag shall be set equal to 1.
  – nal_hrd_parameters_present_flag is present in the bitstream and is equal to 1
  – the need for presence of buffering periods for NAL HRD operation to be present in the bitstream in buffering period SEI messages is determined by the application, by some means not specified in this Recommendation | International Standard.

- Otherwise, the value of NalHrdBpPresentFlag shall be set equal to 0.

vcl_hrd_parameters_present_flag equal to 1 specifies that VCL HRD parameters (pertaining to all bitstream conformance) are present. vcl_hrd_parameters_present_flag equal to 0 specifies that VCL HRD parameters are not present.

NOTE 7 – When vcl_hrd_parameters_present_flag is equal to 0, the conformance of the bitstream cannot be verified without provision of the VCL HRD parameters and all buffering period and picture timing SEI messages, by some means not specified in this Recommendation | International Standard.

When vcl_hrd_parameters_present_flag is equal to 1, VCL HRD parameters (subclauses E.1.2 and E.2.2) immediately follow the flag.

The variable VclHrdBpPresentFlag is derived as follows.

– If any of the following is true, the value of VclHrdBpPresentFlag shall be set equal to 1.
  – vcl_hrd_parameters_present_flag is present in the bitstream and is equal to 1
  – the need for presence of buffering periods for VCL HRD operation to be present in the bitstream in buffering period SEI messages is determined by the application, by some means not specified in this Recommendation | International Standard.
– Otherwise, the value of VclHrdBpPresentFlag shall be set equal to 0.

The variable CpbDpbDelaysPresentFlag is derived as follows.

– If any of the following is true, the value of CpbDpbDelaysPresentFlag shall be set equal to 1.
  – nal_hrd_parameters_present_flag is present in the bitstream and is equal to 1
  – vcl_hrd_parameters_present_flag is present in the bitstream and is equal to 1
  – the need for presence of CPB and DPB output delays to be present in the bitstream in picture timing SEI messages is determined by the application, by some means not specified in this Recommendation | International Standard.
– Otherwise, the value of CpbDpbDelaysPresentFlag shall be set equal to 0.

low_delay_hrd_flag specifies the HRD operational mode as specified in Annex C. When fixed_frame_rate_flag is equal to 1, low_delay_hrd_flag shall be equal to 0.

NOTE 8 – When low_delay_hrd_flag is equal to 1, "big pictures" that violate the nominal CPB removal times due to the number of bits used by an access unit are permitted. It is expected, but not required, that such "big pictures" occur only occasionally.

pic_struct_present_flag equal to 1 specifies that picture timing SEI messages (subclause D.2.2) are present that include the pic_struct syntax element. pic_struct_present_flag equal to 0 specifies that the pic_struct syntax element is not present in picture timing SEI messages. When pic_struct_present_flag is not present, its value shall be inferred to be equal to 0.

bitstream_restriction_flag equal to 1, specifies that the following coded video sequence bitstream restriction parameters are present. bitstream_restriction_flag equal to 0, specifies that the following coded video sequence bitstream restriction parameters are not present.

motion_vectors_over_pic_boundaries_flag equal to 0 indicates that no sample outside the picture boundaries and no sample at a fractional sample position whose value is derived using one or more samples outside the picture boundaries is used to inter predict any sample. motion_vectors_over_pic_boundaries_flag equal to 1 indicates that one or more samples outside picture boundaries may be used in inter prediction. When the motion_vectors_over_pic_boundaries_flag syntax element is not present, motion_vectors_over_pic_boundaries_flag value shall be inferred to be equal to 1.

max_bytes_per_pic_denom indicates a number of bytes not exceeded by the sum of the sizes of the VCL NAL units associated with any coded picture in the coded video sequence.

The number of bytes that represent a picture in the NAL unit stream is specified for this purpose as the total number of bytes of VCL NAL unit data (i.e., the total of the NumBytesInNALunit variables for the VCL NAL units) for the picture. The value of max_bytes_per_pic_denom shall be in the range of 0 to 16, inclusive.

Depending on max_bytes_per_pic_denom the following applies.

– If max_bytes_per_pic_denom is equal to 0, no limits are indicated.
– Otherwise (max_bytes_per_pic_denom is not equal to 0), no coded picture shall be represented in the coded video sequence by more than the following number of bytes.

$$(\text{PicSizeInMbs} * \text{RawMbBits}) \div (8 * \text{max_bytes_per_pic_denom}) \quad \text{(E-35)}$$

When the max_bytes_per_pic_denom syntax element is not present, the value of max_bytes_per_pic_denom shall be inferred to be equal to 2.

max_bits_per_mb_denom indicates the maximum number of coded bits of macroblock_layer( ) data for any macroblock in any picture of the coded video sequence. The value of max_bits_per_mb_denom shall be in the range of 0 to 16, inclusive.

Depending on max_bits_per_mb_denom the following applies.

– If max_bits_per_mb_denom is equal to 0, no limit is specified.

– Otherwise (max_bits_per_mb_denom is not equal to 0), no coded macroblock_layer( ) shall be represented in the bitstream by more than the following number of bits.

$$(128 + RawMbBits) \div max_bits_per_mb_denom \quad \text{(E-36)}$$

Depending on entropy_coding_mode_flag, the bits of macroblock_layer( ) data are counted as follows.

– If entropy_coding_mode_flag is equal to 0, the number of bits of macroblock_layer( ) data is given by the number of bits in the macroblock_layer( ) syntax structure for a macroblock.

– Otherwise (entropy_coding_mode_flag is equal to 1), the number of bits of macroblock_layer( ) data for a macroblock is given by the number of times read_bits(1) is called in subclauses 9.3.3.2.2 and 9.3.3.2.3 when parsing the macroblock_layer( ) associated with the macroblock.

When the max_bits_per_mb_denom is not present, the value of max_bits_per_mb_denom shall be inferred to be equal to 1.

log2_max_mv_length_horizontal and **log2_max_mv_length_vertical** indicate the maximum absolute value of a decoded horizontal and vertical motion vector component, respectively, in ¼ luma sample units, for all pictures in the coded video sequence. A value of n asserts that no value of a motion vector component shall exceed the range from -2^n to 2^n-1, inclusive, in units of ¼ luma sample displacement. The value of log2_max_mv_length_horizontal shall be in the range of 0 to 16, inclusive. The value of log2_max_mv_length_vertical shall be in the range of 0 to 16, inclusive. When log2_max_mv_length_horizontal is not present, the values of log2_max_mv_length_horizontal and log2_max_mv_length_vertical shall be inferred to be equal to 16.

NOTE 9 – The maximum absolute value of a decoded vertical or horizontal motion vector component is also constrained by profile and level limits as specified in Annex A.

num_reorder_frames indicates the maximum number of frames, complementary field pairs, or non-paired fields that precede any frame, complementary field pair, or non-paired field in the coded video sequence in decoding order and follow it in output order. The value of num_reorder_frames shall be in the range of 0 to max_dec_frame_buffering, inclusive. When the num_reorder_frames syntax element is not present, the value of num_reorder_frames value shall be inferred to be equal to max_dec_frame_buffering.

max_dec_frame_buffering specifies the required size of the HRD decoded picture buffer (DPB) in units of frame buffers. The coded video sequence shall not require a decoded picture buffer with size of more than Max(1, max_dec_frame_buffering) frame buffers to enable the output of decoded pictures at the output times specified by dpb_output_delay of the picture timing SEI messages. The value of max_dec_frame_buffering shall be in the range of num_ref_frames to MaxDpbSize (as specified in subclause A.3.1 or A.3.2), inclusive. When the max_dec_frame_buffering syntax element is not present, the value of max_dec_frame_buffering shall be inferred to be equal to MaxDpbSize.

E.2.2 HRD parameters semantics

cpb_cnt_minus1 plus 1 specifies the number of alternative CPB specifications in the bitstream. The value of cpb_cnt_minus1 shall be in the range of 0 to 31, inclusive. When low_delay_hrd_flag is equal to 1, cpb_cnt_minus1 shall be equal to 0. When cpb_cnt_minus1 is not present, it shall be inferred to be equal to 0.

bit_rate_scale (together with bit_rate_value_minus1[SchedSelIdx]) specifies the maximum input bit rate of the SchedSelIdx-th CPB.

cpb_size_scale (together with cpb_size_value_minus1[SchedSelIdx]) specifies the CPB size of the SchedSelIdx-th CPB.

bit_rate_value_minus1[SchedSelIdx **]** (together with bit_rate_scale) specifies the maximum input bit rate for the SchedSelIdx-th CPB. bit_rate_value_minus1[SchedSelIdx] shall be in the range of 0 to 2^{32} - 2, inclusive. For any SchedSelIdx > 0, bit_rate_value_minus1[SchedSelIdx] shall be greater than bit_rate_value_minus1[SchedSelIdx - 1]. The bit rate in bits per second is given by

$$BitRate[SchedSelIdx] = (bit_rate_value_minus1[SchedSelIdx] + 1) * 2^{(6 + bit_rate_scale)} \quad \text{(E-37)}$$

When the bit_rate_value_minus1[SchedSelIdx] syntax element is not present, the value of BitRate[SchedSelIdx] shall be inferred as follows.

– If profile_idc is equal to 66, 77, or 88, BitRate[SchedSelIdx] shall be inferred to be equal to 1000 * MaxBR for VCL HRD parameters and to be equal to 1200 * MaxBR for NAL HRD parameters, where MaxBR is specified in subclause A.3.1.

– Otherwise, BitRate[SchedSelIdx] shall be inferred to be equal to cpbBrVclFactor * MaxBR for VCL HRD parameters and to be equal to cpbBrNalFactor * MaxBR for NAL HRD parameters, where cpbBrVclFactor, cpbBrNalFactor, and MaxBR are specified in subclause A.3.3.

cpb_size_value_minus1[SchedSelIdx **]** is used together with cpb_size_scale to specify the SchedSelIdx-th CPB size. cpb_size_value_minus1[SchedSelIdx] shall be in the range of 0 to $2^{32} - 2$, inclusive. For any SchedSelIdx greater than 0, cpb_size_value_minus1[SchedSelIdx] shall be less than or equal to cpb_size_value_minus1[SchedSelIdx -1].

The CPB size in bits is given by

$$CpbSize[SchedSelIdx] = (cpb_size_value_minus1[SchedSelIdx] + 1) * 2^{(4 + cpb_size_scale)} \quad \text{(E-38)}$$

When the cpb_size_value_minus1[SchedSelIdx] syntax element is not present, the value of CpbSize[SchedSelIdx] shall be inferred as follows.

– If profile_idc is equal to 66, 77, or 88, CpbSize[SchedSelIdx] shall be inferred to be equal to 1000 * MaxCPB for VCL HRD parameters and to be equal to 1200 * MaxCPB for NAL HRD parameters, where MaxCPB is specified in subclause A.3.1.

– Otherwise, CpbSize[SchedSelIdx] shall be inferred to be equal to cpbBrVclFactor * MaxCPB for VCL HRD parameters and to be equal to cpbBrNalFactor * MaxCPB for NAL HRD parameters, where cpbBrVclFactor, cpbBrNalFactor, and MaxCPB are specified in subclause A.3.3.

cbr_flag[SchedSelIdx **]** equal to 0 specifies that to decode this bitstream by the HRD using the SchedSelIdx-th CPB specification, the hypothetical stream delivery scheduler (HSS) operates in an intermittent bit rate mode. cbr_flag[SchedSelIdx] equal to 1 specifies that the HSS operates in a constant bit rate (CBR) mode. When the cbr_flag[SchedSelIdx] syntax element is not present, the value of cbr_flag shall be inferred to be equal to 0.

initial_cpb_removal_delay_length_minus1 specifies the length in bits of the initial_cpb_removal_delay[SchedSelIdx] and initial_cpb_removal_delay_offset[SchedSelIdx] syntax elements of the buffering period SEI message. The length of initial_cpb_removal_delay[SchedSelIdx] and of initial_cpb_removal_delay_offset[SchedSelIdx] is initial_cpb_removal_delay_length_minus1 + 1. When the initial_cpb_removal_delay_length_minus1 syntax element is present in more than one hrd_parameters( ) syntax structure within the VUI parameters syntax structure, the value of the initial_cpb_removal_delay_length_minus1 parameters shall be equal in both hrd_parameters( ) syntax structures. When the initial_cpb_removal_delay_length_minus1 syntax element is not present, it shall be inferred to be equal to 23.

cpb_removal_delay_length_minus1 specifies the length in bits of the cpb_removal_delay syntax element. The length of the cpb_removal_delay syntax element of the picture timing SEI message is cpb_removal_delay_length_minus1 + 1. When the cpb_removal_delay_length_minus1 syntax element is present in more than one hrd_parameters( ) syntax structure within the VUI parameters syntax structure, the value of the cpb_removal_delay_length_minus1 parameters shall be equal in both hrd_parameters( ) syntax structures. When the cpb_removal_delay_length_minus1 syntax element is not present, it shall be inferred to be equal to 23.

dpb_output_delay_length_minus1 specifies the length in bits of the dpb_output_delay syntax element. The length of the dpb_output_delay syntax element of the picture timing SEI message is dpb_output_delay_length_minus1 + 1. When the dpb_output_delay_length_minus1 syntax element is present in more than one hrd_parameters( ) syntax structure within the VUI parameters syntax structure, the value of the dpb_output_delay_length_minus1 parameters shall be equal in both hrd_parameters( ) syntax structures. When the dpb_output_delay_length_minus1 syntax element is not present, it shall be inferred to be equal to 23.

time_offset_length greater than 0 specifies the length in bits of the time_offset syntax element. time_offset_length equal to 0 specifies that the time_offset syntax element is not present. When the time_offset_length syntax element is present in more than one hrd_parameters( ) syntax structure within the VUI parameters syntax structure, the value of the time_offset_length parameters shall be equal in both hrd_parameters( ) syntax structures. When the time_offset_length syntax element is not present, it shall be inferred to be equal to 24.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2005

ITU - Telecommunications Standardization Sector
STUDY GROUP 16 Question 6
Video Coding Experts Group (VCEG)
36th Meeting: San Diego, USA, 8-10 October, 2008

Document VCEG-AJ21
Filename: VCEG-AJ21.doc

Question: Q.6/SG16 (VCEG)

Source: Jaeil Kim, Taeyoung Na, Changhee Kim, Bumshik Lee, Munchurl Kim
Information and Communications Univ.
119 Munji street, Yuseong-Gu, Daejeon, Korea
Tel: +82-42-866-6246
Email: jaeil1203@icu.ac.kr
tyna@icu.ac.kr
changhee21c@icu.ac.kr
bslee@icu.ac.kr
mkim@icu.ac.kr

Changseob Park, Sangjin Hahm, Injoon Cho
Korean Broadcasting System
11 Yoidodong, Youngdeungpo-Gu, Seoul, Korea
Tel: +82-2-781-5963
Email: changseob@kbs.co.kr
caslw@kbs.co.kr
sylphide@kbs.co.kr

Title: **Enlarging MB size for high fidelity video coding beyond HD**

Purpose: Proposal

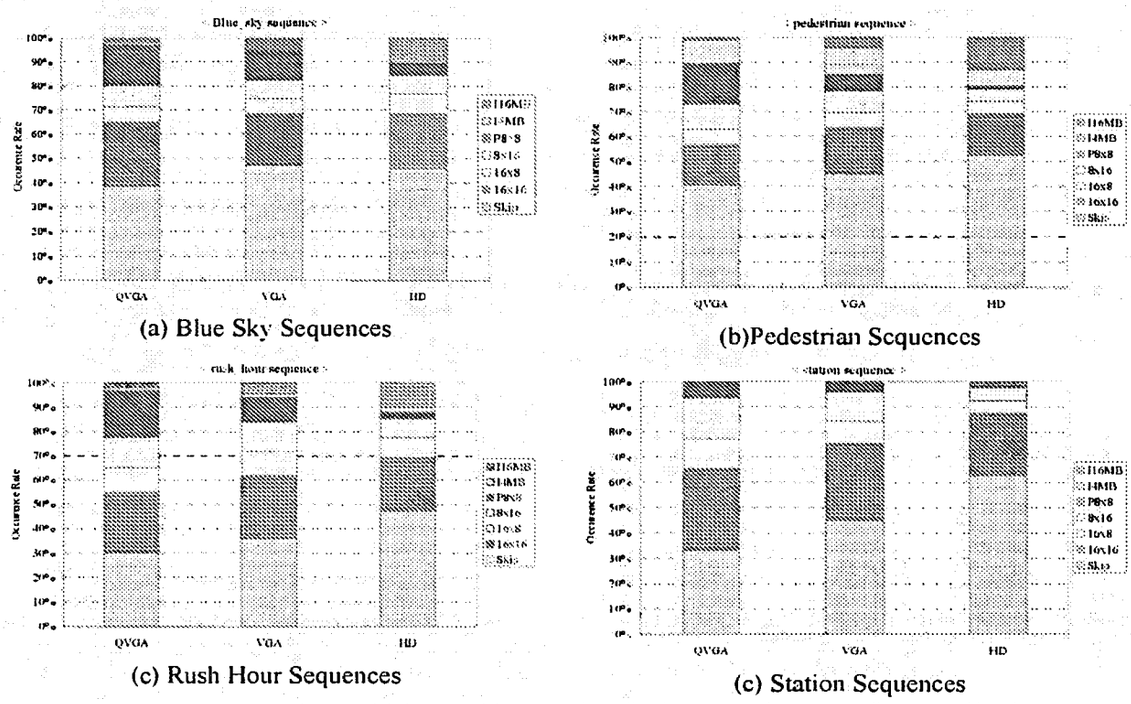
Abstract

In this contribution, we propose to extend the architecture of H.264 codec with enlarged MB sizes. Most of video codecs have lived long with MB based coding, which is now limiting to the coding efficiency toward the video coding of beyond High Definition (HD) resolutions. Even if the High profile of H.264, called FREXt tools, for high bit rate contents has been developed, it still maintains the size of the MB to be 16x16 for ME and MC based predictive coding as well as 16x16 intra coding. To limit the MB size into 16x16 might be one of the factors to restrict the RD performance in the high resolution video. Thus, we attempt to improve the coding efficiency of H.264 by enlarging MB size to the maximum 128x128. This attempt has been made with the extension to JM reference software. For this, the followings are to be newly defined, redefined or extended: new MB mode types for enlarged MB, CBP for new MBs or block indexing etc. In this proposal, the extended architecture with the enlarged MB structure are first proposed for ME&MC but new designs of transform kernels, intra prediction, luma/chroma DC transforms, CABAC context models and de-blocking filters for the enlarged MB are not yet addressed.

1. Introduction

The demand for HD or beyond HD such as ultra high definition videos of 4k and 8k has been increasing with increased network bandwidth and developed display technology. However, the current video encoder has much limitation to support HD or beyond HD video contents. Since the characteristics of the video signals depends heavily upon spatial resolutions, it is necessary to devise an appropriate video coding architecture to reflect the video characteristics with different sizes, especially the high resolution video contents. The High profile of H.264/AVC provides several tools such as 8x8 integer transform, 8x8 intra prediction and 8 to 12 bit-depth processing etc., which allows for better RD performance for high resolution video, for example, SD or HD video contents. Nevertheless, the maximum MB size for ME and MC is maintained with 16x16 pixels. It may not be efficient for the HD or beyond HD since the MB size of 16x16 pixels is relatively small for high resolution video contents. Fig.1 shows the relative portions of the final block mode selections according to the spatial resolutions of different video contents. These statistics of block mode distributions have been collected from the experiments with H.264

reference software, Joint Model ver. 12.0, for *Blue Sky*, *Pedestrian*, *Rush Hour* and *Station* sequences in QVGA, VGA and HD resolutions. As shown Fig.1, it can be noticed that the portions of the large block modes such as SKIP and 16x16 modes increase in proportion to the video resolution sizes. The mode selection trend shows consistent results for all the sequences with different resolutions and different QP values.



(a) Blue Sky Sequences (b)Pedestrian Sequences

(c) Rush Hour Sequences (c) Station Sequences

Fig. 1 Block mode distributions versus spatial resolutions of video

On the other hands, the proportions of the block modes in small spatial resolutions such as P8x8 and I4x4 significantly are reduced, especially for the HD sequences. It implies that relatively large block modes are more advantageous to the high resolution videos in terms of the Rate-distortion optimization. Motivated from these observations, we propose to enlarge the macroblock (MB) of 16x16 for HD or beyond HD video sequences.

Recently, similar works have been introduced in [1-4], and Ma *et. al.* has analyzed the advantage of larger MB with a 16x16 interger transform, a modified RD model and enlarged search ranges for ME in terms of RDO costs for high resolution videos. In this contribution, we attempt to extend with enlarged MB sizes the JM reference software architecture such that the fair comparison and backward compatibility can be supported. Furthermore, the extensions to the current syntax and the required side information for enlarged MB sizes are addressed. In Section 2, how to design the enlarged MB are first explained. Its subsequent works such as new mode definition, CBP for newly enlarged MB's, block indexing and deblocking filters are described.

2. Proposed Architecture

In order to enlarge the MB size, the extensions to H.264 (JM) should be made with a new definition of the added block modes for mb_type, the modification of the CBP definition and block indexing. Fig.2 shows a subsequent chain of the H.264 (JM) elements which are affected by enlarging the MB size. Enlarging the MB size entails the necessity of redefining block indexes of sub-blocks. Subsequently, motion estimation and mode decision are conducted based on the newly numbered block indexing. After the

mode decision is completed, entropy coding is performed. Finally deblocking filters with the block indexing modified are applied.

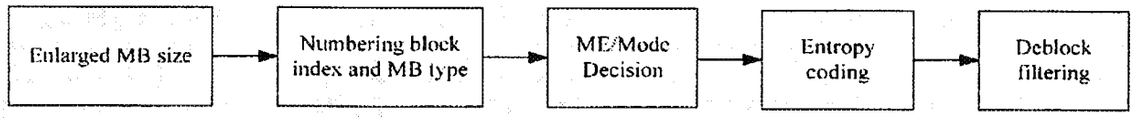


Fig. 2 H.264 (JM) elements affected by the enlarged MB's

Table 1 lists the H.264 elements that must be modified by enlarging the MB sizes. As aforementioned, this is not the exhaustive list. In Table 1, *N* indicates the maximum pixel size for the enlarged MB and can be increased with power of 2 such as 32, 64 or 128.

Table 1. H.264 elements to be modified for enlarged MB's

	H.264/AVC	Proposed enlarged MB
MB size	16x16	NxN
ME size	16x16 ~ 4x4	NxN ~ 4x4
Integer transform	4x4	4x4
CBP computation	8x8	(N/2xN/2), ···, 8x8
The number of SKIP mode	1	$\log_2 N - 3$
Blocking indexing	Fig.4	Fig.5
The number of mode for Inter	8	$4 \cdot \log_2 N - 8$

2.1 Hierarchical block partition structures for enlarged MB's

The proposed block modes with enlarged MB are depicted in Fig. 3. As shown in Fig.3, the MB mode partitions constitute a hierarchical structure in a top-down manner. For example, if the maximum MB size is set to 64x64, the resulting number of the partition levels is 4. The sub-block of 32x32 size constitutes the 2nd level hierarchy which can be partitioned further down to 16x16. Similarly, the 4th level hierarchy ends up with 8x8 block mode. Thus, for the case of 64x64 MB, 13 block types and 3 SKIP modes for 64x64, 32x32 and 16x16 block size can be possible. The number of block modes depends on the maximum MB size as shown in Table 1.

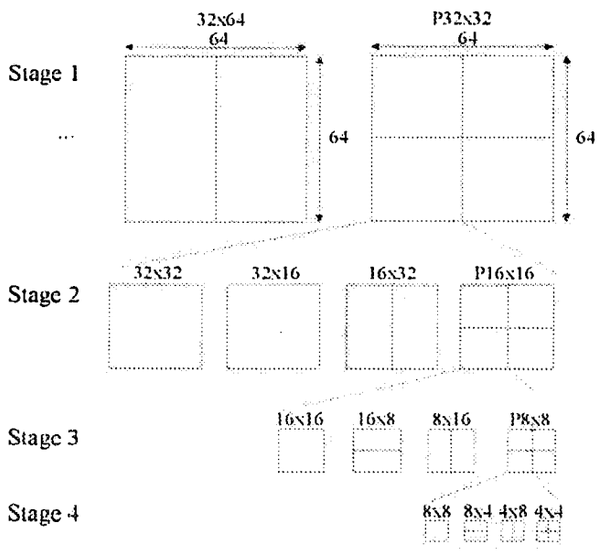


Fig. 3 An example of hierarchical block partitions for enlarged MB of 64x64

2.2 block modes for my_type

The mb_type in the MB syntax of the current H.264 is defined for the MB size of 16x16 and its partitions. The small numbers (0, 1) of mb_type are assigned for the large block modes (SKIP, Inter_16x16) while large numbers (5, 6, 7) are assigned for the block modes (Inter_8x4, Inter_4x8, Inter_4x4) of relatively small sizes. Similarly, in the proposed enlarged MB, the numbering of mb_type should be changed by assigning small numbers to the large block modes such that total bit amounts to represent mb_type become efficient. The mb_type according to the enlarged MB size is defined in Table 2.

Table 2. Possible partition block modes (mb_type) for enlarged MB's of 128x128, 64x64 and 32x32

128x128 MB size	SKIP_128x128(0), 128x128(1), 128x64(2), 64x128(3), SKIP_64x64(4), 64x64(5), 64x32(6), 32x64(7), SKIP_32x32(8), 32x32(9), 32x16(10), 16x32(11), SKIP_16x16(12), 16x16(13), 16x8(14), 8x16(15), 8x8(16), 8x4(17), 4x8(18), 4x4(19),
64x64 MB size	SKIP_64x64(0), 64x64(1), 64x32(2), 32x64(3), SKIP_ 32x32(4), 32x32(5), 32x16(6), 16x32(7), SKIP_16x16(8), 16x16(9), 16x8(10), 8x16(11), 8x8(12), 8x4(13), 4x8(14), 4x4(15),
32x32 MB size	SKIP_32x32(0), 32x32(1), 32x16(2), 16x32(3), SKIP_16x16(4), 16x16(5), 16x8(6), 8x16(7), 8x8(8), 8x4(9), 4x8(10), 4x4(11),

If an enlarged MB size is NxN, '0' is assigned to mb_type for the SKIP_NxN block mode. Hence, the largest number of mb_type is assigned to the smallest block size whose occurrence rate is the smallest among the other block modes.

2.3 Block indexing

As the MB size is enlarged, the block indexing for sub-MB and 4x4 integer transform should be modified. Fig.4 shows the block indexing for the MB of 16x16 size in the 4:2:0 YCbCr format. Once the sub-blocks are partitioned, 4x4 blocks are indexed inside one 8x8 block. However, for an enlarged MB size, the block indexing for the sub-MBs are changed accordingly. Fig. 5 shows an example of the block indexing for the MB size of 32x32. The block indexing for the enlarged MB is numbered hierarchically in a top-down manner. The encoding order according to the proposed block indexing is also shown in Fig.5.

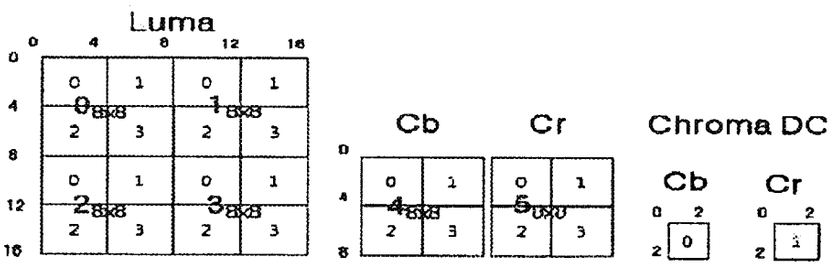


Fig. 4 Block indexing for 16x16 MB

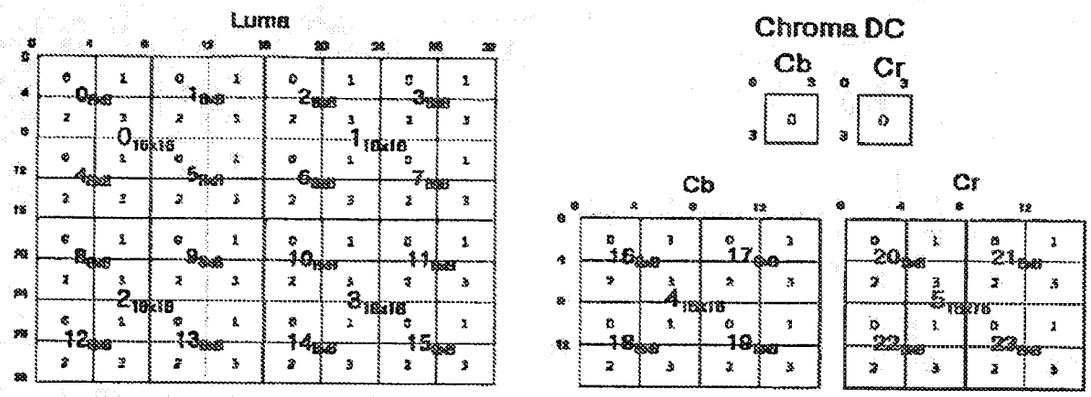


Fig. 5 Block indexing for 32x32 MB

2.4 New SKIP modes in each mode stage.

For the enlarged MB sizes, new SKIP modes are added to the block modes. The total number of SKIP modes in the current H.264 is only one for 16x16 block size while the number of SKIP mode in the enlarged MB depends on the MB size N. Since the block modes are constructed in a hierarchical tree structure, several SKIP modes may be added when the shapes of the block modes are square, for example, 32x32, 64x64, and 128x128 etc. The added SKIP modes can play an important role of enhancing the coding efficiency for high resolution video contents. By incorporating one SKIP mode into each squared block partition, the decoding syntax might be accordingly modified such that the SKIP mode is firstly checked at the beginning of parsing each squared block partition [3]. Fig. 6 shows an example of syntax parsing for enlarged MB with SKIP modes in each level of squared block modes.

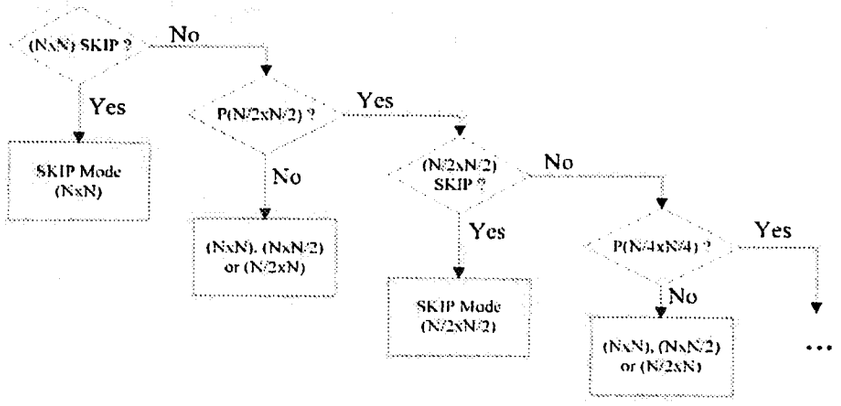


Fig. 6 MB syntax for the enlarged MB

2.5 The CBP value

The CBP value for the 16x16 MB can signal the information about all zero blocks based on 8x8 block in H.264. This is directly extended in a hierarchical manner to an enlarged MB where CBP_EMB (CBP for Enlarged MB) is newly defined to signal the context of quantized IT coefficients based on the NxN MB size. Fig. 7 shows an example of CBP_EMB for the enlarged MB of 32x32 size.

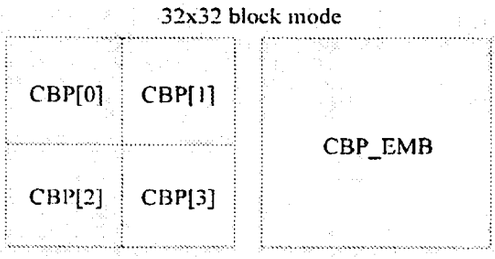


Fig. 7 CBP_EMB for 32x32 MB and CBPs for 16x16 blocks

2.6 Signaling issue for MB size information

An optimal way of signaling MB size information must be investigated by considering MB, Slice, SPS or PPS level.

3 Conclusion

We propose to extend the H.264 with an enlarged MB scheme for high resolution video coding such HD and beyond HD sequences. The statistics show that the enlarged MB can provide enhanced coding efficiency for high resolution video coding. This extension is definitely beneficial to the long-term living of H.264 for high resolution video coding beyond HD resolutions.

For enlarging MB size, the number of block partitions increases accordingly. This may cause to reduce the coding efficiency in terms of signaling the block mode information due to increase number of block partitions. Since the motivation of enlarging the MB size is targeted for high resolution video coding, the number of block modes can be reduced by eliminating the block partitions of small sizes which are rarely selected. This might be a reasonable approach because the interoperability between a very small resolution (QCIF) video and a very large resolution (Ultra HD) video may not be necessary in a single video codec with their application domains far separated.

4 References

[1] S. Naito and A. Koike, "Efficient coding scheme for super high definition video based on extending h.264 high profile," Visual Communications and Image Processing (VCIP) 2006, Jan 2006, San Jose, CA, USA

[2] Siwei Ma and C.-C. Jay Kuo, "High-definition Video Coding with Super-macroblocks," Visual Communications and Image Processing 2007, Sep 16-19, 2007, San Antonio, USA.

[3] Jae-Chool Lee, Jianle Chen, Kyo-Hyuk Lee, and Woo-Jin Han, "Hierarchical Block Mode Structure Extension for Video Coding," Conference 2008 on the Korean Society of Broadcast Engineers, p73-74, 2008

[4] Shun-ichi Sekiguchi, Shuichi Yamagishi, Yusuke Itani, Kohtaro Asai, Tokumichi Murakami., "On motion block size and partition for 4:4:4 video coding", VCEG-AI31, July 2008, Berlin, Germany

IPR Notification: ICU and KBS may have IPR relating to the technology described this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG2008/M15580
July 2008, Hannover, Germany

Title: Technical considerations for Ad Hoc Group on New Challenges in Video Coding Standardization

Purpose: Informative

Authors: Kyohyuk Lee (kyohyuk.lee@samsung.com),
Elena Alshina (elena_a.alshina@samsung.com),
Jeonghoon Park (jeonghoon@samsung.com)
Woojin Han (wjhan.han@samsung.com),
Junghye Min (jh643@samsung.com),

Source: Samsung Electronics Co., Ltd.

Summary

In this contribution, we investigated several aspects of video coding technologies under consideration of new applications, especially with high resolution video contents. As the results of the investigation, we obtained over 20% bit savings with high resolution video sequences compared to the state of the art standard video codec, MPEG-4 AVC High Profile. In the specific test sequence, over 30% bit rate reduction is achieved. Our results imply that there still is enough room for further improvement of the latest standard video codec for high resolution video applications.

Many companies have already developed devices for high resolution video applications over HD resolution. The new video codec standard especially targeting these applications could accelerate the growth of the market of these kinds of applications. Therefore we propose starting of investigation on new video coding standardization as soon as possible under consideration of high resolution video applications.

1. Introduction

In the past decade, high resolution video contents have widely prevailed from internet to broadcasting. For example, DVD applications with SD resolution have already dominated the video contents rental market. Also, DTV technologies with HD resolution not only provide broadcasting service but also try to expand their feasibility towards the mobile market. And besides, resolution and size of display devices tend to increase in the recent consumer electronics market. As we saw in the CES 2008, most major display providers have already introduced 80" LCD and 150" PDP devices. Furthermore, video

capturing devices of 4Kx2K video contents are already in the stage of commercial use meeting the growing demand in digital cinema market. Additionally, commercial projectors already cover 4Kx2K resolution. Even though the current main stream of media contents is HD resolution video, it is quiet clear that the resolution of video contents will expand over HD resolution in the future.

Also, due to the startling development of transmission and storage technologies, quality of video contents is constantly improving. For example, in the case of packaged media applications, Blue Ray exceeds DVD in data capacity. Besides the packaged media application, it is self-evident that all video content services will be inclined to high quality video in the future environment.

For this reason, we investigated several aspects of video coding tools for future video applications.

2. Consideration of video coding for new application domain

2.1. Expanded Prediction Block Unit

It is well-known that in terms of coding efficiency, large motion block size becomes more efficient as image resolution increases. The largest motion block size of state of art video codec, such as MPEG-4 AVC, is 16x16. Additionally, both the motion block for motion compensation and the intra prediction block size are bounded up to 16x16.

We investigated the usefulness of changing prediction block size by expanding the prediction block size over 16x16 to cope with the high resolution image compression, and each separated block has its own displacement vector for inter prediction as conventional video codec.

For the case of intra prediction, most part of the images with high resolution can be described as homogenous regions with variations such as the gradation and directional pattern extension. As the prediction block size is expanded, the characteristics of those spatial variations in one prediction block became more dynamic and diverse. Therefore, conventional 9 directions for intra prediction defined in MPEG-4 AVC may not be enough to express various texture statistics with enlarged prediction block. Thus, we added more detailed directions for intra prediction expecting further performance improvement. So, we have tested directional intra prediction by increasing the number of directions over the conventional.

2.2. Transform size corresponding to Expanded Prediction Block Unit

To maximize the benefits of expanded prediction block units, transform should be designed appropriately. Larger transform has several advantages such as better energy compaction and less quantization error. In large resolution images, most of image patterns in prediction blocks represent small part of objects or backgrounds which can be described as homogeneous texture patterns with little variations. Hence, we tried applying large transforms greater than 8x8 for residual coding.

2.3. Adaptive interpolation filter

In recent years, numerous attempts were done to overcome the limit of coding efficiency with state of the art video coding. For example, many experts are trying to find new evidence for future video coding through KTA activity in VCEG [3]. According to the analysis report from Tandberg, 'adaptive interpolation filter' seems to have the most significant results in terms of coding efficiency [4]. Adaptive interpolation filter exploits image correlation dependent interpolation filter for motion compensation with fractional-pel displacement vector [5].

3. Experimental Results

3.1. Experimental Results without adaptive interpolation filter

We have tested video coding efficiency with the previously introduced methodologies except for adaptive interpolation filter.

Experimental conditions of estimation for coding efficiency were determined under consideration of recent recommendation of ITU-T VCEG [1] and all the numerical expression on coding efficiency is based on BD-PSNR [2]. Coding efficiency of this contribution is compared with state of the art video codec, MPEG-4 AVC. For fair comparison between these two systems, we used well-optimized MPEG-4 AVC encoder; relative Qp control with hierarchical B prediction structure, iterative bi-prediction motion estimation and etc.

Experiments were done with 17 test sequences consisting of various resolutions and video motion characteristics. Test video sequences are listed in Table 1.

Table 1 Test sequences, resolution and test frame numbers

Resolution	Video Sequence Name
CIF	Foreman
	Mobile
	Tempete
	Paris
4CIF	Harbour
	Soccer
	Stockholm
	Shield
HD 720p	BigShips
	City
	Crew
	ShuttleStart
HD 1080p	Rolling Tomatoes

	RushHour
	SunFlower
	PlayingCards
	TableSetting

Test configurations are as follows:

- Hierarchical B prediction structure (GOP size 8)
- One I slice with the first frame
- 4 Qp points for key picture : 22, 26, 30, 34
- CABAC entropy coding

Figure 1 shows average bit saving percentage with 4 Qp points. Total average of bit saving with all 17 test sequences and 4 Qp points is 13% furthermore average bit savings apparently become larger toward large resolution sequences such as HD 1080p.

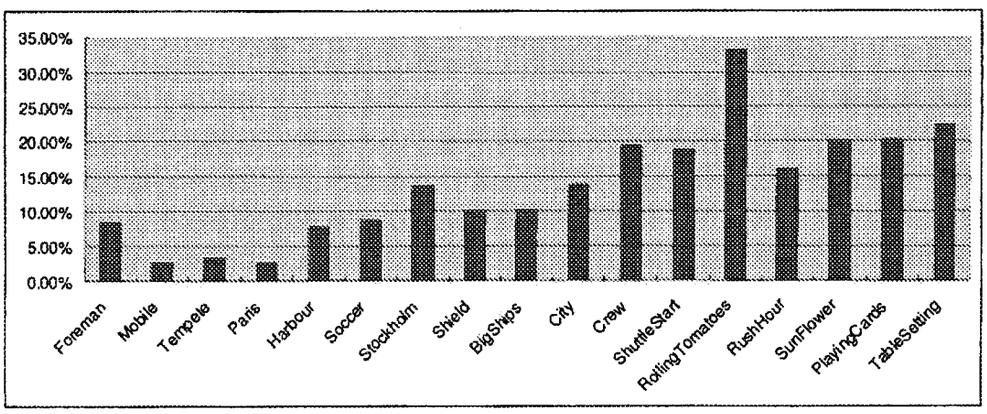


Figure 1 Average bit saving for each sequence

Figure 1 shows average bit saving for each video resolution. It is very easy to see that non-standard video coding tools, such as expanded prediction block unit and large transform, are well-suited for large resolution image compression. With these tools, coding efficiency improves further as the resolution rises. This tendency of coding efficiency might be favorable for the future video applications with high resolution. As we can see in Figure 1, a specific test sequence, RollingTomatoes, showed over 30% average bit saving.

Figure 2 shows the average bit savings for each resolution. As shown in the figure, in case of HD 1080p resolution, we could reach 22.36% average bit saving with 4 Qp points

while larger bit savings are expected if application requires video sequences having above HD resolutions.

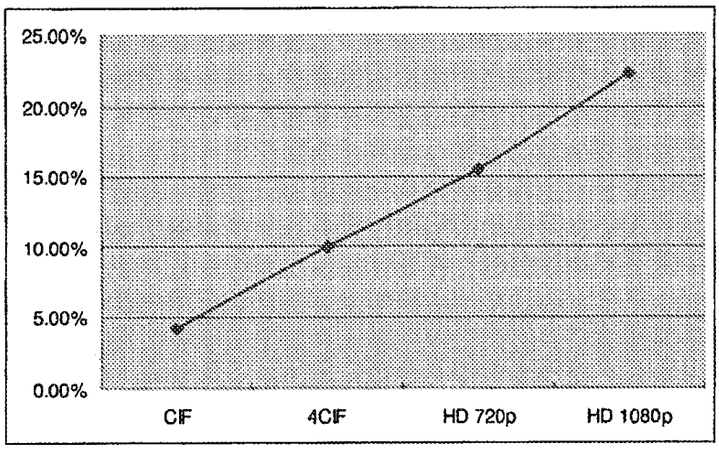
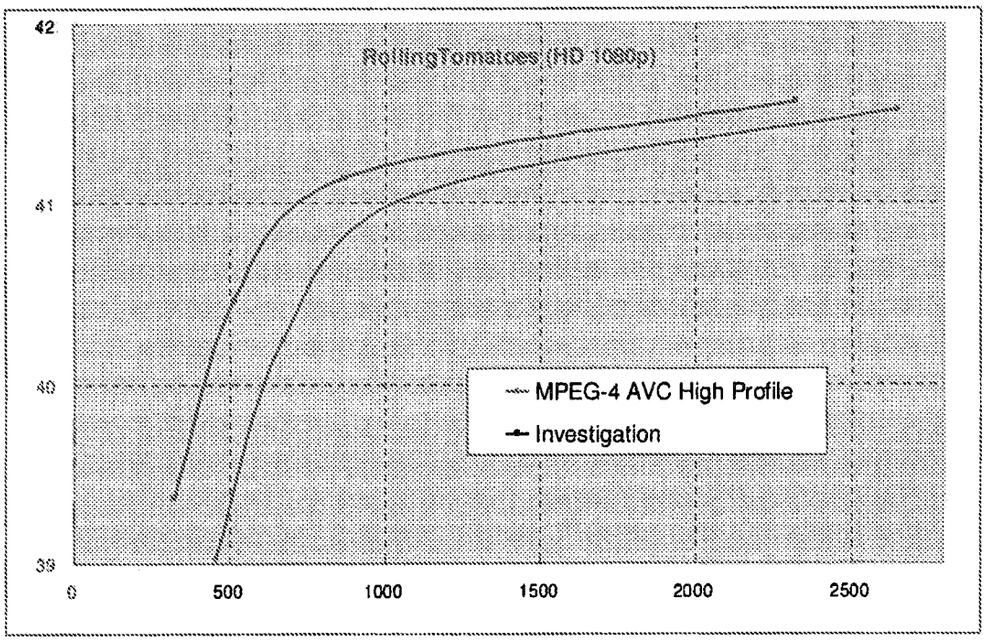
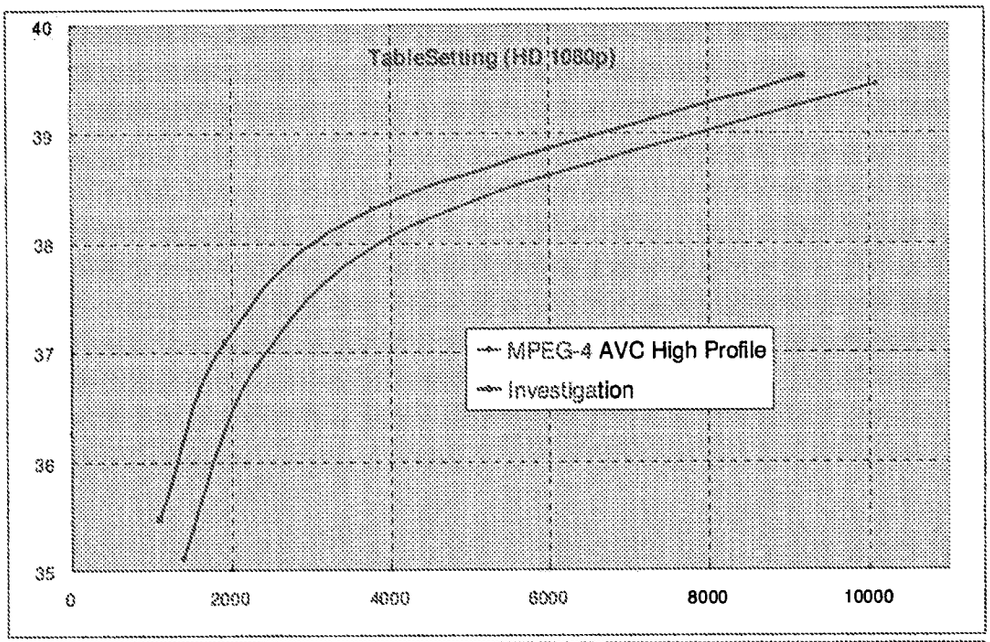
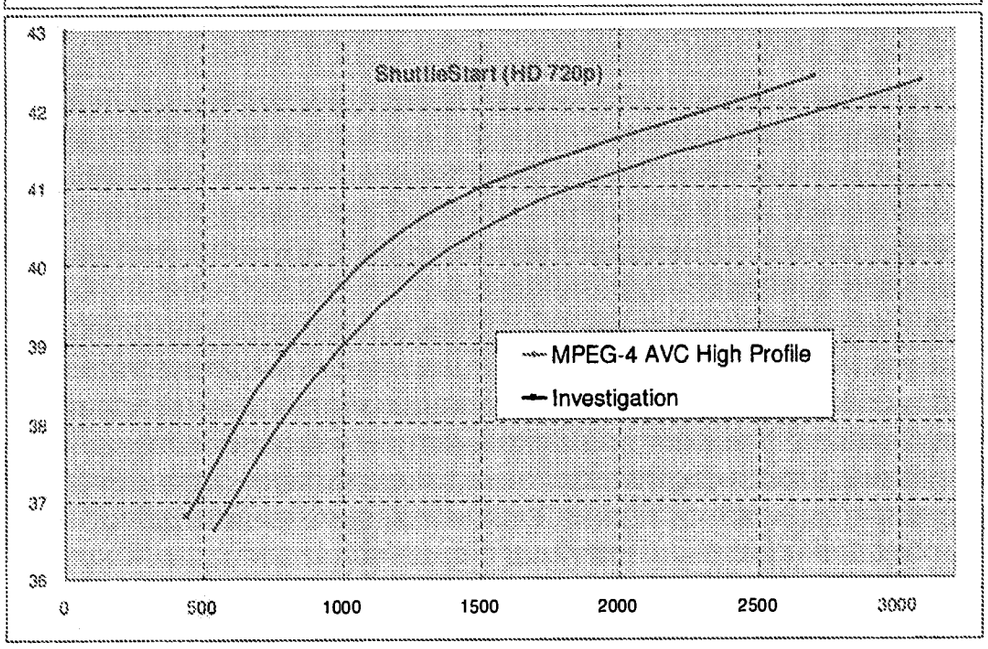


Figure 2 Average bit saving for each resolution.

The following figures show several RD curves of different HD resolutions. From these curves, it is evident that significant bit savings can be achieved from non-standard technologies described in previous sections.




TableSetting (HD 1080p)
40
39
38
37
36
35
0
2000
4000
6000
8000
10000
MPEG-4 AVC High Profile
Investigation


ShuttleStart (HD 720p)
43
42
41
40
39
38
37
36
0
500
1000
1500
2000
2500
3000
MPEG-4 AVC High Profile
Investigation

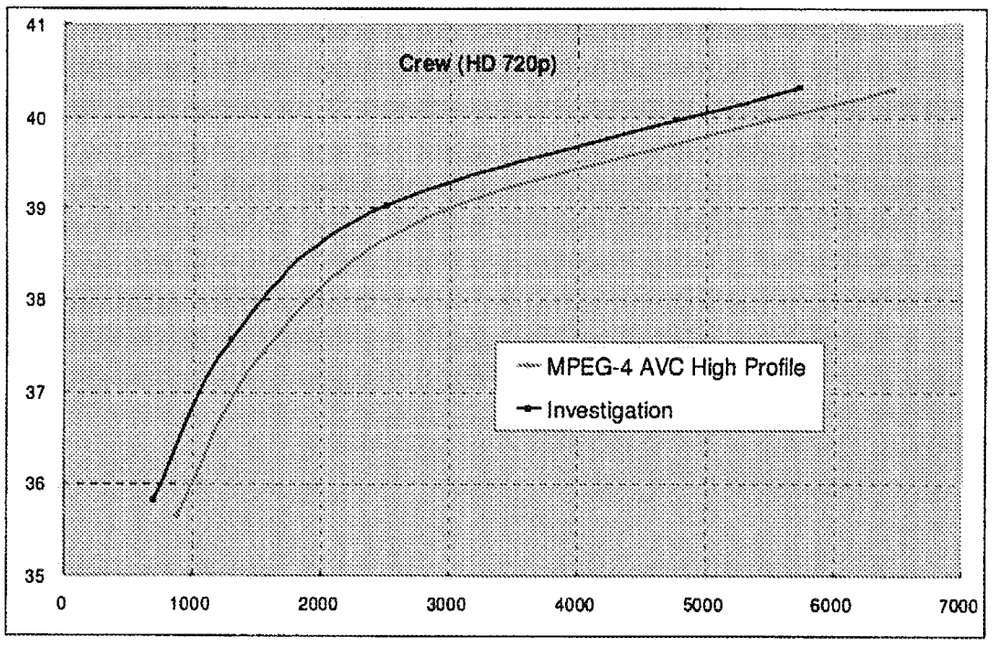


Figure 3 RD curve examples of HD resolution test sequences

3.2. Experimental Results with adaptive interpolation filter

We have tested video coding efficiency with the previously introduced methodologies including adaptive interpolation filter.

Figure 4 shows average bit saving percentage of each test sequence with 4 Qp points and an example of RD curve from HD 1080p resolution. Total average of bit saving with all 17 test sequences and 4 Qp points is 16.94%. Similar to the previous experiment, average bit savings apparently become larger toward large resolution sequences such as HD 1080p. In case of HD 1080p resolution, we could reach 26.32% average bit saving; additional bit saving percentage by adaptive interpolation filter is about 4%. Adaptive interpolation filter seems to work well especially in high bit rate range.

These results imply that we could reach much higher coding efficiency through well-integration of already known new video coding tools.

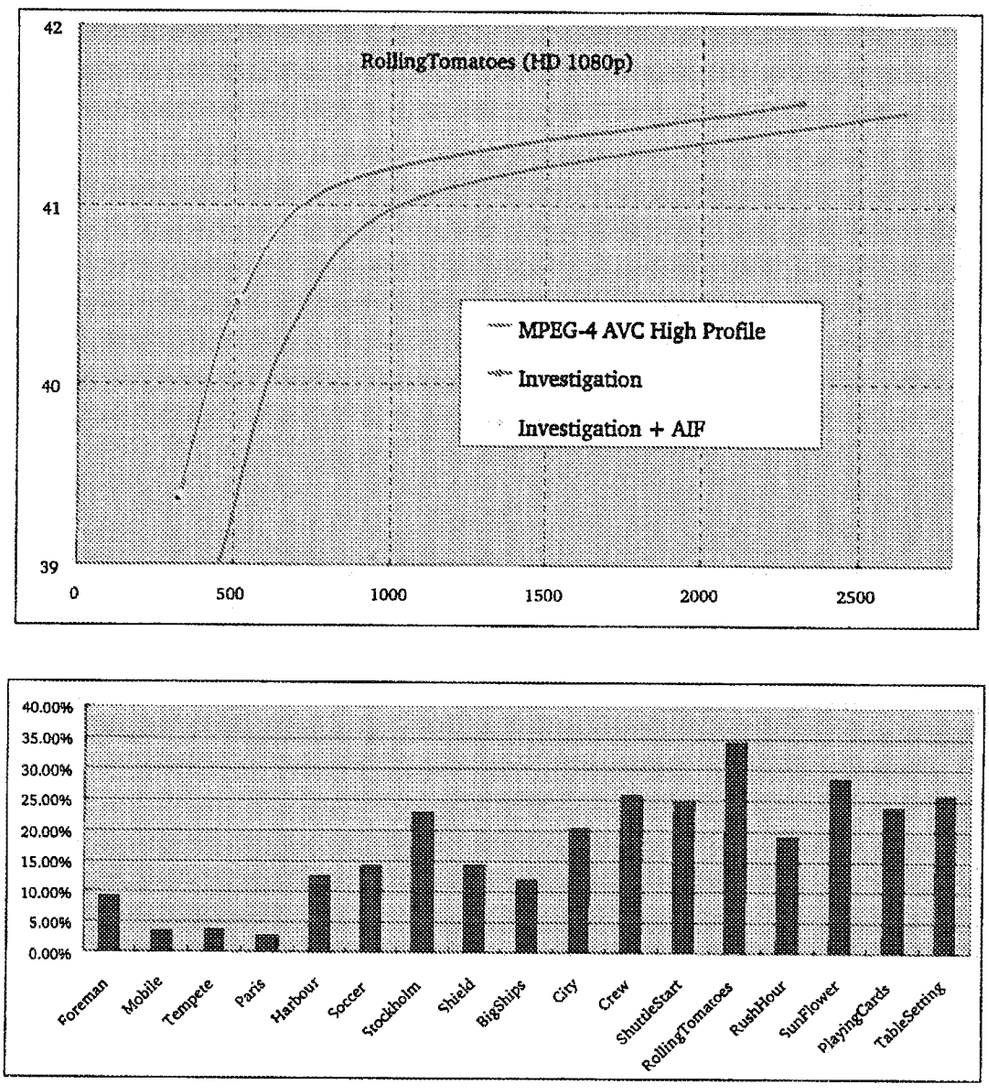


Figure 4 Average bit saving with introduced all techniques

4. Conclusion

From our experimental results, it was proven that it is possible to further improve video coding efficiency of the latest video codec standard for applications requiring high resolution video services. We would like to emphasize that it would be very worth to try launching new video coding standardization for future technical/commercial environments, especially under consideration of high resolution video applications. It could surely accelerate the growth of the market of high definition video services, which already have been prepared by many major companies.

References

[1] TK Tan, G. Sullivan, T. Wedi, "Recommended Simulation Conditions for Coding Efficiency Experiments Revision 2", ITU-T SG16/Q6 Document, ITU-T VCEG-AH010r3, Antalya, January 2008.

[2] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-curves", ITU-T SG16/Q6 Document, ITU-T VCEG-M33, March 2001

[3] http://www.tnt.uni-hannover.de/~vatis/kta/

[4] Tandberg, "A coding efficiency-computational complexity analysis of KTA 1.8 coding tools", ITU-T SG16/Q6 Document, COM 16 – C 409 – E, April 2008

[5] Yuri Vatis, Bernd Edler Dieu Thanh Nguyen and Jörn Ostermann, "Two-dimensional non-separable Adaptive Wiener Interpolation Filter for H.264/AVC", ITU-T SG16/Q6 Document, ITU-T VCEG-Z17, Busan, April 2005

ITU - Telecommunications Standardization Sector
STUDY GROUP 16 Question 6
Video Coding Experts Group (VCEG)
38th Meeting: London, UK / Geneva, CH, 1-8 July, 2009

Document VCEG-AL28
Filename: VCEG-AL28.doc

Question: Q.6/SG16 (VCEG)

Source: Sung-Chang Lim
Hahyun Lee
Jinho Lee
Jongho Kim
Haechul Choi
Seyoon Jeong
Jin Soo Choi

138 Gajeongno, Yuseong-gu, Daejeon,
Korea
ETRI

Tel: +82-42-860-1808
Email: choihc@etri.re.kr
jsy@etri.re.kr
jschoi@etri.re.kr

Title: **Intra coding using extended block size**

Purpose: Proposal

1. Abstract

Inter coding using extended block size [1] was already adopted in KTA. In this contribution, we focus on intra coding using extended block size, which allows intra_32x32 prediction as well as the existing intra predictions. Larger block transforms are also used and more directional prediction modes for intra_16x16 are added. Under VCEG common testing conditions except RDO-Q, the proposed method achieved 1.83% BD-RATE reduction on average over all test sequences for only intra coding.

2. Intra coding using extended block size

We introduce an intra coding using extended block size. The scheme of the intra coding using extended block size can be designed in a various way. First of all, we designed the scheme to be as similar to H.264 as possible, and then we changed a few tools to achieve more coding efficiency. In this contribution, intra coding using extended block size for only I-slice is considered, whereas intra coded block in P- or B-slices are excluded. Since inter coding method using extended block size was proposed and it mixed inter and intra blocks in an extended block such as 32x32, the harmonization and consistency between intra coding process of intra-frame and inter-frame are a quite complicate problem. This problem will be a challenge for further study.

Macroblock is extended from 16x16 to 32x32, which is so-called super macroblock (SMB). An SMB in I-slice can be coded as either one 32x32 block or four 16x16 blocks, for which 1 bit flag, *intra_smb_flag*, is signaled per SMB. When *intra_smb_flag* is equal to 1, 32x32 block is coded as Intra_32x32 with new syntaxes. If *intra_smb_flag* is equal to 0, each 16x16 block is coded in a same way of H.264 in raster scan order whithin 32x32 block. Each 16x16 block within an SMB has individual mb_type so that it can be coded Intra_16x16, Intra_8x8, or Intra_4x4 as Figure 1.

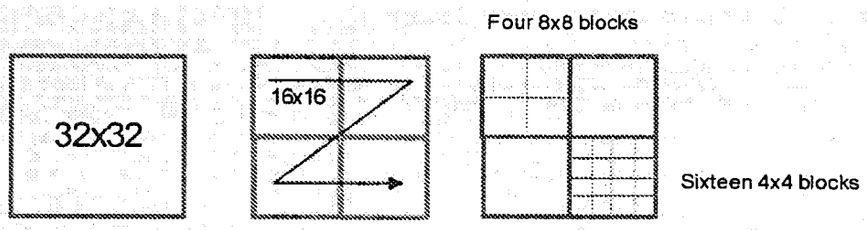


Figure 1. Supported macroblock partitions of intra coding using extended block size

2.1. New Intra_32x32 prediction mode

The Intra_32x32 is coded as follows:

- For Intra_32x32, mb_type32 is introduced in a same way of Intra_16x16 in H.264
- The mb_type32 for Intra_32x32 specifies intra 32x32 prediction mode and values of CodedBlockPatternLuma_32 and CodedBlockPatternChroma_32.
- The Intra_32x32 can be coded as one among four directional prediction modes just like the Intra_16x16 of H.264. Plane prediction mode is revised to be appropriate for 32x32 block size.
- For luma of Intra_32x32, the 8x8 transform of H.264 is used for 32x32 luma block and then 4x4 hadamard transform is applied in a similar way of Intra_16x16 of H.264
- For chroma of Intra_32x32, the 8x8 transform of H.264 is used for 16x16 associated chroma block and then 2x2 hadamard transform is applied in a similar way Intra_16x16 of H.264.

Table 1. Prediction partition size and transforms used for each intra prediction mode (4:2:0 format).

Macroblock partition prediction mode	Luma transform	Chroma transform	Coded Block Pattern
Intra_32x32	8x8 integer cosine transform, and then 4x4 hadamard transform	8x8 integer cosine transform, and then 2x2 hadamard transform	Included in mb_type32
Modified Intra_16x16	16x16 integer cosine transform in KTA 2.3 [1]	4x4 integer cosine transform, and then 2x2 hadamard transform	Luma: 1bit same as cbp16 of KTA2.3 Chroma: 2bit same as H.264
Intra_8x8	8x8 integer cosine transform	4x4 integer cosine transform, and then 2x2 hadamard transform	Luma: 4bit Chroma: 2bit
Intra_4x4	4x4 integer cosine transform	4x4 integer cosine transform, and then 2x2 hadamard transform	Luma: 4bit Chroma: 2bit

2.2. Modified Intra_16x16 prediction mode

For a video with increased resolution and frame rate, Intra_16x16 prediction has probably a higher chance to be used. Also, four Intra_16x16 directional prediction modes may not be an enough number of modes to predict a 16x16 block. Therefore, we extend the number of modes to 9 like Intra_4x4 and Intra_8x8 of H.264 and introduce Most Probable Mode for the modified Intra_16x16 mode (MPM_16).

The modified Intra_16x16 is coded as follows

- All of mb_type for Intra_16x16, which is in the range of 1 to 24 in H.264, are changed like Intra_8x8 and Intra_4x4. The only one mb_type value for specifying Intra_16x16 is assigned. Moreover, intra_16x16_prediction_mode, 1 bit luma CBP (cbp16), and 2bit chroma CBP are signaled individually as flags.
- The 16x16 integer transform used for 16x16 inter coded block of KTA 2.3 is also used for luma 16x16 block.
- The intra_16x16_prediction_mode is predicted from the MPM_16.

3. Experimental results

Under the recommended simulation common condition [2] except RDO-Q, the performance of intra coding using extended block size was preliminarily evaluated. The reason for the exclusion of RDO-Q is that we couldn't yet modify the RDO-Q due to time limitation. Table 3 shows the experimental result for KTA sequences. As Table 2, average 1.83% and up to 5.76% of bit saving is achieved under the KTA QP range, from 22 to 37. Since the Intra_32x32 would have more effect on a homogeneous region, we further experimented for a lower bitrate range. As expected, average 5.19% and up to 17.78% of bit saving is achieved when QP ranges from 32 to 47. To analyze the impact of the proposed method on HD sequences, the further experimental results for other HD sequences are listed in Table 3.

Table 2. Experimental result for KTA sequences under the recommended simulation common condition except RDO-Q.

Sequence		BD-RATE (%)		
		QP: 22~37	QP: 27~42	QP: 32~47
CIF	Paris	-0.24	-0.52	-1.09
	Foreman	-0.70	-1.97	-4.51
	Mobile&Calendar	-0.48	-0.72	-1.13
	Tempete	-0.63	-0.93	-1.66
	CIF average	**-0.51**	**-1.03**	**-2.10**
WVGA	Flower4	-1.74	-2.69	-4.98
	Keiba3	-1.13	-2.35	-4.82
	Nuts5	-5.26	-9.75	-17.78
	WVGA average	**-2.71**	**-4.93**	**-9.19**
XGA	Janine	-1.68	-2.10	-2.98
720p	BigShips	-0.88	-1.79	-4.15
	City	-0.92	-1.77	-3.36
	Crew	-3.48	-6.30	-10.92
	Jets	-5.76	-8.84	-12.01
	Night	-0.96	-1.56	-2.64
	Raven	-1.76	-3.29	-6.47
	720p average	**-2.29**	**-3.93**	**-6.59**
1080p	CrowdRun	-0.99	-1.28	-2.09
	ParkJoy	-0.86	-1.15	-2.03
	Traffic	-1.14	-1.83	-3.30
	Toys&Calendar	-1.83	-2.99	-5.34
	Sunflower	-4.25	-5.83	-7.44
	1080p average	**-1.81**	**-2.62**	**-4.04**
Total Average		**-1.83**	**-3.04**	**-5.19**

Table 3. Experimental result for other HD sequences under the recommended simulation common condition except RDO-Q

Sequence		BD-RATE (%)		
		QP: 22~37	QP: 27~42	QP: 32~47
720p	ShuttleStart	-2.64	-4.99	-9.17
1080p	blue_sky	-1.73	-2.52	-4.15
	pedestrian_area	-2.65	-4.41	-7.81
	Riverbed	-0.79	-2.15	-4.05
	rush_hour	-3.96	-6.97	-11.11
	Station	-2.27	-4.84	-10.27
	Tractor	-0.14	-1.37	-3.58
	rolling_tomatoes	-9.17	-16.15	-23.24
Total average		**-2.92**	**-5.43**	**-9.17**

4. Conclusion

In this contribution, we introduced intra coding using extended block size, which is designed toward the similarity to H.264 and consistency with inter extended block size coding [1]. Although the performance of the proposed method is not much high, we expect that there are a lot of chances to be enhanced by a coupling of MDDT [3], other intra prediction mode, and syntax design, etc. Therefore, we recommend the proposed Intra SMB to be adopted into KTA S/W in order to help a development on video coding tools using extended block size.

References

[1] P. Chen, Y. Ye, and M. Karczewicz, "Video coding using extended block sizes," VCEG, COM 16 – C 123 – E, January 2009.

[2] T.K. Tan, G. Sullivan, and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 4," VCEG-AJ10r1, July 2008.

[3] Y. Ye, and M. Karczewicz, "Improved Intra Coding," VCEG, COM 16 – C 257 – E, June 2007.

IPR Statement:

ETRI may have IPR relating to the technology described this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T patent statement and licensing declaration form).

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas

Detlev Marpe and Thomas Wiegand
Fraunhofer Institute for Telecommunications -
Heinrich Hertz Institute (HHI)
Image Processing Department, Einsteinufer 37
D-10587 Berlin, Germany
{marpe,wiegand}@hhi.fraunhofer.de

Stephen Gordon
Broadcom Cooperation
Consumer Electronics Group
200 Brickstone Sq.
Andover, MA 01810, USA
sgordon@broadcom.com

***Abstract*—The technical work on the first amendment of the H.264/MPEG4-AVC video coding standard has recently been completed. In these so-called Fidelity Range Extensions (FREXt) a set of new coding tools is specified which is primarily targeted at providing significant improvements in coding efficiency for higher-fidelity video material. This paper presents an overview of the corresponding methods, briefly discusses some important aspects regarding profiles and applications, and finally provides experimental results for a performance comparison with existing coding technology.**

***Keywords*-H.264, MPEG4, Advanced Video Coding (AVC), Fidelity Range Extensions (FREXt), video coding, standards, JVT.**

I. INTRODUCTION

The technically aligned specification of ITU-T Recommendation H.264 and ISO/IEC MPEG4-AVC (Advanced Video Coding) [1], abbreviated as *H.264/MPEG4-AVC*, is the latest international video coding standard. As the result of the collective efforts of experts in the *Joint Video Team* (JVT) of ITU-T VCEG and ISO/IEC MPEG, H.264/MPEG4-AVC represents the most significant improvement in coding efficiency since the formerly and jointly developed video coding standard H.262/MPEG-2 Video [2]. Recently published results [3] indicate that H.264/MPEG4-AVC compliant encoders can achieve the same reproduction quality at half or less of the bit rate required by state-of-the-art MPEG-2 compliant encoders.

Technically, the design of the H.264/MPEG4-AVC video coding layer is based on the traditional hybrid concept of block-based motion-compensated prediction (MCP) and transform coding. Within this framework, a number of important innovative ideas have been developed. Some of these key features are given as follows (for more details, please refer to [4]).

- Enhanced MCP capabilities
- Multiple reference pictures and generalized B pictures
- Spatial intra prediction
- Small block-size transform in integer precision
- Context-adaptive in-loop deblocking filter
- Enhanced entropy coding methods

The first version of H.264/MPEG4-AVC, finalized and approved in 2003, was intended to be applied to a wide spectrum of applications ranging from mobile phones to digital cinema. By taking into account different requirements and limitations of typical target applications, three subsets of coding tools were defined as *Baseline*, *Extended*, and *Main* profile in the initial, i.e., 2003 approved H.264/MPEG4-AVC standard. In addition, a set of 15 *levels* is specified in Annex A of the first version of H.264/MPEG4-AVC in order to provide conformance points with certain constraints on parameters such as, e.g., picture size, bit rate or frame rate [1].

After completion of the technical work on version 1 of H.264/MPEG4-AVC in March 2003, the JVT issued a Call for Proposals for support of extended sample bit depth and chroma format in the H.264/MPEG4-AVC standard. This standardization initiative was motivated by the rapidly growing demand for coding of higher-fidelity video material, especially in application areas like professional film production, video post production, or high-definition TV/DVD. The technical work on these so-called *Fidelity Range Extensions* (FREXt) of H.264/MPEG4-AVC was completed in July 2004, and the corresponding final draft amendment text was released in September 2004 [5]. Apart from the new functionalities provided by the FREXt amendment, further improvements in coding efficiency relative to the first version of H.264/MPEG4-AVC were demonstrated by using parts of the FREXt tool set within application domains that were initially targeted by the H.264/MPEG4-AVC Main Profile.

In this paper, an overview of the main distinguishing features of the H.264/MPEG4-AVC FREXt amendment is given. The next section describes the new FREXt coding tools with a particular emphasis on the 8×8 intra prediction and 8×8 transform. Section III presents the new profile definitions for FREXt and highlights some prospective application areas. In Section IV, some representative simulation results are provided for a performance comparison with existing standardized image and video-coding technology.

II. FREXT CODING TOOLS

In this section, we describe those coding tools in more detail that make the FREXt amendment a preferable choice for high-quality broadcast and entertainment applications. In addition, we briefly list all further FREXt specific coding tools, most of them primarily targeted at providing extended functionality in terms of sample bit depth and color format. The main difference between FREXt and non-FREXt H.264/MPEG4-AVC coding is the use of an 8×8 transform in addition to the 4×4 transforms.

0-7803-9134-9/05/$20.00 ©2005 IEEE

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

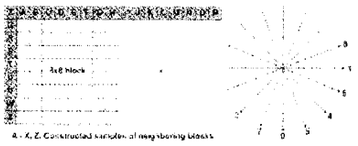


Figure 1: Left: Samples used for 8×8 spatial luma prediction. Right: Directions of spatial luma prediction modes 0, 1 and 3–8.

A. 8×8 Intra Spatial Prediction

Intra prediction in H.264/MPEG4-AVC is conducted by using spatially neighboring samples of a given block, which are already transmitted and decoded. In version 1, two different types of intra prediction for luma values are specified: one type performing a prediction for a whole 16×16 macroblock (MB), and a second type for prediction of 4×4 blocks [1][4].

In the FRExt amendment, an intermediate prediction block size of 8×8 was introduced for spatial luma prediction by extending the concepts for 4×4 intra prediction in version 1. As illustrated in Figure 1, the luma values of each sample in a given 8×8 block are predicted from neighboring constructed reference samples, where 8 different prediction directions plus an additional DC prediction (corresponding to mode 2, not shown in Figure 1) can be selected by the encoder. Note that the choice of the prediction type (4×4, 8×8, or 16×16) is also up to the encoder, subject to certain constraints on the choice of the corresponding luma transform size, as will be explained in Section II.B.

A distinguishing element of the new 8×8 intra prediction is given by a pre-filtering process of the reference samples (denoted by A–X, and Z in Figure 1). This lowpass filtering is conducted prior to the prediction step by applying a simple second-order binomial filter to the constructed luma reference samples.

B. 8×8 Transform

The design of the first version of H.264/MPEG4-AVC includes two different 4×4 transforms for coding of the luma prediction error signal. A 4×4 integer transform is applied to all luma samples with an additional transform stage using the 4×4 Hadamard transform for all resulting 16 DC coefficients of a transformed MB in the case of a preceding 16×16 luma intra prediction only [1][4].

Besides the important property of low computational complexity, the use of small block-size transforms in H.264/MPEG4-AVC has the advantage of significantly reducing ringing artifacts. For high-fidelity video, however, the preservation of fine details and textures which generally requires larger basis functions becomes equally important. For achieving a proper trade-off between these conflicting objectives, the FRExt amendment includes an 8×8 integer transform and allows the encoder to choose adaptively between the 4×4 and 8×8 transform for luma samples on a MB level [6]. This capability is a simplified version of the concept of variable-block size transforms which was already under consideration during the first standardization phase of H.264/MPEG4-AVC [7][8].

The 2-D 8×8 transform in the FRExt amendment is specified in a separable way as a 1-D horizontal (row) transform followed by a 1-D vertical (column) transform, where the corresponding 1-D transform is given by the (non-normalized) transformation matrix $T_{8\times8}$ as follows:

$$T_{8\times8} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}.$$

Interestingly, both the forward and inverse 8×8 transform can be efficiently implemented via fast butterfly operations such that only shift and add operations in $(8+b)$-bit arithmetic precision for b-bit input video are required [6]. In fact, Table I indicates that the computational complexity for the butterfly implementation of the 8×8 inverse transform is only slightly higher than that of the corresponding four 4×4 inverse transforms.

TABLE I: NUMBER OF OPERATIONS (E.G. OPS) REQUIRED FOR THE 2-D 4×4 AND 8×8 INVERSE TRANSFORM IN H.264/MPEG4-AVC FREXT.

Transform size	Operation	# Ops per 8×8 block
4×4	+	64
	>>	16
8×8	+	64
	>>	20

By taking into account the six different squared norm values of the underlying 2-D basis functions, the processes of scaling, quantization, and scanning of 8×8 transform coefficients are specified as straightforward extensions of the corresponding operations already defined for the 4×4 transform in H.264/MPEG4-AVC. Moreover, CABAC is extended by employing 3 additional sets of context models for coding of 8×8 transform levels, whereas CAVLC is applied by regrouping the 8×8 transform levels into four groups of 4×4 levels (for more details, refer to [5]).

As already noted, the encoder can choose for each MB between the 4×4 and 8×8 transform and signal its choice by means of a flag. However, there are two restrictions imposed on the transform size selection process:

- For inter-coded MBs with one or more sub-partitions smaller than 8×8 (i.e., MC blocks of size 4×8, 8×4, or 4×4), the 4×4 luma transform is mandatory.
- For intra-coded MBs, the 8×8 luma transform is chosen if and only if 8×8 luma spatial prediction is used.

C. Further Extensions

In addition to support for extended sample bit depth and monochrome as well as 4:2:2 and 4:4:4 chroma formats, the FRExt amendment contains three other important tools:

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

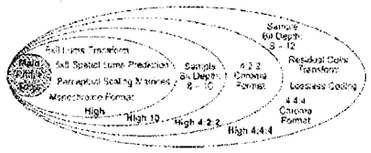


Figure 2: Illustration of the H.264/MPEG4-AVC FRExt profiles.

1. Encoder-specified *scaling matrices* for perceptual based, frequency-dependent quantization
2. A *residual color transform* consisting in a reversible integer-based color conversion from (4:4:4) RGB to the YCgCo color space applied to residual data only
3. A *lossless coding* capability requiring only a relatively simple bypass of transform and quantization

Furthermore, a few new supplemental enhancement information (SEI) messages for enabling enhancements of decoded video have been added to the FRExt amendment [5].

III. FREXT PROFILES AND APPLICATION AREAS

As illustrated in Figure 2, the H.264/MPEG4-AVC FRExt amendment specifies a set of four new profiles which are constructed as nested subsets of capabilities. All four profiles inherit the tool set of the Main profile, and as their common intersection, the *High profile* (HP) additionally contains all major new tools for improving the coding efficiency. Relative to the Main profile (MP), these tools imply only moderate (if any) increases in complexity both in terms of implementation and computational costs (at the decoder side), as briefly discussed for the case of the 8×8 transform in the previous section. Therefore, the High profile with its restriction to 8-bit video in 4:2:0 chroma format (or optionally in monochrome format) is likely to replace the Main profile for prospective applications of H.264/MPEG4-AVC in typical consumer applications.

An indication of this trend is given by the fact that the H.264/MPEG4-AVC High profile has recently been included in some important application standards or specifications of industry consortia:

- The specifications TS 101 154 and TS 102 005 of DVB (Digital Video Broadcasting) specify HP for satellite, cable, and terrestrial broadcasting using MPEG-2 transport stream or IP, respectively
- The HD-DVD specification of the DVD Forum makes HP mandatory for every decoder
- The BD specification of the Blu-Ray Disc Association (BDA) also makes HP mandatory for every decoder

IV. EXPERIMENTAL RESULTS

For a performance evaluation of the HP-specific 8×8 coding tools, as presented in Section II.A and II.B, a set of 7 progressive HD sequences (both at 1280×720@60Hz and 1920×1080@24Hz) with different characteristics has been used. The coding simulations were performed by using the H.264/MPEG4-AVC reference software version JM 9.4 and an MPEG-2 Main profile (MP@HL) conforming encoder, where both encoders have been controlled by using the same rate-distortion (RD) optimized Lagrangian strategy [3]. An I-frame refresh was performed every 500 ms, and two non-referenced B-pictures have been inserted between each two successive P-pictures. Full search motion estimation was performed with a search range of ±32 integer pixels. For H.264/MPEG4-AVC up to 3 reference frames were used.

TABLE 2: AVERAGE BIT-RATE SAVINGS OVER THE WHOLE TEST SET FOR H.264/MPEG4-AVC HP USING CABAC ENTROPY CODING IN COMPARISON WITH HP USING CAVLC, MP USING CABAC, AND MPEG-2.

H.264/MPEG4-AVC HP using CAVLC	H.264/MPEG4-AVC MP using CABAC	MPEG-2 MP@HL
17.9%	9.9%	58.8%

Figure 3 (bottom row) shows the RD curves for the 'Raven' and the 'Book' sequences comparing H.264/MPEG4-AVC HP, MP (both with CABAC and CAVLC), and MPEG-2. Since these sequences are characterized by a predominant highly textured content, relatively large gains can be obtained in favor of HP due to the better frequency selectivity of the 8×8 luma transform. Averaged over the whole HD test set, HP achieves bit-rate savings of about 10% relative to MP (both using CABAC), as shown in Table 2. If, however, the 8×8 tool set of HP is used in conjunction with CAVLC, an average loss of about 1% is observed relative to HP using CABAC, which means that the CAVLC-driven HP leads, on average, to an objectively lower performance than that measured for the CABAC-driven MP (see Table 2). In comparison to MPEG-2, average bit-rate savings of about 59% were observed for the H.264/MPEG4-AVC High profile coder (using CABAC).

In another experiment, the HP-specific support for monochrome input was used in order to investigate the intra coding performance of H.264/MPEG4-AVC HP in comparison with JPEG2000 (Part 1) [9] for a set of 8 photographic monochrome test images (with resolutions from 512×512 up to 2048×3072 samples). Note that these images have been used extensively in the JPEG2000/JPEG-LS standardization work, whereas the H.264/MPEG4-AVC encoder was never before applied to that input data.

For this intra-only coding experiment, the Kakadu implementation [10] (version 2.2) of JPEG2000 was used as a reference, where the Kakadu encoder was driven in default lossy mode using an RD optimization for each given target rate point. The rate points were chosen to cover a range of 0.25 – 2 bits per pixel (bpp). Thus, in order to achieve the maximum compression performance for JPEG2000 and to guarantee a fair comparison, the SNR scalability feature of JPEG2000 was not utilized for our tests.

In Figure 3 (top right), two sets of RD curves are plotted – one for 'Lena' and another for the 'Barbara' image. As can be seen from the graph, the overall objective performance of JPEG2000 and H.264/MPEG4-AVC HP is nearly identical in both test cases. By disabling the 8×8 tool set in the HP encoder, a significant loss in PSNR of about 1 dB is observed for 'Barbara' with its well-known high-frequency

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

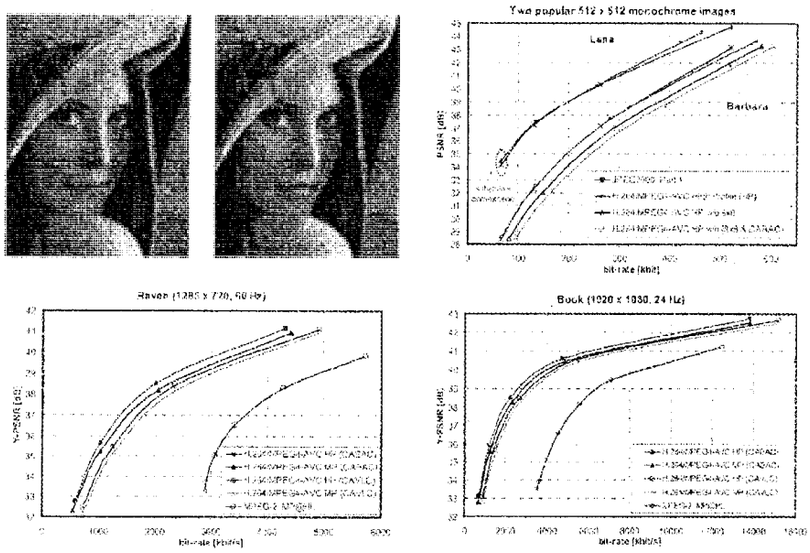


Figure 3: Top row: Detail of "Lena" image containing JPEG2000 (left) and H.264/MPEG4-AVC HP (middle) at a compression ratio of about 32:1 (0.25 bpp). The corresponding RD points are given as the lowest points on the curves in the RD plot (right) showing the 0.6 dB benefit for both "Lena" and "Barbara". Bottom row: RD graphs for the "Raven" and the "Book" sequence comparing H.264/MPEG4-AVC SP, MP (both using CABAC and CAVLC), and MPEG-2 MP@ML.

characteristics. Furthermore, going from CABAC to CAVLC entropy coding, results in an additional loss of 0.5 dB PSNR for that particular image, as shown Figure 3 (top right).

By averaging the intra coding results over the entire test image set and over all bit-rates, a gain of 0.5 dB PSNR was observed in favor of the H.264/MPEG4-AVC HP coder. In all cases, the subjective quality of the H.264 MPEG4-AVC HP encoded reconstruction was at least comparable to that of the corresponding JPEG2000 reconstruction at the same bit-rate. Figure 3 (top left) contains decoded samples of the "Lena" image at 0.25 bpp for a comparison of the visual quality.

V. Conclusions

The H.264/MPEG4-AVC FREXt amendment has been presented with a particular focus on the High profile (HP) and its related coding tools. The High profile-specific tools offer a good trade-off between coding efficiency and implementation cost, especially when compared to the H.264/MPEG4-AVC Main profile (MP). Average bit-rate savings for the H.264/MPEG4-AVC High profile of 10% relative to MP (both using CABAC) and nearly 60% relative to MPEG-2 have been observed for a set of HD sequences. When restricted to intra-only coding, H.264/MPEG4-AVC HP outperforms the state-of-the-art in still-image coding represented by JPEG2000 on a set of monochrome test images by 0.5 dB average PSNR.

References

[1] ITU-T Recommendation H.264 & ISO/IEC 14496-10 (MPEG-4) AVC, "Advanced Video Coding for Generic Audiovisual Services", (version 1: 2003, versions 2: 2004) version 3: 2005.

[2] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), Nov. 1994.

[3] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Trans. CSVT, vol. 13(7), pp. 688-703, July 2003.

[4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264 AVC video coding standard," IEEE Trans. CSVT, vol. 13(7), pp. 560-576, July 2003.

[5] Joint Video Team of ITU-T and ISO/IEC, "Draft Text of H.264/AVC Fidelity Range Extensions Amendment", Doc. JVT-I047, Sept. 2004.

[6] S. Gordon, D. Marpe, and T. Wiegand, "Simplified Use of 8x8 Transforms" JVT Doc. JVT-K028, March-March 2004.

[7] M. Wien, "Variable Block-Size Transforms for H.264/AVC", IEEE Trans. CSVT, vol. 13(7), pp. 604-613, July 2003.

[8] E. Bosveld, "ADT Cleanup and Complexity Reduction", JVT Doc. JVT-D087, Geneva, Oct. 2002.

[9] ITU-T Rec. T.800 and ISO/IEC 15444-1, "JPEG2000 Image Coding System: Core Coding System" (JPEG2000 Part 1), 2000.

[10] Kakadu software available on the http://www.kakadusoftware.com.

Efficient coding scheme for super high definition video based on extending H.264 high profile

S. Naito[a], A. Matsumura[b], A Koike[a]
[a]KDDI R&D Laboratories Inc., 2-1-15, Ohara, Fujimino-shi, Saitama, JAPAN 356-8502
[b]KDDI Corporation, 3-10-10, Iidabashi, Chiyoda-ku, Tokyo, JAPAN 102-8460

ABSTRACT

H.264 High Profile, one of the standard video coding schemes, can be utilized for super high definition video since target image resolution is up to 4096×2304. However, in the case of encoding super high definition video such as the 4Kx2K system whose resolution corresponds to 4096x2160, the coding syntax might be redundant since available macroblock size become quite small compared to the image resolution. The problem is caused by the limitation of the macroblock size by which motion compensated prediction is conducted. To overcome the problem within the framework of the H.264 High Profile encoder, extending the limitation of macroblock size is required. From this perspective, this paper describes an efficient coding scheme applicable to super high definition video based on extending the H.264 High Profile. As the key technology of the proposed coding scheme, an approach for extending the limitation of available macroblock size is introduced. From experimental results, it was confirmed that the outstanding coding gain was achieved by the proposed scheme especially for the low bit-rate condition.

Keywords: H.264 High Profile, macroblock size extension, super high definition video, 4K system, digital cinema

1. INTRODUCTION

The 4Kx2K system has been specified by DCI (Digital Cinema Initiatives, LLC) as a standard for video applications with super high definition such as digital cinema [1]. The original bit-rate of 4Kx2K contents corresponds to 4 times that of conventional HDTV, and the field trial of the 4Kx2K video streaming was conducted based on the Motion JPEG2000 encoder at the bit-rate of 300Mbps [2]. However, highly efficient compression coding is required to realize the 4Kx2K video streaming over practical IP networks. With regard to encoding moving pictures, the representative framework is the hybrid use of motion compensated prediction and DCT. Based on this framework, H.264 is superior in its high coding efficiency and wide coverage of supported image resolutions, and is applicable even to 4Kx2K video [3]. Furthermore, FR-Ext (Fidelity Range Extension) has been included as the High Profile of H.264 in order to improve the coding performance especially for high definition video [4], and its effectiveness has been reported in a previous study [5]. However, the scheme might cause inefficiency in the coding side information part when applied to high resolution images such as 4Kx2K video. The problem is caused by the limitation of the macroblock size by which motion compensated prediction is conducted. In other words, the maximum value of available macroblock size is quite small considering the large image size of 4Kx2K video. Therefore, additional performance improvement can be expected by reducing the bit amount in the side information part as a result of extending the limitation of macroblock size for the H.264 High Profile encoder.

From this perspective, this paper describes an efficient coding scheme applicable for super high definition video based on extending the H.264 High Profile. As the key technology of the proposed coding scheme, the approach for extending the limitation of available macroblock size is introduced. The effectiveness of the proposed technology is evaluated from the experimental results. The macroblock is simply called MB in the following.

2. EXTENDING MACROBLOCK SIZE

In this section, the approach for extending the limitation of MB size is introduced as the key technology of the proposed encoding scheme. The assumed picture level coding process is shown in Fig. 1. Prior to the picture encoding process, the parameter N which is utilized for identifying the maximum value of available MB size is decided for the current picture. The parameter N remains identical while encoding a picture. The parameter N means that the maximum MB

Visual Communications and Image Processing 2006, edited by John G. Apostolopoulos, Amir Said,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6077, 607727, © 2006 SPIE-IS&T · 0277-786X/06/$15

1217954

size is equal to N x N. The value of N is expressed by the power of 2 whose minimum value is 16. Under the limitation identified by N, MB size is selected from the candidates as shown in Fig. 2. For intra and inter coded MBs, the macroblock coding process is conducted under the following constraints.

- Intra coded MB
  MB size is selected from (N x N), ($N/2$ x $N/2$) and ($N/4$ x $N/4$). The DCT process is conducted under the condition as shown in Table 1.
- Inter coded MB
  The MB size is selected from (N x N), ($N/2$ x N), (N x $N/2$) and ($N/2$ x $N/2$), and sub-macroblock size is selected from ($N/2$ x $N/2$), ($N/4$ x $N/2$), ($N/2$ x $N/4$) and ($N/4$ x $N/4$) only when ($N/2$ x $N/2$) is selected as the MB size. The DCT process is applied for each block that contains an MB, where DCT block size is selected from 4x4 and 8x8.

In the proposed coding scheme, the extension of available MB size described above is incorporated into H.264 High Profile coding syntax with a minor revision which adds a short field as an identifier of N in the slice header. The overhead caused by this revision seems to be negligible in terms of the coding efficiency. Based on the proposed coding syntax, the coding process fully equivalent to the standard H.264 High Profile is also available as a result of encoding under N=16. Furthermore, the adaptive selection of N is allowed picture by picture based on the proposed syntax, and the optimization approach is expected to introduce further coding gain. Prior to encoding a current picture, the value for N is updated from the result of coding a previous picture t' with an identical picture type. In this decision, functions $E1$ and $E2$ as shown in (1) are utilized to approximate the picture bit amount when applying $2N'$ and $N'/2$, respectively, where N' is the maximum macroblock size selected for coding the picture t'. In (1), Q' and B' are the quantization parameter and the averaged number of encoded bits, respectively, and those parameters are calculated from the coding result of t'. a_n and b_n (n = 0, 1, 2) are coefficients which are defined dependent on the picture type and the value N'. The output of those functions are normalized by the picture bit amount under N'. The updating algorithm of N is conducted along (2) to (4).

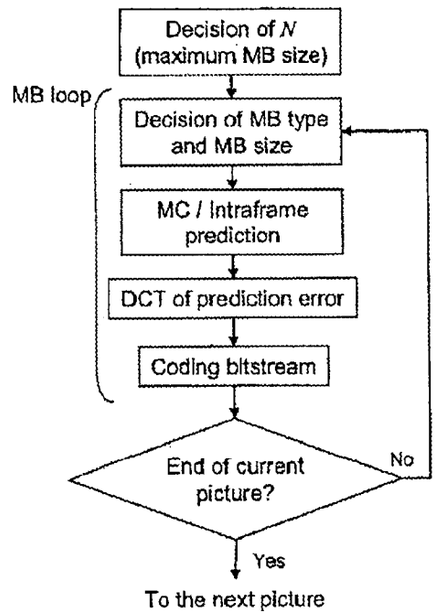


Figure 1: Picture level coding process.

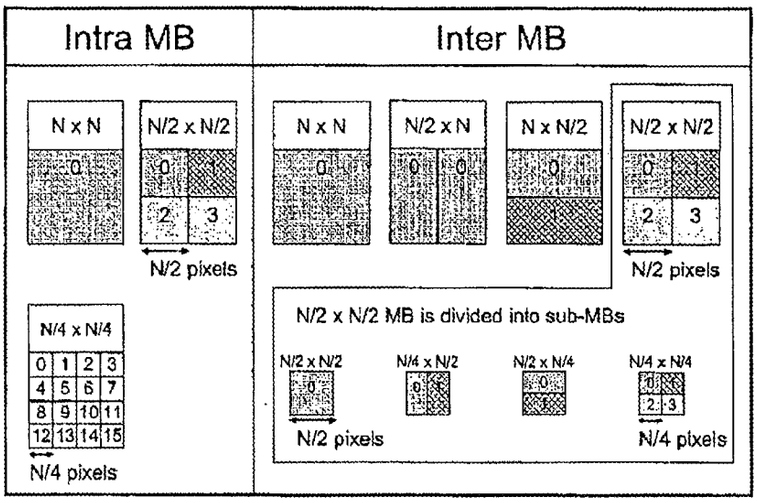


Figure 2: Available candidates for macroblock size decision.

Table 1: DCT process for intra macroblocks.

MB size	DCT type	Re-prediction of DC coefficients
4 x 4	4 x 4	Not applied
8 x 8	8 x 8	Not applied
16 x 16 or larger	4 x 4	Applied

$$(1)\quad \begin{cases} E1(Q',B') = a_0 \times \log_2(Q') + a_1 \times \log_2(B') + a_2 \\ E2(Q',B') = b_0 \times \log_2(Q') + b_1 \times \log_2(B') + b_2 \end{cases}$$

i) $E1 < 1.0$ and $N' < N_{max}$
$N = 2N'$ (2)

ii) $E2 < 1.0$ and $N_{min} < N'$
$N = N'/2$ (3)

iii) otherwise
$N = N'$ (4)

3. EXPERIMENTAL RESULTS

In order to evaluate the coding performance of the proposed scheme, coding experiments were conducted. The coding parameters assumed in those experiments are shown in Table 2. The parameter QP indicates the quantization parameter commonly applied while encoding a sequence. As for video input signals, two sequences (A) and (B) are selected from the digital cinema standard test sequences provided by DCI [6]. For each sequence, the corresponding frame number

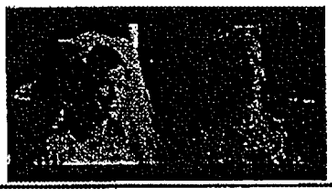
and the still image of the first frame are summarized in Table 3. Those sequences are constituted by 28 frames. As the feature of those materials, the sequence (A) does not include any high speed motion, and the sequence (B) contains comparably high speed motion accompanied by panning operation. The video format is 4096x2160 (YUV, 4:2:0) at the frame rate of 24 frame/sec. The bit-depth is 8bit for each component. With regard to the limitation of available MB size, three different cases, N=16, 32, and 64, were evaluated, where N=16 corresponds to the conventional encoder compliant with the H.264 High Profile.

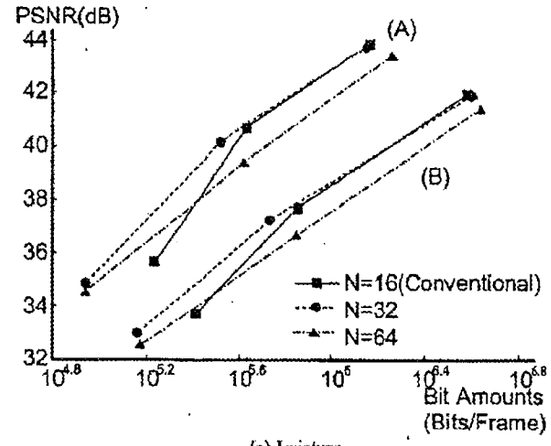
First of all, rate distortion characteristics were observed to evaluate the coding efficiency as shown in Fig. 3. The horizontal and vertical axes indicate the coded bit amount and the PSNR result, respectively. The results of bit amount and PSNR were calculated as the averaged values for a picture. From the results, the coding efficiency under the low bit-rate was improved by selecting a larger value for N compared to the conventional case (N=16). The degree of coding gain was dependent on input signals. On the other hand, the selection of N=64 under smaller QP resulted in the degradation of the coding performance compared to the result by N=16. From this, it was confirmed that the adaptive selection of N conducted picture by picture was required to apply the optimal setting for every picture.

Table 2: Encoder parameters for experiments.

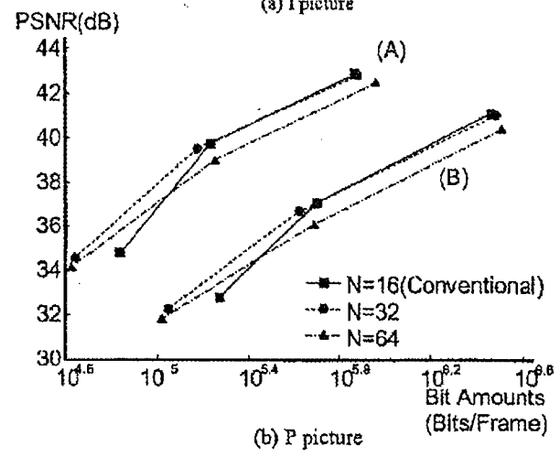
QP (Quantization Parameter)	25, 35, 45
Motion estimation	UMHexagon Search
Search range	128 x 128
GOP structure	IBBPBBPBBPBBPBB
R-D optimization	Enabled
Reference picture	I picture or P picture
Number of reference pictures	2
Coding scheme for prediction error signals	CABAC

Table 3: Test materials.

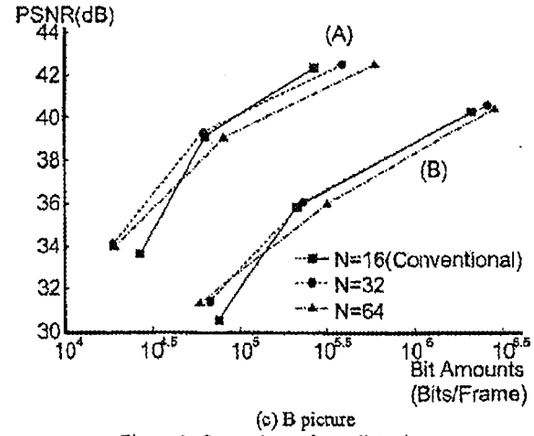
(A) (Frame number :1865 – 1892)	(B) (Frame number : 2157 - 2184)
	
	



(a) I picture



(b) P picture



(c) B picture

Figure 3: Comparison of rate distortion curves.

In order to investigate the effectiveness, details of coded bit assignment were observed. For the sequence (A), the result of coded bit assignment under QP=25 and QP=45 is shown in Fig. 4. In the figure, the coded bits were categorized into the side information part and the DCT part, and the bit amount was averaged for a picture. As the given value for N became large, the bit amount of the side information part was decreased while increasing the bit amount for the DCT part. The coding loss in the DCT part was caused because the prediction error was increased as a result of disabling smaller MB size. Similar results were confirmed from further experiments for the other sequences including the sequence (B). Consequently, the coding gain achieved by the proposed scheme was outstanding under the low bit-rate coding because the side information part becomes dominant in the coded picture.

Furthermore, the distribution of selected MB types was observed in order to evaluate the impact by extending the MB size. The averaged distribution map of MB types selected for the sequence (A) under QP=25 was observed as shown in Fig. 5. From the result, it was confirmed that MBs of smaller size were selected more frequently, and the percentage of selecting skip or direct modes was decreased as the given value for N became large. We also confirmed that this change in the MB type distribution caused the increase in the number of MBs actually encoded for the corresponding parent MB of size N x N. From further experiments for the other combination regarding QP and test sequence, similar results were confirmed.

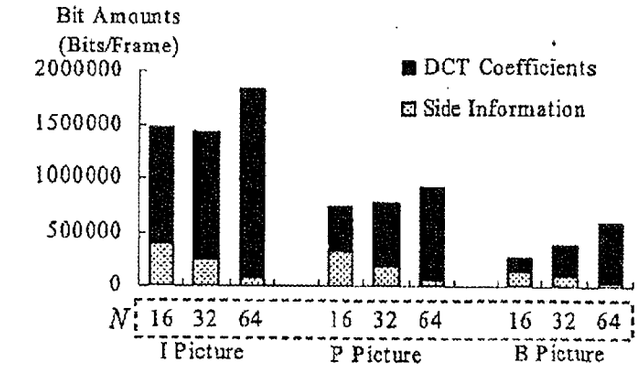
Finally, the proposed adaptive selection scheme for the parameter N was evaluated. Fig. 6 shows PSNR results obtained by the adaptive selection of N whose candidates were assumed to be 16, 32 and 64. The results when N was statically assigned as 16, 32 or 64 were also shown in the figure. This experiment was conducted for two sequences described above. From the results, it was confirmed that the value N selected by the proposed scheme was quite optimal, because the PSNR gain compared to the static assignment case was stably achieved by the proposed scheme independent on the input sequence and the coding bit-rate. The effectiveness was outstanding especially under the low bit-rate coding.

5. CONCLUSIONS

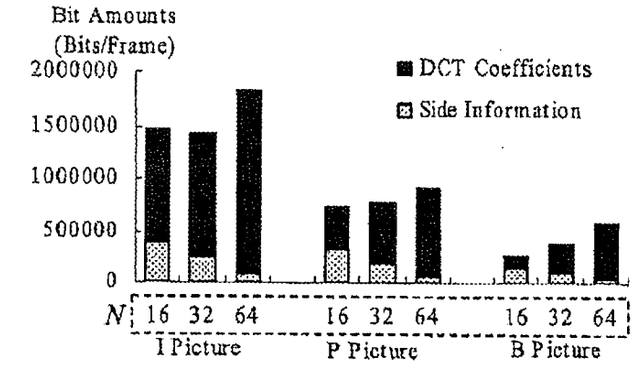
An efficient coding scheme applicable to super high definition video such as 4Kx2K video was proposed based on extending the H.264 High Profile coding mechanism. From experimental results, it was confirmed that the outstanding coding gain was achieved by the proposed scheme especially for the low bit-rate condition. Furthermore, we also confirmed that the adaptive control of extending the macroblock size was effective in maintaining the optimal coding syntax for every picture. The technique to realize the optimal selection of the maximum macroblock size was also introduced and evaluated from coding experiments.

REFERENCES

1. Digital Cinema System Specification V4.3, http://www.dcimovies.com/
2. Takahiro Yamaguchi, Daisuke Shirai, Tatsuya Fujii, Mitsumu Nomura, Tetsuro Fujii and Sadayasu Ono, "SHD Digital Cinema Distribution Over a Long Distance Network of Internet2," Proceedings of SPIE on Visual Communication and Image Processing 2003, Vol. 5150, pp. 1760-1769, July 2003.
3. Text of ISO/IEC 14996 10 Advanced Video Coding 3rd Edition (ISO/IEC JTC1/SCD29/WG11 and ITU-T SG16 Q.6)
4. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 - ISO/IEC 14496-10 AVC), http://www.ncesd.org/vc/docs/JVT-G050.pdf
5. Gary J. Sullivan, Pankaj Topiwala and Ajay Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," SPIE Conference on Applications of Digital Image Processing XXVII, Vol. 5558, pp. 454-474, Aug. 2004.
6. Digital Cinema Initiatives, LLC, ASC/DCI StEM material, http://www.dcimovies.com/StEMAcce.pdf



(a) QP=25



(b) QP=45

Figure 4: Number of coded bits averaged for a picture

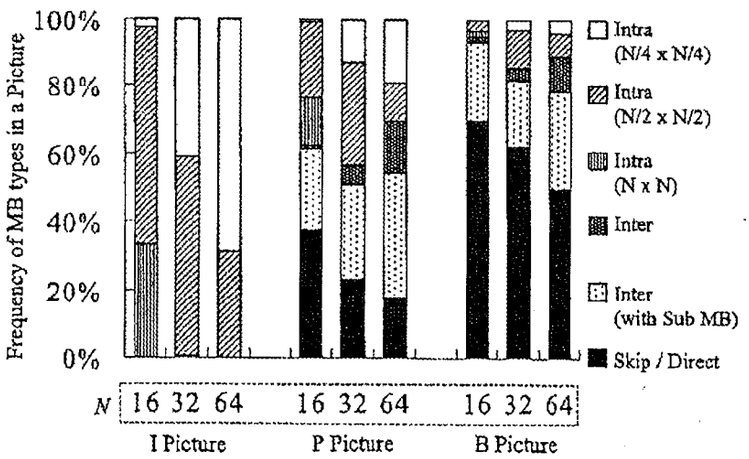
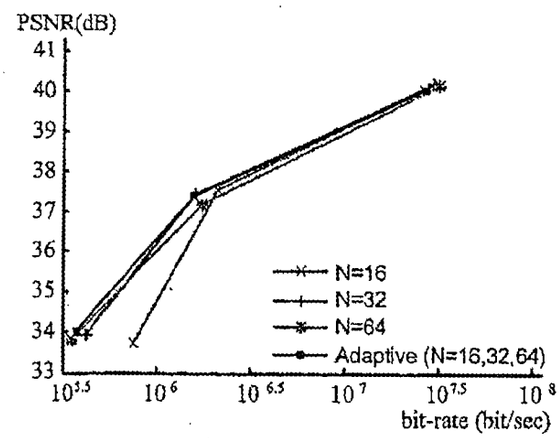
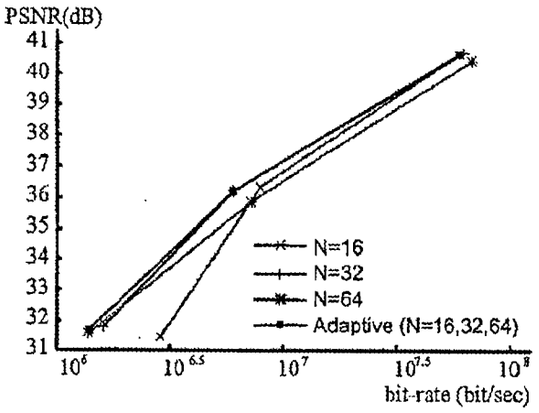


Figure 5: Averaged distribution of macroblock types.



Sequence (A)



Sequence (B)

Figure 6: PSNR results by adaptive selection of the parameter *N*.

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　2

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

Image Compression with Variable Block Size Segmentation

Jacques Vaisey, *Member, IEEE*, and Allen Gersho, *Fellow, IEEE*

***Abstract*—High quality variable-rate image compression is achieved by segmenting an image into regions of different sizes, classifying each region into one of several perceptually distinct categories, and using a distinct coding procedure for each category. Segmentation is performed with a quadtree data structure by isolating the perceptually more important areas of the image into small regions and separately identifying larger "random texture" blocks. Since the important regions have been isolated, the remaining parts of the image can be coded at a lower rate than would be otherwise possible. High quality coding results are achieved at rates between 0.35 and 0.7 b/p depending on the nature of the original image, and satisfactory results have been obtained at 0.25 b/p.**

I. Introduction

NATURAL images can be segmented into regions having widely differing perceptual importance. The importance of a region corresponds to the amount of "information" it conveys to the viewer. Typically, the amount of local "detail," or high frequency content, is considered a reasonable measure of this importance; however, this by itself is an inadequate measure for efficient image coding. It is also desirable to consider other significant characteristics, such as texture and the global context of the local region, in order to assess the local information content. Certain regions are critical to our subjective evaluation of quality, and relatively small quantization errors can perceptually have a major degrading effect on the overall reproduction quality. Such regions tend to dominate the viewer's attention and are intrinsically less compressible than "background" segments. Consequently, when the overall bit rate is low, a uniform allocation of bits across an image implies that the spatial distribution of perceptual degradation is highly nonuniform; some regions have a starvation diet of bits, causing a significant degradation, while other regions have been coded with far more bits than needed for perceptually transparent quality. Thus, we are motivated to examine variable-rate image coders that change the coding resolution (in bits used per unit area) according to the local character and importance of the region being coded. The objective of our approach is to obtain a uniform layer of degradation per unit area over the entire image. Ideally, this layer should be thin enough so that it lies just below the threshold of noticeable distortion, thereby yielding "transparent" quality reproduction. However, even if the available bit quota is inadequate to achieve transparent quality, a much higher overall perceptual quality can be achieved compared to that of coding schemes with spatially uniform resolution at the same average bit rate or variable rate schemes which do not effectively distribute bits to regions according to a perceptually effective criterion.

Image coding algorithms are usually based on one of two rudimentary coding techniques: *transform coding* (TC) [1], [2] or *vector quantization* (VQ) [3], [4]. The signal to be coded may be a raw or processed image, or it may be the result of a signal decomposition such as subband filtering [5]. Both VQ and TC are based on block coding, where the input image, or signal, is partitioned into (usually square) blocks of pixels which are then encoded as separate entities. In each scheme, the block size is a fundamental design parameter. To obtain the flexibility of a variable block size partition while avoiding the excessive overhead or side information needed to characterize more sophisticated image segmentation techniques, we use a top–down quadtree approach to image partitioning. This segmentation divides the image into square regions that vary in size between 32×32 to 4×4 pixels and appropriate coding technique, such as classified VQ [6] and a TC/VQ hybrid, are then used for each block. Pruned-tree structures have also been used in an alternative approach to variable rate algorithms [7]; however, the 2×2 block size used with this technique cannot exploit correlation existing in large low-detailed regions.

This paper extends the preliminary work described in [8] and [9], where variable block size VQ was introduced, and it defines a general philosophy for image coding that is easily combined with other coding techniques (such as subband coding). This work represents a significant departure from traditional image coding methods, since almost all prior studies, for example [2], [6], [10]–[12], have used a fixed block size, usually 4×4 or 5×5 pixels for VQ and 8×8 or 16×16 for TC. Historically, Nasrabadi [13] also mentioned the use of variable block size in image coding; however, the idea was not pursued. Re-

Manuscript received July 28, 1989; revised June 6, 1991. This work was supported in part by the National Science Foundation, Bell Northern Research, Bell Communications Research, Inc., Rockwell International Corporation, and the University of California MICRO Program.

J. Vaisey is with the School of Engineering Science, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada.

A. Gersho is with the Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

IEEE Log Number 9200071

1053-587X/92$03.00 © 1992 IEEE

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。



Fig. 1. The test images. (a) Lenna. (b) Bridge. (c) Barbara.

cently, Chen [14] has adapted our variable block size approach to a coder based entirely on the DCT. Also, although quadtrees have been widely used in image analysis as a convenient data structure, little use has been made of this approach for image coding until recently. Wilson [15] has used quadtrees to generate a multiresolution decomposition of images. Leonardi and Kunt [16] have used them to define regions that can be satisfactorily approximated by 2D polynomials, while Farrelle and Jain [17] used them to flag blocks that are poorly described by their interpolation procedure. More recently, Benedetti *et al.* [18] used quadtrees in lossless image coding, and both Strobach [19] and Nasrabadi *et al.* [20] used them in the coding of image sequences. Bublér and Fortner [21], [22] have independently introduced a quadtree segmentation procedure that is similar to ours.

In this study, we will present coding results for several different 512×512 test images, none of which are contained in the training set of images used to design the coder. These images are shown in Fig. 1, and we refer to them as "Lenna," "Bridge," and "Barbara," respectively.

The organization of this paper proceeds as follows. Section II provides a brief overview of the structure of the algorithm and the quadtree data structure. Section III then describes the first stage of the algorithm, the generation of an approximation to the input image based on information about the local means. This is then followed by Section IV, which describes the second stage segmentation and the actual quantization procedure. Section V presents the smoothing algorithm used to reduce blocking, while Section VI describes specific coder designs and results obtained from computer simulations. Finally, Section VII offers concluding comments and suggestions for further work.

II. An Overview of the Algorithm

The encoding procedure takes place in two stages. In the first stage, an initial quadtree segmentation of the input image, Ω, is performed to locate large regions having homogeneous mean values. Once this mean segmentation is complete, the means of each block are quantized using VQ, with the four components of each vector corresponding to the means of leaf blocks having the same parent.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

These quantized means are used to form the mean image, Ω_q, which is then applied to an interpolation procedure that generates an interpolated image, Ω_i, which by itself offers a low fidelity but very low rate approximation to the input image. The interpolated image is subtracted from the input to obtain the residual image Ω_R.

The residual image is then segmented with a quadtree into square blocks whose size depends on a classification of the region according to a rough qualitative appraisal of its local perceptual importance and coding difficulty. At this stage, the classifier identifies three major categories: low detail; high detail that must be coded with high fidelity; and random texture, a class having high detail that can be coded with low fidelity. The segmentation attempts to group the low detail regions into blocks that are as large as possible, while isolating the high detail regions into small size blocks. These sizes range from 32×32 to 4×4, with the block size (width) always being a power of two. All of the high detail regions are segmented into 4×4 blocks.

Once the segmentation of Ω_R is complete, each block is coded with an algorithm appropriate for its class. The large low-detail regions are quantized using a hybrid TC-VQ approach, while the small high-detail regions are coded with classified VQ. A block diagram showing this processing chain is given in Fig. 2.

At the receiver, the residual and interpolated images are reconstructed and then added together to form the coded output Ω'. This image is then optionally fed into a smoothing algorithm to reduce residual blocking effects, producing the smoothed coded image Ω_S'.

All of the image segmentation procedures described in this paper are based on the quadtree [23], an efficient data structure that provides an effective compromise between the accuracy with which the region boundaries are determined and the number of bits required to specify the segmentation information. Other methods of image segmentation, such as region growing [24], [25], more precisely isolate statistically homogeneous segments; however, in these techniques, both the number of the regions and their shape is determined solely by the image being examined. This fact implies that a very large number of bits may be needed to represent the shape and location information. Quadtrees, on the other hand, require only a small overhead rate by restricting the shape and the number of possible sizes of the final regions from a predetermined set of options. In performing such a segmentation, the coding algorithm must take a series of binary decisions that are controlled by a number of different threshold values, all of which we empirically "optimize" through informal subjective experiments.

A quadtree is a tree structure in which each nonterminal (parent) node has four branches emanating from it. These branches point to nodes that are the children of the parent node. In our application to image coding, every node corresponds to a subblock of the image that is determined, both in size and location, by its position in the tree. The four children of a particular parent node represent the four

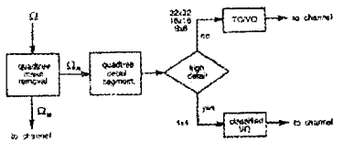


Fig. 2. Operations performed by the encoder.

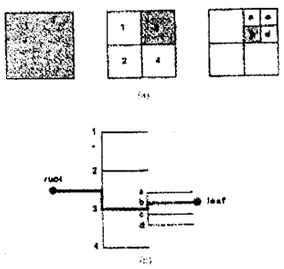


Fig. 3. The structure of the quadtree. (a) The relationship of image regions. (b) The relationship to a tree.

subblocks obtained by splitting the parent block into four equal-size squares. The tree itself consists of several stages, or levels, of nodes, where a node at the nth stage represents a one-quarter subblock of the parent image block from the previous stage.

When using this structure to segment an image, we start at an initial node, the root, corresponding to some particular image block with a relatively large size, and proceed through the tree from level to level. At each node, a test is performed to determine whether the block represented by that node is homogeneous in the property of interest. If the test is positive, then the node becomes a leaf, or terminal node. Otherwise, the segmentation proceeds to the next level and the four children of the node are each examined in turn. The procedure always terminates when it reaches the node level corresponding to a specified minimum allowed block size. With the simplest direct coding method, the final quadtree structure is represented by one bit of side information for each node, which indicates whether that node is a parent or a leaf. The manner in which areas of an image are related to the position in the tree is illustrated in Fig. 3.

III. Generation of the Interpolated Image

The purpose of the first stage of the coder is to generate a "low-pass" version of the input image, Ω_i, that can be coded at a low rate. Subtracting this image from the input

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

allows a second stage coder to concentrate on local details and, as long as the correlation between Ω_R and Ω_I is low, the penalty in efficiency resulting from using a two stage coder will be small. An additional advantage to the two stage approach is that Ω_I can also be used as the first stage in a progressive transmission scheme.

The following section describes the algorithms that are used to generate Ω_I. We start with a definition of Ω_M and then show how the mean segmentation that generates it is done. The interpolation procedure that improves the fidelity of Ω_M is then presented.

A. The Mean Image

The mean image is generated after segmentation by replacing the amplitude of each pixel in a given block by the *quantized* mean amplitude of all pixels in that block. This technique for forming Ω_M can be viewed as a crude form of decimation and interpolation. In fact, given that every pixel amplitude in a block is to be replaced by a constant amplitude, the amplitude that minimizes the squared error is the block mean.

An important property of the block means is that they vary quite slowly over large areas of Ω. This property is shown in Fig. 4, where each 4×4 block has been replaced by its mean value in the image "Lenna." Behavior of this type implies that the local mean values are highly correlated, so that it is possible to encode them at a rather low rate. Most methods used to code the block means use a fixed number of bits for each block, and thus cannot exploit the fact that the means vary faster in some regions of the image than in others. We capitalize on this property by using a quadtree structure to isolate those regions of the image where the mean changes quickly, allowing us to adjust the coding rate appropriately. We next explain our procedure, a variant of the method introduced in [8], in more detail.

1) The Mean Segmentation Algorithm: The purpose of the mean segmentation is to partition Ω into regions that have a homogeneous local mean, where the local mean of an image at a particular pixel is the average pixel amplitude in a small $l\times l$ window centered about that pixel. By translating this window over a region, one can examine how the local mean varies in this region. In general, nonsquare windows could be used; however, we felt that the possible gain in performance is minimal.

Our approach is based on the fact that the local means are fairly constant in the area of the image where the mean is homogeneous. Thus, if the mean pixel amplitude in such a region is subtracted from every pixel in that region, then the mean of any $l\times l$ block in the mean-removed residual will be close to zero. Since the coder that we will use to quantize Ω_R (as will be discussed in Section IV) never uses blocks smaller than 4×4, there is no advantage to ensuring that the mean of a region is homogeneous in local regions smaller than 4×4. We thus use the magnitudes of the means of the 4×4 residual blocks as our measure of mean homogeneity.



Fig. 4. The 4×4 means of Lenna. Each pixel in Lenna has been replaced by the mean values of the 4×4 blocks.

The top–down segmentation algorithm will now be explained step by step. This procedure depends on the values of block variance thresholds to be described in Section IV-B1, since limiting the depth of the mean tree to the depth of the corresponding residual tree will simplify the task of the residual coder. First, we must choose the starting block size and specify the root nodes of the mean tree. An analysis of test images showed that the means of blocks larger that 32×32 are almost always inhomogeneous, and our algorithm thus starts with a partition of the image into blocks of this size. The block mean is then calculated for each block, and the corresponding quantized value subtracted from each pixel to form the 32×32 residuals. At this stage, a decision is made to stop or to descend further into the tree for every 32×32 block. The next step is to examine both the magnitudes of the means of the corresponding 64 4×4 subblocks in the residual block, as well as the variance of the pixels in the 32×32 blocks. If either the variance is too low (according to the detail threshold T_{det}) or all of the 4×4 means are below the mean threshold, T_M, then the block is declared a leaf of the mean tree. Otherwise, we descend further, and four children are generated from the residual of the parent block, each corresponding to a 16×16 block of pixels. The mean values of the four children are related, and we exploit this correlation by quantizing them with a 4-dimensional vector quantizer designed to minimize the MSE between the coded and unaided mean values. The 32×32 mean can be viewed as a prediction of the 16×16 means.

If the 32×32 block does not correspond to a leaf node, then each of the 16×16 children blocks are tested individually to determine whether the mean is homogeneous at this level. As before, the quantized mean (now based on the 16×16 residual block) is subtracted from each pixel to form the new residual. The resulting block is partitioned into 4×4 blocks, whose means are calculated and compared with the threshold. The descent into the tree continues until either the block variance drops below its threshold, the 4×4 means in the residual block are suffi-

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

ciently small, or until we reach the 4×4 level, the deepest level allowed in our tree. The final quantized mean value for each leaf is then calculated by adding up the quantized means back up the tree to the root level. For instance, in Fig. 3(b), the quantized mean for the leaf block at the node "b" is calculated by adding together the means retained at node "3" and the root. The mean image is obtained by replacing all of the pixels in each leaf block by the corresponding quantized mean. A flowchart for the processing done on a $N\times N$ region is provided in Fig. 5, and Ω_M for "Lenna" (coded at 0.079 b/p) is shown in Fig. 6.

The choice of T_M mediates a tradeoff between the number of bits needed to code Ω_M and Ω_R. As the threshold is lowered, more and more 4×4 regions are generated by the mean segmentation. This improves the quality of Ω_M, reducing the information in Ω_R; however, it also increases the number of bits necessary to code the mean information. Through experimentation with several training images we found that an effective compromise was obtained with $T_M \approx 16$. Specific codebook sizes for the mean quantization are given in Section VI.

B. The Interpolation Procedure

The mean image provides a low rate, reasonable quality, approximation to Ω, and effective coders can be built around this result [9]. However, improved performance can be obtained if we realize that there is much more information in the quantized mean values obtained from the quadtree segmentation than is immediately apparent. This data also provides important information as to how detail is distributed in the image. In particular, leaves of the tree representing large blocks of pixels correspond to regions with low detail, and 4×4 leaves generally mark the important edge regions. In fact, the mean values can be viewed as the result of a variably spaced sampling of the local mean, where the sample spacing is large in the background regions of the image and small where the image is busier. Our objective is to find an algorithm that, given these samples, results in the best possible approximation to the input. This result will then generate a Ω_R that is easier to code than when using Ω_D directly. The technique for predicting the AC coefficients in the JPEG coding algorithm [12] also uses the DC values of the surrounding (fixed size 8×8) blocks to predict the AC behavior, however, in this case the prediction is limited to the low frequency AC coefficients and operates in the transform domain instead of the spatial domain.

1) The "Tent" Analogy: We have found it useful to use a *tent* analogy in visualizing the above problem. The idea is to generate an approximate image surface from the mean tree. An image surface refers to the three dimensional model of an image as a surface above the image plane whose height over any pixel is the amplitude of that pixel. In this framework, we start with a blank, or black, image plane blocked in the same manner as Ω_M; this acts as a foundation upon which a "tent" is constructed. Tent-

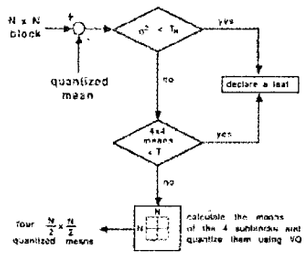


Fig. 5. The flowchart for the mean segmentation.

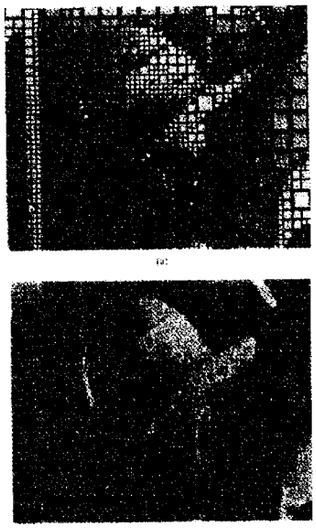


Fig. 6. The mean image for Lenna.

poles are then placed at the center of every block corresponding to a leaf in the mean tree, with the height of each pole given by the amplitude of the mean. The poles are square in cross section in order to follow the block structure; however, different pole areas are required for different block sizes, since a large block in a leaf position indicates that the mean is varying slowly. The task of the interpolation algorithm is to place a canvas over the poles to approximate the surface of the image, with the surface height equal to the pole height at the pole locations.

An example of the pole image Ω_P formed by placing the poles on the image plane is shown together with the result of the interpolation Ω_I in Fig. 7. The pole sizes used for the four possible block sizes are in order of decreasing block size: 24×24; 12×12; 6×6; and 2×2. We note that Ω_P looks very much like Ω with the space between the poles blacked out, and seems to have a higher intrinsic quality than Ω_M; a high quality image seems to lie behind the screen of blacked out pixels. This fact suggests that we are unconsciously performing an extensive interpolation on Ω_P, following some unknown set of rules that allow us to see smooth contours when none exist. Ideally, our interpolation algorithm should be able to per-

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。



(a)

(b)

Fig. 7. Interpolating the mean image. (a) The pole image for Lenna. (b) The interpolated image for Lenna.

form just as well; however, for the moment we will have to settle for a more modest performance that is achieved by higher level visual processes and leave the rest of the problem for future research.

2) The Interpolation: Given Ω_P, we only need to interpolate the regions of the image that lie between the poles. In order to accomplish this task, we utilize the pole intensities in the neighborhood of the pixel to be interpolated f_{ij} (where the ij indicates that the pixel is located at the intersection of the ith row and jth column of the image). Our technique, motivated by that described in [26], is to look in each angular direction, θ, away from f_{ij} in order to find the amplitude of the closest pole (in the sense of the Euclidean distance), $p(\theta)$, in that direction. The value of f_{ij} is then determined by

$$f_{ij} = \int_0^{2\pi} \frac{p(\theta)}{d_E(\theta)}\, d\theta \cdot \frac{1}{\int_0^{2\pi} \frac{d\theta}{d_E(\theta)}} \tag{1}$$

where $d_E(\theta)$ is the Euclidean distance between the centers of f_{ij} and the pixel belonging to the closest pole pixel in the direction θ. Other distance measures could be used; however, $d_E(\theta)$ seems to work well and we have not investigated others at this point.

In practice, we can only search for the closest pole pixels in a finite number, eight in our case, of solid angles. More explicitly, our approach is to divide the region surrounding f_{ij} into sections whose angular extent is $\pi/4$ rad. The center of the kth section occurs at the angle

$$\theta_k = \frac{k\pi}{4}, \qquad k = 0, 1, \cdots, 7. \tag{2}$$

The amplitude of the closest pole pixel in the solid angle defined by θ_k is denoted $p^*(\theta_k)$, and its distance from the interpolated pixel is $d_E^*(\theta_k)$. With these definitions, the interpolated pixel amplitude is given by

$$f_{ij} = \sum_{k=1}^{8} \frac{p^*(\theta_k)}{d_E^*(\theta_k)} \cdot \frac{1}{\sum_{k=1}^{8} \frac{1}{d_E^*(\theta_k)}}. \tag{3}$$

An example of this search is shown in Fig. 8, where the light-gray squares indicate poles, the solid-black square indicates a target pixel that is to be interpolated, the medium-gray squares specify which pole pixel is closest to the target in each solid angle, and the blank regions contain pixels that do not affect the interpolation. The weighted sum of the amplitudes associated with each of these "close" poles then becomes the result of the interpolation. Each of the blank regions will be interpolated successively, with each interpolation being independent of the previous ones.

3) Pole Hardening: The above procedure works reasonably well; however, there is occasionally a problem near the region boundaries in Ω, where these boundaries can generally be identified by searching for contiguous 4×4 leaves in the mean tree. The problem is that pixels to be interpolated on one side of a boundary may have their interpolated amplitudes unduly affected by poles on the other side of the boundary "leaking" through. This effect is demonstrated in Fig. 8, where the poles corresponding to 4×4 leaves, called edge poles, are assumed to occur on a region boundary (the pole sizes are as in Fig. 7(b)). The pole labeled "A" is thus on the opposite side of the boundary to the pixel being interpolated, and will have an undesirable effect on the interpolation.

In order to solve the above problem, the interpolation is done in two stages. The initial stage performs what we call pole hardening, and uses only the poles corresponding to 4×4 blocks to interpolate a subset of the pixels. The objective is to "fill in" the regions between the edge poles so that they can be counted as poles during the next stage, which is identical to the general interpolation discussed in the previous section. Once the boundary is filled in, it is no longer possible for a pole on the wrong side of the boundary to leak through since a boundary pixel will be the closest pole in that direction. We have experimentally found that a pole of size 2×2 for the 4×4 leaves gives the best boundary interpolation.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

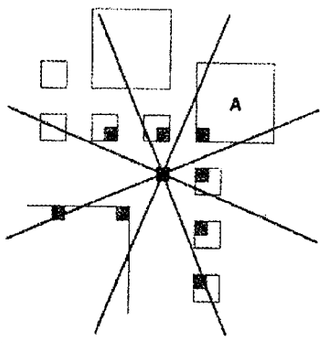


Fig. 8. Interpolating the pole image. The black pixel is to be interpolated, and the medium-gray pixels are the closest pole pixels in each direction.

The pixels interpolated during the hardening procedure are shown as medium-gray, black, and striped blocks in Fig. 9 for the four different possibilities of 4×4 vector configurations. Each target pixel is interpolated according to a simple weighted average of some of its neighboring 2×2 poles. Different numbers of poles are used depending on the configuration of 4×4 blocks: the medium-gray pixels in Fig. 9 use only the two closest poles; the black pixels use the three closest; and the striped pixels the closest four. The weighting scheme is based upon the distance from the target pixel. Thus, in our algorithm, we weight the closest 2×2 pole twice as heavily as the others (which all have equal weight). Because of the geometry, there is a unique "closest pole" for each target pixel. We show the effect of hardening on the example of Fig. 8 in Fig. 10. As can be seen, the pole labeled "A" no longer affects the interpolation.

4) Choosing the Size of the Poles: As was mentioned in Section III-B2, we need to vary the size of the poles to correspond with the size of the region from which they are derived in the mean tree. This choice determines the amount of smoothing done by the interpolation procedure. A pole size equal to the block-size implies no interpolation, while a small pole size implies that areas known to be homogeneous in their means will be overly affected by the surrounding poles. In our approach to this problem, we first set the pole size for the 4×4 regions to be 2×2. This size was chosen in order to get a satisfactory quality first stage, or hardening, interpolation. We then tested a number of possible pole sizes for the other three kinds of regions to find the one providing the lowest squared error between Ω and Ω_I. The results for two cases are shown in Table I for several images from our test data base (see Fig. 17). Case 1 in the table is the one used by our coders,

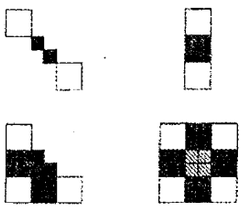


Fig. 9. Hardening boundary pixels. Each of the 4 possible cases displayed. The filled in regions are shown by the small squares.

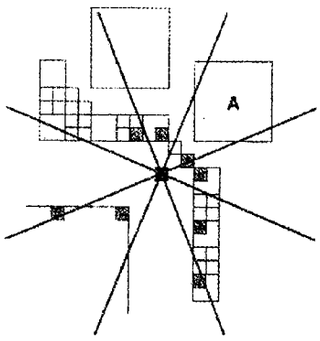


Fig. 10. An example of hardening. The boundary has been hardened and pole A no longer affects the interpolation.

TABLE I
THE CHOICE OF POLE SIZE

Parameter	Region Size	Pole Sizes (Pixels) Case 1	Case 2
Region size	32×32	24×24	16×16
	16×16	12×12	8×8
	8×8	6×6	4×4
	4×4	2×2	2×2
Sena PSNR [dB]		24.16	24.26
Sarnoff PSNR [dB]		23.70	23.85
Cohn PSNR [dB]		26.36	26.03

while case 2 is a reasonable alternative. The size of the poles also affects the appearance of Ω_I, since the amount of smoothing increases as the pole size decreases. Images with higher amounts of smoothing tend to look more nat-

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　9

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

ural; however, they are not optimum in terms of reducing the coding load on the subsequent stages.

IV. Processing of the Residual Image

This section describes how we segment and quantize Ω_R. The discussion starts by introducing visual entropy (VE), which this stage uses as an indicator of perceptual importance. This is followed by the algorithm used to segment Ω_R according to the VE, including a definition of the random texture class of region. The section ends with descriptions of how each region type is quantized.

A. Visual Entropy

The overall perceptual quality of a reconstructed image is difficult to define without extensive subjective testing; however, we do know that subjective quality is highly dependent on the coding fidelity in specific, perceptually important, local areas, such as the eyes in a head-and-shoulders image. This evaluation of quality is further complicated by subtle interactions between degradations in different regions. Generally, we can say that the gestalt quality of an image, that is, the overall effect experienced by an observer, tends to be more closely influenced by the maximum local degradation rather than by the sum of the local degradations.

As a conceptual guide to our study of variable rate perceptually based coding, we define the VE of a local region of an image as the minimum number of bits per pixel (b/p) needed to make the coded version of the region in question indistinguishable from the original. Thus, low detailed regions, which can be encoded with high fidelity at a low rate, have a low VE, while perceptually important detailed regions will have a high VE. However, detail and VE are not synonymous, since some high detail regions, for example those with random structure, can often be coded at a low bit rate without introducing significant perceptual distortion. Although we have no accurate way to compute the VE of a region, it is nevertheless a useful concept in image coding when the goal is high perceptual quality, rather than high SNR (or low MSE). The challenge we face is to find a coding scheme that allocates bits in proportion to the local VE.

Many coding schemes developed over the past ten to fifteen years have attempted to exploit the human visual system (HVS) in various ways. See for example [12], [27]–[29]. None of these, however, have been particularly successful in improving the performance of coding algorithms. The main problem with these methods is that the psychovisual experiments done to measure the properties of the HVS [30] are very simple and controlled in nature and typically do not generalize to the more complex stimuli found in natural images. Also, as explained by Nill [28], the popular method of weighting the importance of DCT-coefficients, see [12], runs into additional problems since the DCT coefficients do not have a direct frequency domain interpretation. In addition, depending on the stimuli, many high frequency DCT-coefficients may be very important for perceptual quality. We have found that the approach of variable-size segmentation combined with tailored coding algorithms for each segment, as described in this paper, is an effective way to exploit the HVS.

B. Segmenting the Residual Image

The residual image is partitioned with a process we call the VE segmentation. This algorithm consists of two parts: a coarse segmentation that determines the block size with which a region is to be encoded; and a further classification for those regions that reach the 4×4 level to determine which of three possible 4×4 codebooks to use. As will be seen, our classes are defined by fairly simple parameters, and it should be possible to obtain much better performance by using semantic information to assist the classification, however, this is a matter for future research. An example of Ω_R for "Lenna" is shown in Fig. 11. Note that the intensities have been shifted upwards by 128 in the figure.

In our classification approach, each image region is assigned to one of four somewhat overlapping categories. Initially, we separate the regions into high detail and low detail classes. Thus, high detail regions having high VE form the high VE class, while the low detail and random texture (having high detail but low VE) regions form the low VE class. These definitions will be clarified as we progress. A flowchart showing the processing for each block size is shown in Fig. 12 and should be referred to in the subsequent descriptions. In this figure, the values of T_V are the variance thresholds used in the segmentation. The technique used to select appropriate threshold values will now be discussed.

A key issue here is the method employed to assess the amount of detail in a block. We have examined two techniques to make this assessment: a calculation of the variance of the pixels in the block, and the relative energies in the high and low frequency bands (obtained using two-dimensional quadrature-mirror filters [5]). The low frequency band was designed to occupy a square region in the frequency domain centered at 0 rad and rolling off at $\pm\pi/2$ rad in each frequency dimension. We performed simple tests by calculating these parameters for regions of images containing different amounts of perceived detail and found no observable benefit with the more complicated ratio-of-energies approach. Our conclusion was that the block variance is a simple and adequate indicator of detail. Other detail indicators have been used in previous studies of adaptive transform-coders. In [31], the energy of the ac DCT-coefficients measures the activity level, while in [32] the sum of the absolute value of these coefficients is used. Buhler [22] investigated the use of a complicated activity detector to discriminate between five different types of image regions.

1) The Coarse Segmentation Strategy: The purpose of this initial stage is to isolate those regions with low detail into blocks that are as large as possible. The algorithm starts with 32×32 blocks and proceeds in much the same

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

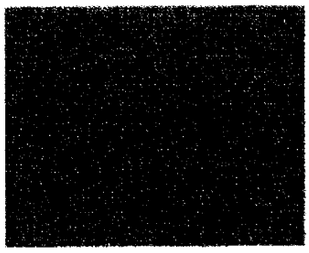


Fig. 11. The residual image for Lenna.

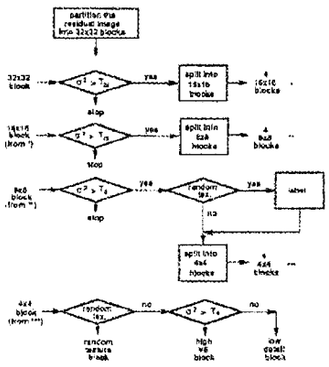


Fig. 12. The flowchart for the VE segmentation.

way as in the mean algorithm, except that the decisions to terminate or to go further into what we call the VE tree are only based upon a calculation of the block variance. If this variance is below the threshold for that block size, either T_{32}, T_{16}, or T_8, then it is declared to be a leaf node. On the other hand, if this variance is above the threshold then the block is split into four pieces, each of whose block variance is examined in turn to see if it should be a leaf. As with the mean tree, the branches never extend below the 4×4 level. The resulting segmentation is dependent on T_W, since a high threshold increases the amount of detail in the blocks at this stage.

After visually examining many kinds of blocks with different amounts and types of detail from our training set, we settled on values for T_{32}, T_{16}, and T_8 of 60, 80, and 80, respectively. These thresholds were chosen conservatively low in order to maintain very low rates for the encoding of the low-detail regions. Higher thresholds would force more bits to be used in the coding of the leaf blocks in order to get the same amount of perceived degradation.

2) The Random-Texture Class: An important type of image region having a large amount of detail but a low VE is the random texture class. This category of region occurs frequently in many different kinds of images, and essentially consists of a high variance homogeneous region without regular structure, such as the fur on an animal or the leaves of a tree seen from a distance. These regions are significant, since it is possible to encode them with a very high squared error without degradation being apparent to the viewer of the image, which is why they have a low VE. The perceptual phenomenon behind this effect has not been extensively studied in the case of random texture; however, it is related to edge masking [30] and the visibility function of Netravali and Prasada [29]. This masking behavior implies that, in many cases, it will be worthwhile to use side information to identify these regions to the receiver, allowing us to encode them in a special manner. We will now describe the technique that we use to detect these areas in an image.

As a first comment, we note that it is impossible to identify small isolated regions as random texture, since they would appear much the same even had they originated in a highly structured area of the image. Thus, to identify a block as belonging to the random texture class, the block must exceed a minimum size and must "fit in" with its surroundings. Heuristic experiments have shown that the minimum size of a random texture region must be on the order of 8×8 pixels. Also, in order to avoid problems with region boundaries or edges, this minimum size block must be surrounded by other pixels that also lie in the same statistical category. Thus, in scanning an image for random texture, we take each 8×8 block and subject it to two fundamental tests: the block must lie in a region of reasonably uniform, and fairly large, variance and must not contain any periodic structures.

The candidates for the random texture class are those 8×8 blocks that were identified as containing high detail during the course segmentation described in the previous section. Since the receiver also knows which vectors are candidates, this knowledge can be exploited to reduce the amount of side information needed to identify the location of random texture blocks. The procedure is to label all 32×32 and 16×16 blocks in the image (nonterminal nodes in the VE tree) that contain a candidate high detail 8×8 block. Each of the labeled 32×32 blocks is then searched to see if any of its high detail 8×8 blocks fall into the random texture class. One bit is used to specify the result of this decision. If random texture was found, then each of the labeled 16×16 blocks belonging to this 32×32 block must be checked to see if it contains random texture. The result of each examination is coded with one bit. All of these 16×16 blocks that contain random texture are then searched to identify which of their 8×8 can-

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

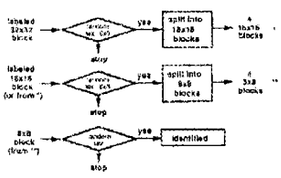


Fig. 13. Coding the random texture segmentation.

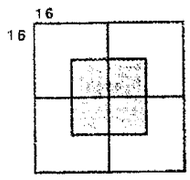


Fig. 14. The texture neighborhood.

diates belong to this class. One bit is used for each 8×8 candidate. A flowchart describing this procedure is shown in Fig. 13.

3) Identifying Random Texture: Random texture is identified in a series of tests. We initiate the first test by examining the neighborhood of each high detail 8×8 region. The neighborhood consists of the 16×16 block of pixels that has the 8×8 block of interest at its center, and is shown schematically in Fig. 14. As can be seen, the neighborhood defines a 4-pixel border zone around the block under analysis. This 16×16 block is then broken up into five pieces of size 8×8, four nonoverlapping blocks and one in the center that overlaps each of the four others equally. The variance of the ith block is denoted by σ_i^2, and the average of the five variances by $\sigma_{i,\text{avg}}^2$. The block of interest is passed onto the next test as a random texture candidate if

$$\sigma_{i,\text{avg}}^2 > 200 \tag{4}$$

and

$$\frac{|\sigma_i^2 - \sigma_{i,\text{avg}}^2|}{\sigma_{i,\text{avg}}^2} < T_{0,i} \qquad i = 1, 2, 3, 4, 5. \tag{5}$$

Through experimentation, we found that a value of $T_{0,i} = 0.65$ successfully passes only those regions that are reasonable candidates for random texture. The region is considered nontexture if either of these tests fails.

In the next test, we take the DCT of both the 8×8 block of interest and the 16×16 neighborhood block. Each set of transform coefficients is then examined for relative peaks by centering a 3×3 window at the locations corresponding to the three coefficients with the highest energy in the transformed block and then measuring the power both inside and outside the window at each position. In both transforms, we ignore several coefficients directly adjacent to the dc term since there is often power here arising from slow variations in the average pixel amplitude. Including these terms in the calculation of the ac power would desensitize the measurement. These experimentally determined "exclusion regions" are shown graphically in Fig. 15, where each square represents a DCT coefficient and the black squares are the ones excluded. The exclusion region for the 8×8 region is larger than that for the 16×16 because we only want to use the 8×8 transform to detect high frequency periodicities (that

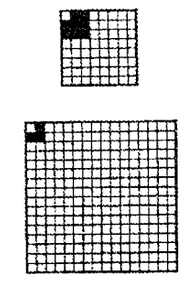


Fig. 15. The DCT exclusion regions for random texture identification.

can change orientation within a 16×16 block). At the lower frequencies, we prefer to use the increased frequency resolution obtained with the 16×16 block size.

The region is rejected as belonging to the random texture class if, in any window, the ratio of the power in the window to the power outside of the window is greater than the allowed threshold. The motivation behind this strategy is that unstructured regions in the spatial domain have their power evenly distributed among many DCT coefficients, while single edges or periodicities result in most of the power appearing in just a few coefficients. The 8×8 DCT is used to search for local periodicities that are unresolvable with the 16×16 transform. An example of this type of texture is shown in Fig. 2(e), where the image Barbara is seen to contain a pattern of high frequency stripes whose orientation changes at the local level. At the 16×16 level the DCT will not show any periodicity due to the changing orientation; however, the block does not look at all like random texture. On the other hand, the periodicity does show up in the 8×8 DCT and we can correctly classify the block. Since mistakes are very costly to the perceptual quality, these thresholds were experimentally set very low to avoid classifying any of the nonrandom texture blocks in the training set as random texture; the final values of 0.85 in the 16×16 case

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

and 2.0 in the 8×8 case gave good performance with all images tested.

The random texture category is very useful for images, such as Bridge (shown in Fig. 2(b)), which have many regions falling into this category. In other images, such as Lenna, the extra side information needed to pinpoint the texture regions may cancel out all of the savings obtained. The random texture regions isolated by our algorithm in Bridge are shown in Fig. 16.

The random texture segmentation described can no doubt be improved through the use of more sophisticated analysis tools such as mathematical morphology [33] and random field models [34], [35]. However, our simple approach gives reasonable results and validates the idea of separate coding for random texture regions.

4) The Classification of the 4×4 Blocks: Once Ω_R is segmented into blocks of different size, we perform a further classification on those blocks that reach the 4×4 level of the VE tree. We shall also refer to these blocks as vectors. At this stage, we classify each of these 4×4 blocks as belonging to one of the high VE, low detail or random texture classes. Vectors of the random texture class always appear in groups of four since only 8×8 blocks are candidates and, as a result, no extra side information is needed at the leaf level to identify them. The nontexture blocks are classified using a variance threshold T_4 to decide on the amount of detail. This threshold was set at 225 through observation of many 4×4 blocks. This classification requires one bit of side information for each encoded nontexture 4×4 block. Since 4×4 blocks always come in groups of four (corresponding to the same 8×8 parent), this side information can be viewed as a 4-ary symbol at the 8×8 level and can be entropy coded.

Classification of the 4×4 blocks allows a significant reduction in the bit rate for two reasons: the low-VE regions may be coded with less than half the number of bits than is needed for those with high VE, and the number of low detail vectors is generally much larger, typically by a factor of two, than the number of high-VE vectors. An additional advantage of the segmentation is that the codebooks become more specialized in their task, improving performance for a given codebook size. This property is due to the fact that the training set for the high-VE codebook design is no longer "corrupted" by many vectors having low VE. An alternative technique for variable rate coding applicable for the 4×4 regions has been recently developed by Riskin *et al.* [7].

C. Quantization of the Residual Image

This section describes how the various regions of Ω_R are coded to produce Ω', the coded image. Our algorithm encodes all blocks larger than 4×4 with a TC/VQ hybrid technique and all 4×4 regions with spatial domain VQ. This strategy was adopted because the ability to use larger block sizes with reasonable computational complexity makes TC inherently superior to direct spatial domain VQ when coding large regions where the detail is low. VQ is inherently superior in small regions where complexity is not such as issue. (In fact, TC is actually a special case of VQ since each combination of possible quantized DCT coefficients represents a specific $N \times N$ vector.) The problem with the DCT in detailed regions is that the DCT performs close to the optimum Karhunen–Loève transform only when the statistical characterization of the block is uniform and the interblock correlation fairly high [2], which is not the case in most areas containing detail. Also, since the amount of energy compaction produced by the DCT drops with the block size, the transform is not very effective for blocks smaller than 8×8. It thus becomes impossible for a DCT-based algorithm to isolate small detailed regions lying in blocks dominated by low detail. The result is either an excessive bit rate for the surrounding low detail or poor quality edges.

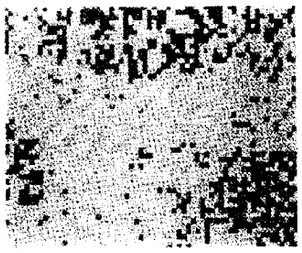


Fig. 16. The random texture regions of bridge

Since a vector quantizer uses a small blocksize and matches patterns directly in the spatial domain, we can reproduce vectors with high-frequency components as needed, enhancing the perceptual quality of the reproduced edge and avoiding spurious high-frequency noise. However, VQ performs poorly relative to TC when it is desirable to code large, fairly homogeneous, regions since complexity limits the size of the blocks that can be handled with VQ to be quite small (usually 4×4 pixels).

We have found that VQ is especially useful in encoding random texture, since the codebook can contain assorted texture patterns, each of which has many high-frequency components. Coded random texture is difficult to distinguish from the original as long as the general pattern, together with the correct amount of high-frequency energy, stays approximately the same. Low-rate transform coders are notoriously bad at reproducing high frequencies and they perform an undesirable lowpass filtering operation on the texture regions.

1) Codebook Design: One of the most difficult tasks in the design of a vector quantizer is the design of the codebook. We base our design on the popular generalized Lloyd algorithm (GLA) [36], with the mean-squared error (MSE) distortion measure. The performance of the GLA

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

is improved through the use of simulated annealing via the "SA-C" algorithm described in [37], which is similar to the "SA-GLA" technique in [38].

The key features of the GLA method (with and without simulated annealing) are a) it uses a training set of typical data to define an implicit source probability density function, and b) it performs an iterative operation where each iteration maps a codebook into a new and improved codebook. The use of a training set is essential in most VQ applications, since it is generally very difficult and impractical to find an analytically tractable model of the source. This point is especially true in image coding, where the fact that the source may be extremely nonstationary causes major modeling difficulties. Typical codebooks may work well when encoding data that has the same properties as the training set, but often result in much worse performance when presented with other data. For this reason, we need a training set that is robust in the sense that every input vector will be similar to at least part of the training set with high probability. In this work, we take our training set from many different 512×512 images representing many types of scenes, from head-and-shoulders photographs to natural landscapes and aerial views, as is shown in Fig. 17. The names of the images in this figure are, from top to bottom and left to right: karen, man, museum, artisle, woodhills, peppers, sailboat, clown, and colet.

2) Coding in the Transform Domain: In this algorithm [8], the DCT coefficients of each block are grouped into vectors that are then individually coded using VQ. The DCT coefficients were initially encoded individually using the Chen–Smith approach [1], but it was later found that it is possible to further reduce the bit rate by grouping coefficients together and using VQ to code them as has been done in [39]. The main advantage of this technique is that fractional bit assignments can now be used for each coefficient. This possibility increases the flexibility of the design and allows us to use lower bit rates than would be possible in scalar quantization. In addition, TC/VQ also allows us to exploit any residual correlation in the block not removed by the DCT.

In grouping the coefficients into vectors, we take into account the relationship between the DCT coefficients and the spatial characteristics of the block. For instance, when there is a horizontal edge in the spatial domain, there is a corresponding group of relatively high magnitude vertically oriented coefficients in the transform domain. On the other hand, when the block has low detail, we find that the coefficients located at a similar distance from the dc coefficient tend to have similar variances. In our case, we know *a priori* that the blocks to be encoded using TC have small variances and low detail. Thus, since every coefficient in a vector is "allocated" the same number of bits, we group the coefficients diagonally so that each vector component will have approximately the same variance. High frequency coefficients in the transform domain are simply set to zero since they never have significant amplitude for these kind of blocks. Also, since the number of coefficients that need quantizing is fairly stable over the ensemble of blocks, we can avoid the need to transmit side-information indicating which coefficients are quantized. We have found that the using VQ instead of scalar quantization allows us to drop the bit rate by 20%–40% without loss in quality. Although subtraction of Ω_1 ensures that the dc coefficient will be fairly small, it is still necessary to quantize this coefficient with additional bits if blocking is to be avoided. The actual assignments of coefficients to vectors will be discussed in Section VI.



Fig. 17. The training images.

We note that an additional advantage of using VQ to code the DCT coefficients is that the linearity of the DCT can be exploited to avoid the necessity of performing an inverse DCT at the receiver. Reconstruction of the coded vector simply requires the precomputation of the inverse DCTs of the k-dimensional blocks formed by setting all of the coefficients to zero except those belonging to a single codevector. The number of precomputations required is equal to the total number of codevectors in the codebooks, with each codevector being involved in one computation. When receiving data for a particular block, the decoder will receive a set of indices $\{i_1, i_2, \ldots\}$, a separate index for each of the vectors in the block of DCT coefficients. Each index points to the k dimensional block in the spatial domain that is the result of the appropriate precomputation. The coded vector is reconstructed as the sum of these blocks.

An interesting alternative approach for coding the low-detail regions would be to apply the generalized-DCT method of Gilge *et al.* [40]. This technique, which would allow direct transform coding on several differently sized contiguous blocks, has not investigated here; however, it offers an interesting topic for future research.

3) Coding in the Spatial Domain: In our coding technique, we use three different spatial domain vector quan-

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　14

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

tizers, each designed to code a different class of 4×4 vector. When the coding rate is low, it is feasible to directly quantize the residual 4×4 blocks using basic VQ with one codebook per class. However, since codebooks have a limited practical size (we set a maximum of 256 codevectors per codebook), we need to alter our strategy if we want to further improve the fidelity. The gain-shape product code method of Sabin and Gray [41] is an effective procedure to achieve this goal. In the first stage of this technique, the shape of each input vector is quantized by finding the vector in the shape codebook that is most highly correlated with the input. After the shape codevector is found, the corresponding correlation value is quantized using the gain codebook. The product of the quantized correlation and the closest shape codevector forms the output of the vector quantizer. Codebook design procedures can be found in [41].

In an additional effort to lower the bit rate, we applied independent Huffman coding to the indices of the three codebooks. The Huffman tables were generated by encoding the training set to estimate the relative probabilities of occurrence between the codevectors. This type of entropy coding typically reduces the number of bits generated by the coder; however, due to mismatches between the input image and the training set, the Huffman assignments were occasionally found to degrade the performance by adding a few hundred bits to the total. To avoid this problem, we compare the bit rate obtained with and without Huffman coding for each spatial domain codebook. A single bit is then used to indicate the best strategy for each codebook.

4) The Supplementary Codebook: It may still be difficult to code many of the 4×4 high-VE blocks of some images even though the codebook has been carefully designed and does an effective job of encoding every vector in the training set. The reason for this problem is that the input images often contain vectors with high VE that are poorly represented in the codebook. This effect is exacerbated when we are striving for a low bit-rate and are forced to use small codebooks. On characterizing the regions that cause problems, we find that they usually have a high VE, and that they are perceptually most critical when they contain well-defined edge segments. Coding can thus be improved by generating and sending side information describing the attributes of the critical areas unique to the image being encoded. As long as the bit rate of this information is kept low, the coder will still achieve a high compression factor. The VQ codebook provides a natural and unique way to generate this side information.

Thus, following the technique we presented in [8], we generate a supplementary training set consisting of those vectors that are coded with the lowest fidelity, as measured by the MSE, and then use this training set to form the 32 vector supplementary codebook (SCB). After being designed, the SCB is then appended to the main codebook and every high-VE 4×4 vector is recoded using the joint codebook. Quantizing each pixel to 5 b allows the SCB to be sent to the receiver using 0.01 b/p for a 512×512 image. A disadvantage of the SCB approach is that it requires an initial pass through the image in order to design the codebook; however, this is the price that must be paid if the codebooks are to be tailored to match the unique characteristics of the input images. If the cost (in memory and processing delay) of implementing this adaptive technique is acceptable, the SCB method can enhance the performance of virtually all VQ schemes.

V. The Smoothing Algorithm

One common problem with basic TC is that low rates often cause the boundaries between blocks to be visible even when the fidelity within each block is high. These blocking effects can easily be removed by quantizing with a larger number of bits; however, since our objective is compression, a better solution is to use a smoothing algorithm. In the following, we describe the heuristic smoothing scheme used to generate a smoothed output image $\hat{\Omega}'_s$ from the coded image $\hat{\Omega}'$. Although not implemented in this work, the JPEG algorithm [12] contains a description of a technique that uses prediction of the ac DCT-coefficients to reduce the blocking effects. Adaptation of this scheme to the variable blocksize environment will likely improve the performance of our coder; however, it is also likely that some additional smoothing will be necessary due to the low rates used here.

A simple approach to smoothing would be to directly postprocess $\hat{\Omega}'$ with a low-pass filter; however, although this works well within the 32×32 blocks, such a filter also unacceptably blurs high detail regions of the image. Since all of the blocks encoded using the TC procedure must belong to the low detail class, these blocks can be selectively smoothed without introducing artifacts. We thus implement smoothing by forcing each low-detail block, including those at the 4×4 level, to fit with the neighboring regions of $\hat{\Omega}'$ by interpolating only the boundary pixels of the block itself. Since the interpolation is only done within low-detail blocks, it is not possible to adversely affect any surrounding high-detail vectors. Degradations due to smoothing may appear in rare cases when edges occur directly along block boundaries and one side or both sides of the edge belong to the low-detail class.

Our smoothing algorithm takes place in two stages. The task of the first stage is to use data from $\hat{\Omega}'$ to smooth the pixels forming the block boundaries, while that of the second is to process the pixels one position away from the boundary. The second stage operates on the output of this first stage; however, due to their small size, the 4×4 blocks with low-detail are omitted from the second stage. We identify pixels in $\hat{\Omega}'$ according to their row number i and column number j. If we describe the pixels in $\hat{\Omega}'$ by f, those in the smoothed image after the kth stage g_k, and call the set of pixels to be smoothed in the kth stage Ψ_k, then the smoothing operation can be described as follows. For the first stage we have

$$g_1(i, j) = \begin{cases} f(i, j) & (i, j) \notin \Psi_1 \\ s(i, j) & (i, j) \in \Psi_1 \end{cases} \tag{6}$$

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

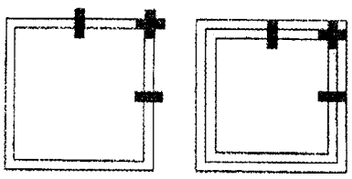


Fig. 18. The two stages of smoothing. (a) First stage. (b) Second stage.

where the $s(i, j)$ are the smoothed pixels given by

$$s(i, j) = \frac{[f(i, j) + f(i, j+1) + f(i, j-1)]}{3} \quad (7)$$

if (i, j) is on the top or bottom of a block;

$$s(i, j) = \frac{[f(i, j) + f(i+1, j) + f(i-1, j)]}{3} \quad (8)$$

if (i, j) is on a block side; or

$$s(i, j) = \frac{[f(i, j) + f(i+1, j) + f(i-1, j) + f(i+1, j+1)]}{4} \quad (9)$$

if (i, j) is a corner pixel of a block. The second stage is implemented in a similar manner, except that the pixels interpolated are different. A diagram illustrating smoothing for a 16×16 block is shown in Fig. 18. In this figure, the lightly shaded regions indicate which pixels are to be smoothed, while the darkly shaded pixels identify which pixels are used in the calculations according to the location of the pixel to be smoothed. The smoothing is done only in the direction perpendicular to the block boundary in order to avoid problems when a neighboring block having a sharp edge comes up to the block boundary. The "cross" pattern is used at the corners because it is a combination of horizontal and vertical smoothing.

The drop in the squared error between Ω and the output image as a result of smoothing is generally between 0.1 and 1.0 dB, and many annoying effects are removed. In all of our experiments, we have never encountered an increase in the squared-error after the application of this algorithm.

VI. Simulation Results and Codebook Sizes

In order to compare the coding fidelity among the different schemes, it is necessary to use an objective measure of distortion. Thus, following common practice we use the peak signal-to-noise-ratio (PSNR)

$$\text{PSNR} = 10 \log \left[255^2 \Big/ \frac{1}{K} \sum_{i=1}^{K} (x_i - c_i)^2 \right] \quad \text{[dB]} \quad (10)$$

where x_i and c_i are the ith pixels from Ω and Ω' (or Ω_3), respectively, and K is the number of pixels in Ω; however, as is well known, this measure is often not indicative of the actual subjective quality and is particularly inadequate when comparing a perceptually based coding method with a method based directly on minimizing mean squared error.

We now discuss the performance of the coding algorithm described in the previous sections. Results are presented for two encoders operating at different rates. Every codebook was designed using the GLA and the appropriate training data as derived by segmenting the training images shown in Fig. 17. Huffman tables were designed in each case to allow entropy coding of the indices; however, the rate reductions obtained were typically less than 5% of the amount needed without Huffman coding for each codebook.

The low rate (LR) coder was designed to achieve an acceptable level of quality at a very low rate, while the high-rate (HR) coder aims at high quality at a reasonable rate. Neither of these coders gives a distortion free result in any class of region. In fact, the coders are designed to balance the fidelity so that it is approximately uniform everywhere. Since the goal of our coders is to "match" Ω' to the original as closely as possible in a perceptual sense, it is not appropriate to strive for perfect reconstruction in some regions when the subjective quality evaluation is dominated by distortion in other regions. This strategy means that the resulting PSNR of the coded images will not be minimized by our algorithm; however, the perceptual quality will be better than in the case where the coders are designed solely on the basis of the MSE. We emphasize that we have designed these coders using a general training set, and that the performance would be substantially improved if the input images could be restricted to certain classes. Typical execution times for our algorithms on a SUN 3/260 computer with a floating point accelerator are: 2 minutes to generate Ω_I and 8 minutes to code the residual. In images with a great deal of detail, most of this 8 minutes of CPU time is devoted to searching the 4×4 codebooks, with approximately 10% of it devoted to the random-texture segmentation.

A. The Codebooks

We now discuss the methodology used to pick codebook sizes in the designs of the coders examined in this chapter.

1) Codebooks for Generating the Interpolated Image: We consider first the number of bits used to define Ω_M, which in turn allows us to form Ω_I. As an objective, we decided to quantize the mean with enough fidelity to force the average of the squared quantization-error to be less than 64 after each stage. The value of 64 was chosen fairly arbitrarily, with the heuristic objective of having the square root of this number equal to half of the peak-error

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　16

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

magnitude. This choice is not really critical since Ω_e is computed in the early stages of the quantizer and many problems are easily repaired in the subsequent stages. Recall that T_M is set to be equal to sixteen.

In order to meet the above goal, we quantized the 32×32 means to 4 b, ensuring an error in the interval $[-8, +8]$, and used Table II to determine the codebook sizes for the vector quantizers operating on the residual means. Using this table, we would initially choose sizes of 64, 128, and 256 for the 16×16, 8×8, and 4×4 and means, respectively; however, we adjust the 16×16 size to be 128 due to the sudden drop in the squared error when going to the larger size and the small number of bits (typically 200) needed for the extra fidelity. Also, the size of the codebook for the 4×4 means was increased to 512 in order to improve the performance in the high-detail regions without increasing the size of the codebook for the spatial-domain quantizer. The results of the mean segmentation with the above bit assignments is shown in Table III for each of the three test images used in this paper. The resulting PSNR's for the Ω_M and Ω_I are given in Table IV, and the bit counts associated with the mean information and quadtree descriptions are given in Table V. This last table also gives data about the VE and texture segmentation, where a double entry indicates a different segmentation for the LR and HR cases respectively. The reason for this difference is explained in the next section. The same Ω_I was used for both coders.

2) The Codebooks for the DCT Coefficients: The number of bits used per vector in the quantization of the DCT coefficients was chosen both in order to meet the desired subjective quality and to obtain balanced distortion over each region of Ω. Most of the computational effort associated with the coding is borne by searches through the 4×4 codebooks, since these predominate in most images (see Table IX) and better quality can easily be obtained by increasing the codebook size if it is thought necessary. For the LR coder we used 32, 23, and 11 b per block for the 32×32, 16×16, and 8×8 blocks, respectively; these numbers were increased to 39, 30 and 14 for the HR coder. In both cases, the training set sizes were 414, 1629, and 2255 for the 32×32, 16×16, and 8×8 blocks, respectively. We also adjusted T_E to 45 from 60 in the HR case; this was done to reduce the variation in the vectors coded in 32×32 blocks allowing higher fidelity without an increase in the codebook size. The codebook size became a problem in this case due to a dearth of 32×32 training vectors (larger training sets are needed for larger codebooks). In Table VI, we show the coefficient groupings that were used to form vectors for quantization. The same groupings were used for both coders and the coefficient labeling was done as in Fig. 19. The codebook sizes are shown in Table VII.

The total number of bits used to encode these low-VE regions is quite low and, as a result, some degree of "blocking" appears in the Ω'. An experiment was performed to determine the cause of this kind of artifact. In this test, we replaced the mean coefficients of the trans-

TABLE II
THE AVERAGE SQUARED-ERROR VERSUS CODEBOOK SIZE CALCULATED BY CODING THE TRAINING SET

Size of Region	Codebook Size 512	256	128	64
16×16			39	80
8×8		40	57	65
4×4	35	53	76	107

TABLE III
THE FREQUENCY OF EACH REGION SIZE IN THE MEAN TREE

Size of Region	Image Name Lenna	Barbara	Bridge
32×32	44	12	0
16×16	216	233	86
8×8	1 197	1 241	1 516
4×4	4 844	6 124	6 824

TABLE IV
COMPARING THE MEAN AND INTERPOLATED IMAGES

Image Name	Bit Rate [b/pl]	PSNR [dB] Mean Image	Interp. Image
Lenna	0.079	24.90	25.48
Barbara	0.098	22.28	22.78
Bridge	0.127	21.67	22.25
Fingerprint	0.075	23.71	24.32

TABLE V
BITS NEEDED FOR THE SIDE INFORMATION

Parameter	Image Name Lenna	Barbara	Bridge
Mean tree info. [bits]	3 256	3 948	4 776
Total mean info. [bits]	20 237	25 373	33 264
Classification info. [bits]	4 870/4 899	5 272/7 331	12 995
Texture-tree info. [bits]	461/489	556/605	1 092
VE tree info. [bits]	2 990/3 046	3 708/3 856	5 886

TABLE VI
THE DECOMPOSITION INTO VECTORS FOR THE DCT BLOCKS

Vector #	Block Size 32×32	16×16	8×8
0	1	1	1
1	2-6	2-4	2-3
2	7-15	5-6	4-10
3	16-28	7-13	
4	29-45	14-28	
5	46-66		
6	67-91		

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　17

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

1	2	4	7	11
3	5	8	12	
6	9	13		
10	14			
15				

Fig. 19. The numbering scheme for the DCT coefficients. The top left corresponds to the dc coefficient.

TABLE VII
THE CODEBOOK SIZES FOR THE DCT COEFFICIENTS

	Block Size					
	32 × 32		16 × 16		8 × 8	
Vector #	LR	HR	LR	HR	LR	HR
0	4	4	4	4	4	4
1	32	128	32	128	8	32
2	32	128	32	128	16	64
3	32	128	32	128		
4	32	64	32	64		
5	32	32				
6	32	32				

formed blocks by their unquantized values; however, we observed no change in the blocking problem. This result implies that enough bits are being used to describe the mean, and that the blocking effect occurs because too few bits are being used to code the ac-coefficients. To confirm this hypothesis we left the coefficients mentioned in Table VII unquantized, and found that all blocking disappeared, implying that we are quantizing a sufficient number of coefficients. This blocking effect is not a serious problem, since it can be significantly reduced with our smoothing algorithm (which was presented in Section V).

3) The Spatial Domain Codebooks: The sizes of the codebooks for the spatial domain VQ were chosen according to the same quality objectives as above. Different sizes were used for each class because of the different resolution requirements imposed by the VE. We attempt to get a similar perceptual quality in each region of Ω'. In the LR coder we use codebooks of size 16, 16, and 256 vectors for the low detail, random texture and high-VE classes respectively; the corresponding training set sizes were 10 688, 3104, and 10 162. The HR coder uses the same random texture codebook, but quadruples the size of the low-VE codebook to 64 vectors and uses 15 b for each vector having high VE. The same basic training sets are used as in the LR coder. In the HR case, each high-VE vector is coded by first quantizing the block mean to four bits. The quantized mean is then removed, and the residual coded using the Sabin–Gray method with a shape codebook of size 256. The correlation term is quantized to three bits. These mean and gain codebooks were experimentally chosen to be the above sizes by finding the best way to partition the allowed seven bit address space. Each of these scalar quantizers was designed using the Lloyd–Max algorithm on data obtained by segmenting the Ω_R corresponding to each training image. The output levels for these mean and gain quantizers are shown in Table VIII. The means to be quantized at this stage are distributed symmetrically about zero (due to the removal of Ω_I) and thus only the positive output levels are shown.

TABLE VIII
THE MEAN AND GAIN QUANTIZERS FOR THE HR CODER USED FOR QUANTIZING THE HIGH VE REGIONS

Output Levels	
Mean	Gain
2	65.9
5	72.9
8	91.8
11	109.4
15	130.9
21	159.1
28	197.8
41	255.9

B. The VE Segmentation

Once Ω_I has been subtracted, a VE segmentation on Ω_R is performed. As described in the body of the paper, the segmentation thresholds T_{16} and $T_{4\times}$, and T_1 are set equal to 60, 66, and 255 for both coders, while (as explained in Section VI-A2) T_{32} is equal to 60 for the LR case and 45 for the HR case. Also, recall that the threshold for checking homogeneity of the variance in random texture regions, $T_{\sigma R}$, was set to 0.65. The resulting number of blocks of each type is shown in Table IX.

The PSNR is very misleading when judging the effect of coding texture regions since, as was mentioned earlier, humans are very insensitive to distortions here, and our attention is usually diverted to other regions when judging quality. It is for this reason that we suggest an equivalent PSNR for coders that ignores the squared-error contributed by the random-texture regions. This measure makes sense, since viewers see essentially no distortion in the coded texture regions unless the images are examined extremely closely. An alternative approach would be to use an equivalent PSNR that is calculated using the image obtained when coding the random-texture regions with the high-VE codebooks; however, this is not attractive since the fidelity of the coded random texture is almost independent of these codebooks.

The PSNR's and the associated number of bits needed to code the images with both encoders are given in Table X, where PSNR_T is the PSNR with the distortion in the texture regions set to zero. PSNR_S is the PSNR after smoothing and PSNR_{ST} is the PSNR after both texture-region removal and smoothing.

C. The Reconstructed Images

The coded images themselves are shown in Figs. 20–22. All of the results include random texture, the supplementary codebook and smoothing unless otherwise indicated in the figure caption. It is seen that both encoders have very high PSNR's for their respective rates, and that

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

TABLE IX
THE NUMBER OF REQUIRED EDGE TYPE AC DECISIONS BY THE VE SEGMENTATION

Size of Region	Image Name		
	Lenna	Barbara	Bridge
32×32	80/53	43/20	1
16×16	249/320	202/235	65
8×8	872/1005	388/500	1934
$4 \times 4_{(a)}$	2698/3904	2655/4666	4785
$4 \times 4_{(b)}$	3253/3792	3473/3922	3875
$4 \times 4_{(c)}$	296	520	3208

TABLE X
THE PSNR'S OF THE TEST IMAGES

Parameter		Image Name					
		Lenna		Barbara		Bridge	
		Low Rate	High Rate	Low Rate	High Rate	Low Rate	High Rate
PSNR (dB) by Class	32×32	33.25	34.99	27.44	35.13	N/A	N/A
	16×16	33.71	33.76	31.93	34.52	32.06	32.52
	8×8	32.90	33.15	28.88	33.065	32.20	32.73
	$4 \times 4_a$	25.69	27.68	22.33	23.91	24.27	25.96
	$4 \times 4_b$	29.13	30.19	26.97	29.09	28.76	29.66
	$4 \times 4_{(c)}$	33.46	33.46	32.48	32.48	32.72	32.72
PSNR [dB]		30.20	31.56	26.30	27.82	23.69	26.45
$PSNR_Y$ [dB]		30.81	31.79	26.65	28.03	26.87	28.20
$PSNR_U$ [dB]		30.43	31.62	26.37	27.29	25.73	26.46
$PSNR_V$ [dB]		30.86	32.09	28.79	28.11	26.94	28.21
rate [b/p]		0.277	0.363	0.389	0.534	0.501	0.630



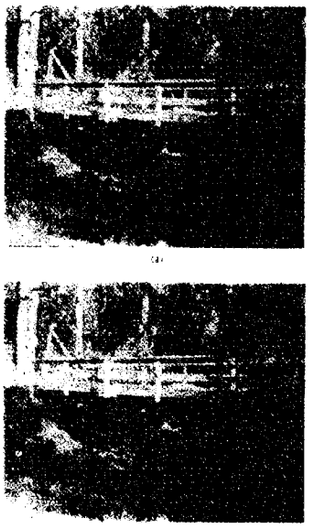
(a)　(b)

Fig. 20. The coded versions of Lenna. (a) The HR case—0.363 b/p. (b) The LR case—0.277 b/p.

neither of them suffers from blocking in the edge regions. There is some blurring in the areas with "medium" detail that have been classified as having low detail, however, this problem is not severe.

The HR coder gives results with a much higher fidelity than the LR coder, however, the quality is saturating and significantly more bits would be necessary if it is desired to clean up the remaining high-VE problem areas. The problem is caused by the fact that complexity limits the VQ codebook size to be quite small. In both the LR and HR cases, the codebook containing the shape information is constrained to contain less than 256 patterns, not enough to obtain truly high fidelity in all cases. This point becomes clearer when we consider the striped trousers in the original "Barbara" image (see Fig. 1(c)). In this region, there are a large number of possible edge orientations.

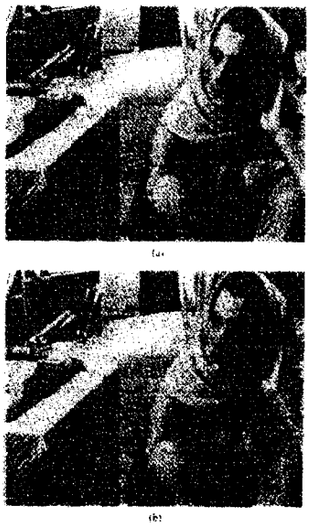
本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。



(a)

(b)

Fig. 21. The coded versions of Bridge. (a) The HR case—0.650 b/p. (b) The LR case—0.363 b/p.



(a)

(b)

Fig. 22. The coded versions of Barbara. (a) The HR case—0.554 b/p. (b) The LR case—0.385 b/p.

certainly more than 256, and it is impossible to find a suitable codevector in each case. There are a few options if better quality is desired, some of these are: to add another level to the VE quadtree and code using 2×2 blocks when necessary; to perform a further classification of the 4×4 blocks into groups with different edge orientations, each to be quantized with a unique codebook [6], or perhaps use a 4×4 DCT (as has been done by Chen [14]), where the rate can be increased without seriously affecting the complexity.

In examining the coded version of Bridge, we note that the perceptual distortion in the texture regions is quite low even though the codebook used is small. Some distortion is visible. Specifically, the texture regions appear somewhat lowpass filtered when examined closely, however, other degradations dominate in determining the perceived fidelity. Including independent coding of the random-texture regions has allowed us to save 0.092 b/p in the HR case and 0.026 b/p in the LR case. The PSNR drops by 0.5 and 1 dB for the LR and HR cases, respectively. The improvements are less dramatic, but still significant, with the other two images, especially with the HR coder. The overall rate for Bridge is high because of the large number of high detail vectors (many of which are not classified as random texture) in the image.

It is interesting to note that the PSNR's obtained in the different regions of the images are very different, but that the "annoyance" due to the distortion is fairly similar over most of each coded image.

D. Comparison with Other Coders

In this section, we present both numerical and qualitative comparisons between our work and other published results. In making these comparisons, we will focus on the 512×512 image Lenna, which is the most popular image to code in the literature. Although PSNR is not an adequate measure of perceived quality and our coding strategy has been oriented toward maximizing perceived quality rather than PSNR, our PSNR results are among the best that have been reported, with superior PSNR values at such low rates being obtained only with subband coding as developed by Woods *et al.* [5]. Woods did not code the 512×512 version of Lenna, but a recent result using a similar scheme [42] gives a PSNR of 31.27 (dB) at a rate of 0.27 b/p. A newer result by Kim *et al.* [43]

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　20

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

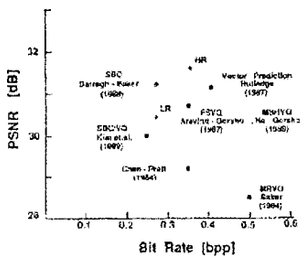
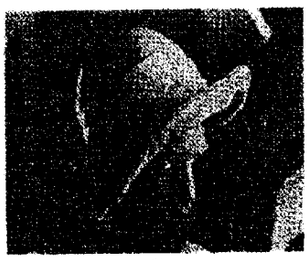


Fig. 23. A performance comparison (Lenna).



(a)



(b)

Fig. 24. The results of Ho and Aravind, rate Ho—0.36 b/p, (b) Aravind—0.35 b/p.

uses a hybrid of subband coding and finite-state vector quantization (FSVQ) to generate a PSNR of 30 dB at a rate of 0.25 b/p. It should be noted, however, that Lenna is quite an easy image to code (especially using subband coders), since it has very little energy at the higher spatial frequencies. Since our coder has been trained on general images with essentially "full bandwidth," we should really define a new training set for low bandwidth images if we desire to obtain the best possible performance when coding Lenna. Actually, we have used the green band from the original USC color image of Lenna, while some researchers have used the monochrome (Y band). The latter tends to give higher PSNR values for a given coder. In Fig. 23, we display our HR and LR results for Lenna together with several results from the literature: subband coding, as implemented by [42]; vector prediction [44]; variable rate finite-state VQ (FSVQ) [45]; subband coding and FSVQ [43]; multistage hierarchical VQ (MSHVQ) [46]; Chen–Pratt transform coding [31] (based on an unpublished implementation); and mean-removed VQ (MRVQ) [10]. Both classified VQ [6] and gain-shape VQ [47] perform essentially the same as MRVQ.

We believe that the perceptual quality of our coded images is better than would be indicated from the PSNR values, but it is difficult to provide a quantitative assessment of perceptual quality other than by informal comparisons with other published coding results. It is possible to make more complete comparisons between our work and that of Aravind and Gersho [1], Aravind [45], and Ho and Gersho [46], since we have access to the actual image files. In Fig. 24 we show the 30.77 (dB), 0.35 b/p variable rate result of Aravind and the 30.95 (dB), 0.36 b/p result of Ho and Gersho. Both of these coders give an output quality that is between our LR and HR results. Aravind's image suffers from a substantial amount of blocking and staircasing in the low- and high-detail regions, respectively. Ho and Gersho's image, on the other hand, has a slightly fuzzy overall appearance, i.e., a lack of crispness, and has a noticeable distortion along the edges as a result of its multistage structure. These artifacts give these images a somewhat unsatisfying artificial quality. On the other hand, the peak errors occurring near sharp edges in our LR result are not present in either of these coders. Our HR result is both objectively and subjectively superior to these results and operates at a similar rate. The superior performance obtained with the new algorithms is due both to the design philosophy, that concentrates the coding power in the regions where it is most needed, and the supplementary codebook that tailors the codebooks to the most difficult parts of the input images.

VII. Conclusions

This study has demonstrated that quadtree-based image segmentation can be an effective and efficient mechanism for isolating blocks of distinct perceptual significance and thereby allowing different coding strategies that are perceptually suited to the individual segment categories. We have introduced a number of novel techniques that not only serve as building blocks for our coding system, but also can be of value in other image coding applications. Central to our coding approach is the use of variable rate coding that conceptually corresponds to the idea of visual

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

entropy, i.e., the spatial distribution of the coding resolution that tends to minimize the maximum perceptual degradation that can occur in the coded image. Perceptually, we have obtained high quality image reproduction at rates between 0.25 and 0.7 b/p, depending on the nature of the test image. Objective measurements of performance using PSNR, while of moderately limited use in assessing quality, has nevertheless given values comparable to the best prior art in image coding.

An important component of our work is the special attention given to random texture regions. Including the random texture class has allowed the identification of high detail patterns where the perceptual importance (and the VE) is low, allowing us to code these regions with a low bit rate. This class occurs quite commonly in "natural" images where one may be looking at the leaves of a tree from a distance, or perhaps the fur of an animal. With this technique, the bit rate necessary to provide a certain level of perceptual quality in the coded image can be substantially reduced in many cases. An additional advantage to identifying classes such as random texture is that it then becomes possible to define a modified PSNR that weights different regions according to their estimated VE. This weighting increases the perceptual relevance of the measure by suppressing the contributions to the PSNR that do not affect the fidelity.

In summary, this work has introduced a new direction of research, where class-specific image coding of regions is achieved by segmenting an image into variable size regions which are then classified into perceptually distinct categories. Our coding algorithm is a first step in a new direction and demonstrates the potential of segmentation based image coding. We anticipate that further research will lead to substantially improved performance with this approach.

References

[1] W. Chen and C. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol. COM-25, no. 11, pp. 1285–1292, Nov. 1977.

[2] R. Clarke, *Transform Coding of Images*. New York: Academic, 1985.

[3] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 157–166, Mar. 1982.

[4] N. Nasrabadi and R. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, pp. 957–971, Aug. 1988.

[5] J. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.

[6] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. COM-34, pp. 1105–1115, Nov. 1986.

[7] E. Riskin, T. Lookabaugh, P. Chou, and R. Gray, "Variable rate vector quantization for medical image compression," *IEEE Trans. Med. Imaging*, vol. 9, pp. 290–298, Sept. 1990.

[8] J. Vaisey and A. Gersho, "Variable block size image coding," in *Proc. ICASSP'87* (Dallas, TX), Apr. 1987, pp. 1051–1054.

[9] J. Vaisey and A. Gersho, "Variable rate image coding using quadtrees and vector quantization," in *Proc. EUSIPCO-88* (Grenoble, France), Sept. 1988, pp. 1133–1136.

[10] R. Baker, "Vector quantization of digital images," Ph. D. dissertation, Interim Syst. Lab., Stanford University, June 1982.

[11] R. Aravind and A. Gersho, "Image compression based on vector quantization with finite memory," *Opt. Eng.*, vol. 26, no. 7, pp. 570–580, July 1987.

[12] *JPEG: Draft Technical Specification*, version 8, Aug. 14, 1990.

[13] S. Nasrabadi, "Use of vector quantizers in image coding," in *Proc. ICASSP'85*, Mar. 1985, pp. 125–128.

[14] C. Chan, "Adaptive transform coding via quadtree-based variable block DCT," in *Proc. ICASSP'89*, May 1989.

[15] R. Wilson, "Quadtree predictive coding: A new class of image data compression algorithms," in *Proc. ICASSP'84*, Apr. 1984, pp. 35.3.1–35.3.4.

[16] R. Lerman and M. Kunt, "Adaptive split and merge for image analysis and coding," in *Proc. SPIE Conf. on Visual Communic. Image Processing* (Cannes, France), Dec. 1988, pp. 2–9.

[17] D. Farrelle and A. Jain, "Quadtree hierarchical image coding," in *Proc. SHLCOM*, 1986, pp. 49.2.1–49.2.5.

[18] C. Buccelati, J. Cobbs, and A. Moser, "Error free compression of medical X-ray images," *Proc. SPIE Int. Soc. Opt. Eng.*, pp. 269–276, 1988.

[19] P. Strobach, "Image coding based on quadtree-structured recursive least squares approximation," in *Proc. ICASSP'89*, 1989, pp. 1961–1964.

[20] N. M. Nasrabadi, S. E. Lin, and Y. Feng, "Interframe hierarchical vector quantization," *Opt. Eng.*, vol. 28, pp. 717–725, July 1989.

[21] Y. Bublet and M. Fontier, "Hierarchical picture coding using quadtree decomposition," in *Proc. 8th Annu. Tech. Symp. Opt. Electron. Opt. Appl. Sci. Eng.* (The Hague, The Netherlands), Apr. 1987.

[22] Y. Bublet, "Etude d'une méthode microchirurgie pour le codage à faible débit des images numériques," CNRS Telecommun. Rep. 87-04, May 1986.

[23] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 188–260, June 1984.

[24] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image coding techniques," *Proc. IEEE*, vol. 73, no. 4, pp. 549–574, Apr. 1985.

[25] M. Kunt, M. Bénard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 11, pp. 1306–1336, Nov. 1987.

[26] J. Clark, M. Palmer, and P. Lawrence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, Oct. 1985.

[27] S. Budge, T. Stockham, D. Chabries, and R. Christiansen, "Vector quantization of color digital images within a human visual model," in *Proc. ICASSP'88*, April 1988, pp. 849–852.

[28] N. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.*, vol. COM-33, pp. 551–557, June 1985.

[29] A. Netravali and B. Prasada, "Adaptive quantization of picture signals using spatial masking," *Proc. IEEE*, vol. 65, pp. 536–548, Apr. 1977.

[30] I. Oliak and I. Theorot, "Seeing spatial patterns," in *Handbook of Perception and Human Performance*, J. P. Thomas, Ed. New York: Wiley, 1986, p. 21.

[31] W. Chen and W. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225–232, Mar. 1984.

[32] J. S. Gethoc, "Use of 'activity classes' in adaptive transform image coding," *IEEE Trans. Commun.*, vol. COM-23, pp. 785–786, July 1975.

[33] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[34] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattr. Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.

[35] T. Simchony and R. Chellappa, "Stochastic and deterministic algorithms for MAP texture segmentation," in *Proc. ICASSP'88*, 1988, pp. 1120–1123.

[36] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[37] A. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Signal Processing*, vol. 40, no. 2, pp. 310–322, Feb. 1992.

[38] J. Vaisey and A. Gersho, "Simulated annealing and codebook design," in *Proc. ICASSP'88* (New York, NY), Apr. 1988, pp. 1176–1179.

[39] T. Gonzalez, Y. Takefman, and H. Ochi, "TCT-VQ coding scheme using categorization with adaptive zonal partition," in *Proc. Picture Coding Symp. 1987*, June 1987, pp. 161–162.

[40] M. Gilge, T. Engelhard, and R. Mehlan, "Coding of arbitrarily shaped

整理番号:T2476K25D1　発送番号:323592　発送日:令和 1年 8月 2日　22/E

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

image segments based on a generalized orthogonal transformation," *Signal Processing*, no. 1, pp. 151–180, 1986.
[41] M. Sabin and R. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 474–488, June 1984.
[42] J. Darragh and R. Baker, "Fixed distortion, variable rate subband coding of images," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 1001, pp. 979–991, Dec. 1988.
[43] C. S. Kim, M. J. T. Smith, and R. M. Mersereau, "An improved SBC/VQ scheme for color image coding," in *Proc. ICASSP 89*, 1989, pp. 1943–1944.
[44] C. Rothwell, "Variable block vector DPCM: Vector predictive coding of color images," in *Proc. ICC'92*, June 1982, pp. 5.2.1–5.2.6.
[45] R. Aravind, "Image compression based on finite-state vector quantization," Ph.D. dissertation, Univ. California, Santa Barbara, Jan. 1988.
[46] Y. Ho and A. Gersho, "Variable-rate multistage vector quantization for image coding," in *Proc. ICASSP'88*, Apr. 1988, pp. 1156–1159.
[47] H. Lee and D. Lee, "A gain-shape vector quantizer for image coding," in *Proc. ICASSP'88*, 1956, pp. 141–144.



Jacques Vaisey (S'77–M'80) received the B.Sc. degree from the University of Manitoba in 1980, the M.Sc. degree from Queen's University, Kingston, Ontario, Canada, in 1982, and the Ph.D. degree from the University of California, Santa Barbara, in 1989, all in electrical engineering.

From February 1982 to September 1984 he was employed in the Transmission Division of Bell-Northern Research (BNR) in Ottawa, Ontario, where he worked on the design of 64-QAM and 256-QAM digital radio systems. Between September 1989 and September 1990 he was a Research Associate at INRS-Telecommunications in Montreal, investigating problems in image coding. He is currently an Assistant Professor of Engineering Science at Simon Fraser University, Burnaby, British Columbia. His current research interests include image coding, image processing, and multidimensional signal processing.

Allen Gersho (S'58–M'70–F'82–SM'82–LM'84) received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1963.

He was at Bell Laboratories from 1963 to 1980. He is now Professor of Electrical and Computer Engineering at the University of California, Santa Barbara, and is the Director of the Center for Information Processing Research there. His current research activities are in signal compression using vector quantization techniques and in pattern recognition and neural networks. He holds patents on speech coding, quantization, adaptive equalization, digital filtering, and modulation and coding for voiceband data modems.

Dr. Gersho served as a Member of the Board of Governors of the IEEE Communications Society from 1982 to 1985 and as a Member of the Communication Theory Technical Committee and the Signal Processing and Communications Electronics Technical Committee of the IEEE Communications Society. He served as Editor of IEEE COMMUNICATIONS MAGAZINE and Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. In 1980, he was awarded the Guillemin-Cauer Prize Award from the Circuits and Systems Society. He received the Donald McCormac Meritorious Service Award from the IEEE Communications Society in 1983 and in 1984 he was awarded an IEEE Centennial Medal.

Variable Block-Size Transforms for H.264/AVC

Mathias Wien, *Member, IEEE*

***Abstract*—A concept for variable block-size transform coding is presented. It is called adaptive block-size transforms (ABT) and was proposed for coding of high resolution and interlaced video in the emerging video coding standard H.264/AVC. The basic idea of inter ABT is to align the block size used for transform coding of the prediction error to the block size used for motion compensation. Intra ABT employs variable block-size prediction and transforms for encoding. With ABT, the maximum feasible signal length is exploited for transform coding. Simulation results reveal a performance increase up to 12% overall rate savings and 0.9 dB in peak signal-to-noise ratio.**

***Index Terms*—Block-based hybrid video coding, H264/AVC, integer transforms, variable block-size transforms.**

I. Introduction

THE Joint Video Team (JVT) of ITU-T SG16 WP3 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) is currently developing a joint video coding standard. It is referenced as recommendation H.264 for the ITU-T and MPEG-4 Part 10 Advanced Video Coding (AVC) for ISO [1]. The work started off as a design process for a new ITU-T recommendation succeeding H.263 [2] and was merged in late 2001 to a joint project with MPEG. The standardization process is reflected in the Joint Model (JM) software that is used for evaluation of the tools presented to JVT. In this paper, the status of the tools is described as of the Joint Final Committee Draft (JFCD) of the standard [3].

The coder is based on the hybrid coding scheme including sophisticated prediction modes and efficient adaptive entropy coding techniques to achieve improved compression performance. Blocks of size 16×16 down to 4×4 pixels can be employed for motion prediction. The prediction modes are organized in a tree-structured manner, allowing even for flexible combination of different motion compensation block sizes inside a 16×16 pixel macroblock. Main features include also multiple reference frames for inter prediction and a generalized B-frame concept.

For improved intra-compression performance, various prediction methods are specified. The adjacent pixels of the already decoded neighboring blocks are used for intra prediction. A fixed 4×4 block transform is applied to the prediction error signal of inter and intra-prediction modes. For intra macroblocks, an additional transform for 4×4 DC coefficients can be applied.

Manuscript received February 26, 2002; revised May 10, 2003. This work was supported by R. Bosch GmbH, Hildesheim.

The author is with the Institut und Lehrstuhl für Nachrichtentechnik, RWTH Aachen University, Aachen, Germany (e-mail: wien@ient.rwth-aachen.de).

Digital Object Identifier 10.1109/TCSVT.2003.815380

Two entropy coding methods are specified for application in different profiles of H.264/AVC. The Base Line profile and the Extended profile employ context-adaptive variable length codes (CAVLC) for the transform coefficients and universal variable length codes (UVLC) for all other symbols. For high coding efficiency context adaptive binary arithmetic coding (CABAC) is employed in the Main profile. The standard is specified to allow for an implementation using only 16 bit multiplications and 16 bit memory access.

In this paper, the concept of variable block-size transform coding is presented. The scheme is called adaptive block-size transforms (ABT), indicating the adaptation of the transform block size to the block size used for motion compensation. For intra coding, the size of the block transform is adapted to the properties of the intra-prediction signal. Thus, for both inter and intra coding, the maximum feasible signal length can be exploited by the transform. With ABT, the block transform as commonly used in the former image and video coding standards is generalized from a fixed-size transform to a signal-adaptive tool for increased overall coding performance.

The large transforms can provide a better energy compaction and a better preservation of detail in a quantized signal than a small transform does. On the other hand, larger transforms introduce more ringing artifacts caused by quantization than small transforms do. By choosing the transform size according to the signal properties, the tradeoff between energy compaction and preserved detail on the one hand and ringing artifacts on the other can be optimized.

Various proposals for variable block-size coding of images and video are known from the literature. Many of them are based on quadtree structures either for prediction or for transform coding. Quadtree-based variable block-size motion compensation was proposed, e.g., in [4], [5]. A quadtree-based variable block-size discrete cosine transform (DCT) was introduced in [6]. In this paper, the choice of the appropriate block size was based on a mean-based decision rule. However, this approach did not lead to rate/distortion (RD) optimal block tilings [7]. An image compression scheme with variable block-size segmentation was presented in [8], [9]. Here, the amount of detail in the blocks was used for determination of the employed block size. Classified vector quantization (VQ) was employed for high detail 4×4 blocks. A combination of transform coding and VQ was used for nonhigh detail 8×8, 16×16, and 32×32 blocks. Quadtree structures were also used with lapped transforms, e.g., in [10]. An RD optimized quadtree-based variable block-size coder using wavelet packets was proposed in [11]. In [12], a variable block-size transform image coder was presented that is not limited to square blocks. Here, block sizes of 8, 16, or 32 pixels per edge were used. The appropriate block size was selected according to a local activity index of the image scene.

1051-8215/03$17.00 © 2003 IEEE

The concept of ABT presented in this paper merges the benefits from variable block-size prediction and variable block-size transform coding.

The organization of the paper is as follows. In Section II, the transformation and quantization used for ABT coding are presented. Section III introduces the application of ABT for inter and intra coding in H.264/AVC. Extensions and modifications of the techniques used in H.264/AVC that are necessary for ABT coding are highlighted. In Section IV, the performance of ABT is analyzed. Finally, conclusions are drawn in Section V.

II. Variable Block Size Transforms

In general, the separable 2-D transform of a 2-D signal can be written as

$$C = T_v \cdot B \cdot T_h^T \tag{1}$$

where B denotes a matrix representing signal block of N pixels and M lines. C is the transform coefficient matrix, whereas T_v and T_h are the $M \times M$ and $N \times N$ transform matrices in vertical and horizontal direction, respectively. The transform matrices comprise a set of orthogonal base functions. T^T is the transposed matrix T.

The ABT proposal for H.264/AVC employs transforms for blocks of size 4×4, 4×8, 8×4, and 8×8 pixels. Hence, employing separable vertical and horizontal transforms, a 4×4 and an 8×8 transform need to be specified. Here, the 4×4 transform specified in H.264/AVC is reused. It is a low-dynamic approximation of the 4×4 DCT [13]. In matrix notation, it can be written as

$$T_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad T_4^I = \begin{pmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{pmatrix}. \tag{2}$$

Both the transform T_4 and the inverse transform T_4^I have different norms for the even and the odd base functions. These have to be regarded in the encoding and decoding process. In the standard, the inverse transform T_4^I is specified as a fast butterfly operation using only additions and bit shift operations. For detail on the 4×4 transform and quantization process (see [14]).

The 8×8 transform matrix proposed for application with ABT is a single-norm approximation of the 8×8 DCT matrix as shown in (3), at the bottom of the page. It was originally proposed in [15]. The transform and the inverse transform $T_8^I = T_8^T$ can be implemented in a butterfly structure, using 36 additions, eight bit-shift operations, and ten multiplications. The algorithm is given in Appendix.

A. Quantization and Normalization

Both, the 4×4 as well as the 8×8 transform are not normalized and therefore require normalization. This operation is integrated into the quantization and de-quantization procedure. Since the decoding process is the relevant operation to the standard, the de-quantization is described here.

In the H.264/AVC specification, only the de-quantization values for the six finest QP need to be stored. The remaining values can be generated by scaling the finest de-quantization values with the appropriate power of 2.

Let $c(m,n)$ denote a decoded coefficient at position (m,n) in an $N \times M$ block. Note that m and n indicate the row (line) and column (pixel) respectively. We use mathematical notation in the equations. The size of blocks in a frame is indicated using pixels $\times$ lines. The de-quantization process can be written as

$$a(m,n) = \left[c(m,n) \cdot R_{M,N}^{i_{QP}}(m,n) \right] \ll \left(\left\lfloor \frac{\mathrm{QP}}{6} \right\rfloor - 2 \right) \tag{4}$$

where $n = 0, \ldots, N-1$, $m = 0, \ldots, M-1$, $i_{QP} = (\mathrm{QP} \bmod 6)$, and $[\cdot]$ denotes rounding with truncation toward zero ("mod" denotes the modulo operation). $R_{M,N}^{i_{QP}}(m,n)$ is the scaling factor corresponding to the position of the coefficient in the reconstructed block.

The even and odd base functions of T_4^I have different norm. Due to these, the norms of coefficients in 4×4 blocks depend on m, n, both being even, both odd, or mixed. For all block sizes, the norms are accounted for in $R_{M,N}^{i_{QP}}(m,n)$. T_8^I has a single norm for all base functions. For 4×8 or 8×4 blocks, the choice of $R_{M,N}^{i_{QP}}(m,n)$ depends only on the parity of n or m, respectively. For 8×8 blocks, the value of $R_{M,N}^{i_{QP}}(m,n)$ only depends on QP. The values for $R_{M,N}^{i_{QP}}(m,n)$ for all block sizes are given in [3].

The low-dynamic 4×4 transform and quantization allow for an implementation using a dynamic range of 16 bit maximum for all operations. The additional ABT transforms can be implemented using only 16-bit multiplications and 16-bit memory access [16].

III. Inter Coding and Intra Coding

A. Inter Coding

In H.264/AVC, a tree-structured motion-prediction scheme is employed. Each 16×16 macroblock can be motion-compensated using one of four macroblock modes. These modes differ

$$T_8 = \begin{pmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{pmatrix} \tag{3}$$

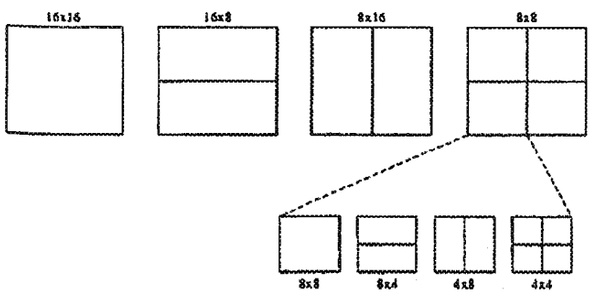


Fig. 1. Block modes available for motion prediction in H.264/AVC. For each subblock, a motion vector is encoded. In the 8 × 8 mode, each 8 × 8 block can be individually divided into subblocks of sizes 8 × 8 down to 4 × 4.

in the number and size of subblocks with common motion vectors. In 8 × 8 mode, the subblocks can be further divided into subblocks of sizes 8 × 8, 8 × 4, 4 × 8, or 4 × 4.

Applying these flexible motion-prediction modes, the resolution of the underlying motion model can be adapted to the content of the encoded video sequence. The block tiling of the motion-prediction modes is depicted in Fig. 1.

The motion vectors are represented in quarter-pel resolution. Multiple reference frames can be exploited for prediction. A generalized concept of B-frames is introduced in H.264/AVC. Besides direct, forward, and backward-prediction modes, bipredictive coding allows employing almost any two frames from the frame store for prediction.

1) ABT Inter Coding: The basic idea of ABT for inter coding is to connect the block size used for transform coding of the prediction error to the block size used for motion compensation. The maximum feasible signal length is exploited for transform coding. As the subblock size is already indicated for motion compensation, no additional side information needs to be encoded for inter ABT.

The maximum transform block size used for ABT is restricted to 8 × 8 pixels. Larger block sizes are omitted due to increase complexity and dynamic range restrictions. Larger transform blocks also tend to spread more ringing artifacts into the reconstructed signal.

2) Motion Estimation: In the JM, the sum of absolute transformed differences (SATD) is applied for subpel motion estimation. The SATD reflects the application of a transform to the prediction error. It accounts for the amount of prediction error, as well as the cost for transformed representation. The cost can be seen as an estimate for the rate spent for encoding. Hence, the SATD operates as an approximative RD cost criterion for the prediction error that can be used for motion vector search.

For SATD calculation, the 4 × 4 Hadamard transform $\boldsymbol{H}_4$ is applied to each 4 × 4 prediction error block. The resulting transformed signal emulates the frequency characteristics of the true transformed block in the subsequent transformation/quantization stage at very low computational cost.

Let $\boldsymbol{D}_{\text{res}}$ be a motion-compensated prediction error block of size 4 × 4 pixel. Then

$$\boldsymbol{D}_H = \boldsymbol{H}_4 \cdot \boldsymbol{D}_{\text{res}} \cdot \boldsymbol{H}_4^T \tag{5}$$

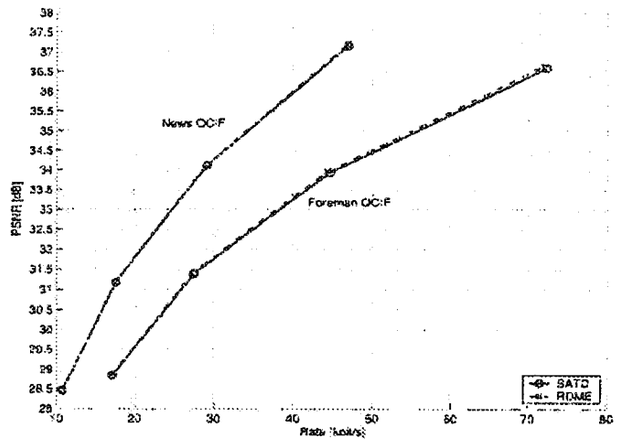


Fig. 2. Comparison of the coding performance for ABT when using the SATD (o) or the RD cost (RDME, ×) for motion estimation. Results are presented for the test sequences FOREMAN and NEWS. The sequences were encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels.

denotes the 2-D transform of the difference block. The SATD of the 4 × 4 block is then given as

$$\text{SATD} = \sum_{i=0}^{3}\sum_{j=0}^{3} |d_H(i,j)| \tag{6}$$

where $d_H(i,j)$ denotes the coefficient of $\boldsymbol{D}_H$ on position (i,j). For motion estimation, the SATD is computed and accumulated for all 4 × 4 blocks of the current subblock. The bit rate needed for encoding of the motion vector is weighted by the quantizer dependent weighting factor λ_{ME} (QP) and added to the SATD to constitute the subblock cost criterion. For details on the weighting factor λ_{ME} (QP) see [17].

In the case of ABT, the calculation of the SATD is modified to account for the new transform sizes. Since the Hadamard transform can be defined for all signal lengths that are multiples of 4 [18], the SATD calculation can be extended straightly to the ABT block sizes.

As stated before, the application of the Hadamard transform emulates the transform step in the transformation/quantization stage. The suitability of the Hadamard transform for SATD calculation of blocks larger than 4 × 4 needs to be verified. The ABT SATD is compared to the performance of a more elaborate RD criterion where the encoded bit rate for the motion vectors and the coefficients is weighted against the sum of the squared reconstructed prediction error. The squared λ_{ME} (QP) was found to give the optimum performance for the RD based motion estimation. Experiments show that the performance of the ABT SATD is very close to the performance of the RD criterion. An example is shown in Fig. 2.

The performance improvement for the single ABT inter coding modes is demonstrated for the test sequence COASTGUARD in Fig. 3. ABT is employed for inter coding only. The RD plots compare the performance for the single prediction block modes when using the 4 × 4 transform or the corresponding $N \times M$ transform employed with ABT. Since

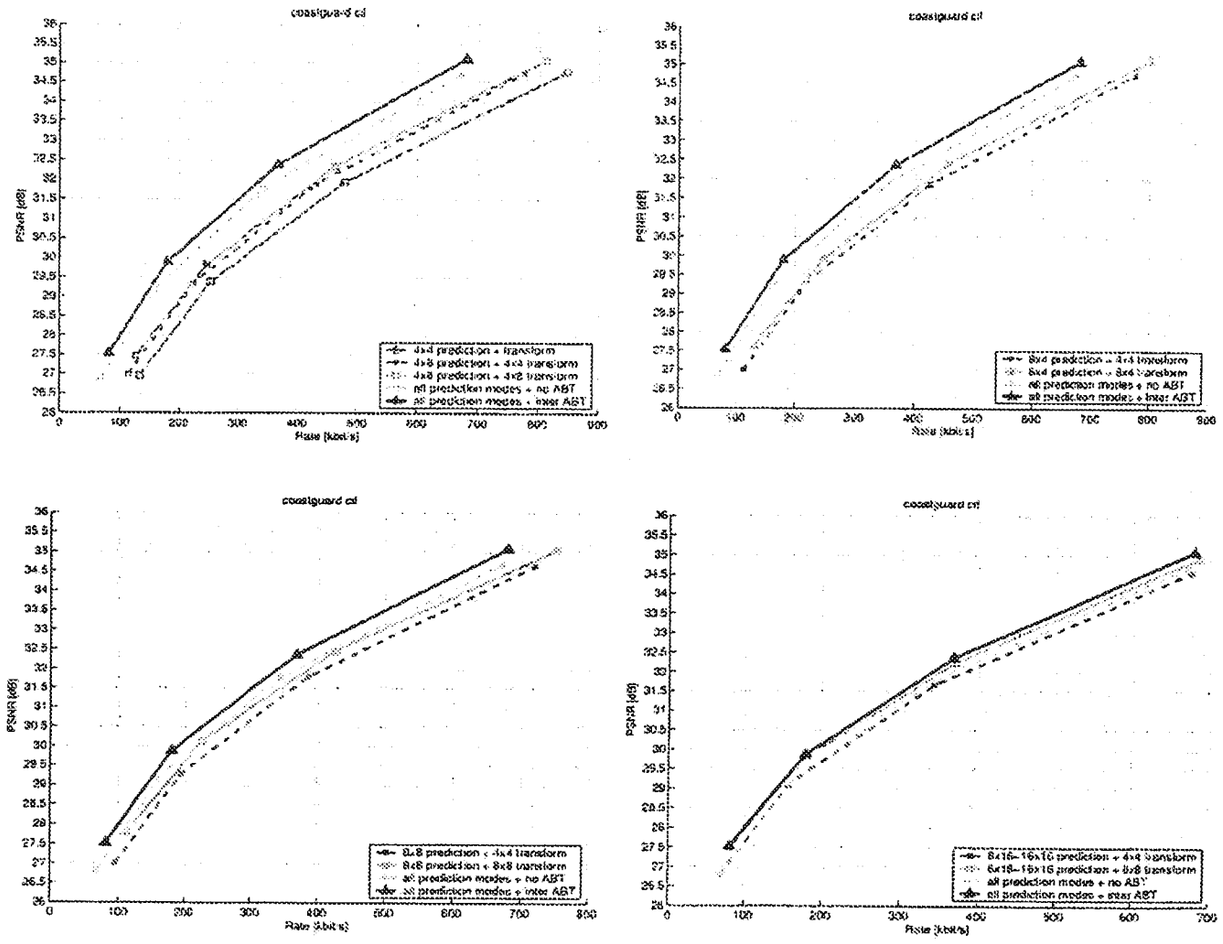


Fig. 3. Comparison of the RD performance with and without ABT for single prediction modes. Results are presented for the test sequence COASTGUARD. The sequence was encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels. ABT was used only for inter-coded blocks. Intra blocks were encoded without ABT.

the 8×8 transform is used for macroblocks in 8×16, 16×8, and 16×16 mode. these modes were combined in one simulation. The RD performance when using all prediction modes is compared to the performance of the single modes. As can be seen from Fig. 3. the performance gain for ABT increases with the block size used for prediction and transformation. At low rates, the performance when using only the modes larger than 8×8 is comparable to the performance when all modes are used. The benefit of the modes with smaller block sizes becomes more pronounced at higher rates where these expensive modes are chosen more frequently.

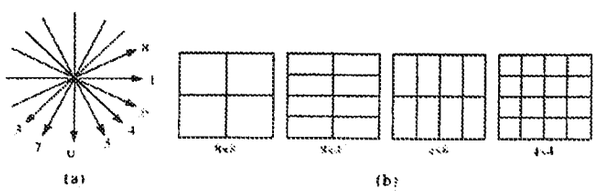


Fig. 4. (a) Intra-prediction directions for non-ABT and ABT coding. (b) Tiling of an intra macroblock for ABT intra coding including the 4×4 tiling for non-ABT coding. For each subblock a separate prediction direction can be employed.

B. Intra Coding

Two basic intra-prediction modes are used in H.264/AVC, comprising 4×4 and 16×16 block-wise prediction from previously decoded blocks. In intra 4×4 prediction mode, each 4×4 block is predicted by the adjacent pixels of the decoded neighboring 4×4 blocks. DC prediction and eight directional prediction modes can be applied to a 4×4 block. The directional prediction modes are shown in Fig. 4. Different prediction modes can be selected for each of the 16 4×4 blocks in a macroblock. The mode information has to be transmitted. To this end, a probability map is generated from the prediction modes of

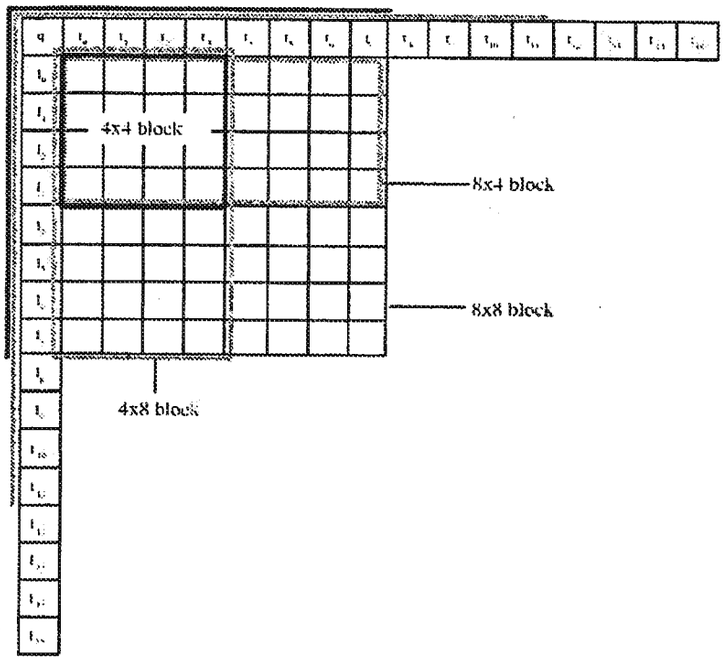


Fig. 5. Pixels I_i, I_j, and q of the top and left neighboring subblocks that may be used for intra prediction of 4×4, 4×8, 8×4, and 8×8 blocks. The maximum number of prediction pixels for an $N \times M$ block is $N + M$ pixels in each direction.

the encoded neighboring blocks. The selected prediction mode is encoded according to its probability in this map.

Smooth macroblocks containing little detail can be predicted more efficiently on a full macroblock basis. This kind of prediction is provided by the intra 16×16 Mode. Here, the current macroblock is predicted by the edge pixels of the adjacent left and top macroblocks. There are four different prediction modes, including DC, horizontal, vertical, and plane prediction.

Blocks in intra 16×16 Mode employ an additional Hadamard transform of the 4×4 DC coefficients after the 4×4 transform of the 16×16 intra prediction error. The two-stage transform decorrelates the signal in the lowest frequencies for the 16×16 block.

1) Intra Coding With Variable Block Sizes: The two-stage transform in the intra 16×16 Mode can be interpreted as a 16×16 transform for the lowest frequencies. Therefore, the choice of two intra modes specified in H.264/AVC can already be seen as a choice between two transform modes. For ABT intra coding, the prediction modes specified for the 4×4 intra-prediction mode are extended to the block sizes used for ABT inter transform coding, see Fig. 4. Hence, the gap between 4×4 intra coding and 16×16 intra coding is closed. With ABT, the encoder can trade energy compaction by larger transforms against the size of the region a single prediction direction is applied to.

An additional codeword has to be transmitted that indicates the block size used for intra prediction. The prediction block size corresponds to the block size used for transform coding. The prediction modes are encoded with probability maps as described for the intra 4×4 Mode. Since the number of subblocks is reduced, the overall overhead information for prediction block size and prediction modes for intra ABT can be smaller than with intra 4×4 Mode.

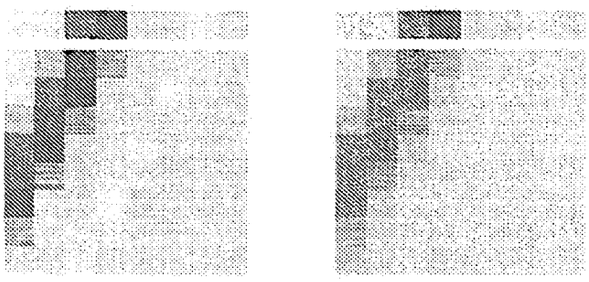
The number of neighboring pixels that are employed for intra prediction varies with the prediction mode. For example, for the horizontal and the vertical prediction mode, exclusively pixels from the left or the top are used, respectively. The reference pixels that may be used for intra-block prediction are shown in Fig. 5. Only pixels of previously decoded subblocks may be used for prediction. If prediction pixel positions required for a prediction mode are not available (e.g., because the corresponding neighboring block has not been not processed yet), the value of the last available pixel position is padded to the unavailable positions.

The maximum number of pixels for directional prediction is used for the *diagonal down/left* prediction (mode 3, see Fig. 4. Here, $N + M$ pixels are used in both the horizontal and vertical direction for one $N \times M$ block.

The prediction from skew directions might introduce visible artifacts into the prediction signal that are unfavorable for an efficient compression. This effect becomes prominent especially for block sizes larger than 4×4 since more pixels are predicted from the same source. Hence, the new ABT intra modes are afflicted. To reduce the artifacts, the pixels used for prediction are filtered with a low-pass filter.

Conceptually, a filter

$$h[n] = \frac{[1 \quad 2 \quad 1]}{4} \tag{7}$$



(a) (b)

Fig. 6. Prediction mode 7 (*vertical-left*) applied to an 8 × 8 block. The last row of the top neighboring 8 × 8 block is used for prediction. (a) Without pre-filtering. (b) Prediction from the same pixels with pre-filtering.

is applied to the vector of reference pixels

$$p[n] = [l_{\max} \quad \cdots \quad l_0 \quad q \quad t_0 \quad \cdots \quad t_{\max}]^T \qquad (8)$$

used for prediction. The reference pixels $p[n]$ are indicated in Fig. 5. The application of the filter sufficiently reduces prediction artifacts for all block sizes. For block sizes larger than 4 × 4, pre-filtering is applied for horizontal and vertical prediction directions as well. In Fig. 6, the impact of the pre-filtering procedure is demonstrated for *vertical-left* prediction of an 8 × 8 block.

2) Mode Decision for ABT Intra Coding: The optimum intra ABT mode for a macroblock is selected in a two-stage process. For all possible block sizes, the best prediction mode for each subblocks is determined. Then, the cost for encoding the macroblock with the selected subblock prediction modes is evaluated for all block sizes. The corresponding mode with the optimum overall cost is chosen for encoding the macroblock.

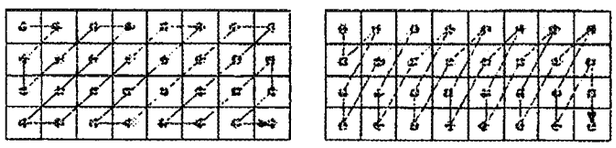
Both decisions can be made based either on the SATD or on a RD cost criterion. The RD criterion weights the rate for the transform coefficients and the corresponding prediction mode against the sum of the squared reconstructed prediction error. The weighting factor used for RD optimized macroblock mode selection in H.264/AVC can be employed here [17].

Examples for the intra mode decision on natural and synthetic test images are presented in Figs. 9 and 11.

C. Deblocking Filter

The application of a 4 × 4 transform tends to promote blocking artifacts in the reconstructed video, especially at lower bit rates. Therefore, a normative in-loop deblocking filter is employed in H.264/AVC. The reconstructed frames are filtered before application for motion compensation in the prediction loop. The functionality of this filter is essential for the subjective performance of the coder, especially at lower bit rates.

The strength of the filter is determined by the chosen macroblock mode, the existence of nonzero transform coefficients, the applied quantizer, and by the motion vector activity. In general, stronger filtering is applied to intra blocks than inter blocks. A maximum of two pixels on each side of the boundary of two adjacent blocks is affected by applying the deblocking filter on 4 × 4 blocks. Intra blocks coded with the intra 16 × 16 mode are filtered only at the macroblock boundaries. Here, a maximum of three boundary pixels may be filtered.



(a) (b)

Fig. 7. (a) Zig-zag scan for frame based coding and (b) alternate scan for field based coding of an 8 × 4 block with ABT.

The mode of operation of the deblocking filter needs to be adjusted for application to ABT-coded blocks. The strength of the filter is increased relative to the strength for 4 × 4 blocks according to the sizes of the adjacent blocks. The strength is increased most for two neighboring 8 × 8 blocks. Deblocking is only performed at transform block boundaries. Hence, the number of filter operations is reduced by ABT coding.

For example, if the 4 × 4 transform is employed, an 8 × 8 block has eight 4 × 4 block boundaries to be filtered. These are four inner boundaries, two on the left and two at the top. The boundaries to the right and at the bottom are encountered for the next blocks. Accordingly, an ABT 8 × 8 block has only two boundaries to be filtered. Similarly, for 4 × 8 and 8 × 4 blocks, the number of filtered boundaries is reduced from four to two.

D. Entropy Coding

H.264/AVC features two entropy coding methods. Variable length coding is supported by all standard compliant decoders. It employs a CAVLC for transform coefficients and a universal VLC for all other symbols. For high coding efficiency, CABAC is employed. Here, nonbinary symbols are binarized, e.g., using a unary code tree. For the separate bins, contexts are generated that are used to adapt the arithmetic coder to the statistics of the encoded symbols. For details, see [19].

Run-length coding is employed for the transform coefficients. Two types of scans can be employed to represent the transform coefficients as (level, run) symbols. The well-known zig-zag scan is employed for frame based encoding. In the case of field-based encoding of interlaced material, an alternate scan is used that accounts for the modified signal statistics in the fields of interlaced video. The alternate scan was originally proposed for ABT in [20]. An example for the zig-zag scan and the alternate scan for an 8 × 4 block is shown in Fig. 7. A VLC design for ABT transform coefficients was proposed in [21].

E. CABAC Transform Coefficient Encoding

The binarization and context generation for the transform coefficients for 4 × 8, 8 × 4,and 8 × 8 blocks is identical to the method specified for 4 × 4 blocks. Separate contexts are used for the symbols of each block size.

- For each encoded block, a flag is signaled to indicate whether the block contains coefficients or not.
- A significance map is generated, indicating the position of nonzero coefficients in the scan. For each significant coefficient a "last" flag is encoded. This flag indicates whether the last significant coefficient in the scan is reached.

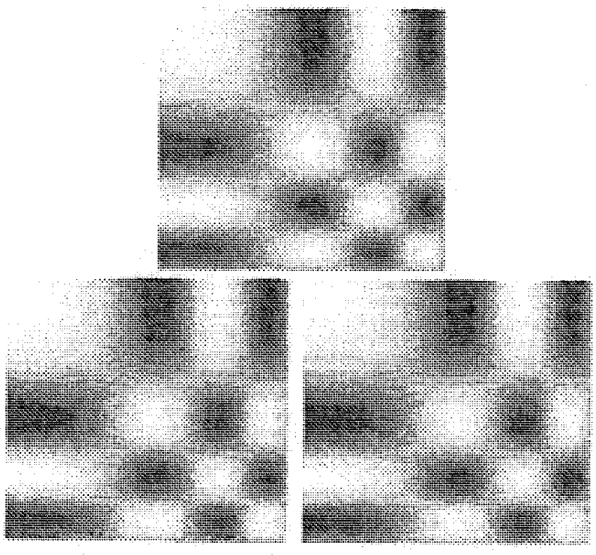


Fig. 8. Subjective comparison: Detail of a chirp image coded with QP = 36. Top: Original. Left: Encoded without ABT (187 720 bit, 30.25 dB). Right: Encoded with ABT (158 304 bit, 32.53 dB).

- The absolute values of the coefficients are binarized using an unary code tree and encoded in reverse order (last coefficient first). The sign of the coefficients is written directly to the bitstream.

IV. Simulation Results

The simulation results given here were generated with the JM 4.2 reference software [22]. ABT is implemented according to [16].

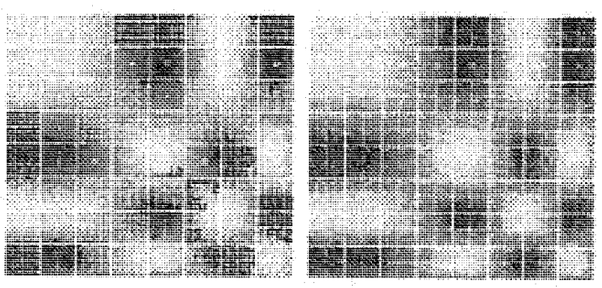
A. Subjective Performance

Two examples for the subjective performance of ABT are given.

Fig. 8 shows an example of the coding performance for a synthetic image. The original gray-level image of size 512 × 512 contains a separable chirp signal with the frequency increasing from left to right and top to bottom. The image was encoded as a single frame sequence with RD optimization at QP = 36. Fig. 8 shows the top left corner of the chirp image. As can be seen, the ABT-coded image reveals a much more pleasing subjective quality than the non-ABT image does. Also, the rate and PSNR numbers indicate superior objective performance for ABT.

In Fig. 9, the selected intra-prediction modes for ABT and non-ABT coding are shown. For non-ABT, the high number of macroblocks in intra 4 × 4 Mode with many different prediction directions is very expensive. The larger transform block sizes employed with ABT give an improved compression performance at a convenient subject quality.

In Fig. 10, a detail of first frame of the the HD sequence Shuttle (1280 × 720 @ 60 Hz) is shown. The sequence was encoded with RD optimization at QP = 36. It can be seen that more detail is preserved by ABT coding. The subjective quality is improved by ABT, even though the PSNR numbers indicate a slight loss.



(a) (b)

Fig. 9. Intra mode selection for the region shown in Fig. 8. (a) Corresponding intra mode selection without ABT. (b) Corresponding intra mode selection with ABT. Macroblocks are marked white, subblocks of 4 × 4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16 × 16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16 × 16 plane prediction is indicated by an empty square (□).

Fig. 11 shows the intra-prediction modes that are chosen for ABT and non-ABT coding. Without ABT, high detail regions are encoded using the intra 4 × 4 Mode. In regions with less activity, the intra 16 × 16 Mode is employed. When ABT is used the number of macroblocks in intra 16 × 16 Mode is largely reduced. The ABT 8 × 8 mode is chosen most frequently. In regions with much variation, smaller block sizes are employed. Blocks of size 4 × 4 are rarely used for intra coding.

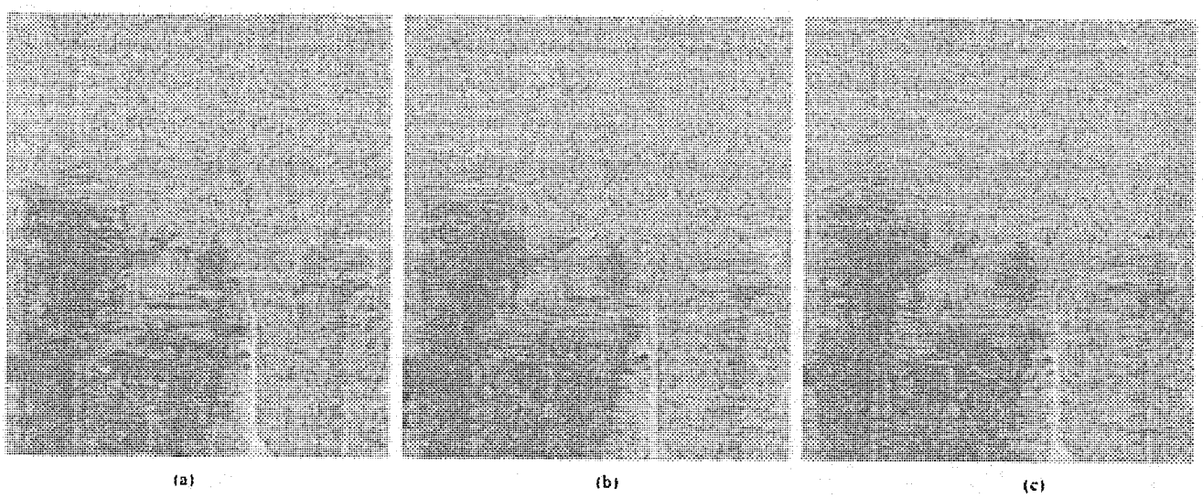
B. RD Performance

The sequences used for the simulations are given in Table I. The sequences RAVEN and KAYAK contain high-definition (HD) video. RAVEN in CIF size was extracted from the full-resolution 1280 × 720 at 60-Hz sequences by windowing the center of the full-resolution frames and temporal down-sampling by a factor of two. No spatial or temporal filtering was applied. This sequence emulates the characteristics of the full-resolution sequences while saving simulation time. All sequences but KAYAK contain progressive video. KAYAK is an interlaced sequence.

1) Simulation Conditions: In all simulations, RD optimization was switched on. Results for two configurations are presented here. In the first simulation, intra only coding is performed. The inter coding performance was evaluated using an IPPP... and an IBBPBBP... frame structure with a search range of 32 pixels and five reference frames. The sequence KAYAK was encoded in field mode using a search range of 16 pixels and two reference frames. The search range was reduced here to shorten the simulation time. ABT was used for intra and inter macroblocks in the ABT simulations.

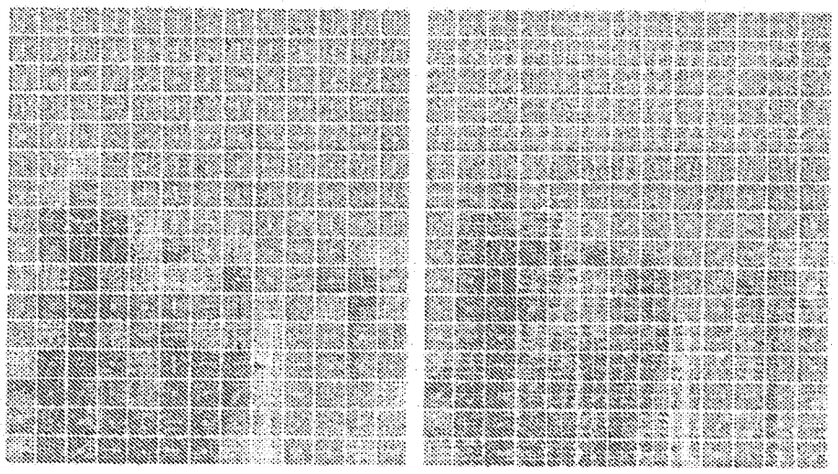
All sequences were encoded using a fixed quantization parameter QP for all frames. The HD sequences were encoded using QP = {20, 24, 28, 32} for I and P frames. The other sequences were encoded using QP = {28, 32, 36, 40} for I and P frames. The quantization parameter QP was increased by two for B frames.

2) Results: Table II shows the rate savings and PSNR gains for the simulation conditions given above. Overall, average rate savings of about 8% and PSNR gains of 0.49 dB can be observed for ABT intra coding. The inter gains are lower than the gains achieved for intra coding. This might be due to the re-



(a) (b) (c)

Fig. 10. Subjective comparison: Detail of the first frame of the Shuttle sequence coded with QP = 30. (a) Original. (b) Encoded without ABT (39128 bit, 38.36 dB). (c) Encoded with ABT (38784 bit, 38.25 dB). More detail is preserved with ABT coding (note the flag staff at the top left of the depicted area).



(a) (b)

Fig. 11. Intra mode selection for the region shown in Fig. 10. (a) Encoded without ABT. (b) Encoded with ABT. Macroblocks are marked white. Subblocks of 4×4 or ABT intra modes are marked in light grey. The directional modes are indicated by corresponding lines for both intra 16×16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16×16 plane prediction is indicated by an empty square (□).

TABLE I
TEST SEQUENCES

Sequence	Frame Size	Freq.	# Frames
FOREMAN	176 × 144	10 Hz	100
COASTGUARD	352 × 288	30 Hz	300
TABLETENNIS	352 × 288	30 Hz	300
RAVEN	352 × 288	30 Hz	300
KAYAK	1920 × 1088	60 Hz	60

TABLE II
OVERALL RATE SAVINGS AND PSNR GAINS FOR ABT CODING

	III		IPPP		IBBP	
Sequence	ΔRate	ΔPSNR	ΔRate	ΔPSNR	ΔRate	ΔPSNR
FOREMAN	-6.69 %	0.43 dB	-5.71 %	0.32 dB	-2.49 %	0.14 dB
COASTG.	-6.20 %	0.33 dB	-8.38 %	0.31 dB	-5.24 %	0.17 dB
TABLET.	-4.33 %	0.23 dB	-4.05 %	0.16 dB	-3.51 %	0.13 dB
RAVEN	-11.65 %	0.58 dB	-8.42 %	0.37 dB	-6.88 %	0.25 dB
KAYAK	-12.35 %	0.90 dB	-9.41 %	0.72 dB	-12.77 %	0.92 dB
Δmean	-8.28 %	0.49 dB	-7.19 %	0.38 dB	-5.68 %	0.32 dB

duced energy and the reduced correlation remaining in the motion predicted signal. Hence, the decorrelation properties of the larger transforms takes less effect. For IPPP, overall average rate savings of about 7% and PSNR gains of 0.38 dB are achieved. When B frames are employed, average rate savings of about 6% and PSNR gains of 0.32 dB are observed for the test set. The delta measurements were performed using the method described in [23]. RD plots for the III and IPPP frame structure are given for the sequences RAVEN and KAYAK in Fig. 12. At high rates, gains of more than 1.0 dB can be observed.

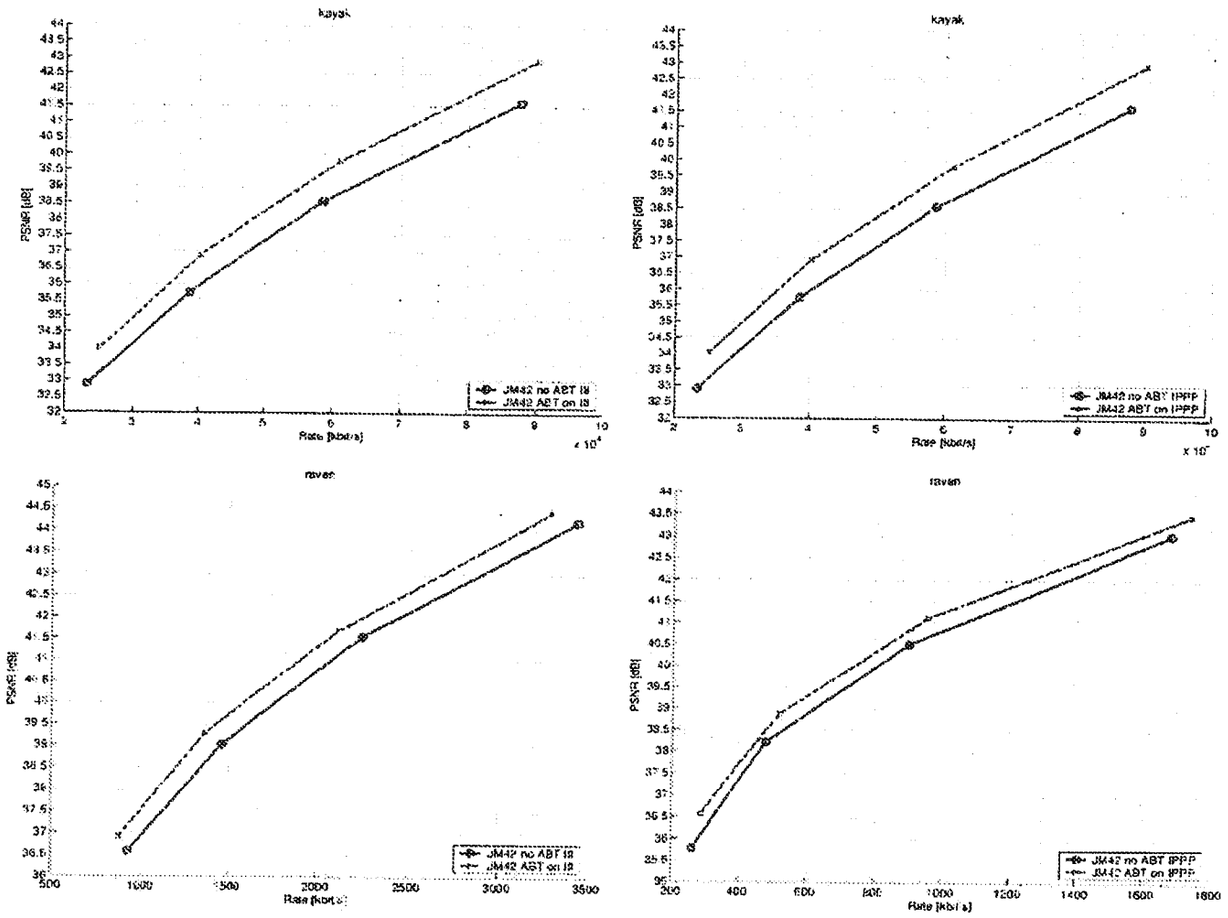


Fig. 12. Simulation results for intra only and inter coding with JM 4.2 without ABT (o) and with ABT (×) for the sequences RAVEN and KAYAK.

V. Conclusion

The concept of ABT for application in H.264/AVC was presented. With ABT, the block size used for transform coding can be aligned to the properties of the encoded signal. Blocks of size 8×8, 8×4, 4×8 and 4×4 can be transformed. For inter coding, no additional overhead information is needed for ABT. For intra coding an additional symbol is transmitted indicating the transform and prediction block size.

The results reveal an increased subjective and objective quality for ABT coding, especially on HD material. Up to 12% rate savings and 0.9 dB increase in PSNR can be observed.

Appendix
Fast Calculation of the 8 × 8 Transform

Let $\boldsymbol{c}$ be a coefficient vector of a transformed signal

$$\boldsymbol{c} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{bmatrix}^T \tag{9}$$

and $\boldsymbol{y}$ be the reconstructed signal itself

$$\boldsymbol{y} = T_8^I \cdot \boldsymbol{c}. \tag{10}$$

The transform can be calculated as follows:

$$\begin{aligned}
k_0 &= 13(c_0 + c_4); \quad k_4 = c_1 + c_7 \\
k_1 &= 17c_2 + 7c_6; \quad k_5 = c_3 + c_5 \\
k_2 &= 13(c_0 - c_4); \quad k_6 = c_1 - c_7 \\
k_3 &= 7c_2 - 17c_6; \quad k_7 = c_3 - c_5 \\
z_0 &= k_0 + k_1 \\
z_1 &= k_2 + k_3 \\
z_2 &= k_2 - k_3 \\
z_3 &= k_0 - k_1 \\
z_4 &= ((k_5 + c_5) \ll 3) + 3k_4 + k_7 + (c_1 \ll 4) \\
z_5 &= ((k_6 + c_1) \ll 3) + 3k_7 - k_4 - (c_5 \ll 4) \\
z_6 &= ((k_4 + c_7) \ll 3) - 3k_5 + k_6 - (c_3 \ll 4) \\
z_7 &= -((k_7 + c_3) \ll 3) + 3k_6 + k_5 - (c_7 \ll 4) \\
y_0 &= z_0 + z_4; \quad y_4 = z_3 - z_7 \\
y_1 &= z_1 + z_5; \quad y_5 = z_2 - z_6 \\
y_2 &= z_2 + z_6; \quad y_6 = z_1 - z_5 \\
y_3 &= z_3 + z_7; \quad y_7 = z_0 - z_4.
\end{aligned}$$

$a \ll n$ denotes the bit shift operation $a \cdot 2^n$.

ACKNOWLEDGMENT

The author would like to thank Prof. J. R. Ohm, Aachen University, and U. Benzler, R. Bosch GmbH, as well as A. Dahlhoff and C. Mayer, Aachen University, for many insightful discussions on the topic. The author would also like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] "Information Technology—Generic Coding of Audio-Visual Objects," ISO/IEC JTC1 IS 14496-2 (MPEG-4), 1998.
[2] Telecommun. Standardization Sector of the ITU, "Video Coding for Low Bitrate Communication (H.263 Version 2)," Sept., 1997.
[3] T. Weigand, "Joint final committee draft (JFCD) of joint video specification," in *Proc. JVT, 4th Meeting*, T. Weigand, Ed., Klagenfurt, Austria, July 2002. ITU-T rec. H.264/ISO/IEC 14 496-10 AVC.
[4] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '91*, Toronto, ON, Canada, May 1991, pp. 2713–2716.
[5] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477–486, Apr. 1990.
[6] C.-T. Chen, "Adaptive transform coding via quadtree-based variable block size DCT," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing ICASSP '89*, vol. 3, Glasgow, Scotland, U.K., May 1989, pp. 1854–1857.
[7] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Feb. 1993.
[8] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 2040–2060, Aug. 1992.
[9] ——, "Variable block size image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '87*, vol. 2, Dallas, TX, Apr. 1987, pp. 1051–1054.
[10] T. J. Klausutis and V. K. Madisetti, "Variable block size adaptive lapped transform-based image coding," in *Proc. IEEE Int. Conf. Image Processing ICIP '97*, vol. 3, Washington, DC, Oct. 1997, pp. 686–689.
[11] K. Ramchandran, Z. Xiong, K. Asai, and M. Vetterli, "Adaptive transforms for image coding using spatially varying wavelet packets," *IEEE Trans. Image Processing*, vol. 5, pp. 1197–1204, July 1996.
[12] I. Dinstein, K. Rose, and A. Heimann, "Variable block-size transform image coder," *IEEE Trans. Commun.*, vol. 38, pp. 2073–2078, Nov. 1990.
[13] H. Malvar, "Low-complexity length-4 transform and quantization with 16-bit arithmetic," in VCEG, 14th Meeting, Santa Barbara, CA, Sept. 2001, Doc. VCEG-N44.
[14] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/avc," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 598–603, July 2003.
[15] G. Bjontegaard, "Addition of 8 × 8 transform to H.26L," in SG16/Q15, 9. Meeting, Red Bank, NJ, Oct. 1999, Doc. Q15-I-39.
[16] M. Wien, "Clean-up and improved design consistency for ABT," in JVT, 5th Meeting, Geneva, Switzerland, Oct. 2002, Document JVT-E025.
[17] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 688–703, July 2003.
[18] F. E. Douglas and K. R. Ramamohan, *Fast Transforms*. San Diego, CA: Academic, 1982.
[19] D. Marpe, H. Schwarz, and T. Wiegand, "Advanced entropy coding in draft h.264/avc video compression standard," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 620–636, July 2003.
[20] W. Limin and B. David, "Alternate coefficient scanning patterns for interlaced ABT coding," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C140.
[21] W. Mathias and D. Achim, "16 bit adaptive block size transforms," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C107.
[22] K. Sühring, Ed., (2002) JM 4.2 Reference Software. [Online]. Available: ftp://ftp.imtc.org/jvt-experts/reference_software/jm42.zip
[23] G. Bjontegaard, "Calculation of average PSNR differences between rd curves," in SG16/Q6 VCEG, 13. Meeting, Austin, TX, Apr. 2001, Doc. VCEG-M33.



Mathias Wien (S'98–M'03) was born in Bonn, Germany, in 1971. He received the Dipl.-Ing. degree in electrical engineering from Aachen University, Aachen, Germany, in 1997, where he is currently working toward the Dr.-Ing. degree in the Institute of Communications Engineering.

He is an active contributor to the ITU-T VCEG and the Joint Video Team of VCEG and ISO/IEC MPEG where he chaired the AdHoc Group on Additional Transforms and Quantization. His research interests are in the area of image and video processing, with emphasis on space-frequency adaptive and scalable video compression.

整理番号: 発送番号:284603 発送日:平成26年 5月28日 2

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

Variable Block-Size Transforms for H.264/AVC

Mathias Wien, *Member, IEEE*

***Abstract*—A concept for variable block-size transform coding is presented. It is called adaptive block-size transforms (ABT) and was proposed for coding of high resolution and interlaced video in the emerging video coding standard H.264/AVC. The basic idea of inter ABT is to align the block size used for transform coding of the prediction error to the block size used for motion compensation. Intra ABT employs variable block-size prediction and transforms for encoding. With ABT, the maximum feasible signal length is exploited for transform coding. Simulation results reveal a performance increase up to 12% overall rate savings and 0.9 dB in peak signal-to-noise ratio.**

***Index Terms*—Block-based hybrid video coding, H264/AVC, integer transforms, variable block-size transforms.**

I. Introduction

THE Joint Video Team (JVT) of ITU-T SG16 WP3 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) is currently developing a joint video coding standard. It is referenced as recommendation H.264 for the ITU-T and MPEG-4 Part 10 Advanced Video Coding (AVC) for ISO [1]. The work started off as a design process for a new ITU-T recommendation succeeding H.263 [2] and was merged in late 2001 to a joint project with MPEG. The standardization process is reflected in the Joint Model (JM) software that is used for evaluation of the tools presented to JVT. In this paper, the status of the tools is described as of the Joint Final Committee Draft (JFCD) of the standard [3].

The coder is based on the hybrid coding scheme including sophisticated prediction modes and efficient adaptive entropy coding techniques to achieve improved compression performance. Blocks of size 16×16 down to 4×4 pixels can be employed for motion prediction. The prediction modes are organized in a tree-structured manner, allowing even for flexible combination of different motion compensation block sizes inside a 16×16 pixel macroblock. Main features include also multiple reference frames for inter prediction and a generalized B-frame concept.

For improved intra-compression performance, various prediction methods are specified. The adjacent pixels of the already decoded neighboring blocks are used for intra prediction. A fixed 4×4 block transform is applied to the prediction error signal of inter and intra-prediction modes. For intra macroblocks, an additional transform for 4×4 DC coefficients can be applied.

Manuscript received February 26, 2002; revised May 10, 2003. This work was supported by R. Bosch GmbH, Hildesheim.

The author is with the Institut und Lehrstuhl für Nachrichtentechnik, RWTH Aachen University, Aachen, Germany (e-mail: wien@ient.rwth-aachen.de).

Digital Object Identifier 10.1109/TCSVT.2003.815380

Two entropy coding methods are specified for application in different profiles of H.264/AVC. The Base Line profile and the Extended profile employ context-adaptive variable length codes (CAVLC) for the transform coefficients and universal variable length codes (UVLC) for all other symbols. For high coding efficiency context adaptive binary arithmetic coding (CABAC) is employed in the Main profile. The standard is specified to allow for an implementation using only 16 bit multiplications and 16 bit memory access.

In this paper, the concept of variable block-size transform coding is presented. The scheme is called adaptive block-size transforms (ABT), indicating the adaptation of the transform block size to the block size used for motion compensation. For intra coding, the size of the block transform is adapted to the properties of the intra-prediction signal. Thus, for both inter and intra coding, the maximum feasible signal length can be exploited by the transform. With ABT, the block transform as commonly used in the former image and video coding standards is generalized from a fixed-size transform to a signal-adaptive tool for increased overall coding performance.

The large transforms can provide a better energy compaction and a better preservation of detail in a quantized signal than a small transform does. On the other hand, larger transforms introduce more ringing artifacts caused by quantization than small transforms do. By choosing the transform size according to the signal properties, the tradeoff between energy compaction and preserved detail on the one hand and ringing artifacts on the other can be optimized.

Various proposals for variable block-size coding of images and video are known from the literature. Many of them are based on quadtree structures either for prediction or for transform coding. Quadtree-based variable block-size motion compensation was proposed, e.g., in [4], [5]. A quadtree-based variable block-size discrete cosine transform (DCT) was introduced in [6]. In this paper, the choice of the appropriate block size was based on a mean-based decision rule. However, this approach did not lead to rate/distortion (RD) optimal block tilings [7]. An image compression scheme with variable block-size segmentation was presented in [8], [9]. Here, the amount of detail in the blocks was used for determination of the employed block size. Classified vector quantization (VQ) was employed for high detail 4×4 blocks. A combination of transform coding and VQ was used for nonhigh detail 8×8, 16×16, and 32×32 blocks. Quadtree structures were also used with lapped transforms, e.g., in [10]. An RD optimized quadtree-based variable block-size coder using wavelet packets was proposed in [11]. In [12], a variable block-size transform image coder was presented that is not limited to square blocks. Here, block sizes of 8, 16, or 32 pixels per edge were used. The appropriate block size was selected according to a local activity index of the image scene.

1051-8215/03$17.00 © 2003 IEEE

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

The concept of ABT presented in this paper merges the benefits from variable block-size prediction and variable block-size transform coding.

The organization of the paper is as follows. In Section II, the transformation and quantization used for ABT coding are presented. Section III introduces the application of ABT for inter and intra coding in H.264/AVC. Extensions and modifications of the techniques used in H.264/AVC that are necessary for ABT coding are highlighted. In Section IV, the performance of ABT is analyzed. Finally, conclusions are drawn in Section V.

II. Variable Block Size Transforms

In general, the separable 2-D transform of a 2-D signal can be written as

$$\boldsymbol{C} = \boldsymbol{T}_v \cdot \boldsymbol{B} \cdot \boldsymbol{T}_h^T \tag{1}$$

where $\boldsymbol{B}$ denotes a matrix representing signal block of N pixels and M lines. $\boldsymbol{C}$ is the transform coefficient matrix, wherens $\boldsymbol{T}_v$ and $\boldsymbol{T}_h$ are the $M \times M$ and $N \times N$ transform matrices in vertical and horizontal direction, respectively. The transform matrices comprise a set of orthogonal base functions. $\boldsymbol{T}^T$ is the transposed matrix $\boldsymbol{T}$.

The ABT proposal for H.264/AVC employs transforms for blocks of size 4×4, 4×8, 8×4, and 8×8 pixels. Hence, employing separable vertical and horizontal transforms, a 4×4 and an 8×8 transform need to be specified. Here, the 4×4 transform specified in H.264/AVC is reused. It is a low-dynamic approximation of the 4×4 DCT [13]. In matrix notation, it can be written as

$$\boldsymbol{T}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad \boldsymbol{T}_4' = \begin{pmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{pmatrix}. \tag{2}$$

Both the transform $\boldsymbol{T}_4$ and the inverse transform $\boldsymbol{T}_4'$ have different norms for the even and the odd base functions. These have to be regarded in the encoding and decoding process. In the standard, the inverse transform $\boldsymbol{T}_4'$ is specified as a fast butterfly operation using only additions and bit shift operations. For detail on the 4×4 transform and quantization process (see [14]).

The 8×8 transform matrix proposed for application with ABT is a single-norm approximation of the 8×8 DCT matrix as shown in (3), at the bottom of the page. It was originally proposed in [15]. The transform and the inverse transform $\boldsymbol{T}_8' = \boldsymbol{T}_8^T$ can be implemented in a butterfly structure, using 36 additions, eight bit-shift operations, and ten multiplications. The algorithm is given in Appendix.

A. Quantization and Normalization

Both, the 4×4 as well as the 8×8 transform are not normalized and therefore require normalization. This operation is integrated into the quantization and de-quantization procedure. Since the decoding process is the relevant operation to the standard, the de-quantization is described here.

In the H.264/AVC specification, only the de-quantization values for the six finest QP need to be stored. The remaining values can be generated by scaling the finest de-quantization values with the appropriate power of 2.

Let $c(m,n)$ denote a decoded coefficient at position (m,n) in an $N \times M$ block. Note that m and n indicate the row (line) and column (pixel) respectively. We use mathematical notation in the equations. The size of blocks in a frame is indicated using pixels $\times$ lines. The de-quantization process can be written as

$$w(m,n) = \left[c(m,n) \cdot R_{M,N}^{i_{QP}}(m,n) \right] \ll \left(\left\lfloor \frac{QP}{6} \right\rfloor - 2 \right) \tag{4}$$

where $n = 0, \ldots, N-1$, $m = 0, \ldots, M-1$, $i_{QP} = (QP \bmod 6)$, and $[\cdot]$ denotes rounding with truncation toward zero ("mod" denotes the modulo operation). $R_{M,N}^{i_{QP}}(m,n)$ is the scaling factor corresponding to the position of the coefficient in the reconstructed block.

The even and odd base functions of $\boldsymbol{T}_4'$ have different norm. Due to these, the norms of coefficients in 4×4 blocks depend on n, m both being even, both odd, or mixed. For all block sizes, the norms are accounted for in $R_{M,N}^{i_{QP}}(m,n)$. $\boldsymbol{T}_8'$ has a single norm for all base functions. For 4×8 or 8×4 blocks, the choice of $R_{M,N}^{i_{QP}}(m,n)$ depends only on the parity of n or m, respectively. For 8×8 blocks, the value of $R_{M,N}^{i_{QP}}(m,n)$ only depends on QP. The values for $R_{M,N}^{i_{QP}}(m,n)$ for all block sizes are given in [3].

The low-dynamic 4×4 transform and quantization allow for an implementation using a dynamic range of 16 bit maximum for all operations. The additional ABT transforms can be implemented using only 16-bit multiplications and 16-bit memory access [16].

III. Inter Coding and Intra Coding

A. Inter Coding

In H.264/AVC, a tree-structured motion-prediction scheme is employed. Each 16×16 macroblock can be motion-compensated using one of four macroblock modes. These modes differ

$$\boldsymbol{T}_8 = \begin{pmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{pmatrix}. \tag{3}$$

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

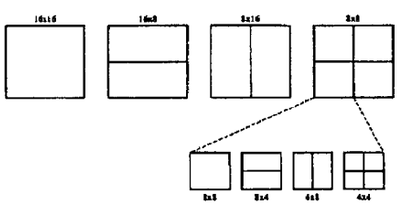


Fig. 1. Block modes available for motion prediction in H.264/AVC. For each subblock, a motion vector is encoded. In the 8 × 8 mode, each 8 × 8 block can be individually divided into subblocks of sizes 8 × 8 down to 4 × 4.

in the number and size of subblocks with common motion vectors. In 8 × 8 mode, the subblocks can be further divided into subblocks of sizes 8 × 8, 8 × 4, 4 × 8, or 4 × 4.

Applying these flexible motion-prediction modes, the resolution of the underlying motion model can be adapted to the content of the encoded video sequence. The block tiling of the motion-prediction modes is depicted in Fig. 1.

The motion vectors are represented in quarter-pel resolution. Multiple reference frames can be exploited for prediction. A generalized concept of B-frames is introduced in H.264/AVC. Besides direct, forward, and backward-prediction modes, bipredictive coding allows employing almost any two frames from the frame store for prediction.

1) ABT Inter Coding: The basic idea of ABT for inter coding is to connect the block size used for transform coding of the prediction error to the block size used for motion compensation. The maximum feasible signal length is exploited for transform coding. As the subblock size is already indicated for motion compensation, no additional side information needs to be encoded for inter ABT.

The maximum transform block size used for ABT is restricted to 8 × 8 pixels. Larger block sizes are omitted due to increase complexity and dynamic range restrictions. Larger transform blocks also tend to spread more ringing artifacts into the reconstructed signal.

2) Motion Estimation: In the JM, the sum of absolute transformed differences (SATD) is applied for subpel motion estimation. The SATD reflects the application of a transform to the prediction error. It accounts for the amount of prediction error, as well as the cost for transformed representation. The cost can be seen as an estimate for the rate spent for encoding. Hence, the SATD operates as an approximative RD cost criterion for the prediction error that can be used for motion vector search.

For SATD calculation, the 4 × 4 Hadamard transform H_4 is applied to each 4 × 4 prediction error block. The resulting transformed signal emulates the frequency characteristics of the true transformed block in the subsequent transformation/quantization stage at very low computational cost.

Let D_{src} be a motion-compensated prediction error block of size 4 × 4 pixel. Then

$$D_H = H_4 \cdot D_{\text{src}} \cdot H_4^T \tag{5}$$

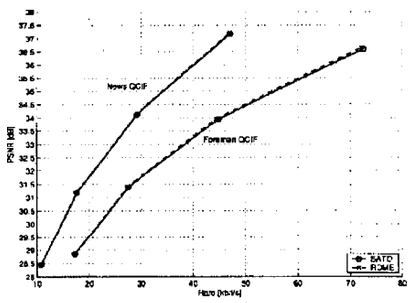


Fig. 2. Comparison of the coding performance for ABT when using the SATD (o) or the RD cost (RDME, ×) for motion estimation. Results are presented for the test sequences FOREMAN and NEWS. The sequences were encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels.

denotes the 2-D transform of the difference block. The SATD of the 4 × 4 block is then given as

$$\text{SATD} = \sum_{i=0}^{3}\sum_{j=0}^{3} |d_H(i,j)| \tag{6}$$

where $d_H(i,j)$ denotes the coefficient of D_H on position (i,j). For motion estimation, the SATD is computed and accumulated for all 4 × 4 blocks of the current subblock. The bit rate needed for encoding of the motion vector is weighted by the quantizer dependent weighting factor λ_{ME} (QP) and added to the SATD to constitute the subblock cost criterion. For details on the weighting factor λ_{ME} (QP) see [17].

In the case of ABT, the calculation of the SATD is modified to account for the new transform sizes. Since the Hadamard transform can be defined for all signal lengths that are multiples of 4 [18], the SATD calculation can be extended straightly to the ABT block sizes.

As stated before, the application of the Hadamard transform emulates the transform step in the transformation/quantization stage. The suitability of the Hadamard transform for SATD calculation of blocks larger than 4 × 4 needs to be verified. The ABT SATD is compared to the performance of a more elaborate RD criterion where the encoded bit rate for the motion vectors and the coefficients is weighted against the sum of the squared reconstructed prediction error. The squared λ_{ME} (QP) was found to give the optimum performance for the RD based motion estimation. Experiments show that the performance of the ABT SATD is very close to the performance of the RD criterion. An example is shown in Fig. 2.

The performance improvement for the single ABT inter coding modes is demonstrated for the test sequence COASTGUARD in Fig. 3. ABT is employed for inter coding only. The RD plots compare the performance for the single prediction block modes when using the 4 × 4 transform or the corresponding $N \times M$ transform employed with ABT. Since

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

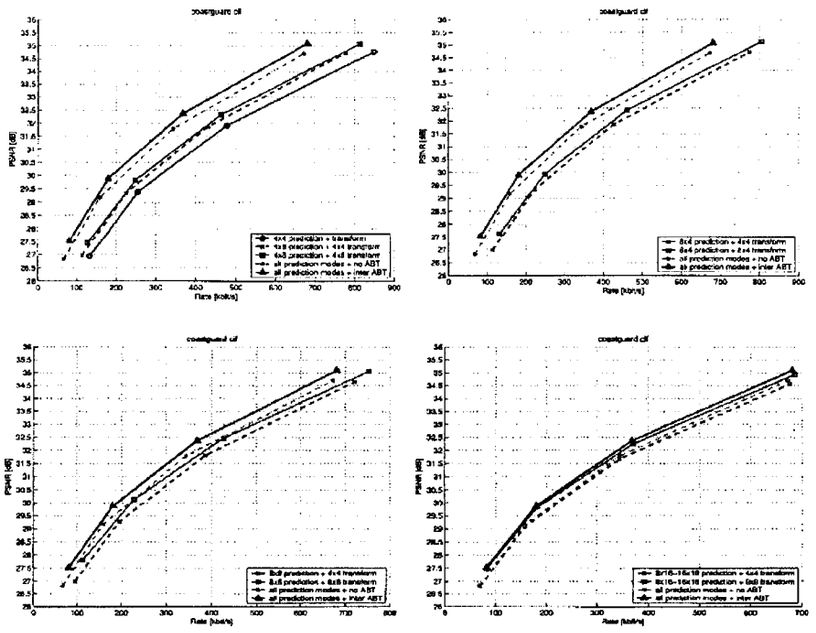


Fig. 3. Comparison of the RD performance with and without ABT for single prediction modes. Results are presented for the test sequence COASTGUARD. The sequence was encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels. ABT was used only for inter-coded blocks. Intra blocks were encoded without ABT.

the 8 × 8 transform is used for macroblocks in 8 × 16, 16 × 8, and 16 × 16 mode, these modes were combined in one simulation. The RD performance when using all prediction modes is compared to the performance of the single modes. As can be seen from Fig. 3, the performance gain for ABT increases with the block size used for prediction and transformation. At low rates, the performance when using only the modes larger than 8 × 8 is comparable to the performance when all modes are used. The benefit of the modes with smaller block sizes becomes more pronounced at higher rates where these expensive modes are chosen more frequently.

B. Intra Coding

Two basic intra-prediction modes are used in H.264/AVC, comprising 4 × 4 and 16 × 16 block-wise prediction from previously decoded blocks. In intra 4 × 4 prediction mode, each 4 × 4 block is predicted by the adjacent pixels of the decoded neighboring 4 × 4 blocks. DC prediction and eight directional prediction modes can be applied to a 4 × 4 block. The directional prediction modes are shown in Fig. 4. Different prediction modes can be selected for each of the 16 4 × 4 blocks in a macroblock. The mode information has to be transmitted. To this end, a probability map is generated from the prediction modes of

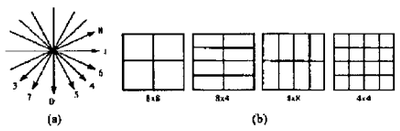


Fig. 4. (a) Intra-prediction directions for non-ABT and ABT coding. (b) Tiling of an intra macroblock for ABT intra coding including the 4 × 4 tiling for non-ABT coding. For each subblock a separate prediction direction can be employed.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

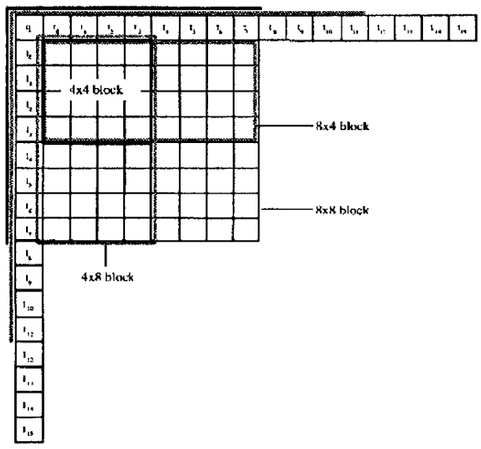


Fig. 5. Pixels t_i, l_i, and q of the top and left neighboring subblocks that may be used for intra prediction of 4×4, 4×8, 8×4, and 8×8 blocks. The maximum number of prediction pixels for an $N \times M$ block is $N + M$ pixels in each direction.

the encoded neighboring blocks. The selected prediction mode is encoded according to its probability in this map.

Smooth macroblocks containing little detail can be predicted more efficiently on a full macroblock basis. This kind of prediction is provided by the intra 16×16 Mode. Here, the current macroblock is predicted by the edge pixels of the adjacent left and top macroblocks. There are four different prediction modes, including DC, horizontal, vertical, and plane prediction.

Blocks in intra 16×16 Mode employ an additional Hadamard transform of the 4×4 DC coefficients after the 4×4 transform of the 16×16 intra prediction error. The two-stage transform decorrelates the signal in the lowest frequencies for the 16×16 block.

1) Intra Coding With Variable Block Sizes: The two-stage transform in the intra 16×16 Mode can be interpreted as a 16×16 transform for the lowest frequencies. Therefore, the choice of two intra modes specified in H.264/AVC can already be seen as a choice between two transform modes. For ABT intra coding, the prediction modes specified for the 4×4 intra-prediction mode are extended to the block sizes used for ABT inter transform coding, see Fig. 4. Hence, the gap between 4×4 intra coding and 16×16 intra coding is closed. With ABT, the encoder can trade energy compaction by larger transforms against the size of the region a single prediction direction is applied to.

An additional codeword has to be transmitted that indicates the block size used for intra prediction. The prediction block size corresponds to the block size used for transform coding. The prediction modes are encoded with probability maps as described for the intra 4×4 Mode. Since the number of subblocks is reduced, the overall overhead information for prediction block size and prediction modes for intra ABT can be smaller than with intra 4×4 Mode.

The number of neighboring pixels that are employed for intra prediction varies with the prediction mode. For example, for the horizontal and the vertical prediction mode, exclusively pixels from the left or the top are used, respectively. The reference pixels that may be used for intra-block prediction are shown in Fig. 5. Only pixels of previously decoded subblocks may be used for prediction. If prediction pixel positions required for a prediction mode are not available (e.g., because the corresponding neighboring block has not been not processed yet), the value of the last available pixel position is padded to the unavailable positions.

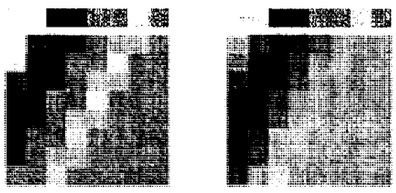
The maximum number of pixels for directional prediction is used for the *diagonal down/left* prediction (mode 3, see Fig. 4. Here, $N + M$ pixels are used in both the horizontal and vertical direction for one $N \times M$ block.

The prediction from skew directions might introduce visible artifacts into the prediction signal that are unfavorable for an efficient compression. This effect becomes prominent especially for block sizes larger than 4×4 since more pixels are predicted from the same source. Hence, the new ABT intra modes are afflicted. To reduce the artifacts, the pixels used for prediction are filtered with a low-pass filter.

Conceptually, a filter

$$h[n] = \frac{[1 \quad 2 \quad 1]}{4} \tag{7}$$

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。



(a)　(b)

Fig. 6. Prediction mode 7 (*vertical/left*) applied to an 8 × 8 block. The last row of the top neighboring 8 × 8 block is used for prediction. (a) Without pre-filtering. (b) Prediction from the same pixels with pre-filtering.

is applied to the vector of reference pixels

$$p[n] = [t_{\max} \quad \dots \quad t_0 \quad q \quad t_0 \quad \dots \quad t_{\max}]^T \qquad (8)$$

used for prediction. The reference pixels $p[n]$ are indicated in Fig. 5. The application of the filter sufficiently reduces prediction artifacts for all block sizes. For block sizes larger than 4 × 4, pre-filtering is applied for horizontal and vertical prediction directions as well. In Fig. 6, the impact of the pre-filtering procedure is demonstrated for *vertical/left* prediction of an 8 × 8 block.

2) Mode Decision for ABT Intra Coding: The optimum intra ABT mode for a macroblock is selected in a two-stage process. For all possible block sizes, the best prediction mode for each subblocks is determined. Then, the cost for encoding the macroblock with the selected subblock prediction modes is evaluated for all block sizes. The corresponding mode with the optimum overall cost is chosen for encoding the macroblock.

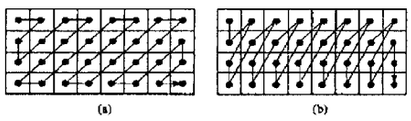
Both decisions can be made based either on the SATD or on a RD cost criterion. The RD criterion weights the rate for the transform coefficients and the corresponding prediction mode against the sum of the squared reconstructed prediction error. The weighting factor used for RD optimized macroblock mode selection in H.264/AVC can be employed here [17].

Examples for the intra mode decision on natural and synthetic test images are presented in Figs. 9 and 11.

C. Deblocking Filter

The application of a 4 × 4 transform tends to promote blocking artifacts in the reconstructed video, especially at lower bit rates. Therefore, a normative in-loop deblocking filter is employed in H.264/AVC. The reconstructed frames are filtered before application for motion compensation in the prediction loop. The functionality of this filter is essential for the subjective performance of the coder, especially at lower bit rates.

The strength of the filter is determined by the chosen macroblock mode, the existence of nonzero transform coefficients, the applied quantizer, and by the motion vector activity. In general, stronger filtering is applied to intra blocks than inter blocks. A maximum of two pixels on each side of the boundary of two adjacent blocks is affected by applying the deblocking filter on 4 × 4 blocks. Intra blocks coded with the intra 16 × 16 mode are filtered only at the macroblock boundaries. Here, a maximum of three boundary pixels may be filtered.



(a)　(b)

Fig. 7. (a) Zig zag scan for frame based coding and (b) alternate scan for field based coding of an 8 × 4 block with ABT.

The mode of operation of the deblocking filter needs to be adjusted for application to ABT-coded blocks. The strength of the filter is increased relative to the strength for 4 × 4 blocks according to the sizes of the adjacent blocks. The strength is increased most for two neighboring 8 × 8 blocks. Deblocking is only performed at transform block boundaries. Hence, the number of filter operations is reduced by ABT coding.

For example, if the 4 × 4 transform is employed, an 8 × 8 block has eight 4 × 4 block boundaries to be filtered. These are four inner boundaries, two on the left and two at the top. The boundaries to the right and at the bottom are encountered for the next blocks. Accordingly, an ABT 8 × 8 block has only two boundaries to be filtered. Similarly, for 4 × 8 and 8 × 4 blocks, the number of filtered boundaries is reduced from four to two.

D. Entropy Coding

H.264/AVC features two entropy coding methods. Variable length coding is supported by all standard compliant decoders. It employs a CAVLC for transform coefficients and a universal VLC for all other symbols. For high coding efficiency, CABAC is employed. Here, nonbinary symbols are binarized, e.g., using a unary code tree. For the separate bins, contexts are generated that are used to adapt the arithmetic coder to the statistics of the encoded symbols. For details, see [19].

Run-length coding is employed for the transform coefficients. Two types of scans can be employed to represent the transform coefficients as (level, run) symbols. The well-known zig-zag scan is employed for frame based encoding. In the case of field-based encoding of interlaced material, an alternate scan is used that accounts for the modified signal statistics in the fields of interlaced video. The alternate scan was originally proposed for ABT in [20]. An example for the zig-zag scan and the alternate scan for an 8 × 4 block is shown in Fig. 7. A VLC design for ABT transform coefficients was proposed in [21].

E. CABAC Transform Coefficient Encoding

The binarization and context generation for the transform coefficients for 4 × 8, 8 × 4,and 8 × 8 blocks is identical to the method specified for 4 × 4 blocks. Separate contexts are used for the symbols of each block size.

- For each encoded block, a flag is signaled to indicate whether the block contains coefficients or not.
- A significance map is generated, indicating the position of nonzero coefficients in the scan. For each significant coefficient a "last" flag is encoded. This flag indicates whether the last significant coefficient in the scan is reached.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

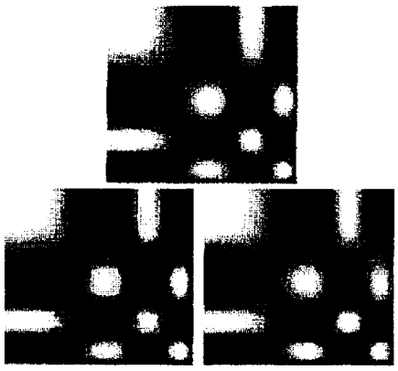


Fig. 8. Subjective comparison: Detail of a chirp image coded with QP = 36. Top: Original. Left: Encoded without ABT (187 720 bit, 30.25 dB). Right: Encoded with ABT (158 304 bit, 32.53 dB).

- The absolute values of the coefficients are binarized using an unary code tree and encoded in reverse order (last coefficient first). The sign of the coefficients is written directly to the bitstream.

IV. Simulation Results

The simulation results given here were generated with the JM 4.2 reference software [22]. ABT is implemented according to [16].

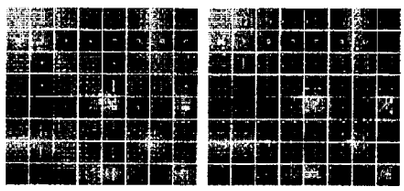
A. Subjective Performance

Two examples for the subjective performance of ABT are given.

Fig. 8 shows an example of the coding performance for a synthetic image. The original gray-level image of size 512 × 512 contains a separable chirp signal with the frequency increasing from left to right and top to bottom. The image was encoded as a single frame sequence with RD optimization at QP = 36. Fig. 8 shows the top left corner of the chirp image. As can be seen, the ABT-coded image reveals a much more pleasing subjective quality than the non-ABT image does. Also, the rate and PSNR numbers indicate superior objective performance for ABT.

In Fig. 9, the selected intra-prediction modes for ABT and non-ABT coding are shown. For non-ABT, the high number of macroblocks in intra 4 × 4 Mode with many different prediction directions is very expensive. The larger transform block sizes employed with ABT give an improved compression performance at a convenient subject quality.

In Fig. 10, a detail of first frame of the the HD sequence Shuttle (1280 × 720 @ 60 Hz) is shown. The sequence was encoded with RD optimization at QP = 36. It can be seen that more detail is preserved by ABT coding. The subjective quality is improved by ABT, even though the PSNR numbers indicate a slight loss.



(a)　(b)

Fig. 9. Intra mode selection for the region shown in Fig. 8. (a) Corresponding intra mode selection without ABT. (b) Corresponding intra mode selection wit ABT. Macroblocks are marked white, subblocks of 4 × 4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16 × 16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16 × 16 plane prediction is indicated by an empty square (□).

Fig. 11 shows the intra-prediction modes that are chosen for ABT and non-ABT coding. Without ABT, high detail regions are encoded using the intra 4 × 4 Mode. In regions with less activity, the intra 16 × 16 Mode is employed. When ABT is used the number of macroblocks in intra 16 × 16 Mode is largely reduced. The ABT 8 × 8 mode is chosen most frequently. In regions with much variation, smaller block sizes are employed. Blocks of size 4 × 4 are rarely used for intra coding.

B. RD Performance

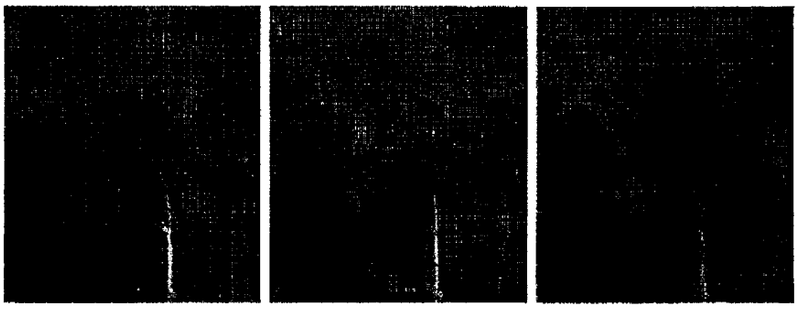
The sequences used for the simulations are given in Table I. The sequences Raven and Kayak contain high-definition (HD) video. Raven in CIF size was extracted from the full-resolution 1280 × 720 at 60-Hz sequences by windowing the center of the full-resolution frames and temporal down-sampling by a factor of two. No spatial or temporal filtering was applied. This sequence emulates the characteristics of the full-resolution sequences while saving simulation time. All sequences but Kayak contain progressive video. Kayak is an interlaced sequence.

1) Simulation Conditions: In all simulations, RD optimization was switched on. Results for two configurations are presented here. In the first simulation, intra only coding is performed. The inter coding performance was evaluated using an IPPP... and an IBBPBBP... frame structure with a search range of 32 pixels and five reference frames. The sequence Kayak was encoded in field mode using a search range of 16 pixels and two reference frames. The search range was reduced here to shorten the simulation time. ABT was used for intra and inter macroblocks in the ABT simulations.

All sequences were encoded using a fixed quantization parameter QP for all frames. The HD sequences were encoded using QP = [20, 24, 28, 32] for I and P frames. The other sequences were encoded using QP = [28, 32, 36, 40] for I and P frames. The quantization parameter QP was increased by two for B frames.

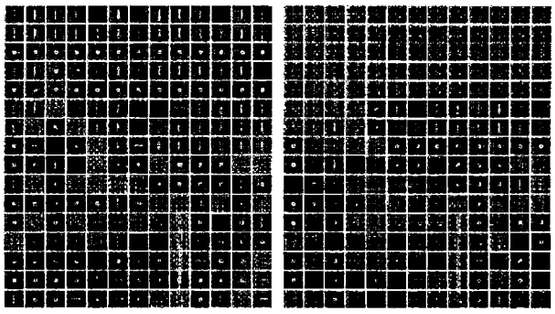
2) Results: Table II shows the rate savings and PSNR gains for the simulation conditions given above. Overall, average rate savings of about 8% and PSNR gains of 0.49 dB can be observed for ABT intra coding. The inter gains are lower than the gains achieved for intra coding. This might be due to the re-

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。



(a)　(b)　(c)

Fig. 10. Subjective comparison: Detail of the first frame of the Shuttle sequence coded with QP = 36. (a) Original. (b) Encoded without ABT (39128 bit, 38.36 dB). (c) Encoded with ABT (38784 bit, 38.25 dB). More detail is preserved with ABT coding (note the flag staff at the top left of the depicted area).



(a)　(b)

Fig. 11. Intra mode selection for the region shown in Fig. 10. (a) Encoded without ABT. (b) Encoded with ABT. Macroblocks are marked white. Subblocks of 4 × 4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16 × 16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16 × 16 plane prediction is indicated by an empty square (□).

TABLE I
TEST SEQUENCES

Sequence	Frame Size	Freq	# Frames
FOREMAN	176 × 144	10 Hz	100
COASTGUARD	352 × 288	30 Hz	300
TABLETENNIS	352 × 288	30 Hz	300
RAVEN	352 × 288	30 Hz	300
KAYAK	1920 × 1088	60 Hz	60

TABLE II
OVERALL RATE SAVINGS AND PSNR GAINS FOR ABT CODING

	III		IPPP		IBBP	
Sequence	ΔRate	ΔPSNR	ΔRate	ΔPSNR	ΔRate	ΔPSNR
FOREMAN	-6.89 %	0.43 dB	-5.71 %	0.32 dB	-2.49 %	0.14 dB
COASTG.	-6.20 %	0.33 dB	-8.38 %	0.31 dB	-5.24 %	0.17 dB
TABLET.	-4.33 %	0.23 dB	-4.05 %	0.16 dB	-3.51 %	0.13 dB
RAVEN	-11.65 %	0.58 dB	-8.42 %	0.37 dB	-5.88 %	0.25 dB
KAYAK	-12.36 %	0.90 dB	-9.41 %	0.72 dB	-12.77 %	0.92 dB
Δmean	-8.28 %	0.49 dB	-7.19 %	0.38 dB	-5.98 %	0.32 dB

duced energy and the reduced correlation remaining in the motion predicted signal. Hence, the decorrelation properties of the larger transforms takes less effect. For IPPP, overall average rate savings of about 7% and PSNR gains of 0.38 dB are achieved. When B frames are employed, average rate savings of about 6% and PSNR gains of 0.32 dB are observed for the test set. The delta measurements were performed using the method described in [23]. RD plots for the III and IPPP frame structure are given for the sequences RAVEN and KAYAK in Fig. 12. At high rates, gains of more than 1.0 dB can be observed.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

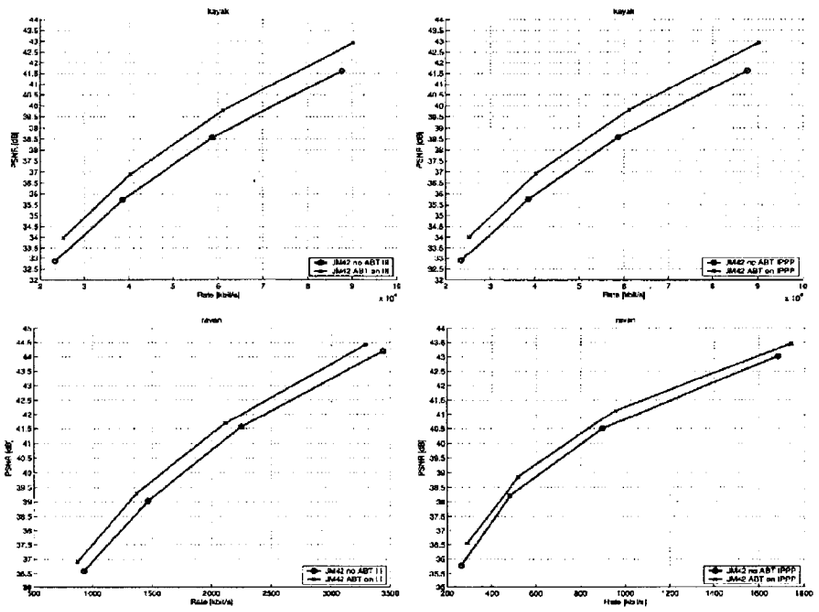


Fig. 12. Simulation results for intra only and inter coding with JM 4.2 without ABT (o) and with ABT (×) for the sequences RAVEN and KAYAK.

V. Conclusion

The concept of ABT for application in H.264/AVC was presented. With ABT, the block size used for transform coding can be aligned to the properties of the encoded signal. Blocks of size 8×8, 8×4, 4×8 and 4×4 can be transformed. For inter coding, no additional overhead information is needed for ABT. For intra coding an additional symbol is transmitted indicating the transform and prediction block size.

The results reveal an increased subjective and objective quality for ABT coding, especially on HD material. Up to 12% rate savings and 0.9 dB increase in PSNR can be observed.

Appendix
Fast Calculation of the 8 × 8 Transform

Let $\mathbf{c}$ be a coefficient vector of a transformed signal

$$\mathbf{c} = [c_0 \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7]^T \tag{9}$$

and $\mathbf{y}$ be the reconstructed signal itself

$$\mathbf{y} = T_8^I \cdot \mathbf{c}. \tag{10}$$

The transform can be calculated as follows:

$$\begin{aligned}
k_0 &\leftarrow 13(c_0 + c_4); \quad k_4 \leftarrow c_1 + c_7 \\
k_1 &\leftarrow 17c_2 + 7c_6; \quad k_5 \leftarrow c_3 + c_5 \\
k_2 &\leftarrow 13(c_0 - c_4); \quad k_6 \leftarrow c_1 - c_7 \\
k_3 &\leftarrow 7c_2 - 17c_6; \quad k_7 \leftarrow c_3 - c_5 \\
z_0 &\leftarrow k_0 + k_1 \\
z_1 &\leftarrow k_2 + k_3 \\
z_2 &\leftarrow k_2 - k_3 \\
z_3 &\leftarrow k_0 - k_1 \\
z_4 &\leftarrow ((k_5 + c_5) \ll 3) + 3k_4 + k_7 + (c_1 \ll 4) \\
z_5 &\leftarrow ((k_6 + c_1) \ll 3) + 3k_7 - k_4 - (c_5 \ll 4) \\
z_6 &\leftarrow ((k_4 + c_7) \ll 3) - 3k_5 + k_6 - (c_3 \ll 4) \\
z_7 &\leftarrow -((k_7 + c_3) \ll 3) + 3k_6 + k_5 - (c_7 \ll 4) \\
y_0 &\leftarrow z_0 + z_4; \quad y_4 \leftarrow z_3 - z_7 \\
y_1 &\leftarrow z_1 + z_5; \quad y_5 \leftarrow z_2 - z_6 \\
y_2 &\leftarrow z_2 + z_6; \quad y_6 \leftarrow z_1 - z_5 \\
y_3 &\leftarrow z_3 + z_7; \quad y_7 \leftarrow z_0 - z_4.
\end{aligned}$$

$a \ll n$ denotes the bit shift operation $a \cdot 2^n$.

本複製物は、特許庁が著作権法第４２条第２項第１号の規定により複製したものです。
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

ACKNOWLEDGMENT

The author would like to thank Prof. J. R. Ohm, Aachen University, and U. Benzler, R. Bosch GmbH, as well as A. Dahlhoff and C. Mayer, Aachen University, for many insightful discussions on the topic. The author would also like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] "Information Technology—Generic Coding of Audio-Visual Objects," ISO/IEC JTC1 IS 14496-2 (MPEG-4), 1998.
[2] Telecommun. Standardization Sector of the ITU, "Video Coding for Low Bitrate Communication (H.263 Version 2)," Sept., 1997.
[3] T. Weigand, "Joint final committee draft (JFCD) of joint video specification," in *Proc. JVT, 4th Meeting*, T. Weigand, Ed., Klagenfurt, Austria, July 2002, ITU-T rec. H.264/ISO/IEC 14 496-10 AVC.
[4] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '91*, Toronto, ON, Canada, May 1991, pp. 2713–2716.
[5] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477–486, Apr. 1990.
[6] C.-T. Chen, "Adaptive transform coding via quadtree-based variable block size DCT," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing ICASSP '87*, vol. 3, Glasgow, Scotland, U.K., May 1989, pp. 1854–1857.
[7] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Feb. 1993.
[8] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 2040–2060, Aug. 1992.
[9] ——, "Variable block size image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '87*, vol. 2, Dallas, TX, Apr. 1987, pp. 1051–1054.
[10] T. J. Kliusutis and V. K. Madisetti, "Variable block size adaptive lapped transform-based image coding," in *Proc. IEEE Int. Conf. Image Processing ICIP '97*, vol. 3, Washington, DC, Oct. 1997, pp. 686–689.
[11] K. Ramchandran, Z. Xiong, K. Asai, and M. Vetterli, "Adaptive transforms for image coding using spatially varying wavelet packets," *IEEE Trans. Image Processing*, vol. 5, pp. 1197–1204, July 1996.
[12] I. Dinstein, K. Rose, and A. Heimann, "Variable block-size transform image coder," *IEEE Trans. Commun.*, vol. 38, pp. 2073–2078, Nov. 1990.
[13] H. Malvar, "Low-complexity length-4 transform and quantization with 16-bit arithmetic," in VCEG, 14th Meeting, Santa Barbara, CA, Sept. 2001, Doc. VCEG-N44.
[14] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/avc," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 598–603, July 2003.
[15] G. Bjontegaard, "Addition of 8 × 8 transform to H.26L," in SG16/Q15, 9. Meeting, Red Bank, NJ, Oct. 1999, Doc. Q15-I-39.
[16] M. Wien, "Clean-up and improved design consistency for ABT," in JVT, 5th Meeting, Geneva, Switzerland, Oct. 2002, Document JVT-E025.
[17] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 688–703, July 2003.
[18] F. E. Douglas and K. R. Rammohan, *Fast Transforms*. San Diego, CA: Academic, 1982.
[19] D. Marpe, H. Schwarz, and T. Wiegand, "Advanced entropy coding in draft h.264/avc video compression standard," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 620–636, July 2003.
[20] W. Limin and B. David, "Alternate coefficient scanning patterns for interlaced ABT coding," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C140.
[21] W. Mathias and D. Achim, "16 bit adaptive block size transforms," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C107.
[22] K. Sühring, Ed., (2002) JM 4.2 Reference Software. [Online]. Available: ftp://ftp.imtc.org/jvt-experts/reference_software/jm42.zip
[23] G. Bjontegaard, "Calculation of average PSNR differences between rd curves," in SG16/Q6 VCEG, 13. Meeting, Austin, TX, Apr. 2001, Doc. VCEG-M33.



Mathias Wien (S'98–M'03) was born in Bonn, Germany, in 1971. He received the Dipl.-Ing. degree in electrical engineering from Aachen University, Aachen, Germany, in 1997, where he is currently working toward the Dr.-Ing. degree in the Institute of Communications Engineering.

He is an active contributor to the ITU-T VCEG and the Joint Video Team of VCEG and ISO/IEC MPEG where he chaired the AdHoc Group on Additional Transforms and Quantization. His research interests are in the area of image and video processing, with emphasis on space-frequency adaptive and scalable video compression.

Variable Block-Size Transforms for H.264/AVC

Mathias Wien, *Member, IEEE*

***Abstract*—A concept for variable block-size transform coding is presented. It is called adaptive block-size transforms (ABT) and was proposed for coding of high resolution and interlaced video in the emerging video coding standard H.264/AVC. The basic idea of inter ABT is to align the block size used for transform coding of the prediction error to the block size used for motion compensation. Intra ABT employs variable block-size prediction and transforms for encoding. With ABT, the maximum feasible signal length is exploited for transform coding. Simulation results reveal a performance increase up to 12% overall rate savings and 0.9 dB in peak signal-to-noise ratio.**

***Index Terms*—Block-based hybrid video coding, H264/AVC, integer transforms, variable block-size transforms.**

I. Introduction

THE Joint Video Team (JVT) of ITU-T SG16 WP3 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) is currently developing a joint video coding standard. It is referenced as recommendation H.264 for the ITU-T and MPEG-4 Part 10 Advanced Video Coding (AVC) for ISO [1]. The work started off as a design process for a new ITU-T recommendation succeeding H.263 [2] and was merged in late 2001 to a joint project with MPEG. The standardization process is reflected in the Joint Model (JM) software that is used for evaluation of the tools presented to JVT. In this paper, the status of the tools is described as of the Joint Final Committee Draft (JFCD) of the standard [3].

The coder is based on the hybrid coding scheme including sophisticated prediction modes and efficient adaptive entropy coding techniques to achieve improved compression performance. Blocks of size 16×16 down to 4×4 pixels can be employed for motion prediction. The prediction modes are organized in a tree-structured manner, allowing even for flexible combination of different motion compensation block sizes inside a 16×16 pixel macroblock. Main features include also multiple reference frames for inter prediction and a generalized B-frame concept.

For improved intra-compression performance, various prediction methods are specified. The adjacent pixels of the already decoded neighboring blocks are used for intra prediction. A fixed 4×4 block transform is applied to the prediction error signal of inter and intra-prediction modes. For intra macroblocks, an additional transform for 4×4 DC coefficients can be applied.

Manuscript received February 26, 2002; revised May 10, 2003. This work was supported by R. Bosch GmbH, Hildesheim.

The author is with the Institut und Lehrstuhl für Nachrichtentechnik, RWTH Aachen University, Aachen, Germany (e-mail: wien@ient.rwth-aachen.de)

Digital Object Identifier 10.1109/TCSVT.2003.815380

Two entropy coding methods are specified for application in different profiles of H.264/AVC. The Base Line profile and the Extended profile employ context-adaptive variable length codes (CAVLC) for the transform coefficients and universal variable length codes (UVLC) for all other symbols. For high coding efficiency context adaptive binary arithmetic coding (CABAC) is employed in the Main profile. The standard is specified to allow for an implementation using only 16 bit multiplications and 16 bit memory access.

In this paper, the concept of variable block-size transform coding is presented. The scheme is called adaptive block-size transforms (ABT), indicating the adaptation of the transform block size to the block size used for motion compensation. For intra coding, the size of the block transform is adapted to the properties of the intra-prediction signal. Thus, for both inter and intra coding, the maximum feasible signal length can be exploited by the transform. With ABT, the block transform as commonly used in the former image and video coding standards is generalized from a fixed-size transform to a signal-adaptive tool for increased overall coding performance.

The large transforms can provide a better energy compaction and a better preservation of detail in a quantized signal than a small transform does. On the other hand, larger transforms introduce more ringing artifacts caused by quantization than small transforms do. By choosing the transform size according to the signal properties, the tradeoff between energy compaction and preserved detail on the one hand and ringing artifacts on the other can be optimized.

Various proposals for variable block-size coding of images and video are known from the literature. Many of them are based on quadtree structures either for prediction or for transform coding. Quadtree-based variable block-size motion compensation was proposed, e.g., in [4], [5]. A quadtree-based variable block-size discrete cosine transform (DCT) was introduced in [6]. In this paper, the choice of the appropriate block size was based on a mean-based decision rule. However, this approach did not lead to rate/distortion (RD) optimal block tilings [7]. An image compression scheme with variable block-size segmentation was presented in [8], [9]. Here, the amount of detail in the blocks was used for determination of the employed block size. Classified vector quantization (VQ) was employed for high detail 4×4 blocks. A combination of transform coding and VQ was used for nonhigh detail 8×8, 16×16, and 32×32 blocks. Quadtree structures were also used with lapped transforms, e.g., in [10]. An RD optimized quadtree-based variable block-size coder using wavelet packets was proposed in [11]. In [12], a variable block-size transform image coder was presented that is not limited to square blocks. Here, block sizes of 8, 16, or 32 pixels per edge were used. The appropriate block size was selected according to a local activity index of the image scene.

1051-8215/03$17.00 © 2003 IEEE

BNSDOCID <XP_____1089253A_1_>

The concept of ABT presented in this paper merges the benefits from variable block-size prediction and variable block-size transform coding.

The organization of the paper is as follows. In Section II, the transformation and quantization used for ABT coding are presented. Section III introduces the application of ABT for inter and intra coding in H.264/AVC. Extensions and modifications of the techniques used in H.264/AVC that are necessary for ABT coding are highlighted. In Section IV, the performance of ABT is analyzed. Finally, conclusions are drawn in Section V.

II. Variable Block Size Transforms

In general, the separable 2-D transform of a 2-D signal can be written as

$$\boldsymbol{C} = \boldsymbol{T}_v \cdot \boldsymbol{B} \cdot \boldsymbol{T}_h^T \tag{1}$$

where $\boldsymbol{B}$ denotes a matrix representing signal block of N pixels and M lines. $\boldsymbol{C}$ is the transform coefficient matrix, whereas $\boldsymbol{T}_v$ and $\boldsymbol{T}_h$ are the $M \times M$ and $N \times N$ transform matrices in vertical and horizontal direction, respectively. The transform matrices comprise a set of orthogonal base functions. $\boldsymbol{T}^T$ is the transposed matrix $\boldsymbol{T}$.

The ABT proposal for H.264/AVC employs transforms for blocks of size 4×4, 4×8, 8×4, and 8×8 pixels. Hence, employing separable vertical and horizontal transforms, a 4×4 and an 8×8 transform need to be specified. Here, the 4×4 transform specified in H.264/AVC is reused. It is a low-dynamic approximation of the 4×4 DCT [13]. In matrix notation, it can be written as

$$\boldsymbol{T}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad \boldsymbol{T}_4^I = \begin{pmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{pmatrix}. \tag{2}$$

Both the transform $\boldsymbol{T}_4$ and the inverse transform $\boldsymbol{T}_4^I$ have different norms for the even and the odd base functions. These have to be regarded in the encoding and decoding process. In the standard, the inverse transform $\boldsymbol{T}_8^I$ is specified as a fast butterfly operation using only additions and bit shift operations. For detail on the 4×4 transform and quantization process (see [14]).

The 8×8 transform matrix proposed for application with ABT is a single-norm approximation of the 8×8 DCT matrix as shown in (3), at the bottom of the page. It was originally proposed in [15]. The transform and the inverse transform $\boldsymbol{T}_8^I = \boldsymbol{T}_8^T$ can be implemented in a butterfly structure, using 36 additions, eight bit-shift operations, and ten multiplications. The algorithm is given in Appendix.

A. Quantization and Normalization

Both, the 4×4 as well as the 8×8 transform are not normalized and therefore require normalization. This operation is integrated into the quantization and de-quantization procedure. Since the decoding process is the relevant operation to the standard, the de-quantization is described here.

In the H.264/AVC specification, only the de-quantization values for the six finest QP need to be stored. The remaining values can be generated by scaling the finest de-quantization values with the appropriate power of 2.

Let $c(m,n)$ denote a decoded coefficient at position (m,n) in an $N \times M$ block. Note that m and n indicate the row (line) and column (pixel) respectively. We use mathematical notation in the equations. The size of blocks in a frame is indicated using pixels $\times$ lines. The de-quantization process can be written as

$$w(m,n) = \left[c(m,n) \cdot R_{M,N}^{i_{QP}}(m,n) \right] \ll \left(\left\lfloor \frac{\mathrm{QP}}{6} \right\rfloor - 2 \right) \tag{4}$$

where $n = 0, \ldots, N-1$, $m = 0, \ldots, M-1$, $i_{QP} = (\mathrm{QP} \bmod 6)$, and $\lfloor \cdot \rceil$ denotes rounding with truncation toward zero ("mod" denotes the modulo operation). $R_{M,N}^{i_{QP}}(m,n)$ is the scaling factor corresponding to the position of the coefficient in the reconstructed block.

The even and odd base functions of $\boldsymbol{T}_4'$ have different norm. Due to these, the norms of coefficients in 4×4 blocks depend on n, m both being even, both odd, or mixed. For all block sizes, the norms are accounted for in $R_{M,N}^{i_{QP}}(m,n)$. $\boldsymbol{T}_8'$ has a single norm for all base functions. For 4×8 or 8×4 blocks, the choice of $R_{M,N}^{i_{QP}}(m,n)$ depends only on the parity of n or m, respectively. For 8×8 blocks, the value of $R_{M,N}^{i_{QP}}(m,n)$ only depends on QP. The values for $R_{M,N}^{i_{QP}}(m,n)$ for all block sizes are given in [3].

The low-dynamic 4×4 transform and quantization allow for an implementation using a dynamic range of 16 bit maximum for all operations. The additional ABT transforms can be implemented using only 16-bit multiplications and 16-bit memory access [16].

III. Inter Coding and Intra Coding

A. Inter Coding

In H.264/AVC, a tree-structured motion-prediction scheme is employed. Each 16×16 macroblock can be motion-compensated using one of four macroblock modes. These modes differ

$$\boldsymbol{T}_8 = \begin{pmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{pmatrix}. \tag{3}$$

BNSDOCID <XP.....1099253A_1_>

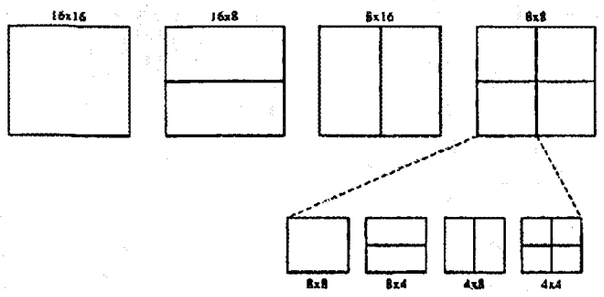


Fig. 1. Block modes available for motion prediction in H.264/AVC. For each subblock, a motion vector is encoded. In the 8 × 8 mode, each 8 × 8 block can be individually divided into subblocks of sizes 8 × 8 down to 4 × 4.

in the number and size of subblocks with common motion vectors. In 8 × 8 mode, the subblocks can be further divided into subblocks of sizes 8 × 8, 8 × 4, 4 × 8, or 4 × 4.

Applying these flexible motion-prediction modes, the resolution of the underlying motion model can be adapted to the content of the encoded video sequence. The block tiling of the motion-prediction modes is depicted in Fig. 1.

The motion vectors are represented in quarter-pel resolution. Multiple reference frames can be exploited for prediction. A generalized concept of B-frames is introduced in H.264/AVC. Besides direct, forward, and backward-prediction modes, bipredictive coding allows employing almost any two frames from the frame store for prediction.

1) ABT Inter Coding: The basic idea of ABT for inter coding is to connect the block size used for transform coding of the prediction error to the block size used for motion compensation. The maximum feasible signal length is exploited for transform coding. As the subblock size is already indicated for motion compensation, no additional side information needs to be encoded for inter ABT.

The maximum transform block size used for ABT is restricted to 8 × 8 pixels. Larger block sizes are omitted due to increase complexity and dynamic range restrictions. Larger transform blocks also tend to spread more ringing artifacts into the reconstructed signal.

2) Motion Estimation: In the JM, the sum of absolute transformed differences (SATD) is applied for subpel motion estimation. The SATD reflects the application of a transform to the prediction error. It accounts for the amount of prediction error, as well as the cost for transformed representation. The cost can be seen as an estimate for the rate spent for encoding. Hence, the SATD operates as an approximative RD cost criterion for the prediction error that can be used for motion vector search.

For SATD calculation, the 4 × 4 Hadamard transform $\boldsymbol{H}_4$ is applied to each 4 × 4 prediction error block. The resulting transformed signal emulates the frequency characteristics of the true transformed block in the subsequent transformation/quantization stage at very low computational cost.

Let $\boldsymbol{D}_{\text{mc}}$ be a motion-compensated prediction error block of size 4 × 4 pixel. Then

$$\boldsymbol{D}_H = \boldsymbol{H}_4 \cdot \boldsymbol{D}_{\text{mc}} \cdot \boldsymbol{H}_4^T \tag{5}$$

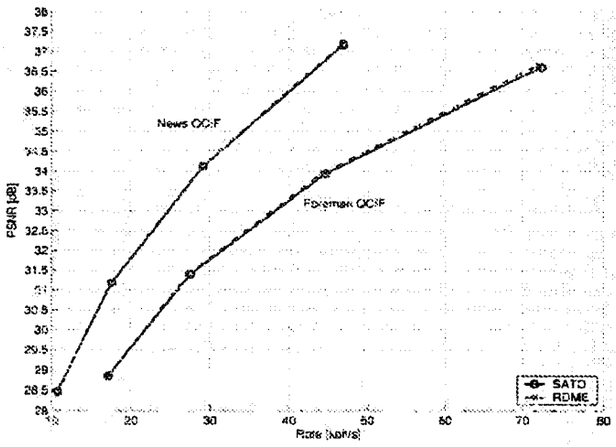


Fig. 2. Comparison of the coding performance for ABT when using the SATD (○) or the RD cost (RDME, ×) for motion estimation. Results are presented for the test sequences FOREMAN and NEWS. The sequences were encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels.

denotes the 2-D transform of the difference block. The SATD of the 4 × 4 block is then given as

$$\text{SATD} = \sum_{i=0}^{3}\sum_{j=0}^{3} |d_H(i,j)| \tag{6}$$

where $d_H(i,j)$ denotes the coefficient of $\boldsymbol{D}_H$ on position (i,j). For motion estimation, the SATD is computed and accumulated for all 4 × 4 blocks of the current subblock. The bit rate needed for encoding of the motion vector is weighted by the quantizer dependent weighting factor λ_{MF} (QP) and added to the SATD to constitute the subblock cost criterion. For details on the weighting factor λ_{MF} (QP) see [17].

In the case of ABT, the calculation of the SATD is modified to account for the new transform sizes. Since the Hadamard transform can be defined for all signal lengths that are multiples of 4 [18], the SATD calculation can be extended straightly to the ABT block sizes.

As stated before, the application of the Hadamard transform emulates the transform step in the transformation/quantization stage. The suitability of the Hadamard transform for SATD calculation of blocks larger than 4 × 4 needs to be verified. The ABT SATD is compared to the performance of a more elaborate RD criterion where the encoded bit rate for the motion vectors and the coefficients is weighted against the sum of the squared reconstructed prediction error. The squared λ_{ME} (QP) was found to give the optimum performance for the RD based motion estimation. Experiments show that the performance of the ABT SATD is very close to the performance of the RD criterion. An example is shown in Fig. 2.

The performance improvement for the single ABT inter coding modes is demonstrated for the test sequence COASTGUARD in Fig. 3. ABT is employed for inter coding only. The RD plots compare the performance for the single prediction block modes when using the 4 × 4 transform or the corresponding $N \times M$ transform employed with ABT. Since

BNSDOCID: <XP_____1099253A_>

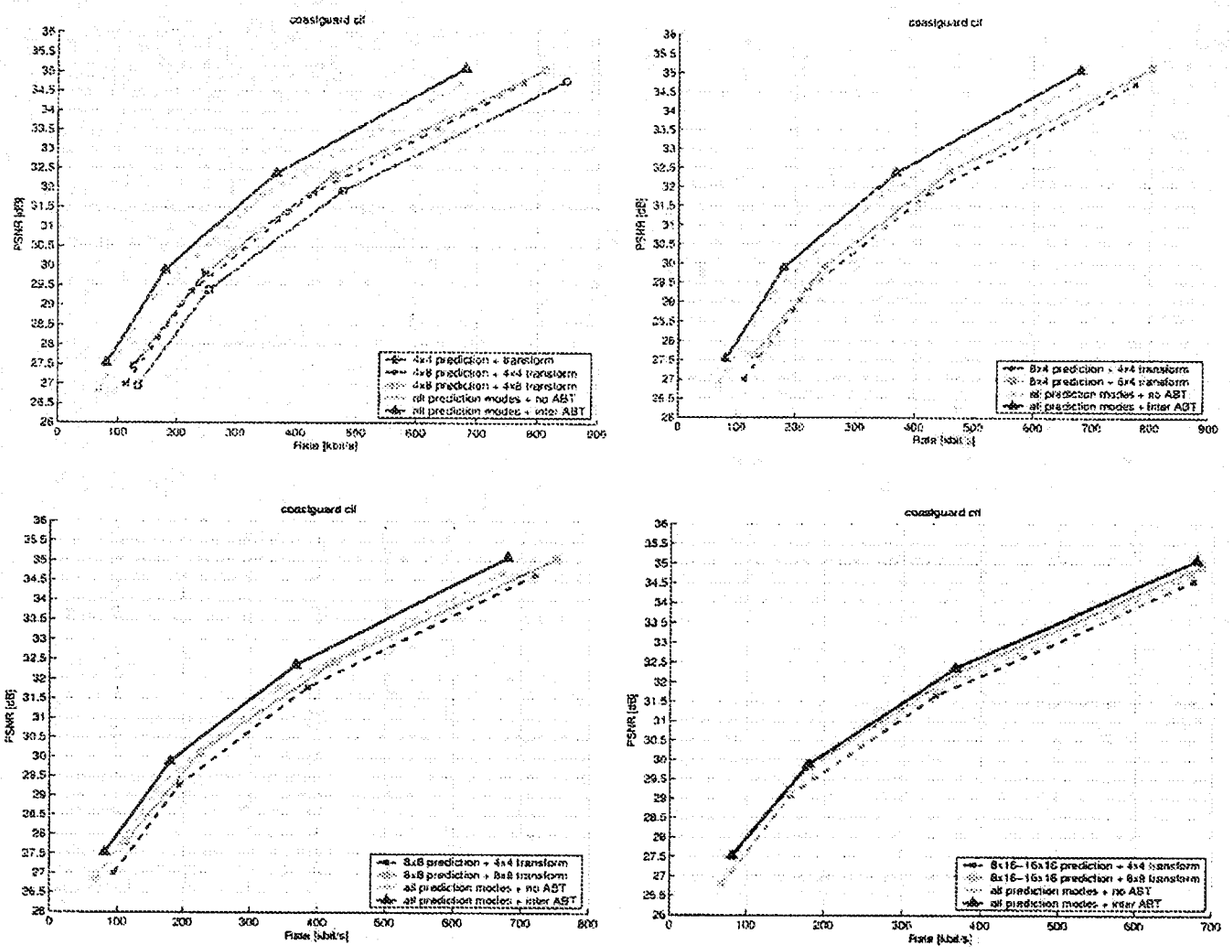


Fig. 3. Comparison of the RD performance with and without ABT for single prediction modes. Results are presented for the test sequence COASTGUARD. The sequence was encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels. ABT was used only for inter-coded blocks. Intra blocks were encoded without ABT.

the 8×8 transform is used for macroblocks in 8×16, 16×8, and 16×16 mode, these modes were combined in one simulation. The RD performance when using all prediction modes is compared to the performance of the single modes. As can be seen from Fig. 3, the performance gain for ABT increases with the block size used for prediction and transformation. At low rates, the performance when using only the modes larger than 8×8 is comparable to the performance when all modes are used. The benefit of the modes with smaller block sizes becomes more pronounced at higher rates where these expensive modes are chosen more frequently.

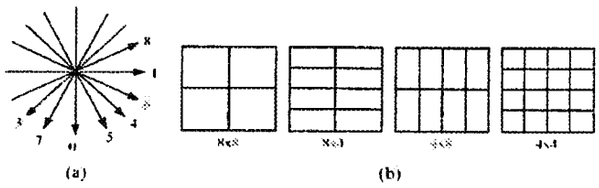


Fig. 4. (a) Intra-prediction directions for non-ABT and ABT coding. (b) Tiling of an intra macroblock for ABT intra coding including the 4×4 tiling for non-ABT coding. For each subblock a separate prediction direction can be employed.

B. Intra Coding

Two basic intra-prediction modes are used in H.264/AVC, comprising 4×4 and 16×16 block-wise prediction from previously decoded blocks. In intra 4×4 prediction mode, each 4×4 block is predicted by the adjacent pixels of the decoded neighboring 4×4 blocks. DC prediction and eight directional prediction modes can be applied to a 4×4 block. The directional prediction modes are shown in Fig. 4. Different prediction modes can be selected for each of the 16 4×4 blocks in a macroblock. The mode information has to be transmitted. To this end, a probability map is generated from the prediction modes of

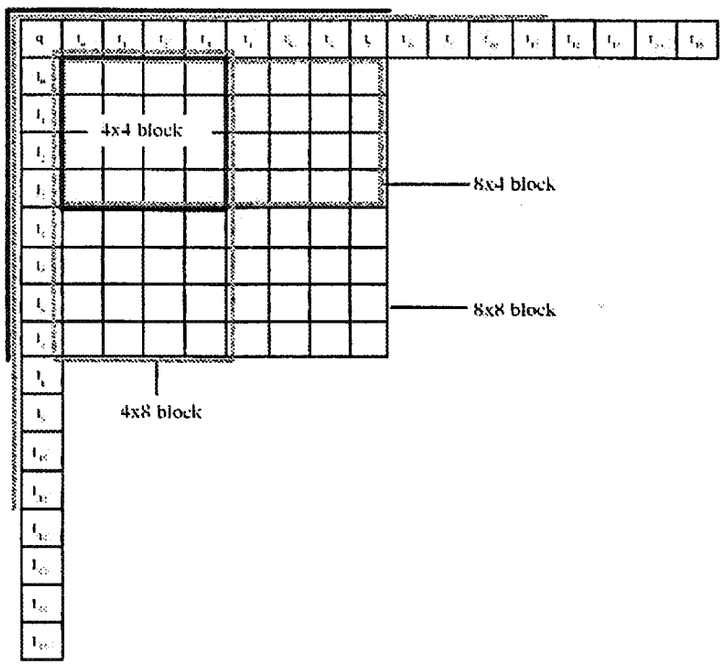


Fig. 5. Pixels I_x, I_y, and q of the top and left neighboring subblocks that may be used for intra prediction of 4×4, 4×8, 8×4, and 8×8 blocks. The maximum number of prediction pixels for an $N \times M$ block is $N + M$ pixels in each direction.

the encoded neighboring blocks. The selected prediction mode is encoded according to its probability in this map.

Smooth macroblocks containing little detail can be predicted more efficiently on a full macroblock basis. This kind of prediction is provided by the intra 16×16 Mode. Here, the current macroblock is predicted by the edge pixels of the adjacent left and top macroblocks. There are four different prediction modes, including DC, horizontal, vertical, and plane prediction.

Blocks in intra 16×16 Mode employ an additional Hadamard transform of the 4×4 DC coefficients after the 4×4 transform of the 16×16 intra prediction error. The two-stage transform decorrelates the signal in the lowest frequencies for the 16×16 block.

1) Intra Coding With Variable Block Sizes: The two-stage transform in the intra 16×16 Mode can be interpreted as a 16×16 transform for the lowest frequencies. Therefore, the choice of two intra modes specified in H.264/AVC can already be seen as a choice between two transform modes. For ABT intra coding, the prediction modes specified for the 4×4 intra-prediction mode are extended to the block sizes used for ABT inter transform coding, see Fig. 4. Hence, the gap between 4×4 intra coding and 16×16 intra coding is closed. With ABT, the encoder can trade energy compaction by larger transforms against the size of the region a single prediction direction is applied to.

An additional codeword has to be transmitted that indicates the block size used for intra prediction. The prediction block size corresponds to the block size used for transform coding. The prediction modes are encoded with probability maps as described for the intra 4×4 Mode. Since the number of subblocks is reduced, the overall overhead information for prediction block size and prediction modes for intra ABT can be smaller than with intra 4×4 Mode.

The number of neighboring pixels that are employed for intra prediction varies with the prediction mode. For example, for the horizontal and the vertical prediction mode, exclusively pixels from the left or the top are used, respectively. The reference pixels that may be used for intra-block prediction are shown in Fig. 5. Only pixels of previously decoded subblocks may be used for prediction. If prediction pixel positions required for a prediction mode are not available (e.g., because the corresponding neighboring block has not been not processed yet), the value of the last available pixel position is padded to the unavailable positions.

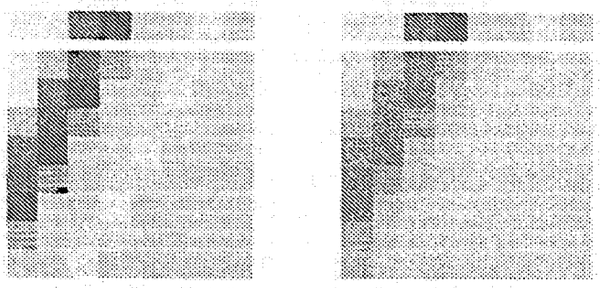
The maximum number of pixels for directional prediction is used for the *diagonal down/left* prediction (mode 3, see Fig. 4. Here, $N + M$ pixels are used in both the horizontal and vertical direction for one $N \times M$ block.

The prediction from skew directions might introduce visible artifacts into the prediction signal that are unfavorable for an efficient compression. This effect becomes prominent especially for block sizes larger than 4×4 since more pixels are predicted from the same source. Hence, the new ABT intra modes are afflicted. To reduce the artifacts, the pixels used for prediction are filtered with a low-pass filter.

Conceptually, a filter

$$h[n] = \frac{[1 \quad 2 \quad 1]}{4} \tag{7}$$

BNSDOCID: <XP.....S1099283A...L.>



(a) (b)

Fig. 6. Prediction mode 7 (*vertical/left*) applied to an 8 × 8 block. The last row of the top neighboring 8 × 8 block is used for prediction. (a) Without pre-filtering. (b) Prediction from the same pixels with pre-filtering.

is applied to the vector of reference pixels

$$p[u] = [t_{\max} \quad \dots \quad t_0 \quad q \quad t_0 \quad \dots \quad t_{\max}]^T \qquad (8)$$

used for prediction. The reference pixels $p[u]$ are indicated in Fig. 5. The application of the filter sufficiently reduces prediction artifacts for all block sizes. For block sizes larger than 4 × 4, pre-filtering is applied for horizontal and vertical prediction directions as well. In Fig. 6, the impact of the pre-filtering procedure is demonstrated for *vertical/left* prediction of an 8 × 8 block.

3) Mode Decision for ABT Intra Coding: The optimum intra ABT mode for a macroblock is selected in a two-stage process. For all possible block sizes, the best prediction mode for each subblocks is determined. Then, the cost for encoding the macroblock with the selected subblock prediction modes is evaluated for all block sizes. The corresponding mode with the optimum overall cost is chosen for encoding the macroblock.

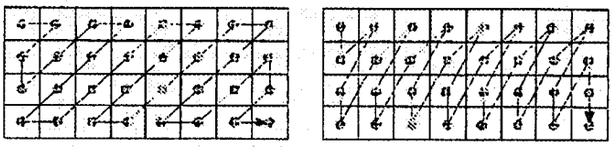
Both decisions can be made based either on the SATD or on a RD cost criterion. The RD criterion weights the rate for the transform coefficients and the corresponding prediction mode against the sum of the squared reconstructed prediction error. The weighting factor used for RD optimized macroblock mode selection in H.264/AVC can be employed here [17].

Examples for the intra mode decision on natural and synthetic test images are presented in Figs. 9 and 11.

C. Deblocking Filter

The application of a 4 × 4 transform tends to promote blocking artifacts in the reconstructed video, especially at lower bit rates. Therefore, a normative in-loop deblocking filter is employed in H.264/AVC. The reconstructed frames are filtered before application for motion compensation in the prediction loop. The functionality of this filter is essential for the subjective performance of the coder, especially at lower bit rates.

The strength of the filter is determined by the chosen macroblock mode, the existence of nonzero transform coefficients, the applied quantizer, and by the motion vector activity. In general, stronger filtering is applied to intra blocks than inter blocks. A maximum of two pixels on each side of the boundary of two adjacent blocks is affected by applying the deblocking filter on 4 × 4 blocks. Intra blocks coded with the intra 16 × 16 mode are filtered only at the macroblock boundaries. Here, a maximum of three boundary pixels may be filtered.



(a) (b)

Fig. 7. (a) Zig-zag scan for frame based coding and (b) alternate scan for field based coding of an 8 × 4 block with ABT.

The mode of operation of the deblocking filter needs to be adjusted for application to ABT-coded blocks. The strength of the filter is increased relative to the strength for 4 × 4 blocks according to the sizes of the adjacent blocks. The strength is increased most for two neighboring 8 × 8 blocks. Deblocking is only performed at transform block boundaries. Hence, the number of filter operations is reduced by ABT coding.

For example, if the 4 × 4 transform is employed, an 8 × 8 block has eight 4 × 4 block boundaries to be filtered. These are four inner boundaries, two on the left and two at the top. The boundaries to the right and at the bottom are encountered for the next blocks. Accordingly, an ABT 8 × 8 block has only two boundaries to be filtered. Similarly, for 4 × 8 and 8 × 4 blocks, the number of filtered boundaries is reduced from four to two.

D. Entropy Coding

H.264/AVC features two entropy coding methods. Variable length coding is supported by all standard compliant decoders. It employs a CAVLC for transform coefficients and a universal VLC for all other symbols. For high coding efficiency, CABAC is employed. Here, nonbinary symbols are binarized, e.g., using a unary code tree. For the separate bins, contexts are generated that are used to adapt the arithmetic coder to the statistics of the encoded symbols. For details, see [19].

Run-length coding is employed for the transform coefficients. Two types of scans can be employed to represent the transform coefficients as (level, run) symbols. The well-known zig-zag scan is employed for frame based encoding. In the case of field-based encoding of interlaced material, an alternate scan is used that accounts for the modified signal statistics in the fields of interlaced video. The alternate scan was originally proposed for ABT in [20]. An example for the zig-zag scan and the alternate scan for an 8 × 4 block is shown in Fig. 7. A VLC design for ABT transform coefficients was proposed in [21].

E. CABAC Transform Coefficient Encoding

The binarization and context generation for the transform coefficients for 4 × 8, 8 × 4,and 8 × 8 blocks is identical to the method specified for 4 × 4 blocks. Separate contexts are used for the symbols of each block size.

- For each encoded block, a flag is signaled to indicate whether the block contains coefficients or not.
- A significance map is generated, indicating the position of nonzero coefficients in the scan. For each significant coefficient a "last" flag is encoded. This flag indicates whether the last significant coefficient in the scan is reached.

BNSDOCID <XP.....11099253A...t..s

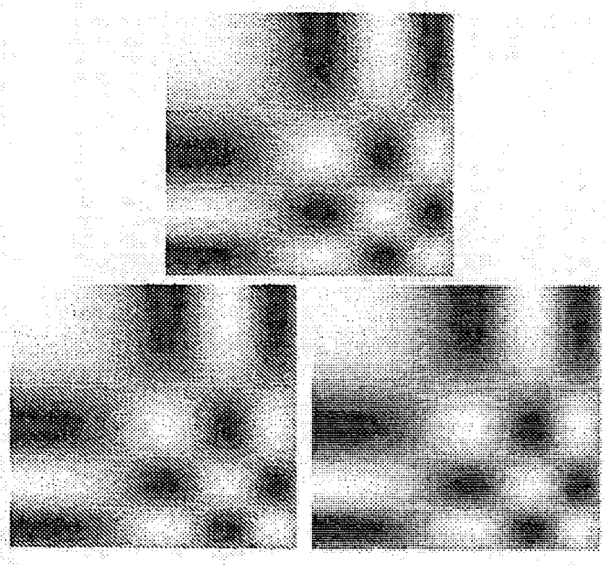


Fig. 8. Subjective comparison: Detail of a chirp image coded with QP = 36. Top: Original. Left: Encoded without ABT (187 720 bit, 30.25 dB). Right: Encoded with ABT (158 304 bit, 32.53 dB).

- The absolute values of the coefficients are binarized using an unary code tree and encoded in reverse order (last coefficient first). The sign of the coefficients is written directly to the bitstream.

IV. Simulation Results

The simulation results given here were generated with the JM 4.2 reference software [22]. ABT is implemented according to [16].

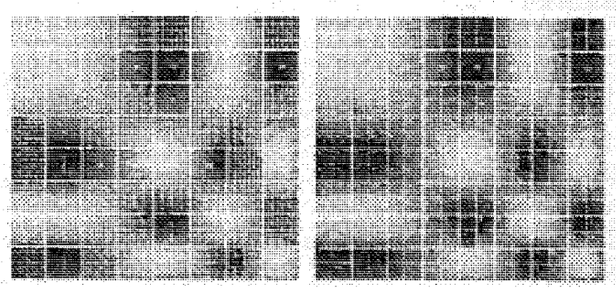
A. Subjective Performance

Two examples for the subjective performance of ABT are given.

Fig. 8 shows an example of the coding performance for a synthetic image. The original gray-level image of size 512 × 512 contains a separable chirp signal with the frequency increasing from left to right and top to bottom. The image was encoded as a single frame sequence with RD optimization at QP = 36. Fig. 8 shows the top left corner of the chirp image. As can be seen, the ABT-coded image reveals a much more pleasing subjective quality than the non-ABT image does. Also, the rate and PSNR numbers indicate superior objective performance for ABT.

In Fig. 9, the selected intra-prediction modes for ABT and non-ABT coding are shown. For non-ABT, the high number of macroblocks in intra 4 × 4 Mode with many different prediction directions is very expensive. The larger transform block sizes employed with ABT give an improved compression performance at a convenient subject quality.

In Fig. 10, a detail of first frame of the the HD sequence Shuttle (1280 × 720 @ 60 Hz) is shown. The sequence was encoded with RD optimization at QP = 36. It can be seen that more detail is preserved by ABT coding. The subjective quality is improved by ABT, even though the PSNR numbers indicate a slight loss.



(a) (b)

Fig. 9. Intra mode selection for the region shown in Fig. 8. (a) Corresponding intra mode selection without ABT. (b) Corresponding intra mode selection wit ABT. Macroblocks are marked white, subblocks of 4 × 4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16 × 16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16 × 16 plane prediction is indicated by an empty square (□).

Fig. 11 shows the intra-prediction modes that are chosen for ABT and non-ABT coding. Without ABT, high detail regions are encoded using the intra 4 × 4 Mode. In regions with less activity, the intra 16 × 16 Mode is employed. When ABT is used the number of macroblocks in intra 16 × 16 Mode is largely reduced. The ABT 8 × 8 mode is chosen most frequently. In regions with much variation, smaller block sizes are employed. Blocks of size 4 × 4 are rarely used for intra coding.

B. RD Performance

The sequences used for the simulations are given in Table I. The sequences RAVEN and KAYAK contain high-definition (HD) video. RAVEN in CIF size was extracted from the full-resolution 1280 × 720 at 60-Hz sequences by windowing the center of the full-resolution frames and temporal down-sampling by a factor of two. No spatial or temporal filtering was applied. This sequence emulates the characteristics of the full-resolution sequences while saving simulation time. All sequences but KAYAK contain progressive video. KAYAK is an interlaced sequence.

1) Simulation Conditions: In all simulations, RD optimization was switched on. Results for two configurations are presented here. In the first simulation, intra only coding is performed. The inter coding performance was evaluated using an IPPP... and an IBBPBBP... frame structure with a search range of 32 pixels and five reference frames. The sequence KAYAK was encoded in field mode using a search range of 16 pixels and two reference frames. The search range was reduced here to shorten the simulation time. ABT was used for intra and inter macroblocks in the ABT simulations.

All sequences were encoded using a fixed quantization parameter QP for all frames. The HD sequences were encoded using QP = {20, 24, 28, 32} for I and P frames. The other sequences were encoded using QP = {28, 32, 36, 40} for I and P frames. The quantization parameter QP was increased by two for B frames.

2) Results: Table II shows the rate savings and PSNR gains for the simulation conditions given above. Overall, average rate savings of about 8% and PSNR gains of 0.49 dB can be observed for ABT intra coding. The inter gains are lower than the gains achieved for intra coding. This might be due to the re-