

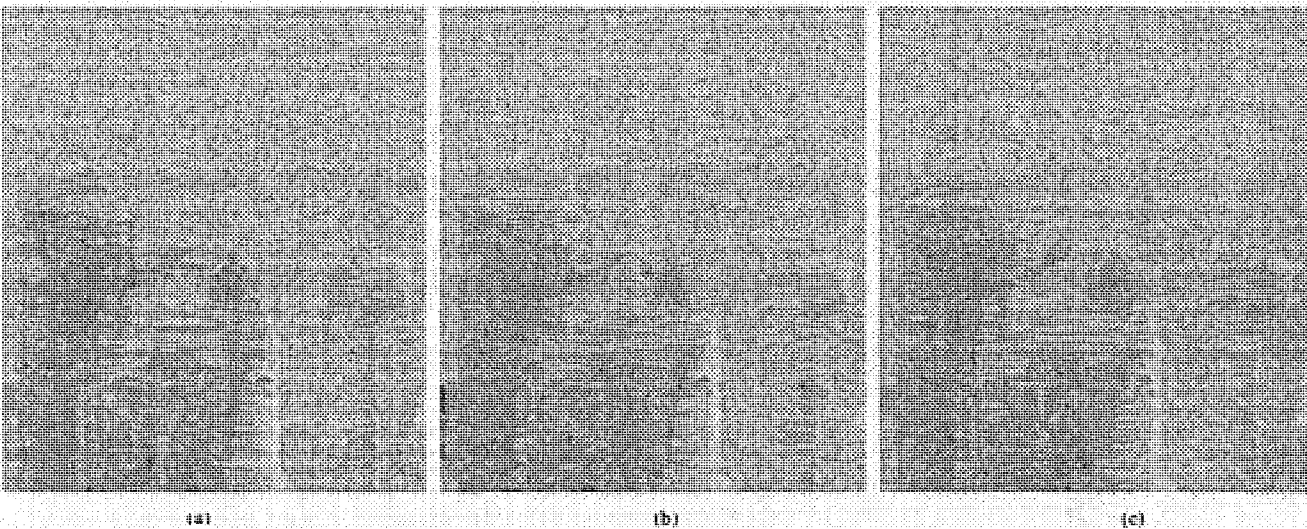


Fig. 10. Subjective comparison: Detail of the first frame of the Shuttle sequence coded with QP = 36. (a) Original. (b) Encoded without ABT (39128 bit, 38.36 dB). (c) Encoded with ABT (38784 bit, 38.25 dB). More detail is preserved with ABT coding (note the flag staff at the top left of the depicted area).

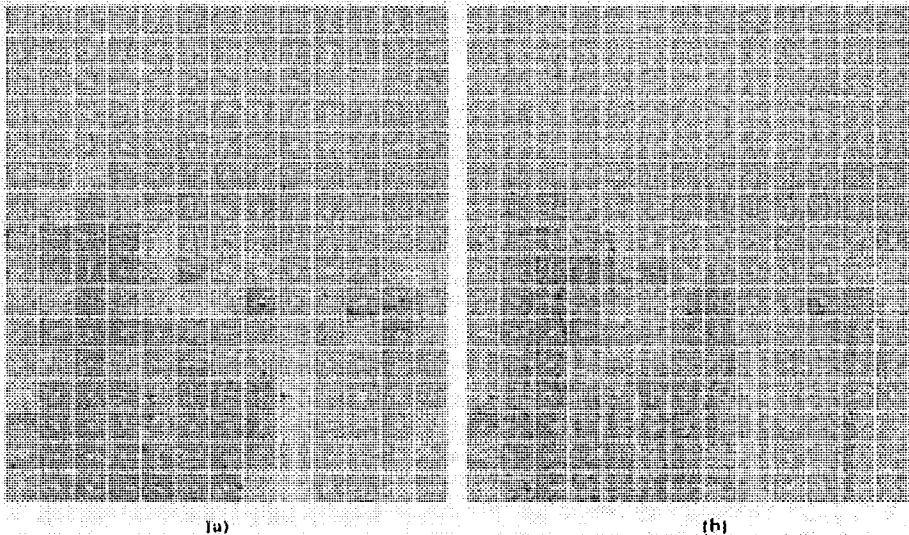


Fig. 11. Intra mode selection for the region shown in Fig. 10. (a) Encoded without ABT. (b) Encoded with ABT. Macroblocks are marked white. Subblocks of 4×4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16×16 and subblock modes. DC prediction is indicated by a dot (•). Intra 16×16 plane prediction is indicated by an empty square (□).

TABLE I
TINY 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

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

TABLE II
OVERALL RATE SAVINGS AND PSNR GAINS FOR ABT CODING

Sequence	IPI		IPPP		IBBP	
	Δ Rate	Δ PSNR	Δ Rate	Δ PSNR	Δ Rate	Δ PSNR
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	-8.92 %	0.37 dB	-5.88 %	0.25 dB
KAYAK	-12.35 %	0.90 dB	-9.41 %	0.72 dB	-12.77 %	0.92 dB
Δ mean	-8.28 %	0.49 dB	-7.19 %	0.35 dB	-5.98 %	0.32 dB

delta measurements were performed using the method described in [23]. RD plots for the IPI 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.

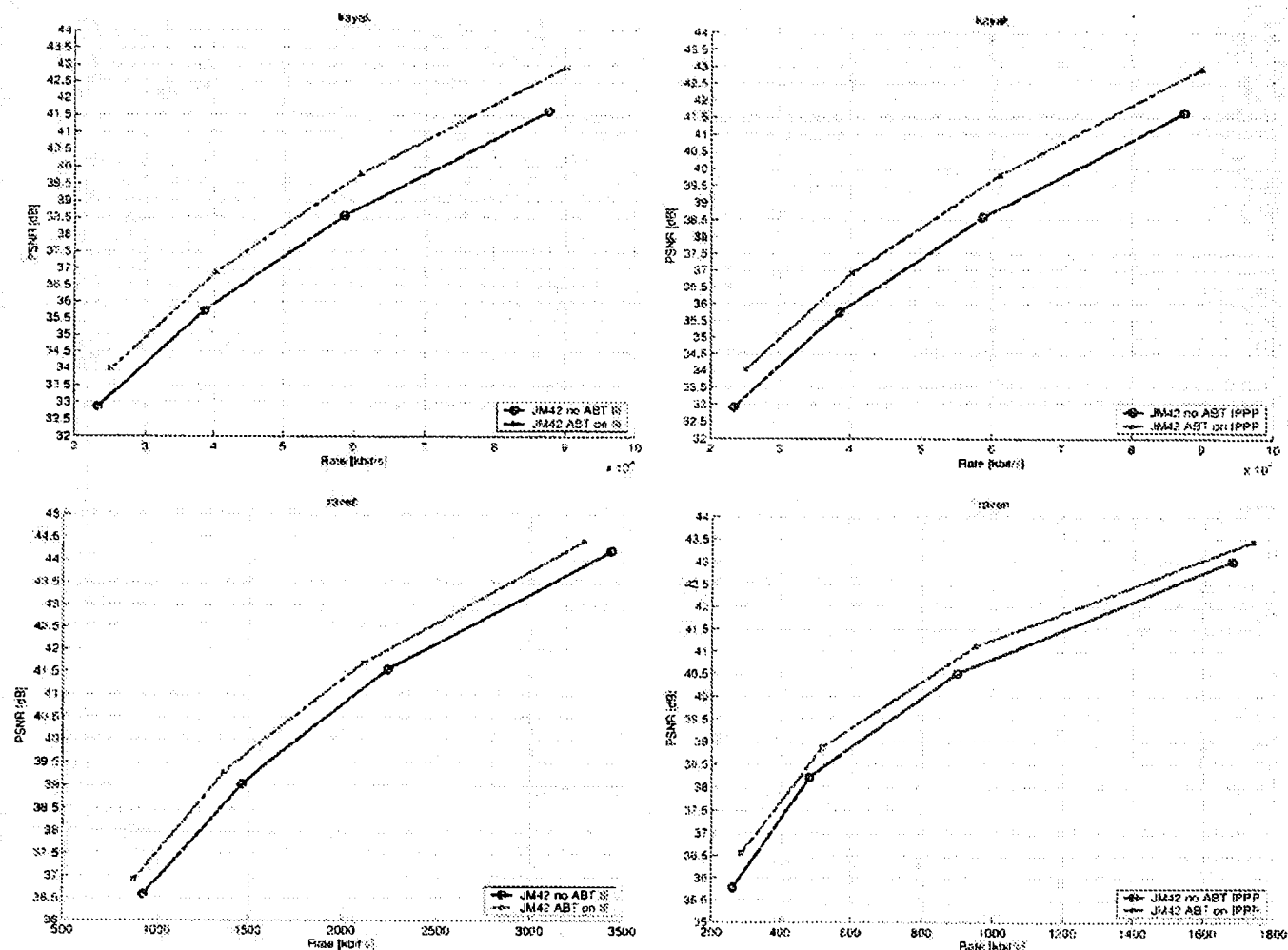


Fig. 12. Simulation results for intra only and inter coding with JM 4.2 without ABT (o) and with ABT (x) for the sequences RAVEN and KAYAK.

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 c be a coefficient vector of a transformed signal

$$c = [c_0 \ c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6 \ c_7]^T \quad (9)$$

and y be the reconstructed signal itself

$$y = T_{\pi}^T \cdot c \quad (10)$$

The transform can be calculated as follows:

$$\begin{aligned} k_0 &= 13(c_0 + c_3); & k_4 &= c_1 + c_7 \\ k_1 &= 17c_2 + 7c_6; & k_5 &= c_3 + c_5 \\ k_2 &= 13(c_0 - c_4); & k_6 &= c_1 - c_7 \\ k_3 &= 7c_2 - 17c_6; & k_7 &= c_3 - c_5 \\ z_0 &= k_0 + k_1 \\ z_1 &= k_2 + k_3 \\ z_2 &= k_2 - k_1 \\ z_3 &= k_0 - k_1 \\ z_4 &= ((k_4 + c_5) \ll 3) + 3k_4 + k_7 + (c_1 \ll 4) \\ z_5 &= ((k_6 + c_3) \ll 3) + 3k_7 - k_5 - (c_3 \ll 4) \\ z_6 &= ((k_4 + c_7) \ll 3) - 3k_5 + k_6 - (c_3 \ll 4) \\ z_7 &= -((k_7 + c_3) \ll 3) + 3k_6 + k_5 - (c_7 \ll 4) \\ y_0 &= z_0 + z_4; & y_4 &= z_3 - z_7 \\ y_1 &= z_1 + z_5; & y_5 &= z_2 - z_6 \\ y_2 &= z_2 + z_6; & y_3 &= z_1 - z_5 \\ y_3 &= z_3 + z_7; & y_7 &= z_0 - z_4 \end{aligned}$$

$a \ll n$ 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. Dahloff 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. Wiegand, "Joint final committee draft (JFCD) of joint video specification," in *Proc. JVT, 4th Meeting*, T. Wiegand, Ed., Klagenfurt, Austria, July 2002, ITU-T rec. H.264/ISO/IEC 14496-10 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. Strohach, "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. Klausitis 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. Beimann, "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. 2003, Doc. VCEG-N44.
- [14] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/ave," *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. Q15-1-39.
- [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/ave 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. Stühling, Ed., (2002) JM 4.2 Reference Software. [Online]. Available: http://ip.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.

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/N116082
February 2009, Lausanne, Switzerland

Source KDDI Corp. (KDDI R&D Laboratories Inc.)
Status Informative
Title Preliminary response for Draft Call for Evidence on High Performance Video Coding
Author Tomonobu Yoshino (to-yoshino@kddi.com)
Sei Naito (se-naitou@kddi.com)
Shigeyuki Sakazawa (sh-sakazawa@kddi.com)

1. Introduction

In the previous meeting, further study of the possibility of coding performance improvement by extending H.264 coding technology was discussed, and the text of the Draft Call for Evidence [1] was issued. As the preliminary response to that CfE, an extended H.264 coding technology applicable for ultra-high resolution video is reported with the result of coding performance evaluation.

We studied extended H.264 coding technology [2] to achieve high coding efficiency especially for ultra-high resolution video such as 4K. Our previous work [2] proposed the approach of extending MB size and showed its effectiveness. The coding gain achieved by eliminating the spatial redundancy of coded side information. Considering the characteristics of UHDTV such as higher correlation between neighboring pixels, the approach can work more effectively.

From this perspective, this document evaluates the extended MB size scheme for ultra-high resolution video coding, and we address its potential effectiveness as part of the HVC tool.

2. Enhancements on H.264

In H.264 coding framework, 16x16 is adopted as the MB size for motion compensation and intra prediction. In this document, we employ the extended MB size over 32x32, and 8x8 DCT, while our previous work adopted 4x4 DCT. Other coding processes are similar to H.264 including inter/intra mode, MB partitioning, motion compensation, and intra prediction. MB partitioning for motion compensation is depicted in Figure.1 for NxN MB size. When the MB is divided into sub-MB (N/2 x N/2), every sub-MB can also be divided into smaller blocks as shown in Fig.2.

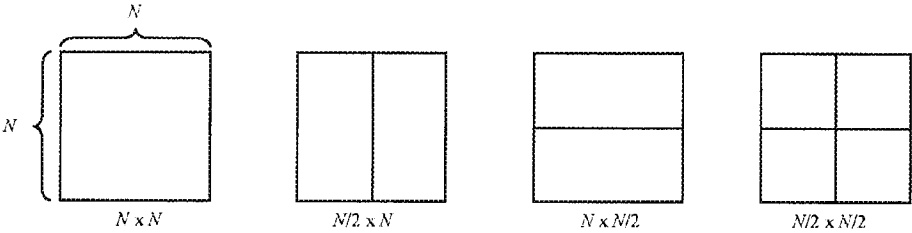


Fig. 1 Macroblock partitioning

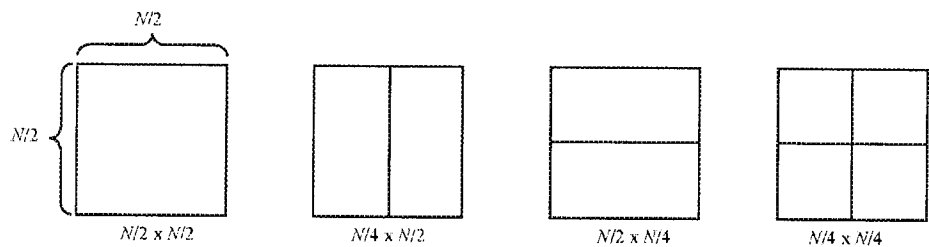


Fig. 2 Sub-MB partitioning

3. Coding Experiments

Coding experiments were conducted to evaluate the coding performance improvement achieved by the proposed enhancements. It was additionally implemented based on H.264 reference software JM13.2 [3].

In the computer simulation, we use a trimming part of ultra-high resolution (7680 pixels x 4320 lines, 60 fps, progressive, 8bit) video in order to analyze local characteristics with distinctive features. The trimming size is 1920x1080.

The coding performance of the extended MB size scheme is strongly dependent on the input image feature, and its effectiveness is expected to be significant especially for an object that belongs to a static or monotonous feature. In this experiment, we focus the opposite case with high speed motion and complicated texture in order to evaluate the guaranteed minimum gain for each material.

As the test material, six test sequences comprising 100 frames were used. The image features of those sequences are shown in Table 1.

Table 1 Feature of test materials

Test material	Feature
A	No camera work. The whole frame is occupied by the sun. Motionless.
B	No camera work. Looking down at clouds from a high mountain. The motion of clouds is slow.
C	No camera work. Japanese cherry and sky are displayed. Motionless.
D	Tilt camera work with slow motion. A traditional Japanese temple is positioned at the center.
E	Pan camera work with slow motion. Autumn leaves are displayed.
F	Static camera work. The rocket is launched.

The MB size N can be switched slice by slice and the control information of MB size is attached for every slice header. Additionally, bitstream syntax of the MB layer is assumed to be fully compliant with the specification in H.264 standard. However, the frequency of updating the MB size and the coding scheme of the control information should be determined depending on the coding characteristics by further study.

Detailed encoder configuration was selected to conform to the recommended common condition [4] in VCEG except for the number of reference and QP that were set as follows:

- The number of reference = 1
- QP = 27, 32, 37, 42

The video input signal was sampled into YUV4:2:0 format, and 4:2:0 encoding based on High Profile was applied under the IPPP (One I followed by all P pictures) coding structure.

The improvement of coding performance in the case of $N = 32$ is shown in Table 2. The result was obtained for the sequence with a comparatively high speed motion and a complicated texture. The results below indicate the BD-bitrate and BD-PSNR [5] for the 32x32 MB size against the 16x16 MB size.

Table 2. Coding results		
Test material	Δ bitrate [%]	Δ PSNR [dB]
A	31.89	0.55
B	28.23	0.44
C	8.79	0.15
D	1.79	0.17
E	1.95	0.25
F	17.08	0.19

These results confirm that the coding efficiency strongly depends on the input image feature, and the bit reduction gain up to 30% is achieved for those selected test sequences.

As the most significant case, the bit reduction ratio reaches 53.30% for a trimming part with low speed motion and monotonous texture in sequence B.

Furthermore, it is confirmed that the larger MB size can introduce significant improvement in coding efficiency especially for the sequence without camera operation. From further experiments focusing on the 64x64 MB size, the averaged bit reduction ratio of 39.32% is achieved even for complicated texture area in sequence B. The detailed analysis of the experiment reveals that the significant gain is found especially in the low bitrate case. Thus, it can be expected that the extended MB size scheme is one of the promising approaches to realize extremely efficient compression coding for ultra-high resolution video.

4. Conclusion

This document introduced the extended MB size scheme for H.264 AVC, and reported the coding efficiency for ultra-high resolution video. The experimental results confirmed that the scheme was effective for ultra-high resolution video coding and its effectiveness was strongly dependent on the input image feature. Furthermore, when adopting the 64x64 MB size, the averaged bitrate reduction ratio of more than 39.32% was achieved for the specific video sequence selected in the experiment, and the proposed scheme is expected to become one of the promising approaches to maintain sufficient coding performance by the next generation video coding scheme applicable for ultra-high definition video. In the near future, further improvement evidence achieved by several extension components including the extended MB size scheme will be reported as the final response for CIE.

The coding performance evaluation for 8K video was conducted by using coding technology for 4K video, and we confirmed the technology was also effective for 8K video. Therefore, we believe that HVC is potentially effective for ultra-high definition video that cannot be realized by existing coding standard. From this fundamental perspective, we will support the HVC standardization activity.

This research was conducted under the research project, “Ultra-high definition video compression coding technology”, organized by the National Institute of Information and Communications Technology, Japan.

[1] "Draft Call for Evidence on High-Performance Video Coding (HVC)", N10177, Oct. 2008.
 [2] S. Naito, A. Matsumura 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
 [3] Karsten Suhring, H.264/AVC JM Reference Software, <http://iphone.hhi.de/suehring/tmnl/>

- [4] TK Tan, Gary Sullivan, Thomas Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 2", ITU-T SG16/Q6, 34th VCEG Meeting, VCEG-AH10r1, Antalya, Turkey, 12-13 Jan. 2008.
- [5] Gisle Bjontegaard, "Improvements of the BD-PSNR model", ITU-T SG16/Q6, 34th VCEG Meeting, VCEG-AI11, Berlin, Germany, 16-18 July 2008.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: Not Yet Assigned

Application No.: Not Yet Assigned

Group Art Unit: Not Yet Assigned

Filing Date: Herewith

Examiner: Not Yet Assigned

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

PRELIMINARY AMENDMENT PURSUANT TO 37 CFR § 1.115

Preliminary to examination of the above-captioned patent application, please amend the application as follows:

- ☒ **Amendments to the Specification** begin on page 2 of this paper.
- ☒ **Amendments to the Claims** are reflected in the listing of the claims which begins on page 6 of this paper.
- ☐ **Amendments to the Drawings** begin on page of this paper and include an attached replacement sheet.
- ☒ **Remarks** begin on page 10 of this paper.
- ☒ The Commissioner is hereby authorized to charge any fee deficiency, charge any additional fees, or credit any overpayment of fees, associated with this application in connection with this filing, or any future filing, submitted to the U.S. Patent and Trademark Office during the pendency of this application, to Deposit Account No. 50-5519.

Amendments to the Specification:

Please amend the Cross-reference to Related Applications at paragraph beginning at page 1, line 6 of the specification as follows:

This application is a Continuation of co-pending U.S. Application Serial No. 15/806,893 filed November 8, 2017, which is a continuation of 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, each of which is are incorporated by reference herein in its entirety their entireties.

Please amend the paragraph beginning at page 1, line 19 of the specification by removing the underlining as follows:

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 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.

Please amend the paragraph beginning at page 13, line 30 of the specification as follows:

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 ~~split types~~ split 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.

Please amend the paragraph beginning at page 14, line 21 of the specification as follows:

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~~ sip_type 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.

Please amend the paragraph beginning at page 18, line 3 of the specification as follows:

In another embodiment, we introduce a ~~SIP type~~ 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.

Please amend the paragraph beginning at page 21, line 4 of the specification as follows:

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 ("Yes"), then control is passed to a function block 885. Otherwise ("No"), control is passed to a loop limit block 825.

Please amend the paragraph beginning at page 21, line 15 of the specification as follows:

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 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.

Please amend the paragraph beginning at page 21, line 23 of the specification as follows:

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 ("Yes"), then control is passed to a function block 835. Otherwise ("No"), control is passed to a loop limit block 845.

Please amend the paragraph beginning at page 21, line 27 of the specification as follows:

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 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.

Please amend the paragraph beginning at page 22, line 3 of the specification as follows:

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 ("Yes"), then control is passed to a function block 855. Otherwise ("No"), control is passed to a function block 860.

Please amend the Abstract as follows:

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.

Claims:

1 - 19. (Cancelled)

20. (New) 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 32x32, 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 sub-blocks;

partitioning the large block into sub-blocks in response to the decoded binary split signaling syntax element; and

decoding 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.

21. (New) The method of claim 20, further comprising decoding an intra prediction mode for the basic coding unit sized sub-block.

22. (New) The method of claim 20, further comprising decoding a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.

23. (New) The method of claim 20, wherein the binary split signaling syntax element specifies that the large block is further split into four equally sized sub-blocks.

24. (New) The method of claim 20, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

25. (New) 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 32x32, 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 sub-blocks;

partition the large block into sub-blocks in response to the decoded binary split signaling syntax element; and

decode 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.

26. (New) The apparatus of claim 25, wherein the one or more processors are configured to decode an intra prediction mode for the basic coding unit sized sub-block.

27. (New) The apparatus of claim 25, wherein the one or more processors are configured to decode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.

28. (New) The apparatus of claim 25, wherein the binary split signaling syntax element specifies that the large block is further split into four equally sized sub-blocks.

29. (New) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit.

30. (New) The apparatus of claim 25, 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.

31. (New) The apparatus of claim 25, 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.

32. (New) 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 32x32, 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 sub-blocks;

partitioning the large block into sub-blocks in response to the decoded binary split signaling syntax element; and

encoding 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.

33. (New) The method of claim 32, further comprising encoding an intra prediction mode for the basic coding unit sized sub-block.

34. (New) The method of claim 32, further comprising encoding a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.

35. (New) The method of claim 32, wherein a spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

36. (New) 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 32x32, 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 sub-blocks;
- partition the large block into sub-blocks in response to the binary split signaling syntax element; and
- encode 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.

37. (New) The apparatus of claim 36, wherein, the one or more processors are configured to encode an intra prediction mode for the basic coding unit sized sub-block.

38. (New) The apparatus of claim 36, wherein, the one or more processors are configured to encode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.

39. (New) The method of claim 36, wherein a spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

DOCKET NO.: ICVW_2009P00073 US02
Application No.: Not Yet Assigned

PATENT

REMARKS

This application is a continuation of US Serial No. 15/806,893.

Claims 1-19 are cancelled without prejudice or disclaimer. Claims 20-39 have been added. The new claims find support throughout the specification and figures. No new matter has been added.

Applicant has amended the application to update the cross-reference and correct typographical errors. Upon information and belief, these corrections were accepted in the parent application and thus no substitute specification is necessary. Applicant has amended the Abstract to match the parent application and appended a clean copy on a separate page to this preliminary amendment. No new matter has been added to the specification nor the abstract.

The Examiner is cordially invited to contact the undersigned at any point if it will facilitate prosecution.

Date: June 18, 2021

/Brian J. Hubbard/

Brian J. Hubbard
Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Boulevard, Suite 1700
Philadelphia, PA 19103
Telephone: (215) 558-5727
Facsimile: (215) 558-5676

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.

Electronic Patent Application Fee Transmittal				
Application Number:				
Filing Date:				
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		
Filer:		Brian Hubbard/Cassandra Katz		
Attorney Docket Number:		ICVW_2009P00073 US02		
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
UTILITY APPLICATION FILING	1011	1	320	320
UTILITY SEARCH FEE	1111	1	700	700
UTILITY EXAMINATION FEE	1311	1	800	800
Pages:				
Claims:				
INDEPENDENT CLAIMS IN EXCESS OF 3	1201	1	480	480
Miscellaneous-Filing:				
Petition:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				2300

File History Content Report

The following content is missing from the original file history record obtained from the United States Patent and Trademark Office. No additional information is available.

Document Date - 2021-06-18

Document Title - Applicant Communication Re: Entity Status Set to Undiscounted
(Initial Default Setting or Status Change)



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

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
17/351,649	06/18/2021	2482	2300	ICVW_2009P00073 US02	20	4

CONFIRMATION NO. 2654

FILING RECEIPT



15800
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

Date Mailed: 06/30/2021

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

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;

Applicant(s)

InterDigital VC Holdings, Inc., Wilmington, DE;

Power of Attorney: The patent practitioners associated with Customer Number 15800

Domestic Priority data as claimed by applicant

This application is a CON of 15/806,893 11/08/2017
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 for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 06/30/2021

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/351,649**

Projected Publication Date: To Be Determined - pending completion of Corrected Papers

Non-Publication Request: No

Early Publication Request: No

Title

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

Preliminary Class

375

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor

community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.



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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02

CONFIRMATION NO. 2654

FORMALITIES LETTER



15800
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

Date Mailed: 06/30/2021

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given **TWO MONTHS** from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment:

- A substitute specification excluding claims in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125 is required. The substitute specification must be submitted with markings and be accompanied by a clean version (without markings) as set forth in 37 CFR 1.125(c) and a statement that the substitute specification contains no new matter (see 37 CFR 1.125(b)). Since a preliminary amendment was present on the filing date of the application and such amendment is part of the original disclosure of the application, the substitute specification must include all of the desired changes made in the preliminary amendment. See 37 CFR 1.115 and 1.215.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice".
<https://portal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at
(571) 272-4000 or (571) 272-4200 or 1-888-786-0101.

/wtsige/

Electronic Acknowledgement Receipt	
EFS ID:	43577241
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:	23-AUG-2021
Filing Date:	18-JUN-2021
Time Stamp:	16:00:52
Application Type:	Utility under 35 USC 111(a)

Payment information:					
Submitted with Payment			no		
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Applicant Response to Pre-Exam Formalities Notice	ICVW_2009P00073US02_RespN otFileCorrAppPapers_Substitut eSpec.pdf	71692	no	1
			81caa01cc60fc603f21893f39c85596d4ca0c5f3		
Warnings:					

Information:					
2	Specification	ICVW_2009P00073US02_Substitute_Specification-marked.pdf	254053	no	24
			429fc8559701f8c2a53023483c94115340e630b0		
Warnings:					
Information:					
3	Specification	ICVW_2009P00073US02_Substitute_Specification-clean.pdf	237531	no	24
			f5c81323d4ddf4fb8b2435a717cfd02b10f605c3		
Warnings:					
Information:					
Total Files Size (in bytes):			563276		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

DOCKET NO.: ICVW_2009P00073 US02

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2482

Filing Date: June 18, 2021

Examiner: Not Yet Assigned

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

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

In response to the Notice to File Corrected Application Papers dated June 30, 2021, a substitute specification excluding claims in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125 is being filed herewith.

This substitute specification incorporates the preliminary amendments to the specification as filed with the application on June 18, 2021. In compliance with 37 CFR 1.125(c) the clean version of the substitute specification is accompanied by a marked version.

Although the Preliminary Amendment filed with the application on November 14, 2012 included both amendments to the specification and the claims, the substitute sheets filed here consists of only the substitute specification and excludes the claims, as requested.

In compliance with 37 CFR 1.125(b), Applicant confirms that this substitute specification contains no new matter.

Date: August 23, 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) 558-5727
Facsimile: (215) 558-5676

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

5 CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of co-pending U.S. Application Serial No. 15/806,893 filed November 8, 2017, which is a continuation of 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
10 June 29, 2010, which claims the benefit of U.S. Provisional Application Serial No. 61/222,177, filed July 1, 2009, each of which is ~~is~~ are incorporated by reference herein in ~~its entirety~~ their entireties.

TECHNICAL FIELD

15 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

20 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~~ Moving Picture Experts
Group-4 (MPEG-4) Part 10 Advanced Video Coding (AVC) standard/International
25 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
30 INTRA8x8 support 9 intra prediction modes and INTRA16x16 supports 4 intra prediction modes.

SUBSTITUTE SPECIFICATION - MARKED

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.

5 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
10 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
15 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
20 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
25 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.

30

(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.

- 5
- (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.

10

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.

5 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

10 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
15 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
20 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
25 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.
30 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;

5 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;

10 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;

15 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

20 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.

25 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.

30 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.

SUBSTITUTE SPECIFICATION - MARKED

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
5 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
10 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
15 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
20 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.
25 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.

30 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
5 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,
10 the appearances of the phrase “in one embodiment” or “in an embodiment”, as well as 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
15 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
20 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.

5 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.

10 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

SUBSTITUTE SPECIFICATION - MARKED

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.

5 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
10 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
15 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
20 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
25 570.

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
30 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.

5 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.

10 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.

15 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

30 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

5 621. 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~~ sip_types 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

10 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

15 (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,

20 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~~

25 sip_type 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

30 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.

5

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.

10

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
5 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.

10

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
15 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.

5 **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.

10 **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.

15 **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.

20

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

Another Embodiment

25 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.

30 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
5 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 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.
10 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
15 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
20 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:

5

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.

10

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.

15

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.

20

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.

25

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.

30

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.

5 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.

10 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.

15 The decision block 750 determines whether or not $RD_{32} < TotRD_{16}$ (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.

20 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.

25 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.

30 The decision block 770 determines whether or not $RD_{64} < TotRD_{32}$ (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.

5 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 ("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 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 ("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 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 ("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 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.

5 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
10 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
15 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
20 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
25 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
30 ("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.

5 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
10 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
15 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.

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

5 CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of co-pending U.S. Application Serial No. 15/806,893 filed November 8, 2017, which is a continuation of 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
10 June 29, 2010, which claims the benefit of U.S. Provisional Application Serial No. 61/222,177, filed July 1, 2009, each of which are incorporated by reference herein in their entireties.

TECHNICAL FIELD

15 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

20 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,
25 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
30 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

SUBSTITUTE SPECIFICATION - CLEAN

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.

5 (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
5 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
10 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
15 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
20 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.
25 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
30 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
5 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
10 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
15 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
20 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,
25 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:
30 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;

SUBSTITUTE SPECIFICATION - CLEAN

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.

5 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
10 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,
15 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
20 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,
25 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
30 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
5 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
10 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
15 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
20 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
25 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.

30 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.

5 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.

10 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.

15

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.

20

25

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

30

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.

5 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.

10 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.

15 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

20 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.

20

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 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.

30

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

5 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
5 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.

10

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
15 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.

20

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.

5 **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.

10

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

15 **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.

20 **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.

25 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

30

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
5 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
10 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
15 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
20 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
25 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
30 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.

5 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)
10 over the 16x16 blocks, and passes control to the decision block 750.

The decision block 750 determines whether or not $RD_{32} < TotRD_{16}$ (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
15 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
20 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 $RD_{64} < TotRD_{32}$ (i.e.,
25 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
30 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
5 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
10 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 ("Yes"), then control is passed to a function block 885. Otherwise ("No"), control is passed to a loop limit block 825.

15 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 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
20 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
25 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
30 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 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.

5 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 ("Yes"), then control is passed to a function block 855. Otherwise ("No"), control is passed to a function block 860.

10 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.

15 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.

20 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.

25 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
5 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
10 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
15 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
20 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
25 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.
30 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
5 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.

10 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
15 and modifications are intended to be included within the scope of the present principles as set forth in the appended claims.



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

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
17/351,649	06/18/2021	2482	2300	ICVW_2009P00073 US02	20	4

15800
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

CONFIRMATION NO. 2654
UPDATED FILING RECEIPT



Date Mailed: 09/02/2021

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

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;

Applicant(s)

InterDigital VC Holdings, Inc., Wilmington, DE;

Power of Attorney: The patent practitioners associated with Customer Number 15800

Domestic Priority data as claimed by applicant

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 for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 06/30/2021

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/351,649**

Projected Publication Date: 12/09/2021

Non-Publication Request: No

Early Publication Request: No

Title

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

Preliminary Class

375

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor

community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.

Electronic Acknowledgement Receipt	
EFS ID:	44127787
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:	26-OCT-2021
Filing Date:	18-JUN-2021
Time Stamp:	16:10:44
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	ICVW_2009P00073_US02_IDS_Transmittal.pdf	101743	no	1
			6aba0e3d4e4843b557c60b085338d89b1dba673b		

Warnings:

Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	ICVW_2009P00073US02_IDS-1449.pdf	152639	no	1
			402657d2934ac3ed8bf1c013d1497222d63b5294		
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	74-EP_2293567_A2.pdf	16063933	no	88
			911ad12be403eb31fd96afdc29f4a559e469b2a1		
Warnings:					
Information:					
Total Files Size (in bytes):			16318315		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

DOCKET NO.: ICVW_2009P00073 US02

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: InterDigital VC Holdings, Inc. **Confirmation No.:** 2654
Application No.: 17/351,649 **Group Art Unit:** Not Yet Assigned
Filing Date: June 18, 2021 **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

☒ A copy of reference number 74 listed on the attached Form PTO-1449 are not required to be submitted pursuant to 37 CFR § 1.98(a)(2)(ii).

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: Oct. 25, 2021

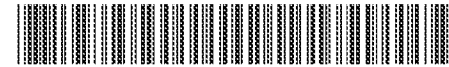
/Brian J. Hubbard/
Brian J. Hubbard
Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
Telephone: (215) 558-5740
Facsimile: (215) 558-5676

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				Application Number		17/351,649	
				Filing Date		Jun 18, 2021	
				First Named Inventor		Yunfei Zheng	
				Art Unit		Not Yet Assigned	
				Examiner Name		Not Yet Assigned	
Sheet	1	of	1	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
	74	2293567	EP	A2	2011-03-09	SAMSUNG ELECTRONICS CO., LTD.		

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.			



(11) EP 2 293 567 A2

(12) EUROPEAN PATENT APPLICATION
published in accordance with Art. 153(4) EPC

- (43) Date of publication:
09.03.2011 Bulletin 2011/10

(21) Application number: 09762619.6

(22) Date of filing: 03.06.2009
- (51) Int Cl.:
H04N 7/24 (2011.01)

(86) International application number:
PCT/KR2009/002942

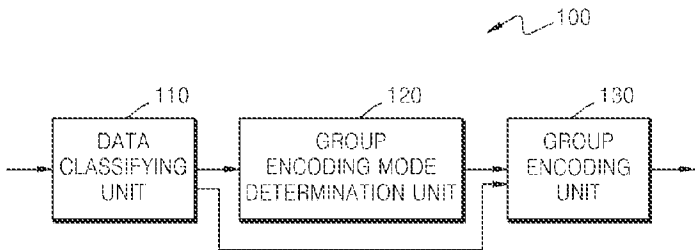
(87) International publication number:
WO 2009/151232 (17.12.2009 Gazette 2009/51)

<p>(84) Designated Contracting States: AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK TR Designated Extension States: AL BA RS</p> <p>(30) Priority: 13.06.2008 US 61175 P 21.07.2008 US 82338 P 20.10.2008 KR 20080102464 11.05.2009 KR 20090040890</p> <p>(71) Applicant: Samsung Electronics Co., Ltd. Suwon-si, Gyeonggi-do 442-742 (KR)</p> <p>(72) Inventors: • CHEN, Jianle Suwon-si Gyeonggi-do 443-725 (KR)</p>	<ul style="list-style-type: none">• LEE, Jae-Chool Suwon-si Gyeonggi-do 443-744 (KR)• LEE, Sang-Rae Suwon-si Gyeonggi-do 443-745 (KR)• CHEON, Min-Su Suwon-si Gyeonggi-do 441-400 (KR)• LEE, Kyo-Hyuk Yongin-si Gyeonggi-do 448-536 (KR) <p>(74) Representative: Fearnside, Andrew Simon Appleyard Lees 15 Clare Road Halifax, HX1 2HY (GB)</p>
---	--

(54) IMAGE-ENCODING METHOD AND A DEVICE THEREFOR, AND IMAGE-DECODING METHOD AND A DEVICE THEREFOR

(57) An image encoding method is provided, in which image data divided into basic blocks is classified in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; an encoding mode for a predetermined group is determined in order to encode the predetermined group, wherein the encoding mode represents a mode for encoding data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and the data of the predetermined group is encoded according to the determined encoding mode. Detailed operations in the image encoding method are performed in consideration of the encoding mode of the group.

FIG. 1



Printed by Jouve, 75001 PARIS (FR)

EP 2 293 567 A2

DescriptionTechnical Field

5 **[0001]** The present invention relates to image encoding and image decoding.

Background Art

10 **[0002]** In conventional block-based video encoding and decoding methods such as Moving Pictures Experts Group (MPEG)2, MPEG4, and H.264/ Advanced Video Coding (AVC), the size of a macroblock is fixed to 16x16. In particular, in H.264/AVC, image data may be divided into blocks less than or equal to 8x8 blocks, namely, into 4x4 blocks, in order to achieve more accurate estimation or prediction.

Brief Description of the Drawings

15 **[0003]** The above and other features and advantages of the present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

FIG. 1 is a block diagram of an image encoding apparatus according to an embodiment of the present invention;
 FIG. 2 is a block diagram of an image decoding apparatus according to an embodiment of the present invention;
 FIG. 3 illustrates a concept of a group which is a data processing unit proposed by the present invention;
 FIG. 4 illustrates an example of a subgroup which is a data processing unit proposed by the present invention;
 FIG. 5 illustrates another example of a subgroup which is a data processing unit proposed by the present invention;
 FIG. 6 is a flowchart for explaining a syntax of an encoding mode according to an embodiment of the present invention;
 FIG. 7 illustrates an example of an encoding method performed in units of the data processing unit;
 FIG. 8 is a flowchart of an encoding method of determining a group encoding mode, according to an embodiment of the present invention;
 FIG. 9 is a flowchart of a method of reading and decoding an encoding mode, according to an embodiment of the present invention;
 FIG. 10 illustrates a hierarchical structure of a data processing unit which is available in the present invention;
 FIG. 11 is a flowchart for explaining a syntax of an encoding mode in which encoding is performed in units of super macroblocks and macroblocks, according to another embodiment of the present invention;
 FIG. 12 illustrates a motion vector estimating method which is performed when an encoding mode is a skip mode, according to another embodiment of the present invention;
 FIG. 13 illustrates a sequence of encoding and decoding each data processing unit, according to another embodiment of the present invention;
 FIG. 14 illustrates a method of arranging frequency coefficients, according to another embodiment of the present invention;
 FIG. 15 illustrates an existing encoding method (on the left of FIG. 15) and a syntax encoding method (on the right of FIG. 15) according to another embodiment of the present invention, wherein the syntax encoding method is associated with a coded block pattern (CBP) and the size of a coefficient block;
 FIG. 16 is a flowchart for explaining a CBP-related syntax according to another embodiment of the present invention;
 FIG. 17 illustrates an encoding mode for a B frame, according to another embodiment of the present invention;
 FIG. 18 is a flowchart of an image encoding method according to an embodiment of the present invention;
 FIG. 19 is a flowchart of an image decoding method according to an embodiment of the present invention;
 FIGS. 20A, 20B, 20C, 20D, 20E, and 20F are block diagrams of modified embodiments of the image encoding apparatus of FIG. 1 that consider a data processing unit dynamically determined according to an embodiment;
 FIG. 20G illustrates an outline of image encoding methods that consider a data processing unit dynamically determined for a large data processing unit according to an embodiment, according to an embodiment of the present invention;
 FIG. 21 illustrates a scanning method performed for each type of data processing unit in order to perform motion estimation designed for a large data processing unit, according to an embodiment of the present invention;
 FIG. 22 illustrates a scanning method performed for each type of data processing unit in order to perform motion estimation designed for a large data processing unit, according to an embodiment of the present invention;
 FIG. 23A illustrates simplified hierarchical data processing units from among a large data processing unit, according to an embodiment of the present invention;
 FIG. 23B illustrates a syntax designed for a large data processing unit, according to an embodiment of the present invention;

FIGS. 24A, 24B, and 24C illustrate several methods of constructing a skip group, according to an embodiment of the present invention;

FIGS. 25A and 25B are block diagrams for illustrating a method of inducing a motion vector of a large data processing unit, and a method of inducing a motion vector of a lower data processing unit, according to an embodiment of the present invention;

FIGS. 26A and 26B are block diagrams for illustrating several motion estimation methods for a chroma component of a large data processing unit, according to an embodiment of the present invention;

FIG. 27 is a table for illustrating a method of inducing the size of a data processing unit for a chroma component of a large data processing unit, according to an embodiment of the present invention;

FIG. 28 is a flowchart illustrating a method of using a CBP that considers a large data processing unit, according to an embodiment of the present invention;

FIGS. 29A and 29B illustrate several methods of scanning a CBP that considers a large data processing unit, according to an embodiment of the present invention;

FIGS. 30A, 30B, 30C, 30D, 30E, and 30F are block diagrams of modified embodiments of the image decoding apparatus of FIG. 2 that consider a data processing unit dynamically determined according to an embodiment;

FIGS. 31A, 31B, 31C, 31D, 31E, and 31F are flowcharts of modified embodiments of the image encoding method of FIG. 18 that consider a data processing unit dynamically determined according to an embodiment; and

FIGS. 32A, 32B, 32C, 32D, 32E, and 32F are flowcharts of modified embodiments of the image decoding method of FIG. 19 that consider a data processing unit dynamically determined according to an embodiment.

Disclosure of the Invention

Technical Problem

[0004] The present invention provides encoding and decoding of image data.

Technical Solution

[0005] According to an aspect of the present invention, there is provided an image encoding method comprising classifying image data divided into basic blocks, in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; determining an encoding mode for a predetermined group in order to encode the predetermined group, wherein the encoding mode represents a mode for encoding data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and encoding the data of the predetermined group according to the determined encoding mode.

Best mode

[0006] According to an aspect of the present invention, there is provided an image encoding method comprising classifying image data divided into basic blocks, in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; determining an encoding mode for a predetermined group in order to encode the predetermined group, wherein the encoding mode represents a mode for encoding data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and encoding the data of the predetermined group according to the determined encoding mode.

[0007] The image encoding method may further comprise encoding a flag that represents the encoding mode for the predetermined group.

[0008] The classifying of the image data may comprise determining a size of the predetermined group and a shape of the predetermined group, wherein the size represents the number of basic blocks included in the predetermined group and the shape is formed by the basic blocks included in the predetermined group.

[0009] The classifying of the image data may comprise determining subgroups for the predetermined group based on a level representing the number of subgroups included in the predetermined group and a scan sequence in which the basic blocks are scanned.

[0010] The encoding mode may comprise at least one of information about the type of data processing units in which the data of the predetermined group is encoded, wherein the type is one of a group, a subgroup, and a basic block; information about the size of the data processing unit, wherein the size represents the number of basic blocks included in the data processing unit; and information about an prediction mode that represents a mode for determining reference information corresponding to the data processing unit in order to perform motion estimation on the data processing unit.

[0011] The determining of the encoding mode for the predetermined group may comprise encoding the predetermined group according to at least one prediction mode in units of different types of data processing units which are groups, subgroups, and basic blocks; determining error rates that are generated by encoding performed on the predetermined group in units of combinations of the different types of data processing units and the at least one prediction mode; and
 5 determining a prediction mode for a predetermined data processing unit generating a smallest error rate from among the determined error rates.

[0012] The determining of the encoding mode for the predetermined group may comprise comparing encoding error rates generated by considering the at least one prediction mode in units of the different types of data processing units with one another and determining, as the encoding mode for the predetermined group, a combination of a data processing
 10 unit and an prediction mode, which generates the smallest error rate.

[0013] The image encoding method may further comprise rearranging frequency-transformed coefficients of each group or each subgroup, in units of a plurality of basic blocks.

[0014] The image encoding method may further comprise encoding a flag that represents whether the frequency-transformed coefficients have been rearranged in units of basic blocks.

15 [0015] The image encoding method may further comprise encoding a flag that represents a data processing unit of units in which the data of the predetermined group is frequency-transformed.

[0016] In the classifying of the image data, the sizes, shapes, and maximum levels of groups may be determined in units of a data unit selected from pictures, sequences, and groups of the image data.

[0017] According to an aspect of the present invention, there is provided an image decoding method comprising
 20 receiving and parsing image data encoded in units of at least one data processing unit selected from a basic block, a group, and a subgroup, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; reading an encoding mode for a predetermined group from the parsed image data, wherein the encoding mode represents a mode for encoding image data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and decoding the image
 25 data of the predetermined group according to the read encoding mode.

[0018] The reading of the encoding mode for the predetermined group may comprise reading information about a shape of a group set as a data processing unit in which decoding is performed, information about a size of a group representing the number of basic blocks included in the group, and information about a maximum level associated with a maximum number of subgroups included in the group; and determining the group which is the data processing unit
 30 by which decoding is performed, by classifying the parsed image data by using the information about the shape and size of the group.

[0019] In the image decoding method, the groups may be set in units of a data unit selected from a picture, a sequence, and a group of the image data.

[0020] In the image decoding method, the subgroups may be set based on a level representing the number of subgroups
 35 included in the group, and a scan sequence in which basic blocks included in the group are scanned.

[0021] The reading of the encoding mode for the predetermined group may comprise extracting an encoding mode flag that represents the encoding mode for the predetermined group, from parsed data associated with the predetermined group; and reading the encoding mode for the predetermined group on the basis of the encoding mode flag.

[0022] The encoding mode for the predetermined group may comprise at least one of information about the type of
 40 data processing units in which the data of the predetermined group is encoded, wherein the type is one of a group, a subgroup, and a basic block; information about the size of the data processing unit, wherein the size represents the number of basic blocks included in the data processing unit; and information about a prediction mode that represents a mode for determining reference information corresponding to the data processing unit in order to perform motion estimation on the data processing unit.

45 [0023] The decoding of the data of the predetermined group may comprise determining a data processing unit in units in which decoding is to be performed, based on the information about the type and size of the data processing unit; determining a prediction mode of the data processing unit on the basis of the information about the prediction mode; and decoding the data of the group according to the determined data processing unit and the determined prediction mode.

[0024] In the image decoding method, the predetermined group may be encoded according to at least one prediction
 50 mode in units of different types of data processing units, namely, groups, subgroups, and basic blocks so as to calculate error rates for the at least one prediction mode, a combination of a predetermined data processing unit and a prediction mode, which corresponds to a smallest error rate from among the calculated error rates, may be determined; and encoding error rates generated by considering the at least one prediction mode in units of the different types of data processing units, may be compared with one another so as to determine, as the encoding mode for the predetermined
 55 group, a combination of a data processing unit and a prediction mode, which correspond to the smallest error rate.

[0025] In the image decoding method, the basic blocks may be decoded in a zigzag scan sequence.

[0026] The image decoding method may further comprise rearranging frequency transformed coefficients of a plurality of basic blocks, in units of one of groups and subgroups.

[0027] The image decoding method may further comprise reading, from the parsed data, a flag that represents a data processing unit of units in which the data included in the predetermined group is frequency transformed.

[0028] According to an aspect of the present invention, there is provided an image encoding apparatus comprising a data classifying unit classifying image data divided into basic blocks, in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; a group encoding mode determination unit determining at least one data processing unit selected from a group, a subgroup, and a basic block and determining an encoding mode for a predetermined group in order to encode the predetermined group, wherein the encoding mode represents a mode for encoding data included in the predetermined group in units of the at least one data processing unit; and a group encoding unit encoding the data of the predetermined group according to the determined encoding mode.

[0029] According to an aspect of the present invention, there is provided an image decoding apparatus comprising a receiving unit receiving and parsing image data encoded in units of at least one data processing unit selected from a basic block, a group, and a subgroup, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; a group encoding mode reading unit reading an encoding mode for a predetermined group from the parsed image data, wherein the encoding mode represents a mode for encoding image data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and a group decoding unit decoding the image data of the predetermined group according to the read encoding mode.

[0030] When an encoding mode representing the type of a data processing unit for an image and a motion prediction method is determined, the image is encoded considering the data processing unit and a motion prediction mode that correspond to the encoding mode.

[0031] The image encoding method based on a data processing unit dynamically determined according to an embodiment of the present invention further comprises performing frequency transformation and quantization for encoding of the input image on the basis of a large data processing unit corresponding to the determined group encoding mode; performing motion estimation for the encoding of the input image, based on the large data processing unit; performing in-loop filtering for the encoding of the input image, based on the large data processing unit; and performing entropy encoding for the encoding of the input image, based on the large data processing unit.

[0032] The image decoding method based on the data processing unit dynamically determined according to an embodiment of the present invention further comprises performing entropy decoding for decoding of data of the group on the basis of a large data processing unit corresponding to the encoding mode for the determined group; performing inverse quantization and inverse frequency transformation for the decoding of the data of the group, based on the large data processing unit; performing motion compensation for the decoding of the data of the group, based on the large data processing unit; and performing in-loop filtering for the decoding of the data of the group, based on the large data processing unit.

[0033] The image encoding apparatus based on the data processing unit dynamically determined according to an embodiment of the present invention may further comprise a frequency transformation and quantization unit for performing frequency transformation and quantization for encoding of the input image on the basis of a large data processing unit corresponding to the determined group encoding mode; a motion estimation unit for performing motion estimation for the encoding of the input image, based on the large data processing unit; an in-loop filtering unit for performing in-loop filtering for the encoding of the input image, based on the large data processing unit; and an entropy encoding unit for performing entropy encoding for the encoding of the input image, based on the large data processing unit.

[0034] The image decoding apparatus based on the data processing unit dynamically determined according to an embodiment of the present invention may further comprise an entropy decoding unit for performing entropy decoding for decoding of data of the group on the basis of a large data processing unit corresponding to the determined group encoding mode; an inverse quantization and inverse frequency transformation unit for performing inverse quantization and inverse frequency transformation for the decoding of the data of the group, based on the large data processing unit; a motion compensation unit for performing motion compensation for the decoding of the data of the group, based on the large data processing unit; and an in-loop filtering unit for performing in-loop filtering for the decoding of the data of the group, based on the large data processing unit.

[0035] In the image encoding method or the image decoding method, a size of a data processing unit for the frequency transformation may be determined based on a size of a data processing unit for the motion estimation.

[0036] In connection with the performing of the motion estimation in the image encoding method or the image decoding method, a scanning sequence for each data processing unit for sequential motion estimation may be determined based on the large data processing unit, and neighbor availability corresponding to whether a neighboring data processing unit located on a right upper end of a current data processing unit can be used as a reference block may be determined based on the large data processing unit.

[0037] The motion estimation based on the large data processing unit may use a square-shaped data processing unit from among the data processing units. The size of the data processing unit for the frequency transformation based on

the square-shaped data processing unit may be restricted to a predetermined size smaller than or equal to the square-shaped data processing unit.

[0038] The motion estimation based on the large data processing unit may comprise intra prediction based on the large data processing unit, and comprise intra prediction performed in units of the large data processing units in an intra mode of a complex plain type. A data processing unit whose motion can be estimated in an intra mode may be previously determined based on the large data processing unit.

[0039] The motion estimation based on the large data processing unit may comprise motion estimation performed in a prediction mode in which an intra mode and an inter mode are mixed, based on the large data processing unit. A motion vector for motion estimation of an inter mode may be determined based on the large data processing unit. A motion vector predictor for an inter mode corresponding to the motion vector determined based on the large data processing unit may be induced. In connection with the motion estimation based on the large data processing unit, a skip mode of an extended shape may be determined based on the large data processing unit. A skip group comprising adjacent data processing units of at least one skip mode may be determined, and motion estimation may be performed in units of skip groups in a skip mode.

[0040] The frequency transformation based on the large data processing unit may comprise integer discrete cosine transformation based on a large block, based on the large data processing unit. Frequency transformation may be performed with respect to a chroma component of a current data processing unit, to a predetermined size determined based on the large data processing unit. Modified frequency transformation may be performed to reduce calculation complexity of frequency transformation considering the large data processing unit.

[0041] The entropy encoding based on the large data processing unit may use a coded block pattern hierarchically determined according to the size of each data processing unit based on the large data processing unit.

[0042] In the entropy encoding based on the large data processing unit, the coded block pattern may be scanned for each of regions into which a current data processing unit is classified based on the possibility that a quantized coefficient which is not 0 from among the quantized coefficients of the current data processing unit exists. A coefficient strip of a region including a coefficient which is not 0 may be scanned based on the coded block pattern for each region of the current data processing unit. A flag representing the coefficient strip of a region including a coefficient which is not 0 may be encoded to be included in a bitstream, and the flag may be read during decoding of the bitstream.

[0043] The entropy encoding or the entropy decoding may use Context-based Adaptive Binary Arithmetic Coding (CABAC) or Context-based Adaptive Variable Length Coding (CAVLC) modified based on the large data processing unit.

[0044] The entropy encoding or the entropy decoding based on the large data processing unit may use a coded block pattern determined as a single context with respect to an intra mode or a chroma component based on the large data processing unit.

[0045] In the in-loop filtering based on the large data processing unit, it may be determined based on the large data processing unit whether deblocking filtering is performed on a lower data processing unit in a current data processing unit, based on the large data processing unit. The deblocking filtering with respect to the current data processing unit may not comprise deblocking filtering with respect to a left or upper boundary of a lower data processing unit in the current data processing unit, and deblocking filtering with respect to a boundary of a lower data processing unit in the current data processing unit on which frequency-transformation has been performed in units of a predetermined data processing unit.

[0046] In the in-loop filtering based on the large data processing unit, deblocking filtering with respect to a boundary of a chroma component of the current data processing unit may not comprise deblocking filtering with respect to a boundary of a chroma component of the lower data processing unit in the current data processing unit. In the in-loop filtering based on the large data processing unit, deblocking filtering with respect to a boundary of the current data processing unit on which motion estimation has been performed in a mixed mode in which an inter mode and an intra mode are mixed may not comprise deblocking filtering with respect to a boundary of the lower data processing unit in the current data processing unit.

[0047] In connection with the in-loop filtering based on the large data processing unit, the degree of deblocking filtering with respect to each boundary may be set for a boundary of at least one lower data processing unit in the large data processing unit.

[0048] The in-loop filtering based on the large data processing unit may comprise additional filtering for reducing a ringing effect is performed on a current data processing unit having a large size, based on the large data processing unit.

[0049] According to an aspect of the present invention, there is provided a computer readable recording medium having recorded thereon a program for executing the image encoding method based on the data processing unit dynamically determined according to an embodiment of the present invention.

[0050] According to an aspect of the present invention, there is provided a computer readable recording medium having recorded thereon a program for executing the image decoding method based on the data processing unit dynamically determined according to an embodiment of the present invention.

Mode for Invention

[0051] Hereinafter, the table of embodiments of the present invention will be introduced to facilitate easy understanding of the specification.

[0052] In the exemplary embodiments, "unit" may or may not refer to a unit of size, depending on its context.

[0053] First, a data processing unit and an encoding mode which are dynamically determined according to an embodiment of the present invention will be described in section [I . dynamic determination of data processing unit].

[0054] An apparatus for encoding image based on the data processing unit and the encoding mode dynamically determined above will be described in section [II . Image encoding apparatus based on dynamically determined data processing unit].

[0055] Detailed operations of the image encoding apparatus based on the data processing unit dynamically determined according to an embodiment will now be described in sections [III . Motion prediction process], [III-1. Intra prediction], [III -2. Inter prediction], [IV. Frequency transformation and quantization], and [V .entropy encoding].

[0056] An apparatus for decoding image based on the data processing unit dynamically determined according to an embodiment, which corresponds to the image encoding apparatus based on the data processing unit dynamically determined according to an embodiment, will be described in section [VII. Image decoding apparatus based on dynamically determined data processing unit].

[0057] An image encoding method based on the data processing unit dynamically determined according to an embodiment, and an image decoding method corresponding to the image encoding method will be described in section [VIII Image encoding method and image decoding method based on dynamically determined data processing unit].

[I dynamic determination of data processing unit]

[0058] FIG. 1 is a block diagram of an image encoding apparatus 100 according to an embodiment of the present invention. Referring to FIG. 1, the image encoding apparatus 100 includes a data classifying unit 110, a group encoding mode determination unit 120, and a group encoding unit 130.

[0059] When received image data has been divided into basic blocks, the data classifying unit 110 classifies pieces of the received image data in units of basic blocks, in units of groups each including at least one basic block, or in units of subgroups each including at least one basic block and being included in a group. The data classifying unit 110 outputs the classified image data to the group encoding mode determination unit 120 and the group encoding unit 130.

[0060] The basic block, the group, and the subgroup are types of data processing units, that is, units by which image data is encoded.

[0061] A group may have a property "size" representing the number of basic blocks and a property "shape" representing a shape formed by the basic blocks included in the group. A subgroup is determined according to a level representing the number of subgroups included in a group and a scan sequence in which the basic blocks included in the group are scanned. At least one property of each group from among a size, a shape, a maximum level, and a scan sequence may be determined individually for each group, each picture or each sequence.

[0062] A method of determining such a group will be described later with reference to FIG. 3, and a method of determining such a subgroup will be described later with reference to FIGS. 4 and 5.

[0063] The group encoding mode determination unit 120 determines a group encoding mode for each group, which includes information about a method of encoding the data included in each group in units of at least one of data processing units, namely, a group, a subgroup, and a basic block.

[0064] In an embodiment of the present invention, the group encoding mode determination unit 120 may determine error rates that are generated by encoding the data of each group in units of respective data processing units, namely, in units of groups, subgroups, and basic blocks, and determine the group encoding mode for each group on the basis of the error rates corresponding to the respective data processing units. Accordingly, the image data is encoded in units of a data processing unit dynamically determined based on the error rates of each group for the respective data processing units.

[0065] The group encoding mode determined by the group encoding mode determination unit 120 includes at least one piece of information from among information about the type of a data processing unit which is used to encode the data of each group from among a group, a subgroup, and a basic block, information about the size of the data processing unit, the size representing the number of basic blocks included in the data processing unit, and information about a prediction mode that represents a method of determining reference information required to perform motion estimation based on the data processing unit.

[0066] The group encoding mode determination unit 120 may calculate and compare an error rate generated when the data of each group is encoded in units of basic blocks, an error rate generated when the data of each group is encoded in units of subgroups, and an error rate generated when the data of each group is encoded in units of groups, and determine a data processing unit having the smallest error rate as a unit by which the data of each group is encoded.

In other words, the encoding and the error rate calculation are performed in units of groups, subgroups, and basic blocks.

[0067] In an embodiment of the present invention, the group encoding mode determination unit 120 may compare error rates generated when each group has been encoded in several prediction modes in units of several types of data processing units, and determine a prediction mode having the smallest error rate for each type of data processing unit.

[0068] For example, the group encoding mode determination unit 120 may encode the data of each group in a skip mode, an inter mode, and an intra mode as prediction modes in units of different types of data processing units, calculate and compare encoding error rates for the skip mode, the inter mode, and the intra mode, and determine a prediction mode that generates the smallest error rate as a representative prediction mode for each of the data processing units. In other words, the comparison of the error rates generated due to the encoding in the respective prediction modes and the determination of the representative prediction mode may be performed in units of groups, subgroups, and basic blocks.

[0069] The group encoding mode determination unit 120 may compare error rates of the representative prediction modes for the respective data processing units with one another and determine a data processing unit having the smallest error rate as a representative data processing unit for the data of each group. Based on a combination of a representative data processing unit and a representative prediction mode both generating the smallest error rates, information about the type, the size, and the representative prediction mode of the representative data processing unit for each group is determined so as to be included in the group encoding mode for each group.

[0070] The group encoding unit 130 encodes and outputs the image data received from the data classifying unit 110 according to the group encoding mode for each group which is determined by the group encoding mode determination unit 120. The image encoding apparatus 100 may also encode a group encoding mode flag which represents the group encoding mode for each group, in addition to the image data.

[0071] The image encoding apparatus 100 may encode data included in one picture according to a zigzag scanning method instead of a raster scanning method in which raster scanning is performed between basic blocks.

[0072] The image encoding apparatus 100 may rearrange coefficients into which each group or each subgroup is frequency-transformed, in units of a plurality of basic blocks. The image encoding apparatus 100 may frequency-transform the image data in units of basic blocks or subgroups. At this time, a data processing unit by which frequency transformation is performed may not be identical with a data processing unit by which motion estimation is performed.

[0073] The image encoding apparatus 100 may encode a flag that represents whether frequency-transformed coefficients of the data of a predetermined data processing unit have been rearranged according to a variable data processing unit.

[0074] FIG. 2 is a block diagram of an image decoding apparatus 200 according to an embodiment of the present invention. Referring to FIG. 2, the image decoding apparatus 200 includes a receiving unit 210, a group encoding mode reading unit 220, and a group decoding unit 230. The concepts of a basic block, a group, a subgroup, an encoding mode, etc. in relation to FIG. 2 are the same as those of the basic block, the group, the subgroup, the encoding mode, etc. described above in relation to the image encoding apparatus 100.

[0075] The receiving unit 210 receives and parses data which has been encoded in units of basic blocks, groups, or subgroups, extracts data pieces to be decoded from the parsed data, and outputs the extracted data pieces to the group encoding mode reading unit 220 and the group decoding unit 230.

[0076] In an embodiment of the present invention, the group encoding mode reading unit 220 reads a group encoding mode from the parsed data output by the receiving unit 210 and outputs the group encoding mode to the group decoding unit 230. The group encoding mode reading unit 220 may read information about the shape, size, and maximum level of each group from the parsed data and determine each group based on the read-out information about the group shape and size. The properties of the group may be set differently for each group, or each unit of data units, such as, pictures and sequences.

[0077] In another embodiment, the group encoding mode reading unit 220 may extract from the parsed data a group encoding mode flag that represents a group encoding mode for each group, and read the group encoding mode for each group by using the group encoding mode flag. The read-out group encoding mode for each group includes information about the type, size, prediction mode, etc. of a data processing unit which is to be used when the data of each group is decoded.

[0078] The group decoding unit 230 decodes and outputs the data of each group, received from the receiving unit 210, by using the group encoding mode for each group received from the group encoding mode reading unit 220.

[0079] The group decoding unit 230 may determine a data processing unit to be used to decode the data of each group, based on the information about the type and size of the data processing unit, from among the pieces of information included in the group encoding mode for each group, and determine a prediction mode for the determined data processing unit on the basis of the information about a prediction mode, from among the pieces of information included in the group encoding mode for each group. The group decoding unit 230 may decode the data of each group on the basis of the data processing unit and the prediction mode which have been determined based on the group encoding mode for each group.

[0080] The group decoding unit 230 may read and decode the group encoding mode for each group by scanning the

data of a single picture according to a zigzag scanning method in which scanning is performed in units of basic blocks.

[0081] The image decoding apparatus 200 may rearrange frequency-transformed coefficients of a plurality of basic blocks, in units of either groups or subgroups. The image decoding apparatus 200 may inversely frequency-transform coefficients classified in units of basic blocks or subgroups.

5 [0082] The image decoding apparatus 200 may also extract, from the parsed data, a flag that represents whether frequency-transformed coefficients of either each group or each subgroup, have been rearranged in units of basic blocks during encoding. Based on the flag representing whether the coefficients have been rearranged, the coefficients may be re-arranged in units of either groups or subgroups during decoding.

[0083] FIG. 3 illustrates a concept of a group which is a data processing unit proposed by the present invention.

10 [0084] As hardware and a network environment that allow large-capacity data processing are established, an existing macroblock may be considered as a data processing unit which is relatively too small to store useful information compared with large-capacity data. For example, in a high-definition (HD) resolution or higher resolution, which is higher than a standard definition (SD) resolution, bits which are desirable to be used in actual texture coding due to small-sized macroblocks being allocated to additional information corresponding to a macroblock symbol such as a macroblock mode, a motion vector, or the like, and thus a rate-distortion (RD) cost may be reduced.

15 [0085] In the present invention, a group including at least one basic block is used as a data processing unit.

[0086] A basic block 300 is a data processing unit that includes at least one piece of data. For example, the basic block 300 may be an 8×8 macroblock, a 16×16 macroblock, or the like in standard MPEG or H.264.

20 [0087] In a group, which is a data processing unit including at least one basic block, its shape and its size which represents the number of basic blocks may be arbitrarily determined. For example, there exist a group 310 having a size of 16 and a square shape, groups 320 and 325 each having a size of 8 and a rectangular shape, and a group 330 having a size of 13 and a diamond shape. The shapes and sizes of the groups may be set constantly in units of pictures and picture sequences. Alternatively, the shapes and sizes of the groups may be set individually for each group.

[0088] FIG. 4 illustrates an example of a subgroup which is a data processing unit proposed by the present invention.

25 [0089] A subgroup is included in a group and includes at least one basic block. In a first method of determining a subgroup, a subgroup is determined according to a level and a scanning sequence. FIG. 4 illustrates groups that have square shapes, sizes of 16, and levels of 1 through 4.

[0090] The groups may be divided into a class of groups with a level of 4, namely, groups 410, 420, and 450, a class of groups with a level of 2, namely, groups 430 and 460, and a class of groups with a level of 1, namely, a group 440, according to the number of sub groups included in each group.

30 [0091] The groups may also be divided into a class of groups with a zigzag scanning sequence, namely, the group 410, a class of groups with a vertical scanning sequence, namely, the groups 420, 430, and 440, and a class of groups with a raster scanning sequence, namely, the groups 450 and 460, according to a scan sequence in which the basic blocks included in each group are scanned.

35 [0092] Accordingly, if subgroups for the group 410 with a size of 16 and a square shape are set according to a level of 4 and a zigzag scanning sequence, four subgroups with sizes of 4 and square shapes may be formed. If subgroups for the group 420 with a size of 16 and a square shape are set according to a level of 4 and a vertical scanning sequence, four subgroups 422, 424, 426, and 428 with sizes of 4 and rectangular shapes may be formed. Similarly, if subgroups for the group 450 with a size of 16 and a square shape are set according to a level of 4 and a raster scanning sequence, four subgroups 452, 454, 456, and 458 with sizes of 4 and rectangular shapes may be formed.

40 [0093] If subgroups are set for the group 430 with a size of 16 and a square shape according to a level of 2 and a vertical scanning sequence, two subgroups with sizes of 8 and rectangular shapes may be formed. Similarly, if subgroups for the group 460 with a size of 16 and a square shape are set according to a level of 2 and a raster scanning sequence, two subgroups 462 and 464 with sizes of 8 and rectangular shapes may be formed.

45 [0094] If subgroups for the group 440 with a size of 16 and a square shape are set according to a level of 1 and a vertical scanning sequence, a single subgroup 445 with a size of 16 and a square shape may be formed.

[0095] Although several scanning sequences may be set for each level as illustrated in FIG. 4, a predetermined scanning sequence may be set for each level. A data processing unit for frequency transformation may be equal to or smaller than the predefined sizes of the group and the subgroup.

50 [0096] FIG. 5 illustrates another example of a subgroup which is a data processing unit proposed by the present invention.

[0097] A second method of determining a subgroup is to divide a sequence of basic blocks which are arrayed according to a sequence in which the basic blocks included in each group are scanned. The second method of determining a subgroup will now be described in greater detail with reference to FIG. 5, which illustrates groups 500 and 550 that have square shapes and sizes of 16.

55 [0098] Basic blocks 1 through 16 of the group 500 may be rearranged according to a zigzag scan sequence so as to generate a basic block sequence 510 in which the basic blocks 1 through 16 are arranged in a sequence of basic blocks 1, 2, 5, 9, 6, 3, 4, 7, 10, 13, 14, 11, 8, 12, 15, and 16. If the level of the group 500 is 3, the basic block sequence 510

may be divided into a first subgroup including the basic blocks 1, 2, 5, 9, and 6, a second subgroup including the basic blocks 3, 4, 7, 10, and 13, and a third subgroup including the basic blocks 14, 11, 8, 12, 15, and 16. Accordingly, if the group 500 is divided into subgroups with a level of 3 according to the zigzag scan sequence, the group 500 may be divided into subgroups included in a group 520.

[0099] Basic blocks 1 through 16 of the group 550 may be rearranged according to an inverse zigzag scan sequence so as to generate a basic block sequence 560 in which the basic blocks are arranged in a sequence of basic blocks 4, 8, 3, 2, 7, 12, 16, 11, 6, 1, 5, 10, 15, 14, 9, 13. If the level of the group 550 is 4, the basic block sequence 560 may be divided into a first subgroup including the basic blocks 4, 8, 3, and 2, a second subgroup including the basic blocks 7, 12, 16, and 11, a third subgroup including the basic blocks 6, 1, 5, and 10, and a fourth subgroup including the basic blocks 15, 14, 9, and 13. Accordingly, if the group 550 is divided into subgroups with a level of 4 according to the inverse zigzag scan sequence, the group 550 may be divided into subgroups as in a group 570.

[0100] FIG. 6 is a flowchart for explaining a syntax that represents an encoding mode for a predetermined group, according to an embodiment of the present invention.

[0101] FIG. 6 is a flowchart of a process in which the group encoding mode reading unit 220 of the image decoding apparatus 200 reads out a group encoding mode according to the syntax of the encoding mode for the predetermined group.

[0102] In operation 610, the group encoding mode reading unit 220 checks a flag Group_SKIP that represents whether the data included in the predetermined group has been encoded in a skip mode of a group unit. If the value of the flag 'Group_SKIP' is 1, the data of the predetermined group is decoded in the skip mode of a group unit, in operation 620.

On the other hand, if the value of the flag 'Group_SKIP' is 0, the process proceeds to operation 630.

[0103] In operation 630, the group encoding mode reading unit 220 determines whether subgroups have been set based on a level. If subgroups based on a level are set, the group encoding mode reading unit 220 checks a method of performing decoding in units of subgroups, in operation 640. On the other hand, if subgroups have not been set based on a level, the group encoding mode reading unit 220 checks a method of performing decoding in units of basic blocks, in operation 650.

[0104] In an embodiment of the present invention, the level of subgroups should be greater than 1 and smaller than the size of a group corresponding to the subgroups, because if the level is 1, a set subgroup is identical with a group.

[0105] In operation 640, the group encoding mode reading unit 220 checks a flag Subgroup_SKIP that represents whether the data of the predetermined group has been encoded in a skip mode of a subgroup unit. If the value of the flag 'Subgroup_SKIP' is 1, the group decoding unit 230 decodes the data of the predetermined group in the skip mode of a subgroup unit, in operation 660. If the value of the flag 'Subgroup_SKIP' is 0, the group decoding unit 230 decodes the data of the predetermined group in units of subgroups in a prediction mode other than the skip mode, in operation 670.

[0106] To differentiate the skip mode of the subgroup unit from the skip mode of the group unit, all the values of the flags 'Subgroup_SKIP' of subgroups should not be set to be 1.

[0107] In an embodiment of the present invention, in the decoding on a subgroup-by-subgroup basis (Subgroup_Mode), a general mode is determined according to a level and a scan sequence of a subgroup. The level needs to be less than or equal to a maximum level.

[0108] In operation 650, the group encoding mode reading unit 220 checks a flag Unit_SKIP that represents whether the data of the predetermined group has been encoded in a skip mode of a basic block unit. If the value of the flag 'Unit_SKIP' is 1, the group decoding unit 230 decodes the data of the predetermined group in the skip mode of a basic block unit, in operation 680. If the value of the flag 'Unit_SKIP' is 0, the group decoding unit 230 decodes the data of the predetermined group in units of basic blocks in a prediction mode other than the skip mode, in operation 690.

[0109] In an embodiment of the present invention, to differentiate the skip mode of the basic block unit from the skip mode of the subgroup unit or the skip mode of the group unit, all of the values of the flags 'Unit_SKIP' of subgroups should not be set to be 1.

[0110] FIG. 7 illustrates an example of methods of encoding different types of data processing units.

[0111] The methods of encoding the different types of data processing units may be determined arbitrarily according to circumstances. An example of a method of encoding a group 710 with a square shape, a size of 4, and a maximum level of 2 in units of each type of data processing unit will now be described.

[0112] When the group 710 has a level of 2, a subgroup 720 with a vertical scan sequence and a subgroup 730 with a raster scan sequence may be set. When the group 710 has a level of 1, a subgroup 740 is set.

[0113] In a method of encoding basic blocks 700, a 16×16 skip mode may be used as a skip mode Unit_SKIP, and one of 16×16 , 8×8 , and 4×4 intra modes and 16×16 , 16×8 , 8×16 , 8×8 , 4×8 , 8×4 , and 4×4 inter modes may be used as a general mode Unit_Mode.

[0114] In a method of encoding the subgroup 740 with the level of 1, a 32×32 inter mode or a 32×32 intra mode may be used as a general mode Subgroup_Mode. In a method of encoding the subgroups 720 and 730 with the level of 2, a 16×32 inter mode or a 32×16 inter mode may be used as a general mode Subgroup_Mode.

[0115] In a method of encoding the group 720, a 32×32 skip mode may be used as a skip mode Group_SKIP.

[0116] FIG. 8 is a flowchart of an encoding method 800 of determining a group encoding mode, according to an embodiment of the present invention. A method of determining an optimal data processing unit and an optimal prediction mode in connection with an encoding mode for each data processing unit will now be described with reference to FIG. 8.

[0117] The encoding method 800 is a process in which the image encoding apparatus 100, particularly, the group encoding mode determination unit 120, determines an encoding mode for a predetermined group.

[0118] In operation 810, the predetermined group is determined from data to be encoded, based on information about a group shape, a group size, and a group maximum level which is acquired from the to-be-encoded data. The information about the group shape, the group size, and the group maximum level may be set individually for each group, or each unit of data units such as pictures or sequences.

[0119] Encoding performed on the predetermined group in units of each data processing unit will now be described.

[0120] First, in operation 820, encoding is performed on the predetermined group in a skip mode of a group unit. In operation 830, an RD cost is calculated as an error rate generated due to the encoding performed on the predetermined group in the skip mode of the group unit. For example, an RD cost generated when a 32×32 group is encoded in a 32×32 skip mode is calculated.

[0121] In operation 840, encoding is performed on the predetermined group in units of basic blocks. The encoding in units of basic blocks may be performed in each of a skip mode, an intra mode, and an inter mode. Since encoding should be performed on each of the basic blocks included in each group, operation 840 is repeated a number of times equal to the number of basic blocks included in each group, wherein the number of basic blocks corresponds to the size of each group.

[0122] In operation 850, RD costs generated due to the encoding of the basic blocks included in the predetermined group are calculated. For example, a basic block having the smallest RD cost from among RD costs which are generated due to encoding of the basic blocks included in the predetermined group in a 16×16 skip mode, a $16 \times 16/8 \times 8/4 \times 4$ intra mode, and a $16 \times 16/16 \times 8/8 \times 16/8 \times 8/4 \times 8/8 \times 4/4 \times 4$ inter mode is detected. In other words, a combination of a basic block size and a prediction mode that correspond to the smallest RD cost is selected for each basic block.

[0123] In operation 860, encoding is performed on the predetermined group in units of subgroups. More specifically, subgroups for the predetermined group are set by checking a level in operation 862 and checking a scan sequence in operation 864. Then, in operation 860, encoding is performed on each of the subgroups corresponding to the checked level. The encoding in units of subgroups may be performed in each of an intra mode and an inter mode. Since encoding needs to be performed on each of the subgroups included in the predetermined group, operations 862, 864, and 860 are repeated a number of times corresponding to the value of the checked level.

[0124] In operation 870, RD costs generated due to the encoding of the subgroups included in the predetermined group are calculated. An optimal prediction mode having the smallest RD cost from among RD costs generated due to encoding performed on the subgroups included in the predetermined group in an intra mode and an inter mode is determined, and a combination of a subgroup having the smallest RD cost from among the subgroups included in the predetermined group and the determined optimal prediction mode is determined. For example, a subgroup having the smallest RD cost from among RD costs which are generated due to encoding of the subgroups in a 32×32 inter mode/ 32×32 intra mode or a $16 \times 32/32 \times 16$ inter mode is detected. In other words, a combination of a subgroup size and a prediction mode that correspond to the smallest RD cost is selected for each subgroup.

[0125] In operation 880, the group encoding mode determination unit 120 compares the RD cost of the predetermined group calculated in operation 830, the minimum RD cost of the basic block combination selected in operation 850, and the minimum RD cost of the subgroup combination selected in operation 870 with one another and selects a smallest RD cost from the three RD costs.

[0126] In operation 890, the group encoding mode determination unit 120 determines the encoding mode for the predetermined group on the basis of a data processing unit corresponding to the smallest RD cost selected in operation 880 or on the basis of a data processing unit combination which corresponds to the smallest RD cost selected in operation 880. For example, the encoding mode for the predetermined group may include information about the type of a data processing unit from among the group, the subgroup, and the basic block, information about the size (for example, 32×32 , 32×16 , 16×32 , 16×16 , 16×8 , 8×16 , 8×8 , 4×8 , 8×4 , 4×4) of the data processing unit, and information about a prediction mode (for example, a skip mode, an intra mode, and an inter mode).

[0127] FIG. 9 is a flowchart of a method 900 of reading an encoding mode for a predetermined group in order to perform decoding, according to an embodiment of the present invention.

[0128] The method 900 illustrates a process in which the image decoding apparatus 200, particularly, the group encoding mode reading unit 220, reads the encoding mode for the predetermined group in order to decode the predetermined group.

[0129] In operation 910, the group encoding mode reading unit 220 acquires information about the shape, size, and maximum level of the predetermined group from data to be decoded. The information about the shape, size, and maximum level of the predetermined group may be set individually for each group, or each unit of data units such as pictures or sequences.

[0130] In operation 920, the group encoding mode reading unit 220 checks a flag Group_Skip that represents whether the predetermined group has been encoded in a group-by-group skip mode. If it is determined that the predetermined group has been encoded in the group-by-group skip mode, the predetermined group is decoded in the group-by-group skip mode, in operation 930. On the other hand, if it is determined that the predetermined group has not been encoded in the group-by-group skip mode, the method 900 proceeds to operation 940.

[0131] In operation 940, the group encoding mode reading unit 220 checks the maximum level of the predetermined group. If it is determined that subgroups have not been set, the group encoding mode reading unit 220 investigates a method of decoding each basic block, in operation 990. On the other hand, if it is determined that subgroups have been set, the group encoding mode reading unit 220 checks a level for the set subgroups in operation 960, and determines a scan sequence in operation 970. In operation 950, the group encoding mode reading unit 220 performs decoding on subgroups determined according to the level and the scan sequence. Since the decoding is performed on each of the subgroups of the predetermined group, operations 960, 970, and 950 are repeated a number of times corresponding to the value of the level of the predetermined group.

[0132] In operation 990, the group encoding mode reading unit 220 checks a flag Unit_mode that represents a prediction mode of a basic block unit. In operation 980, the group encoding mode reading unit 220 decodes a corresponding basic block according to the prediction mode determined according to the result of the checking. Since the decoding is performed on each of the basic blocks included in the predetermined group, operations 990 and 980 are repeated a number of times that corresponds to the size of the predetermined group.

[0133] Various embodiments of encoding and decoding in units of various data processing units such as a group, a subgroup, and a basic block according to the present invention will now be described in detail.

[0134] In an embodiment of the present invention, in a motion vector estimating method based on an inter mode, a median value of motion vectors of all data processing units neighboring a current data processing unit may be determined as a motion vector of the current data processing unit.

[0135] In an embodiment of the present invention, the motion vector estimating method based on an inter mode may vary according to the sizes of a group, a subgroup, and a basic block. For example, the size, type, etc. of neighboring blocks which are to be referred to, may vary according to the size of a data processing unit.

[0136] In an embodiment of the present invention, in a motion vector estimating method based on an intra mode, 5 adjacent pixels located at a certain angle or in a certain direction of a current data processing unit are referred to, and thus a weighted average of the values of the 5 adjacent pixels may be estimated as a current pixel value.

[0137] In an embodiment of the present invention, in an example of the motion vector estimating method based on an intra mode, the data of a predetermined group may be rearranged in units of a data processing unit by which the predetermined group is frequency transformed, and the rearranged data pieces may be used as reference pixels. In another example of the motion vector estimating method based on an intra mode, an x-axis variation and a y-axis variation of a motion vector may be represented as a gradation V.

[0138] In a block motion estimating method based on a 32×32 intra mode, a current block may be estimated using a weighted average of two estimation signals such as motion vectors. The weighted average may be obtained according to a method of giving a greater weighted value to a reference pixel closest to each pixel than to other reference pixels neighboring each pixel.

[0139] In an embodiment of the present invention, flags that represent skip modes for groups may be collected in units of pictures and processed all at once. Only flags that represent skip modes for the groups included in a B picture may be processed all at once.

[0140] An embodiment of the present invention proposes a method of inducing information corresponding to a current data processing unit, as a method compared with a skip mode. For example, both motion vector information and texture information may be induced instead of being received. The motion vector information may be received, and the texture information may be induced. Alternatively, the motion vector information may be induced, and the texture information may be received. A prediction mode in which only information regarding direct current (DC) coefficients is received may be possible. Various motion vector information inducing methods may be defined for each quantization parameter (Qp).

[0141] In an embodiment of the present invention, a data processing unit of a skip mode may have a hierarchical structure. For example, if a 16×16 block included in a 32×32 block of a skip mode is processed in a skip mode, a flag that represents a hierarchical structure of a 32×32 skip flag and a 16×16 skip flag may be set. In addition, a group that is a collection of basic blocks corresponding to a skip mode may be formed.

[0142] According to embodiments of the present invention, video compression performance may be improved by increasing the size of an existing macroblock to bigger sizes so as to efficiently cope with encoding/decoding of the increasing resolution of video contents.

[0143] By contrast with an existing H.264 moving picture encoding/decoding method, there is proposed another embodiment of the present invention in which a hierarchical concept of a group, a subgroup, and a basic block, which is proposed by the present invention, is implemented as a hierarchical structure of data processing units, namely, a 32×32 super macroblock ('SMB' or 'S-MB'), a 16×16 macroblock (MB), and an 8×8 macroblock. This proposed embodiment

of the present invention will now be described with reference to FIGS. 10 through 17.

[0144] FIG. 10 illustrates a hierarchical structure 1000 of a data processing unit which is available in the present invention.

[0145] The hierarchical structure 1000 may have three layers, namely, a first layer 1010, a second layer 1020, and a third layer 1030. The first layer 1010 is a 32×32 super macroblock which is the largest. For example, the first layer 1010 may be a single macroblock of a 32×32 skip mode, a single macroblock of a 32×32 inter mode, two macroblocks of a 32×16 inter mode, two macroblocks of a 16×32 inter mode, or four macroblocks of a 16×16 inter mode. The first layer 1010 may be frequency-transformed in units of 16×16 blocks, 8×8 blocks, and 4×4 blocks.

[0146] The second layer 1020 is a 16×16 macroblock which is a part of the 32×32 super macroblock of the first layer 1010. For example, the second layer 1020 may be a single macroblock of a 16×16 intra mode, a single macroblock of a 16×16 skip mode, a single macroblock of a 16×16 inter mode, two macroblocks of a 16×8 inter mode, two macroblocks of an 8×16 inter mode, or four macroblocks of an 8×8 inter mode. The second layer 1020 may be frequency-transformed in units of 8×8 blocks and 4×4 blocks.

[0147] The third layer 1030 corresponds to an 8×8 macroblock which is a part of the 16×16 macroblock of the second layer 1020. For example, the 8×8 macroblock of the third layer 1030 may be a single block of an 8×8 skip mode, a single block of an 8×8 inter mode, two blocks of an 8×4 inter mode, two blocks of a 4×8 inter mode, or four blocks of a 4×4 inter mode. The third layer 1030 may be frequency-transformed in units of 4×4 blocks.

[0148] According to an extended hierarchical structure of a macroblock proposed by the present invention, encoding and decoding may be performed in a 32×32 super macroblock mode for large-capacity video data and also in a macroblock mode less than or equal to an 8×8 macroblock mode. In addition, since prediction modes (such as a skip mode, an intramode, etc.) and processing of macroblocks smaller than or equal to a 16×16 macroblock in an extended hierarchical structure of a macroblock according to the present invention may be the same as those in an existing encoding/decoding method, costs and time required to establish a new encoding/decoding system may be reduced.

[0149] FIG. 11 is a flowchart for explaining a syntax of an encoding mode in which encoding is performed in units of SMBs and MBs, according to another embodiment of the present invention.

[0150] In operation 1110, the group encoding mode reading unit 220 of the image decoding apparatus 200 checks a flag SMB_SKIP that represents whether a current group has been encoded in a skip mode for super macroblocks in order to determine an encoding mode for the current group. If the value of the flag 'SMB_SKIP' is 1, the group decoding unit 230 performs decoding through motion estimation in a skip mode for 32×32 super macroblocks, in operation 1120.

[0151] On the other hand, if the value of the flag 'SMB_SKIP' is 0, the group encoding mode reading unit 220 determines whether the current group has been encoded in units of super macroblocks, in operation 1130. If it is determined in operation 1130 that the current group has been encoded in units of super macroblocks in prediction modes other than a skip mode, a result of the determination is transmitted to the group decoding unit 230, and the group decoding unit 230 performs decoding through motion estimation in units of super macroblocks of a 32×32 inter mode, a 32×16 inter mode, a 16×32 inter mode, and a 16×16 inter mode according to a flag 'SMB_mode', in operation 1140.

[0152] A method of performing motion estimation in units of 16×16 inter mode super macroblocks according to an embodiment of the present invention may be the same as an existing method of performing motion estimation in units of 16×16 inter mode macroblocks.

[0153] On the other hand, if it is determined in operation 1130 that the current group has not been encoded in an inter mode for super macroblocks, the group encoding mode reading unit 220 checks a flag MB_SKIP that represents whether the current group has been encoded in a macroblock-by-macroblock skip mode, in operation 1150.

[0154] If it is determined in operation 1150 that the current group has been encoded in the macroblock-by-macroblock skip mode, the result of the determination is transmitted to the group decoding unit 230, and the group decoding unit 230 performs decoding through motion estimation of a skip mode in units of 16×16 super macroblocks according to the flag 'MB_SKIP', in operation 1160.

[0155] On the other hand, if it is determined in operation 1150 that the current group has not been encoded in the macroblock-by-macroblock skip mode, the result of the determination is transmitted to the group decoding unit 230, and the group decoding unit 230 performs decoding in units of 16×16 intra mode macroblocks, 16×16 inter mode macroblocks, 16×8 inter mode macroblocks, 8×16 inter mode macroblocks, or 8×8 inter mode macroblocks or in units of sub macroblocks smaller than 8×8 macroblocks according to a flag 'SMB_mode', in operation 1170.

[0156] FIG. 12 illustrates a motion vector estimating method which is performed when an encoding mode is a skip mode, according to another embodiment of the present invention.

[0157] The image encoding apparatus 100 and the image decoding apparatus 200 may use a motion vector estimating method based on super macroblocks or a motion vector estimating method based on macroblocks, in order to estimate motion vectors of blocks corresponding to a skip mode.

[0158] In an example of the motion vector estimating method based on super macroblocks, a motion vector of a 32×32 skip mode super macroblock may be estimated using motion vectors of blocks neighboring the 32×32 skip mode super macroblock. The neighboring blocks may be macroblocks or super macroblocks. For example, a motion vector of a

super macroblock is estimated according to Equation 1:

$$MV = \text{Median}(MVa, MVb, MVc) \quad \dots (1)$$

wherein a function Median(·) denotes a function of outputting a median value of an input. In other words, a motion vector MV 1210 of a super macroblock is determined to be a motion vector having a median value of a motion vector MVa 1220 of a left block located on the left side of the super macroblock from among blocks neighboring the super macroblocks, a motion vector MVb 1225 of an upper block located on the upper side of the super macroblock, and a motion vector MVc1230 of a right block located on the right side of the super macroblock.

[0159] In an example of the motion vector estimating method based on macroblocks, a motion vector of a 16×16 skip mode macroblock may be estimated using motion vectors of blocks neighboring the 16×16 skip mode macroblock. For example, a motion vector of a macroblock may be estimated according to Equations 2, 3, 4, and 5:

$$MV0 = \text{Median}(MVa, MVb, MVc) \quad \dots (2)$$

$$MV1 = \text{Median}(MV0, MVc, MVd) \quad \dots (3)$$

$$MV2 = \text{Median}(MVe, MV0, MV1) \quad \dots (4)$$

$$MV3 = \text{Median}(MV0, MV1, MV2) \quad \dots (5)$$

[0160] Each of the motion vectors of 16×16 skip mode macroblocks MV0 1240, MV1 1245, MV2 1250, and MV3 1255 is determined to be a motion vector having a median value of respective motion vectors of a left block, an upper block, and a left upper block, from among blocks neighboring each of the 16×16 skip mode macroblocks MV0 1240, MV1 1245, MV2 1250, and MV3 1255. In other words, the motion vector of the macroblock MV0 1240 is determined to be a median value of motion vectors of blocks MVa 1260, MVb 1265, and MVc 1270 according to Equation 2, the motion vector of the macroblock MV1 1245 is determined to be a mean value of motion vectors of the blocks MV0 1240 and MVc 1270 and a block MVd 1275 according to Equation 3, the motion vector of the macroblock MV2 1250 is determined to be a mean value of motion vectors of a block MVe 1280 and the blocks MV0 1240 and MV1 1245 according to Equation 4, and the motion vector of the macroblock MV3 1255 is determined to be a mean value of motion vectors of the blocks MV2 1250, MV0 1240, and MV1 1245 according to Equation 5.

[0161] The image encoding apparatus 100 may encode a flag that represents whether a motion vector estimating method is performed in units of super macroblocks or in units of macroblocks, and insert the flag into slices or data pieces of a predetermined data unit. The image decoding apparatus 200 may extract the flag associated with the motion vector estimating method from each slice or each predetermined data unit and use the flag when estimating a motion vector of a super macroblock or a macroblock.

[0162] FIG. 13 illustrates a sequence of encoding and decoding each data processing unit, according to another embodiment of the present invention.

[0163] To support an extended macroblock mode having a three-layered hierarchical structure, a zigzag scanning method 1320 in units of 16×16 macroblocks, instead of a raster scanning method 1310 that horizontally scans each macroblock, is used in an image encoding or decoding operation of the present invention.

[0164] For example, a current macroblock may be a 16×16 macroblock(MB) included in a collection of 16×16 MBs like a block 1330, an MB included in a 32×32 super macroblockS-MB like a block 1340, an MB included in a pair of 32×16 or 16×32 super macroblocks $32 \times 16/16 \times 32$ S-MB like a block 1350, an MB included in a 32×16 or 16×32 half-super macroblock $32 \times 16/16 \times 32$ S-MB half like a block 1370, or a 16×16 MB like a block 1360.

[0165] The group encoding mode determination unit 120 of the image encoding apparatus 100 may compare an RD cost generated due to encoding of a single 32×32 super macroblock with that generated due to encoding of four 16×16 macroblocks, and determine whether encoding is performed in either a 32×32 super macroblock mode or a 16×16 macroblock mode.

[0166] Since an existing encoding or decoding system can be used without big changes in its structure, that is, by only changing a scanning method, due to the use of a macroblock zigzag-scanning method, the present invention can

be implemented using an existing encoding or decoding system.

[0167] An edge portion of an image that does not conform to the size of a super macroblock may be processed using a 32×16 or 16×32 half-super macroblock like the block 1370 or a 16×16 macroblock like the block 1360.

[0168] FIG. 14 illustrates a method of arranging frequency coefficients, according to another embodiment of the present invention.

[0169] In an image encoding process and an image decoding process according to the present invention, frequency coefficients 1400 of a 16×16 block may be rearranged into 8×8 macroblocks 1450 according to an interleaving method. The frequency coefficients 1400 rearranged in units of 8×8 macroblocks 1450 may be entropy-encoded or decoded in units of 8×8 macroblocks 1450.

[0170] In an example of a method of rearranging the coefficients of a 16×16 block according to an interleaving method performed in units of 8×8 macroblocks, the coefficients of the 16×16 block may be rearranged into a left upper 8×8 macroblock, a right upper 8×8 macroblock, a left lower 8×8 macroblock, and a right lower 8×8 macroblock according to the positions of the coefficients of the 16×16 block.

[0171] According to the coefficient rearrangement sequence, a left upper coefficient from among the coefficients included in a 2×2 block of a 16×16 block is arranged into a left upper 8×8 macroblock 1460. In other words, a coefficient 1410 of the 16×16 block is arranged as a coefficient 1462 of the left upper 8×8 macroblock 1460.

[0172] A right upper coefficient from among the coefficients included in the 2×2 block of the 16×16 block is arranged into a right upper 8×8 macroblock 1470. In other words, a coefficient 1430 of the 16×16 block is arranged as a coefficient 1472 of the right upper 8×8 macroblock 1470.

[0173] Similarly, a left lower coefficient from among the coefficients included in the 2×2 block of the 16×16 block is arranged into a left lower 8×8 macroblock 1480. In other words, a coefficient 1420 of the 16×16 block is arranged as a coefficient 1482 of the left lower 8×8 macroblock 1480.

[0174] Similarly, a right lower coefficient from among the coefficients included in the 2×2 block of the 16×16 block is arranged into a right lower 8×8 macroblock 1490. In other words, a coefficient 1440 of the 16×16 block is arranged as a coefficient 1492 of the right lower 8×8 macroblock 1490.

[0175] FIG. 15 illustrates an existing encoding method 1510 (on the left of FIG. 15) and a syntax encoding method 1550 (on the right of FIG. 15) according to another embodiment of the present invention, wherein the latter is associated with a coded block pattern (CBP) and the size of a coefficient block.

[0176] In the existing encoding method 1510 of encoding a coefficient obtained in consideration of a CBP, in operation 1520, the value of the CBP is checked. If the CBP value is 0, the method 1510 is concluded. On the other hand, if the CBP value is not 0, a flag TFlag representing the size of a coefficient block is encoded, in operation 1530. In operation 1540, a current coefficient is encoded. In this way, the encoding of the coefficient obtained in consideration of the CBP is completed. When coefficients obtained by frequency transformation in units of 8×8 or 4×4 blocks are arranged according to an existing encoding and decoding method, the flag 'Tflag' may represent whether the coefficients have been obtained in units of 8×8 or 4×4 blocks.

[0177] In the syntax encoding method 1550 which considers a CBP, in operation 1520, the value of the CBP is checked. If the CBP value is 0, the method 1550 is concluded. On the other hand, if the CBP value is not 0, a flag TFlag8 representing where a current coefficient is a coefficient obtained in units of 8×8 or 4×4 blocks is encoded, in operation 1560. In operation 1570, it is determined whether the current coefficient is a coefficient obtained in units of 4×4 blocks. If the current coefficient is a coefficient obtained in units of 4×4 blocks, the current coefficient is encoded, in operation 1590. In this way, the encoding of the coefficient obtained in consideration of the CBP is completed.

[0178] On the other hand, if the current coefficient is not obtained in units of 4×4 blocks, a flag TFlag16 representing where the current coefficient is a coefficient obtained in units of 8×8 or 16×16 blocks is encoded, in operation 1580. In operation 1590, the current coefficient is encoded. In this way, the encoding of the coefficient obtained in consideration of the CBP is completed.

[0179] Accordingly, encoding by performing frequency transformation, and decoding by performing frequency inverse-transformation in data processing units of variable block sizes may be performed by using the two flags TFlag8 and TFlag16 associated with the data processing units into which frequency-transformed coefficients are arranged.

[0180] In connection with frequency transformation, an embodiment of the present invention may use not only the size of a block into which frequency-transformed coefficients are arranged, but also frequency transformation in units of integers. The length of an End of Block (EOB) may be reduced through a coefficient block obtained according to an 8×8 interleaving method. When the value of the flag 'Tflag8' is 1, an overhead for a 16×16 inter mode additionally requires only one bit. Since energy compression efficiency is increased due to various size blocks, a frequency transformation frequency may be reduced.

[0181] FIG. 16 is a flowchart for explaining a CBP-related syntax according to another embodiment of the present invention.

[0182] The image encoding apparatus 200 uses a method of encoding a hierarchical CBP. In operation 1600, a flag MB_CBP_flag representing whether the CBP has been encoded is encoded. In operation 1610, it is determined whether

the value of the flag 'MB_CBP_flag' is 0. If it is determined that the value of the flag 'MB_CBP_flag' is 0, additional information is not needed, and thus the CBP encoding is concluded. On the other hand, if it is determined that the value of the flag 'MB_CBP_flag' is 1, encoding of the CBP for a corresponding macroblock is performed, in operation 1620.

[0183] The image decoding apparatus 200 may extract the flag 'MB_CBP_flag' from received data and determine whether CBP decoding with respect to a macroblock to be decoded is needed.

[0184] FIG. 17 illustrates an encoding mode for a B frame, according to another embodiment of the present invention.

[0185] The image encoding apparatus 100 and the image decoding apparatus 200 may use an S-MB direct mode as the encoding mode for a B frame. If an encoding mode for a current super macroblock is the S-MB direct mode, texture coding should be performed.

[0186] For example, flags MB_CBP_FLAG respectively included in 16×16 macroblocks 1720, 1730, 1740, and 1750 of a 32×32 super macroblock 1710, each flag representing whether CBP encoding has been performed, represent whether the 16×16 macroblocks 1720, 1730, 1740, and 1750 have been encoded in a skip mode or a direct mode.

[0187] That is, since the 16×16 macroblocks 1720 and 1750 having flags 'MB_CBP_FLAG' of 1 are macroblocks 1760 and 1790 of the direct mode, a motion vector of 16×16 macroblocks 1720 and 1750 may be predicted using a motion prediction method which is the same as a direct mode conforming to the H.264 standard.

[0188] Since the 16×16 macroblocks 1730 and 1740 having flags 'MB_CBP_FLAG' of 0 are macroblocks 1770 and 1780 of the direct mode, further information for motion vector estimation is not needed.

[0189] In an example of the image encoding apparatus 100 and the image decoding apparatus 200, a block which is a data processing unit for a dynamic range division method may be determined based on a quantization parameter, a resolution, or sequence characteristics. In addition, data pieces corresponding to a data processing unit may be classified in not only horizontal and vertical directions but also various other directions.

[0190] In the above-described three-level hierarchical structure, a first level may include not only super macroblocks of a skip mode and an intermode but also a super macroblock of a prediction mode corresponding to a combination of an intramode and an intermode.

[0191] An encoding mode or a data processing unit may be set for each data unit, such as a picture, a sequence, a slice, and a macroblock, and may be represented as a maximum or minimum basic block size and a branched layer level. A binary tree of the hierarchical structure of data processing units may be extended into an n-ary tree (where n is an integer more than 2).

[0192] In an embodiment of the image encoding apparatus 100 and the image decoding apparatus 200, block sizes may be maintained in a decreasing order from the size of a block for motion estimation, to the size of a block for frequency transformation, through to the size of a block for deblocking filtering. In addition, a data processing unit for motion estimation and a data processing unit for frequency transformation may be incorporated or separated when represented.

[0193] Intra- or inter-mode macroblocks that constitute a group may be formed as a group including an arbitrary number of basic blocks. In this case, block estimation may be performed based on the size of a group, or frequency transformation may be performed based on the size of a group.

[0194] An arbitrary number of skip-mode macroblocks may constitute a group.

[0195] FIG. 18 is a flowchart of an image encoding method according to an embodiment of the present invention.

[0196] In operation 1810, basic blocks into which image data is divided are classified according to an extended concept including a group, a subgroup, and a basic block. A group may have properties such as a size representing the number of basic blocks included in the group, a shape of the group, and a maximum level of the group. A subgroup may be set according to a level and a scan sequence.

[0197] In operation 1820, a group encoding mode for each group is determined, including information about at least one of data processing units, namely, a group, a subgroup, and a basic block, and information about a method of encoding the data included in each group in units of the at least one data processing unit.

[0198] The group encoding mode for each group may be determined based on error rates that are generated due to encoding of the data of each group in units of different types of data processing units that are each smaller than or equal to the size of each group. An encoding error rate of a predetermined group, a lowest error rate of the encoding error rates of the subgroups included in the group, and a lowest error rate of the encoding error rates of the basic blocks included in the group are compared with one another, and thus an optimal data processing unit and an optimal prediction mode are selected. The group encoding mode for each group includes information about an optimal data processing unit and an optimal prediction mode for each group.

[0199] In operation 1830, the data of each group is encoded according to the group encoding mode. Not only information representing image data and an encoding mode for each group, but also information about whether frequency-transformed coefficients of the data of each group have been rearranged in units of a predetermined data processing unit, CBP encoding information, etc. may be encoded and inserted into transmission data.

[0200] FIG. 19 is a flowchart of an image decoding method according to an embodiment of the present invention.

[0201] In operation 1910, data encoded in data processing units of one of a basic block, a group, and a subgroup, is received, and information required during encoding is parsed from the received data. Image data which is to be decoded

may be parsed from the received data. Parameters required for image decoding may be parsed from the received data.

[0202] In operation 1920, an encoding mode for a predetermined group, which represents a method of encoding data included in the predetermined group in units of one of a group, a subgroup, and a basic block, is read from the data parsed in operation 1910.

[0203] To determine each group, information representing the size, shape, and maximum level of the group is extracted and read from the parsed data. If each group is determined, an encoding mode for the determined group is extracted and read.

[0204] In operation 1930, data included in the determined group corresponding to the read encoding mode is decoded.

[0205] The data included in the determined group is decoded using information about a data processing unit, indicating in which units the determined group should be decoded, information about the size and shape of the data processing unit, and information about a motion prediction mode. These pieces of information are extracted and read from the encoding mode for the predetermined group.

[II. image encoding apparatus based on data processing unit]

[0206] FIGS. 20A, 20B, 20C, 20D, 20E, and 20F are block diagrams of modified embodiments of the image encoding apparatus 100 that consider a data processing unit dynamically determined according to an embodiment.

[0207] The group encoding unit 130 of the image encoding apparatus 100 performs frequency transformation and quantization, motion estimation including inter prediction and intra prediction, in-loop filtering, and entropy encoding, in order to encode the data of a group based on a data processing unit and a prediction mode that are based on the group encoding mode determined by the group encoding mode determination unit 120.

[0208] The image encoding apparatus 100 dynamically determines a data processing unit of the group. Accordingly, the data processing unit of the group may be a group, a subgroup, or a basic block that form a hierarchical structure.

[0209] Since the data processing unit dynamically determined by the image encoding apparatus 100 may be a data processing unit that is not defined in an existing image encoding format, a unique process distinguished from an existing image encoding process is required. For example, when a large data processing unit having a size larger than a 16×16 or 8×8 macroblock is used, the group encoding unit 130 should perform image encoding through a process that considers the large data processing unit.

[0210] FIG. 20A is a block diagram of a case where frequency transformation and quantization considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0211] A first embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes a frequency transformation and quantization unit 2001 considering a large data processing unit, a motion estimation unit 133 including an intra prediction unit and an inter prediction unit, an in-loop filtering unit 137, and an entropy encoding unit 139.

[0212] The frequency transformation and quantization unit 2001 considering the large data processing unit according to the first embodiment in the group encoding unit 130 may perform integer discrete cosine transformation and quantization based on the large data processing unit in order to perform frequency transformation and quantization by a large data processing unit based on the group encoding mode determined by the group encoding mode determination unit 120.

[0213] The frequency transformation and quantization unit 2001 considering the large data processing unit according to the first embodiment in the group encoding unit 130 may perform discrete cosine transformation and quantization on a chroma component of a large data processing unit, in order to perform frequency transformation and quantization by the large data processing unit.

[0214] The frequency transformation and quantization unit 2001 considering the large data processing unit according to the first embodiment in the group encoding unit 130 may perform low-complexity discrete cosine transformation, in order to reduce the number of calculations that may be added by discrete cosine transformation based on the large data processing unit.

[0215] FIG. 20B is a block diagram of a case where motion estimation processing considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0216] A second embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes a frequency transformation and quantization unit 131, a motion estimation unit 2003 by processing considering the large data processing unit, the in-loop filtering unit 137, and the entropy encoding unit 139. Intra prediction and inter prediction may be performed based on the processing considering the large data processing unit.

[0217] The motion estimation unit 2003 by the processing considering the large data processing unit according to the second embodiment in the group encoding unit 130 may previously determine a combination of a large data processing unit for motion estimation and a large data processing unit for frequency transformation, in consideration of a large data processing unit based on the group encoding mode determined by the group encoding mode determination unit 120. The size of the data processing unit for frequency transformation may be smaller than that of the data processing unit for motion estimation.

[0218] The motion estimation unit 2003 by the processing considering the large data processing unit according to the

second embodiment in the group encoding unit 130 may determine a scanning method for motion estimation for each data processing unit and neighbor availability in which a neighboring block can be used as a reference block, in consideration of the large data processing unit.

[0219] The motion estimation unit 2003 by the processing considering the large data processing unit according to the second embodiment in the group encoding unit 130 may use a simplified encoding mode using only a square-shaped data processing unit in consideration of the large data processing unit, in order to reduce bit rate of bits representing an encoding mode.

[0220] FIG. 20C is a block diagram of a case where intra prediction considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0221] A third embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes the frequency transformation and quantization unit 131, an intra prediction unit 2004 considering a large data processing unit, an inter prediction unit 135, the in-loop filtering unit 137, and the entropy encoding unit 139.

[0222] The intra prediction unit 2004 considering the large data processing unit according to the third embodiment in the group encoding unit 130 may additionally define an intra mode for performing intra prediction with respect to a predetermined large data processing unit.

[0223] The intra prediction unit 2004 considering the large data processing unit according to the third embodiment in the group encoding unit 130 may use a prediction mode using a complex plane into the intra mode, in order to perform more accurate intra prediction with respect to a large data processing unit.

[0224] The intra prediction unit 2004 considering the large data processing unit according to the third embodiment in the group encoding unit 130 may restrict the type of large data processing unit on which intra prediction can be performed.

[0225] FIG. 20D is a block diagram of a case where inter prediction considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0226] A fourth embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes the frequency transformation and quantization unit 131, an intra prediction unit 134, an inter prediction unit 2005 considering a large data processing unit, the in-loop filtering unit 137, and the entropy encoding unit 139.

[0227] The inter prediction unit 2005 considering the large data processing unit according to the fourth embodiment in the group encoding unit 130 may use a combination mode in which inter prediction and intra prediction can be mixed, as a prediction mode of the large data processing unit.

[0228] The inter prediction unit 2005 considering the large data processing unit according to the fourth embodiment in the group encoding unit 130 may determine a motion vector (MV) for each large data processing unit and induce a motion vector predictor (MVP) according to the MV.

[0229] The inter prediction unit 2005 considering the large data processing unit according to the fourth embodiment in the group encoding unit 130 may newly use a skip mode for a large data processing unit and set a skip group including a plurality of data processing units of a plurality of skip modes. All of the data processing units of the skip modes included in the skip group may be skipped all together.

[0230] FIG. 20E is a block diagram of a case where in-loop filtering considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0231] A fifth embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes the frequency transformation and quantization unit 131, the motion estimation unit 133 including an intra prediction unit and an inter prediction unit, an in-loop filtering unit 2007 considering a large data processing unit, and the entropy encoding unit 139.

[0232] The in-loop filtering unit 2007 considering the large data processing unit according to the fifth embodiment in the group encoding unit 130 may determine whether a boundary of a lower data processing unit in a large data processing unit is filtered, and adjust the degree of the filtering.

[0233] The in-loop filtering unit 2007 considering the large data processing unit according to the fifth embodiment in the group encoding unit 130 may perform in-loop filtering for reducing a ringing effect generated within the large data processing unit.

[0234] FIG. 20F is a block diagram of a case where entropy encoding considering a large data processing unit is applied to the group encoding unit 130 of the image encoding apparatus 100.

[0235] A sixth embodiment of the group encoding unit 130 of the image encoding apparatus 100 includes the frequency transformation and quantization unit 131, the motion estimation unit 133 including an intra prediction unit and an inter prediction unit, the in-loop filtering unit 137, and an entropy encoding unit 2009 considering a large data processing unit.

[0236] The entropy encoding unit 2009 considering the large data processing unit according to the sixth embodiment in the group encoding unit 130 defines a coded block pattern (CBP) for the large data processing unit. The entropy encoding unit 2009 may also scan the CBP based on the probability that a CBP, which is not 0, exists within a data processing unit.

[0237] The entropy encoding unit 2009 considering the large data processing unit according to the sixth embodiment in the group encoding unit 130 may apply context-based adaptive binary arithmetic coding (CABAC) or context-based adaptive variable length coding (CAVLC) to a large data processing unit.

[0238] The entropy encoding unit 2009 considering the large data processing unit according to the sixth embodiment in

the group encoding unit 130 may set a new context of a CBP for a large data processing unit.

[0239] The first through sixth embodiments of the group encoding unit 130 are unique processes that can be performed considering a large data processing unit for each encoding process. Hereinafter, a method of encoding an image sequence based on a large block from among large data processing units on the entire image encoding process will be described with reference to FIG. 20G.

[0240] FIG. 20G illustrates an outline 2000 of image encoding methods that consider a data processing unit dynamically determined for a large data processing unit according to an embodiment, according to an embodiment of the present invention.

[0241] According to the outline 2000 of image encoding methods, an input image sequence 2010 is encoded by using a block corresponding to a predetermined encoding mode from a hierarchical structure 2020 of blocks. The hierarchical structure 2020 of blocks is the same as that of groups, subgroups, and basic blocks described above with reference to FIG. 10.

[0242] According to the outline 2000 of image encoding methods, an image bitstream output operation s2098 is performed after a frequency transformation and quantization operation s2030, a motion estimation operation s2040, an in-loop filtering operation s2080, an entropy encoding operation s2090. The motion estimation operation s2040 includes an intra prediction operation s2060 and an inter prediction operation s2070 which are based on predetermined processing 2050.

[0243] Since the image encoding method s2000 considers a large block, a unique process considering the large block may be performed in the frequency transformation and quantization operation s2030, the processing 2050 for motion estimation, the intra prediction operation s2060, the inter prediction operation s2070, the in-loop filtering operation s2080, and the entropy encoding operation s2090.

[0244] For example, the unique process considering the large block in the frequency transformation and quantization operation s2030 may include at least one selected from the group consisting of integer discrete cosine transformation (DCT) and quantization 2032 based on the large block, DCT and quantization 2034 with respect to a chroma component of the large block, and a low-complexity DCT 2036 based on the large block.

[0245] The unique process considering the large block in the processing 2050 for motion estimation may include at least one selected from the group consisting of a combination 2052 of a large block for motion estimation and a large block for frequency transformation, a block scanning method and neighbor availability 2054 for motion estimation, and a simplification mode 2056 of a hierarchical structure large block.

[0246] The unique process considering the large block in the intra prediction operation s2060 may include at least one selected from the group consisting of introduction 2062 of an intra mode based on the large block, introduction 2064 of a complex plain for intra prediction, and determination 2066 of a predictable intra mode.

[0247] The unique process considering the large block in the inter prediction operation s2070 may include at least one selected from the group consisting of introduction 2072 of a prediction mode in which an intra mode and an inter mode are mixed, an operation 2074 of determining an MV of the large block and inducing an MVP, and introduction 2076 of a new skip mode and a skip group of the large group.

[0248] The unique process considering the large block in the in-loop filtering operation s2080 may include at least one selected from the group consisting of determination 2082 of filtering or non-filtering of a boundary within the large block and the degree of the filtering, and filtering 2084 for reduction of the ringing effect generated in the large block.

[0249] The unique process considering the large block in the entropy encoding operation s2090 may include at least one selected from the group consisting of an operation 2092 corresponding to a CBP for the large block and an CBP scanning method, CABAC or CAVLC 2094 both considering the large block, and a method 2096 of setting a context for the CBP of the large block.

[0250] In the image encoding apparatus 100 or the outline 2000 of image encoding methods both considering a large data processing unit, described above with reference to FIGS. 20A through 20G, an image can be encoded by adding or replacing unique processes considering a large data processing unit, without greatly changing an existing image encoding format.

[0251] Although only image encoding has been described above with reference to FIGS. 20A through 20G, the unique processes considering a large data processing unit may be applied to image decoding corresponding to the image encoding. Accordingly, a unique process for each operation which will be described below may be applied not only to image encoding but also to image decoding.

[III. Motion prediction processing]

[0252] Unique processing of the motion estimation unit 133 of the group encoding unit 130 of the image encoding apparatus 100, and the processing 2050 for motion estimation in the image encoding method 2000 will now be described in greater detail. The processing 2050 for motion estimation, which will be described below, may also be applied to an image decoding apparatus corresponding to the image encoding apparatus 100.

[0253] The processing 2050 for motion estimation may be characterized by the combination 2052 of a large data processing unit for motion estimation and a data processing unit for frequency transformation, the block scanning method and neighbor availability 2054 for motion estimation, and the simplification mode 2056 of a large block.

[0254] First, as the data processing unit for motion estimation is enlarged to be equal to or greater than an existing 16×16 or 8×8 macroblock, the data processing unit for frequency transformation may be enlarged in proportion to the data processing unit for motion estimation, so as to be equal to or greater than a maximum size of the data processing unit for motion estimation.

[0255] In other words, when the data processing unit for motion estimation is a 32×32 or 16×16 macroblock, the data processing unit for frequency transformation may be combined into a 32×32 block, a 16×16 block, a 8×8 block, or a 4×4 block according to the data processing unit for motion estimation.

[0256] Second, the scanning method and neighbor availability of a neighboring block 2054 will be described below with reference to FIGS. 21 and 22.

[0257] FIG. 21 illustrates a scanning method for each data processing unit to perform motion estimation that considers a large data processing unit according to an embodiment of the present invention. As the scanning method in units of super macroblocks has been described above with reference to FIG. 13, a large data processing unit within the frame 1320 is scanned in a zigzag scanning method, so that an encoding mode is determined.

[0258] As in FIG. 13, super macroblocks within the frame 1320 may be arranged parallel to each other in vertical and horizontal directions, so that blocks are arranged in units of 64×64 large blocks. The scanning method for motion estimation according to an embodiment may also be applied to a case where super macroblocks are not arranged within a frame 2100 as in FIG. 21, besides a case where super macroblocks are arranged within the frame 1320 as in FIG. 13.

[0259] For example, a first scan target 2110 of the frame 2100 may not be a square-shaped super macroblock. Super macroblocks 2140 having square shapes, which are scanned in a zigzag manner, may not necessarily be arranged parallel to each other. A super macroblock 2120 and a super macroblock 2130 are adjacent to each other but are not arranged in a line.

[0260] Although not scanned in units of square-shaped super macroblocks 32×32 S-MB and 32×16/16×32 S-MB, scanning may be performed in units of 16×16 macroblocks 2140 at a location where scanning may be performed in units of a rectangular super macroblock 32×16/16×32 S-MB_half.

[0261] When scan target blocks are arranged as in the frame 1320 of FIG. 13, a neighboring block may be used as a reference block in the same method as an existing method related with a neighboring block estimation and context. However, when no scan target blocks are arranged as in the frame 2100 of FIG. 21, since a reference block varies according to circumstances, an MVP or an intra predictor of a current block may be set to indicate a block that is closest to the current block and has the same size as the current block.

[0262] In a simplification mode where only square-shaped blocks 32×32 S-MB, 32×16/16×32 S-MB, and 16×16 MB can be set, calculation complexity and signaling that are additionally generated may be controlled.

[0263] FIG. 22 illustrates availability of a neighboring data processing unit in motion estimation considering a large data processing unit according to an embodiment of the present invention. An existing 16×16 macroblock 2200 and a 32×32 super macroblock 2250 are used as examples of a data processing unit.

[0264] During MVP and intra prediction, whether a neighboring block located on the right upper side of a current block can be used as a reference block (hereinafter, referred to as neighbor availability) may be determined according to a location index unit and whether the neighboring block has already been scanned.

[0265] The neighbor availability will now be described with reference to FIG. 22, illustrating a case where a square block is used as a data processing unit. The blocks included in a frame are classified into a first block group 2202 capable of referring to right upper neighbor blocks in units of 4×4 blocks, being a minimum data unit, a second block group 2204 incapable of referring to right upper neighbor blocks only in a 4×4 index case, a third block group 2206 incapable of referring to right upper neighbor blocks only in a 8×8 index case, and a fourth block group 2208 incapable of referring to right upper neighbor blocks only in a 16×16 index case.

[0266] For example, in the existing 16×16 macroblock 2200, 4×4 blocks $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3, d_1$, and d_2 correspond to the first block group 2202 capable of referring to right upper neighbor blocks. 4×4 blocks a_4, b_4, c_4 , and d_4 correspond to the second block group 2204 incapable of referring to right upper neighbor blocks only in a 4×4 index case. A 4×4 blocks d_2 corresponds to the third block group 2206 incapable of referring to right upper neighbor blocks only in a 8×8 index case.

[0267] Accordingly, in the existing 16×16 macroblock 2200, the blocks may be classified into the first block group 2202, the second block group 2204, and the third block group 2206 in relation to the neighbor availability.

[0268] When a large data processing unit according to an embodiment is used, an 16×16 index may be added in addition to a 4×4 or 8×8 index in order to determine a reference block. Accordingly, the fourth block group 2208 incapable of referring to right upper neighbor blocks only in a 16×16 index case may exist.

[0269] In the 32×32 super macroblock 2250, 4×4 blocks $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, D_1, D_2, E_1, E_2, E_3, F_1, F_2, F_3, G_1, G_2, G_3, H_1, H_2, H_3, I_1, I_2, I_3, J_1, J_2, J_3, K_1, K_2, K_3, L_1, L_2, L_3, M_1, M_2, M_3, N_1, N_2, N_3, O_1, O_2, O_3, P_1$, and P_2 correspond

to the first block group 2202 capable of referring to right upper neighbor blocks.

[0270] 4×4 blocks A₄, B₄, C₄, D₄, E₄, F₄, G₄, H₄, I₄, J₄, K₄, L₄, M₄, N₄, O₄ and P₄ correspond to the second block group 2204 incapable of referring to right upper neighbor blocks only in a 4×4 index case. 4×4 blocks D₂, H₂, L₂ and P₂ correspond to the third block group 2206 incapable of referring to right upper neighbor blocks only in a 8×8 index case.

[0271] A 4×4 blocks N₂ corresponds to the fourth block group 2208 incapable of referring to right upper neighbor blocks only in a 16×16 index case.

[0272] The neighbor availability of right upper neighboring blocks may be used as a condition for determining an MVP or an intra predictor, or as input parameters of an induction function of an MVP or an intra predictor. Accordingly, when a large data processing unit is used according to an embodiment, the neighbor availability of right upper neighboring blocks by the 16×16 index may also be used as the condition for determining an MVP or an intra predictor or as an input parameter of the induction function of the MVP or the intra predictor.

[0273] Third, an simplified encoding mode using only a square-shaped data processing unit will now be described with reference to FIGS. 23A and 23B.

[0274] FIG. 23A illustrates a simplified hierarchical data processing unit from among large data processing units according to an embodiment of the present invention.

[0275] According to the hierarchical structure 1000 of a data processing unit described above in FIG. 10, a data processing unit according to an embodiment includes both square blocks 32×32, 16×16, 8×8, and 4×4 and rectangular blocks 32×16, 16×32, 16×8, 8×16, 8×4, and 4×8. According to the hierarchical structure 1000 of the data processing unit, encoding based on various data processing units may be performed.

[0276] The image encoding apparatus 100 may be simplified into a hierarchical structure 2300 of data processing units including only square blocks.

[0277] A first layer 2310 may include a 32×32 super macroblock of an intra mode, a 32×32 super macroblock of a skip mode, a 32×32 super macroblock of an inter mode, and a 16×16 macroblock. The first layer 2310 may be frequency-transformed in units of at least one selected from the group consisting of a 32×32 block, a 16×16 block, a 8×8 block, and a 4×4 block.

[0278] A second layer 2320 may include a 16×16 super macroblock of an intra mode, a 16×16 super macroblock of a skip mode, a 16×16 super macroblock of an inter mode, and a 8×8 macroblock of an inter mode. The second layer 2320 may be frequency-transformed in units of at least one selected from the group consisting of a 16×16 block, a 8×8 block, and a 4×4 block.

[0279] A third layer 2330 may include a 8×8 super macroblock of a skip mode, a 8×8 super macroblock of an inter mode, and a 4×4 block of an inter mode. The third layer 2330 may be frequency-transformed in units of at least one selected from the group consisting of a 8×8 block and a 4×4 block.

[0280] According to the simplified hierarchical structure 2300 of data processing units, an inter mode block and an intra mode block both having the same size and the same shape exist. As the number of encoding modes decreases, bits representing the encoding modes may decrease.

[0281] FIG. 23B illustrates a syntax 2350 considering a large data processing unit according to an embodiment of the present invention. The syntax 2350 includes various pieces of information about an encoding mode of a predetermined data processing unit.

[0282] A value 'mode' 2352 indicates the encoding mode. A value 'ref_idx/mvd' 2354 indicates a reference index and a deviation of an MV. A value 'coded_block pattern' 2356 indicates a CBP. A value 'transform_size_flag' 2358 indicates the size of a data processing unit for frequency transformation. A value 'residual()' 2360 indicates information about a residual component.

[0283] According to the simplified hierarchical structure 2300 of data processing units, a bit for the value 'mode' 2352 may be saved. When the size of the data processing unit for frequency transformation based on the encoding mode is restricted, the value 'coded_block_pattern' 2356 and the value 'transform_size_flag' 2358 in the syntax 2350 may be changed.

[III-1. Intra prediction]

[0284] The intra prediction unit 2004 of the image encoding apparatus 100, and the intra estimation operation s2060 of the image encoding method 2000 may use a unique process considering a large data processing unit.

[0285] First, an intra mode 2062 based on the large block is included in the intra prediction operation s2060. For example, a 32×32 super macroblock of an intra mode and a 16×16 macroblock of an intra mode may be added.

[0286] Second, a prediction mode using a complex plain for delicate intra prediction may be used.

[0287] Third, an encoding mode representing the possibility of intra prediction may be determined for each data processing unit.

[III-2. Interprediction]

[0288] The inter prediction unit 2005 of the image encoding apparatus 100, and the inter prediction operation s2070 of the image encoding method 2000 may use a unique process considering a large data processing unit.

5 [0289] First, a prediction mode in which an intra mode and an inter mode are mixed may be set for an arbitrary large data processing unit. Also, an intra mode and an inter mode may be mixed and set for each lower data processing unit in a large data processing unit.

[0290] Second, a skip mode for a large data processing unit may be newly defined, and a skip group including a data processing unit of at least one skip mode may be used. For example, a 32×32 super macroblock of a skip mode, and

10 [0291] FIGS. 24A, 24B, and 24C illustrate several methods of constructing a skip group, according to an embodiment of the present invention. For convenience of explanation, a 32×32 super macroblock is used as the large data processing units of FIGS. 24A, 24B, and 24C.

[0292] In image encoding and image decoding according to an embodiment, adjacent skip mode blocks may be bound into a skip group and thus skipping may be performed in units of skip groups. Accordingly, when a skip group is read while a frame is being scanned, it needs to be determined whether neighboring blocks adjacent to the skip group are skip mode blocks included in the skip group.

15 [0293] Accordingly, a mark indicating whether a predetermined block is a skip mode block and is included in a skip group is needed. In the embodiment illustrated in FIGS. 24A, 24B, and 24C, a skip mode flag is used as the mark indicating whether a predetermined block is a skip mode block and is included in a skip group. When the skip mode flag is '0', a corresponding block is not a skip mode block. When the skip mode flag is '2', the corresponding block is the start block of a skip group. When the skip mode flag is '1', the corresponding block is included in the skip group.

[0294] In relation to a skip group according to an embodiment, the skip group may be located to search for a skip mode block from the frame and scan the skip group. Moreover, the maximum size of a skip group including adjacent skip mode blocks may be restricted.

20 [0295] Accordingly, in inter prediction based on a large data processing unit according to an embodiment, a case where the position and size of a skip group are fixed (a frame 2400 of FIG. 24A), a case where the position of a skip group is fixed and the skip group is formed in an arbitrary size (a frame 2440 of FIG. 24B), and a case where a skip group may be positioned at any location and may be formed in any size (a frame 2470 of FIG. 24C) may be considered to perform skip processing.

25 [0296] In the first frame 2400 of FIG. 24A, the size of a skip group is fixed to a 64×64 block. Accordingly, to scan a scan group, only a 32×32 super macroblock located on the left upper side of each 64×64 block needs to be checked to determine whether the 32×32 super macroblock is a skip mode block. For example, since the skip mode flag of a 32×32 super macroblock 2412 is '0', the 32×32 super macroblock 2412 is not a skip mode block, and a 64×64 block 2410 including the super macroblock 2412 is not a skip group.

30 [0297] Next, when it is determined from scanning of a super macroblock located on the right upper side of a 64×64 block 2420 that the skip mode flag thereof is '0', the 64×64 block 2420 is not a skip group either.

[0298] Since the skip mode flag of a super macroblock 2432 is '2', the super macroblock 2432 is the start block of a skip group. Since the size of a skip group of the first frame 2400 is fixed to the 64×64 block size, a 64×64 block 2430 including the super macroblock 2432 at the left upper side thereof is a skip group. The other super macroblocks 2434 and the like in the 64×64 block 2430 may be predicted to be skip mode blocks each having a skip mode flag of '1'.

35 [0299] In a skip group having a fixed position and a fixed size illustrated in FIG. 24A, once it is determined whether the skip mode flag of a 32×32 block at a fixed location is '0' or '2', it may also be determined whether neighboring blocks are skip mode blocks. Thus, when a decoder receives only the skip mode flag of a corresponding block at a fixed location from an encoder, the decoder may sufficiently ascertain whether all of the blocks of a corresponding frame are skip mode blocks by using the received skip mode flag.

40 [0300] In the frame 2440 of FIG. 24B, the size of a skip group is not fixed, but a location where the skip group can start is fixed. In other words, since a skip group may start from a 32×32 block located on the left upper side of each 64×64 block, when the frame 2440 is scanned, it has only to be determined whether the skip mode flag of a 32×32 block located on the left upper side of each 64×64 block is '0' or '2'. For example, when a 32×32 block located on the left upper side of a 64×64 block 2450 is scanned to have a skip mode flag of '0', the 64×64 block 2450 is not a skip group.

45 [0301] When a 32×32 block 2461 located on the left upper side of another 64×64 block is scanned, the scan mode flag of the 32×32 block 2461 is '2', and thus a 64×64 block 2460 starting from the 32×32 block 2461 should be determined. When 32×32 blocks adjacent to the 32×32 block 2461 rightwards and downwards are scanned, 32×32 blocks 2462, 2463, 2464, 2465, 2466, 2467, 2468, and 2469 consecutively have skip mode flags of '1'. Thus, they are included in the skip group 2460 in which the 32×32 block 2461 is included.

50 [0302] Although the skip mode flags of 32×32 blocks at fixed positions are supposed to be checked during scanning of the frame 2440, blocks that have been already scanned and determined to be included in a skip group are not scanned

again. For example, since 32×32 blocks 2463, 2467, and 2469 are located on the respective left upper sides of 64×64 blocks, they are position-fixed blocks which are to be scanned, but they have already been checked as blocks included in the skip group 2460 according to a scanning sequence. Accordingly, scanning of the 32×32 blocks 2463, 2467, and 2469 is skipped, and scanning is performed on the next block.

[0303] In a skip group having a fixed position and any size illustrated in FIG. 24B, the skip mode flag of a 32×32 block at a fixed location and the size of a skip group may be transmitted from an encoder and received by a decoder. The decoder may determine whether a corresponding block is a skip mode block, by reading the skip mode flags of the blocks of a corresponding frame. However, the skip mode flag of a block that is indicated as a block included in another skip block through the skip mode flag of '1' although it is a block having a fixed position does not need to be transmitted and received.

[0304] In the frame 2470 of FIG. 24C, the size of a skip group is not fixed, and a location where the skip group can start is not fixed either. Accordingly, when the frame 2470 is scanned, the skip mode flags of all 32×32 blocks need to be checked.

[0305] For example, since the skip mode flag of a 32×32 block 2472 is '0', a 32×32 block 2474 right next to the 32×32 block 2472 is then scanned. During scanning, all of the 32×32 blocks are sequentially scanned until a 32×32 block having a skip mode flag of '2' is found.

[0306] When a 32×32 block 2482 having a skip mode flag of '2' is scanned, a skip group 2480 starting from the 32×32 block 2482 should be determined. Consecutive 32×32 blocks checked to have skip mode flags of '1' by scanning 32×32 blocks adjacent to the 32×32 block 2482 rightwards and downwards may be all included in the skip group 2480 in which the 32×32 block 2482 is included.

[0307] After the skip group 2480 is determined, scanning of the 32×32 blocks included in the skip group 2480 is skipped, and scanning continues on a 32×32 block 2492 not included in the skip group 2480. Similarly, a skip group 2490 starting from the 32×32 block 2492 may be determined.

[0308] In a skip group having any position and any size illustrated in FIG. 24C, the skip mode flags of 32×32 blocks at all locations and the size of a skip group may be transmitted from an encoder and received by a decoder. The decoder may determine whether each block in a corresponding frame is a skip mode block, by using the skip mode flags of all of the blocks of the corresponding frame. However, the skip mode flag of a 32×32 block checked to be included in a skip group other than the skip group does not need to be transmitted and received.

[0309] Accordingly, since skip processing may be performed on skip mode blocks in units of groups according the various embodiments described above with reference to FIGS. 24A, 24B, and 24C, the number of bits for representing the skip modes of blocks may be reduced, and the skip modes may be efficiently searched for without needing to scan all blocks.

[0310] Third, a MV depending on inter prediction may be determined in units of large data processing units, and an MVP may also be induced in units of large data processing units.

[0311] FIGS. 25A and 25B are block diagrams for illustrating a method of inducing a MV of a large data processing unit according to an embodiment of the present invention, and a method of inducing a MV of a lower data processing unit of the large data processing unit.

[0312] A MV and an MVP generated by inter prediction according to an embodiment are induced in units of large data processing units. However, FIGS. 25A and 25B illustrate a method of inducing an MV of a large data processing unit (a group of 16×16 macroblocks, namely, a 32×32 block) by using the MV of an existing data processing unit (a 16×16 macroblock), without adding an encoding mode for each large data processing unit.

[0313] For example, in inter prediction of a 32×32 super macroblock 2500 being a large data processing unit according to an embodiment, a motion vector mv_s of a 32×32 super macroblock corresponding to an inter mode is estimated or induced.

[0314] In image encoding or image decoding where an encoding mode of a large data processing unit is not individually defined, a motion vector of a super macroblock may be induced using the MV of a macroblock.

[0315] For example, when the MV_s of 16×16 macroblocks 2502, 2504, 2506, and 2508 are mv_0 , mv_1 , mv_2 , and mv_3 , respectively, the MVs mv_1 , mv_2 , and mv_3 may be induced using the MV mv_0 . For example, the mv_1 , mv_2 , and mv_3 may be estimated using Equation 6.

【Equation 6】

$$mv_1 = mv_2 = mv_3 = f(mv_0), \text{ (where } f(x) = \text{median}(x, \text{neighbors}) \text{)}$$

where a function $\text{median}(x, y)$ denotes a function for obtaining a median value of x and y and an input parameter 'neighbors' denotes a representative MV of a neighboring large data processing unit. The MV of a 16×16 macroblock may be estimated as a median value of the MV mv_0 of a representative 16×16 macroblock 2502 and a representative

MV of a neighboring 32×32 block.

[0316] Accordingly, in the case of FIG. 25A, without needing to measure all MVs of 16×16 macroblocks, a single MV may be considered as the MV of each 16×16 macroblock, and the MV mv_s of each 32×32 super macroblock may be estimated from the MV of each 16×16 macroblock.

5 [0317] In the case of FIG. 25A, an additional flag for signaling the MV mv_0 of a macroblock (or the MV mv_s of a 32×32 block) needs to be transmitted between an encoder and a decoder in order to achieve signaling of the MV of a 32×32 block. The decoder may induce the other MVs mv_1 , mv_2 , and mv_3 not signaled, through three inducing processes using received MVs.

10 [0318] FIG. 25B proposes another method of estimating an MV of a 32×32 block by using the MV of a 16×16 macroblock. In another estimating method, a 32×32 block 2510 is sub-sampled into 16×16 macroblocks so as to estimate the MV of the 32×32 block 2510 by using the MVs of the 16×16 macroblocks.

[0319] For example, a left upper pixel 2512, a right upper pixel 2514, a left lower pixel 2516, and a right lower pixel 2518 may be sequentially mapped into a left upper macroblock 2522, a right upper macroblock 2524, a left lower macroblock 2526, and a right lower macroblock 2528, respectively, thereby generating the sub-sampled 16×16 macroblocks 2522, 2524, 2526, and 2528.

15 [0320] The MVs mv_0' , mv_1' , mv_2' , and mv_3' of the sub-sampled 16×16 macroblocks 2522, 2524, 2526, and 2528 may be induced as follows.

20 [Equation 7]

$$mv_1' = mv_2' = mv_3' = mv_0'$$

25 [0321] Accordingly, when only one of the MVs mv_0' , mv_1' , mv_2' , and mv_3' of the 16×16 macroblocks 2522, 2524, 2526, and 2528 is measured, the other MVs may be estimated.

[0322] The MV of the 32×32 block 2510 may be estimated as a representative MV $MV (= mv_1' = mv_2' = mv_3' = mv_0')$ of the 16×16 macroblocks 2522, 2524, 2526, and 2528. Alternatively, the MV may be estimated using an MVP using the MVs mv_1' , mv_2' , mv_3' , and mv_0' . Accordingly, in the case of FIG. 25B, an additional flag for signaling the MV of a macroblock (mv_0' , mv_1' , mv_2' , or mv_3') needs to be transmitted between an encoder and a decoder in order to achieve signaling of the MV of a 32×32 block.

[IV. frequency transformation and quantization]

35 [0323] The frequency transformation and quantization unit 2001 of the image encoding apparatus 100, and the frequency transformation and quantization operation S2030 of the image encoding method 2000 may use a unique process considering a large data processing unit.

[0324] In image encoding or image decoding according to an embodiment, since the size of a data processing unit for each encoding mode may be greater than the size of an existing macroblock, data processing units other than a 8×8 or 4×4 block, which are data processing units for existing frequency transformation, may be used.

40 [0325] Accordingly, in the image encoding or image decoding according to an embodiment, (i-1) integer DCT and quantization based on a large block other than the 8×8 or 4×4 block may be used, and (i-2) the size of a data processing unit may be changed according to a luma component in consideration of a color standard in DCT on a chroma component based on a large block. Moreover, (i-3) low-complexity DCT may be modified and used to reduce the number of calculations generated by the DCT based on a large block.

45 [0326] FIGS. 26A and 26B are block diagrams for illustrating several motion estimation methods for a chroma component of a large data processing unit according to an embodiment of the present invention. The DCT 2034 with respect to a chroma component based on a large block (i-2) will be described later. A YUV color standard format includes a YUV 420 format and a YUV 444 format. FIG. 26A illustrates a frequency transformation method for each prediction mode with respect to the YUV 420 format, and FIG. 26B illustrates a frequency transformation method for each prediction mode with respect to the YUV 444 format. A case where a data processing unit for each prediction mode with respect to a luma component is a 32×32 super macroblock will be illustrated.

[0327] A block diagram 2600 of FIG. 26A corresponds to an embodiment of the frequency transformation and quantization unit 2001 considering a large data processing unit, and illustrates a frequency transformation method that can be applied for each prediction mode to a chroma component with the YUV 420 format.

55 [0328] Since a ratio of the numbers of bits allocated to a Y component, a U component, and a V component, respectively, is 4:1:1 in the YUV 420 format, the number of bits allocated to a chroma component corresponding to each of the U and V components is a quarter or less the number of bits allocated to a luma component corresponding to the Y component. Accordingly, the size of a data processing unit for a chroma component may be a quarter or less that of a data processing

unit for a luma component. For example, the maximum size of a data processing unit for frequency transformation with respect to a chroma component may be a 16×16 block which is a quarter of a 32×32 block.

[0329] In an intra mode 2610, a new method (ii) in which 16×16 DCT 2618 is performed after 16×16 directional intra prediction 2616 may be added to an existing method (i) in which 4×4 DCT and 2×2 Hadamard transformation 2614, that is, 8×8 frequency transformation, is performed after 8×8 directional intra prediction 2612.

[0330] In an inter mode 2620, a new method (ii) in which 16×16 DCT 2626 is performed may be added to an existing method (i) in which 4×4 DCT and 2×2 Hadamard transformation 2624, that is, 8×8 frequency transformation, is performed after $N \times M$ block inter prediction 2622.

[0331] A chroma component of the large data processing unit according to an embodiment may be encoded through quantization 2630 and entropy encoding 2640 after motion prediction and frequency transformation based on the intra mode 2610 or the inter mode 2620.

[0332] A block diagram 2650 of FIG. 26B corresponds to another embodiment of the frequency transformation and quantization unit 2001 considering a large data processing unit, and illustrates a frequency transformation method that can be applied for each prediction mode to a chroma component with the YUV 444 format.

[0333] Since a ratio of the numbers of bits allocated to a Y component, a U component, and a V component, respectively, is 4:4:4 in the YUV 444 format, the number of bits allocated to each chroma component is equal to the number of bits allocated to a luma component. For example, since the maximum size of a data processing unit for a luma component is a 32×32 block size, the maximum size of a data processing unit for a chroma component may be a 32×32 block size.

[0334] In an intra mode 2660, a new method (ii) in which 16×16 DCT 2664 is performed after 16×16 directional intra prediction 2663, and a new method (iii) in which 32×32 DCT 2666 is performed after 32×32 directional intra prediction 2665 may be added to an existing method (i) in which 4×4 DCT and 2×2 Hadamard transformation 2662, that is, 8×8 frequency transformation, is performed after 8×8 directional intra prediction 2661.

[0335] In an inter mode 2670, a new method (ii) in which 16×16 DCT 2676 is performed may be added to an existing method (i) in which 4×4 DCT and 2×2 Hadamard transformation 2674, that is, 8×8 frequency transformation, is performed after $N \times M$ block inter prediction 2672.

[0336] A chroma component of the large data processing unit according to an embodiment may be encoded through quantization 2680 and entropy encoding 2690 after motion prediction and frequency transformation based on the intra mode 2660 or the inter mode 2670.

[0337] A table summarizing in detail a relationship between a prediction mode and a frequency transformation block according to the block diagrams 2600 and 2650 of FIGS. 26A and 26B is illustrated in FIG. 27. Thus, the size of a frequency transformation block for a chroma component for each prediction mode may be induced from the size of a frequency transformation block for a luma component for each prediction mode.

[0338] FIG. 27 is a table for illustrating a method of inducing the size of a data processing unit for a chroma component of a large data processing unit according to an embodiment of the present invention. The integer DCT and quantization 2032 based on a large block (i-1) and the DCT 2034 with respect to a chroma component based on a large block (i-2) will be described later.

[0339] In the YUV 420 format, the size of a frequency transformation block for a chroma component is a quarter or less that of a frequency transformation block for a luma component. However, the minimum size of a frequency transformation block is a 8×8 block.

[0340] In the YUV 420 format, a data processing unit for DCT for a luma component of an intra mode includes 32×32 , 16×16 , 8×8 , and 4×4 blocks. Accordingly, when a DCT block for a luma component of an intra mode with the YUV 420 format is a 32×32 block, a DCT block for a corresponding chroma component may be induced to a 16×16 block. Moreover, when the DCT block for a luma component of an intra mode with the YUV 420 format is a 16×16 , 8×8 , or 4×4 block, DCT blocks for a corresponding chroma component may be induced to a 8×8 block which is a minimum data unit.

[0341] In the YUV 420 format, a data processing unit for DCT for a luma component of an inter mode includes 16×16 , 8×8 , and 4×4 blocks. Accordingly, when a DCT block for a luma component of an inter mode with the YUV 420 format is a 16×16 , 8×8 , or 4×4 block, a DCT block for a corresponding chroma component may be induced to a 8×8 block which is a minimum data unit.

[0342] In the YUV 444 format, the size of a frequency transformation block for a chroma component may be equal to that of a frequency transformation block for a luma component, in the intra mode. Similarly, the minimum size of a frequency transformation block is a 8×8 block.

[0343] In the YUV 444 format, a data processing unit for DCT for a luma component of an intra mode includes 32×32 , 16×16 , 8×8 , and 4×4 blocks. Accordingly, when DCT blocks for a luma component of an intra mode with the YUV 444 format are a 32×32 block and a 16×16 block, DCT blocks for a corresponding chroma component may be induced to a 32×32 block and a 16×16 block, respectively.

[0344] Moreover, when DCT blocks for a luma component of an intra mode with the YUV 444 format are 8×8 and 4×4 blocks, DCT blocks for a corresponding chroma component may be induced to a 8×8 block which is a minimum

data unit.

[0345] In the YUV 444 format, a data processing unit for DCT for a luma component of an inter mode includes 16×16 , 8×8 , and 4×4 blocks. Accordingly, when a DCT block for a luma component of an inter mode with the YUV 444 format is a 16×16 block, a DCT block for a corresponding chroma component may be induced to a 16×16 block. Moreover, when DCT blocks for a luma component of an inter mode with the YUV 444 format are 8×8 and 4×4 blocks, DCT blocks for a corresponding chroma component may be induced to a 8×8 block which is a minimum data unit.

[0346] Accordingly, a data processing unit for DCT on a chroma component may be induced from a data processing unit for DCT on a luma component. Although frequency transformation for each prediction mode has been illustrated using only the YUV 420 format and the YUV 444 format from among the YUV color standard formats, image encoding or image decoding based on a data processing unit dynamically determined according to an embodiment is not limited to the YUV 420 format and the YUV 444 format.

[V. In-loop filtering]

[0347] The in-loop filtering unit 2007 of the image encoding apparatus 100, and the frequency transformation and quantization operation s2080 of the image encoding method 2000 may use a unique process considering a large data processing unit.

[0348] In the image encoding or image decoding according to an embodiment, since the size of a large data processing unit may be greater than the size of an existing macroblock, filtering of the inside of the large data processing unit needs to be considered.

[0349] Accordingly, a unique process associated with in-loop filtering considering a large data processing unit according to an embodiment may include a method of determining whether a boundary of a lower data processing unit included in a large data processing unit and determining the degree of the filtering of each boundary, and filtering for reducing the ringing effect caused by encoding of the large data processing unit.

[0350] Accordingly, the method of determining whether the boundary of the lower data processing unit of the large data processing unit and determining the degree of the filtering of each boundary, in the in-loop filtering considering a large data processing unit according to an embodiment, will now be described as follows by using an in-loop filtering-related flag used in in-loop filtering performed in existing image encoding or image decoding.

[0351] According to an existing method, it is determined whether in-loop filtering is performed on the boundary of each 16×16 or 8×8 macroblock. For example, in an existing H.264 image encoding/decoding format, an in-loop filtering-related flag is set. The in-loop filtering-related flag includes a first flag 'filterLeftMbEdgeFlag' indicating whether in-loop filtering is performed on a boundary between a current macroblock and a neighboring macroblock located on the left side of the current macroblock, and a second flag 'filterTopMbEdgeFlag' indicating whether in-loop filtering is performed on a boundary between the current macroblock and a neighboring macroblock located on the upper side of the current macroblock.

[0352] The in-loop filtering-related flag may also include a third flag 'filterInternalEdgeflag' indicating whether in-loop filtering is performed on the inside of a 16×16 or 8×8 macroblock, in order to reduce a block artifact generated by frequency transformation based on a 8×8 or 4×4 block within the 16×16 or 8×8 macroblock.

[0353] The in-loop filtering-related flag may also include a fourth flag 'chromaEdgeflag' indicating whether in-loop filtering is performed on a boundary of a block for a chroma component.

[0354] In the image encoding or image decoding according to an embodiment, since encoding is performed in units of data processing units larger than a 16×16 or 8×8 macroblock, whether in-loop filtering is performed on the inside of a large data processing unit should be added to conditions for performing in-loop filtering, in order to use an existing in-loop filtering-related flag.

[0355] For example, in the in-loop filtering performed in the image encoding or image decoding according to an embodiment, when a current boundary is within a large data processing unit, the first, second, and fourth flags may be set to be '0', and when the current boundary is outside the large data processing unit, the first, second, and fourth flags may be set to be '1'. In other words, an in-loop filtering-related flag may be set so that in-loop filtering is not performed on boundaries of macroblocks included in a large data processing unit.

[0356] Moreover, in the in-loop filtering performed in the image encoding or image decoding according to an embodiment, when frequency transformation is performed in units of 16×16 macroblocks, the third flag may be set to be '0'. In other words, when frequency transformation is performed in units of 16×16 macroblocks in an encoding mode of 16×16 macroblocks, an in-loop filtering-related flag may be set so that in-loop filtering for reducing a block artifact caused by frequency transformation is not performed on the inside of a 16×16 macroblock.

[0357] The in-loop filtering-related flag may also include a fifth flag 'mixedModeEdgeFlag' indicating whether in-loop filtering is performed on a boundary between neighboring blocks of different prediction modes, in the case of a macroblock of a prediction mode in which an intra mode and an inter mode are mixed.

[0358] In the in-loop filtering performed in the image encoding or image decoding according to an embodiment, since

neighboring blocks of an intra mode and an inter mode may exist within a large data processing unit, a case where a current data processing unit is the large data processing unit may be added to the conditions of the fifth flag. The degree of in-loop filtering may be set for each boundary of a lower data processing unit in the large data processing unit.

[0359] In the image encoding or image decoding according to an embodiment, since various processes are performed in units of large data processing units, a ringing effect may be generated in addition to the block artifact. In the in-loop filtering performed in the image encoding or image decoding according to an embodiment, filtering for reducing a ringing effect caused by encoding of the large data processing unit may be performed.

[VI. Entropy encoding]

[0360] The entropy encoding unit 2009 of the image encoding apparatus 100, and the entropy encoding operation s2090 of the image encoding method 2000 may use a unique process considering a large data processing unit.

[0361] The unique process for entropy encoding or decoding, considering the large data processing unit according to an embodiment, includes the operation 2092 corresponding to introduction of a hierarchical CBP considering the large data processing unit and an CBP scanning method, CABAC or CAVLC 2094 both considering the large data processing unit, and a single context for the CBP 2096 for the large data processing unit of an intra mode or a chroma component.

[0362] FIG. 28 is a flowchart illustrating a method of using a CBP that considers a large data processing unit according to an embodiment of the present invention.

[0363] Entropy encoding and entropy decoding according to an embodiment may use not only a CBP for a 8×8 block, CBP8, which is an existing CBP, but also a hierarchical CBP considering the large data processing unit.

[0364] For example, when an encoding mode of a data processing unit according to an embodiment is an encoding mode of a data processing unit that is equal to or greater than a 16×16 block inter mode such as an inter mode Inter_32x32 of a 32×32 super macroblock, an inter mode Inter_32x16 of a 32×16 super macroblock, an inter mode Inter_16x32 of a 16×32 super macroblock, and the like, a CBP for a 16×16 block, CBP16, is added. Accordingly, entropy encoding or decoding may be performed according to a hierarchical CBP including the CBP CBP16 and the CBP CBP8.

[0365] Accordingly, in operation 2800, encoding should be performed starting from the CBP CBP16 in the case of entropy encoding with respect to the encoding modes Inter_32x32, Inter_32x16, and Inter_16x32. First, in operation 2810, it is determined whether a CBP CBP16 for a luma component, Luma_CBP16, is '1'. When the CBP Luma_CBP16 is '0', texture processing for entropy encoding is concluded. When the CBP Luma_CBP16 is '1', a flag Tr_type indicating the size of a data processing unit is encoded in operation 2820, and then a CBP CBP8 is encoded in operation 2830. In operation 2840, texture processing for a luma component is performed.

[0366] In operation 2850, it is determined whether a CBP CBP16 for a chroma component, Chroma_CBP16, is '1'. In operation S2860, texture processing for the chroma component is performed.

[0367] As described above, in an encoding mode of a data processing unit equal to or greater than a 16×16 block inter mode according to an embodiment, the operation 2800 of performing entropy encoding by using the CBP CBP16 illustrates a case where the CBP CBP16 for a 16×16 block is added. However, even when the data processing unit is additionally deformed, a CBP deformed considering the data processing unit may exist.

[0368] FIGS. 29A and 29B illustrate several methods of scanning a CBP that considers a large data processing unit according to an embodiment of the present invention. The operation 2092 corresponding to a CBP scanning method considering the large data processing unit will now be described.

[0369] In entropy encoding according to an embodiment, a data processing unit 2900 is scanned to search for a CBP that is not '0'. In an existing scanning method, a square 8×8 block is scanned in an interleaved manner.

[0370] The entropy encoding according to an embodiment uses a method of scanning the data processing unit 2900 back and forth in a zigzag manner about a diagonal line from the left upper end of the data processing unit 2900 to the right lower end thereof. As a CBP is located closer to the left upper end of the data processing unit 2900, the possibility that a CBP is not 0 is higher. Thus, the efficiency of entropy encoding for encoding effective bits that are not 0 may be increased.

[0371] To obtain the efficiency of scanning CBPs according to an embodiment, the data processing unit 2900 may be partitioned by a plurality of boundaries 2910, 2920, 2930, 2940, 2950, 2960, and 2970. According to a zigzag scanning method, it may be determined whether a CBP which is not 0 (a location marked with 'x') exists in a strip of a region between boundaries, and only a strip including a CBP which is not 0 may be encoded.

[0372] In other words, in the method of scanning CBPs according to an embodiment, the data processing unit 2900 is scanned to find a strip of a region including a CBP which is not 0 from among a first region before the boundary 2910, a second region between the boundaries 2910 and 2920, a third region between the boundaries 2920 and 2930, a fourth region between the boundaries 2930 and 2940, a fifth region between the boundaries 2940 and 2950, a sixth region between the boundaries 2950 and 2960, a seventh region between the boundaries 2960 and 2970, and an eighth region after the boundary 2970 so as to encode the strip of a region including a CBP which is not 0.

[0373] Two methods of representing a strip of a region including a CBP which is not 0 in order to reduce the number

of transmission bits are illustrated in FIGS. 29A and 29B, respectively.

[0374] In FIG. 29A, an effective strip and a concluded strip are defined to indicate that only the data of at least one of the regions between the region of the effective strip and the region of the concluded strip is effective data. In FIG. 29B, a concluded strip may be defined to indicate that only the data of at least one of the regions until the region of the concluded strip is effective data.

[0375] In the case of FIG. 29A, flags representing the effective strip and the concluded strip have only to be defined on a syntax in order to determine CBP scan regions of the data processing unit 2900. Although an ineffective strip exists between the regions of the effective strip and the concluded strip, a flag representing the ineffective strip is not needed, and a flag representing a not-concluded strip is not needed either. Accordingly, only two bits for flags representing an effective strip and a concluded strip, respectively, in relation to a CBP are needed as the transmission bits of the single data processing unit 2900.

[0376] In the case of FIG. 29B, a flag representing the concluded strip has only to be defined on a syntax in order to determine CBP scan regions of the data processing unit 2900. A flag representing a not-concluded strip before the region of a concluded strip is not needed. Accordingly, only one bit for a flag representing a concluded strip in relation to a CBP is needed as the transmission bits of the single data processing unit 2900.

[0377] In the entropy encoding according to an embodiment, a new CBP for a large data processing unit of an intra mode or a chroma component, and a single context of the new CBP may be defined according to an encoding mode of the large data processing unit.

[0378] For example, the encoding mode of the large data processing unit according to an embodiment may newly include an intra mode Intra_32×32 of a 32×32 super macroblock, an intra mode Intra_16×16 of a 16×16 macroblock, and intra modes Cr_16×16 and Cb_16×16 for chroma components of a 16×16 macroblock.

[0379] A large data processing unit of an intra mode has a very low possibility that all of the internal data is 0. Accordingly, instead that the context of a CBP of a current data processing unit uses the context of a neighboring data processing unit, a single context for a CBP may be set for each large data processing unit.

[0380] The context of a CBP of an inter mode may be set in the same manner as the manner in which the context of a CBP for a 8×8 macroblock of an existing inter mode is set. Accordingly, the context of a current 8×8 macroblock may be determined using the context of a neighboring 8×8 macroblock of an inter mode.

[0381] In a decoder according to an embodiment, when an image bitstream encoded by entropy encoding based on a large data processing unit is decoded, entropy decoding considering the large data processing unit is needed. For example, the decoder according to an embodiment restores an image through entropy decoding using a CBP for each 16×16 macroblock.

[VII. Image decoding apparatus based on dynamically-determined data processing unit]

[0382] FIGS. 30A, 30B, 30C, 30D, 30E, and 30F are block diagrams of modified embodiments of the image decoding apparatus 200 that consider a data processing unit dynamically determined according to an embodiment.

[0383] To apply a large data processing unit to the image encoding apparatus 200, the group decoding unit 230 generates a restored image through entropy decoding, inverse quantization and inverse frequency transformation, intra prediction and inter prediction, and in-loop filtering by considering the large data processing unit.

[0384] The receiving unit 210 may receive an image bitstream from the image encoding apparatus 100, and the group encoding mode reading unit 220 may read an encoding mode from the image bitstream. The encoding mode may be read from a syntax, a flag, or the like. The group decoding unit 230 decodes group data based on the read-out encoding mode.

[0385] The group decoding unit 230 may perform a unique process distinguished from an existing image encoding and decoding format, in order to perform decoding considering the dynamically-determined data processing unit. Unique processes considering a large data processing unit from among dynamically-determined data processing units correspond to the techniques described above with reference to FIGS. 21 through 29B. A unique process for each operation of the group decoding unit 230 will be described in detail.

[0386] FIG. 30A is a block diagram of a case where entropy decoding considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0387] A first embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes an entropy encoding unit 3010 considering the large data processing unit, an intra prediction unit and motion compensation unit 233, an inverse quantization and inverse frequency transformation unit 237, and an in-loop filtering unit 239.

[0388] The entropy decoding unit 3010 considering a large data processing unit according to the first embodiment in the group decoding unit 230 may perform entropy decoding in units of the large data processing unit based on the encoding mode read out by the group encoding mode reading unit 220. The entropy decoding performed by the entropy decoding unit 3010, corresponds to decoding of a bitstream obtained by the entropy encoding described above in section [VI. Entropy encoding].

[0389] The CBP described above with reference to FIGS. 28 through 29B may be used in the entropy decoding unit 3010. The read-out CBP may be a result obtained by scanning in a direction where the possibility of generating a CBP which is not '0' increases and encoding the CBP which is not '0'.

[0390] The entropy decoding unit 3010 considering the large data processing unit according to the first embodiment in the group decoding unit 230 may apply CABAC or CAVLC to a large data processing unit in order to perform entropy decoding on the large data processing unit.

[0391] The entropy decoding unit 3010 considering the large data processing unit according to the first embodiment in the group decoding unit 230 may decode a context newly set for a CBP based on the large data processing unit.

[0392] FIG. 30B is a block diagram of a case where motion prediction considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0393] A second embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes an entropy decoding unit 231, an intra prediction unit and motion compensation unit 3030 following a processing considering a large data processing unit, an inverse quantization and inverse frequency transformation unit 237, and an in-loop filtering unit 239.

[0394] The intra prediction unit and motion compensation unit 3030 following the processing considering a large data processing unit according to the second embodiment in the group decoding unit 230 may perform motion prediction or compensation according to a process by the large data processing unit based on the encoding mode read out by the group encoding mode reading unit 220. The processing considering the large data processing unit according to an embodiment has been described above in section [III. Motion prediction processing].

[0395] The data processing unit according to an embodiment may be determined by a combination of a large data processing unit for motion prediction and a data processing unit for frequency transformation.

[0396] The intra prediction unit and motion compensation unit 3030 following the processing considering the large data processing unit according to the second embodiment in the group decoding unit 230 may use a scanning method for motion prediction or compensation for each type of data processing unit and neighbor availability of a neighboring block, which are determined in consideration of the large data processing unit. The scanning method for motion prediction or compensation for each data processing unit and the neighbor availability of a neighboring block have been described above with reference to FIGS. 13, 21, and 22.

[0397] The intra prediction unit and motion compensation unit 3030 following the processing considering the large data processing unit according to the second embodiment in the group decoding unit 230 may use the simplified encoding mode using only a square-shaped large data processing unit. An exemplary embodiment of the simplified encoding mode has been described above with reference to FIGS. 23A and 23B.

[0398] FIG. 30C is a block diagram of a case where intra prediction considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0399] A third embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes the entropy decoding unit 231, an intra prediction unit 3040 considering the large data processing unit, a motion compensation unit 235, the inverse quantization and inverse frequency transformation unit 237, and the in-loop filtering unit 239. When an image has been encoded according to the unique process according to an embodiment described above in section [III. Intra prediction], the image decoding apparatus 200 may perform decoding according to the process described above in section [III. Intra prediction].

[0400] The intra prediction unit 3040 considering the large data processing unit according the third embodiment in the group decoding unit 230 may use an intra mode added for a predetermined large data processing unit. For example, the intra prediction unit 3040 may use a 32×32 super macroblock of an intra mode, a 16×16 super macroblock of an intra mode, or the like.

[0401] The intra prediction unit 3040 considering the large data processing unit according the third embodiment in the group decoding unit 230 may perform a prediction mode using a complex plain into the intra mode, in order to perform more accurate intra prediction with respect to a large data processing unit.

[0402] The intra prediction unit 3040 considering the large data processing unit according the third embodiment in the group decoding unit 230 may perform intra prediction by using only a restricted type of large data processing unit.

[0403] FIG. 30D is a block diagram of a case where motion compensation considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0404] A fourth embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes the entropy decoding unit 231, an intra prediction unit 234, a motion compensation unit 3050 considering a large data processing unit, the inverse quantization and inverse frequency transformation unit 237, and the in-loop filtering unit 239. An image obtained by encoding in units of large data processing units according to an embodiment may undergo motion prediction and motion compensation according to the unique process according to an embodiment described above in section [III-2 Inter prediction].

[0405] The motion compensation unit 3050 considering the large data processing unit according the fourth embodiment in the group decoding unit 230 may use the combination mode in which inter prediction and intra prediction can be mixed,

as an prediction mode for the large data processing unit.

[0406] The motion compensation unit 3050 considering the large data processing unit according the fourth embodiment in the group decoding unit 230 may perform motion compensation by using an MV determined for each large data processing unit and an MVP.

[0407] The motion compensation unit 3050 considering the large data processing unit according the fourth embodiment in the group decoding unit 230 may use a new skip mode defined for each large data processing unit and may perform skip processing in units of skip groups including a plurality of data processing units of a plurality of skip modes.

[0408] FIG. 30E is a block diagram of a case where inverse quantization and inverse frequency transformation considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0409] A fifth embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes the entropy decoding unit 231, the intra prediction unit and motion compensation unit 233, an inverse quantization and inverse frequency transformation unit 3070 considering the large data processing unit, and the in-loop filtering unit 239. The inverse quantization and inverse frequency transformation unit 3070 considering the large data processing unit may perform an inverse operation of the frequency transformation and quantization according to an embodiment described above in section [IV. frequency transformation and quantization].

[0410] The inverse quantization and inverse frequency transformation unit 3070 considering the large data processing unit according to the fifth embodiment in the group decoding unit 230 may perform integer DCT and inverse quantization based on a large data processing unit, in order to perform inverse quantization and inverse frequency transformation by the large data processing unit.

[0411] The inverse quantization and inverse frequency transformation unit 3070 considering the large data processing unit according to the fifth embodiment in the group decoding unit 230 may perform inverse DCT and inverse quantization on a chroma component of the large data processing unit, in order to perform inverse frequency transformation and inverse quantization by the large data processing unit.

[0412] The inverse quantization and inverse frequency transformation unit 3070 considering the large data processing unit according to the fifth embodiment in the group decoding unit 230 may perform inverse DCT aiming at lowly complex calculation, in order to reduce the number of calculations that may increase due to the inverse DCT based on the large data processing unit.

[0413] FIG. 30F is a block diagram of a case where in-loop filtering considering a large data processing unit is applied to the group decoding unit 230 of the image decoding apparatus 200.

[0414] A sixth embodiment of the group decoding unit 230 of the image decoding apparatus 200 includes the entropy decoding unit 231, the intra prediction unit and motion compensation unit 233, the inverse quantization and inverse frequency transformation unit 237, and an in-loop filtering unit 3090 considering the large data processing unit. The in-loop filtering unit 3090 considering the large data processing unit may perform an operation corresponding to the in-loop filtering according to an embodiment described above in section [V. In-loop filtering].

[0415] The in-loop filtering unit 3090 considering the large data processing unit according the sixth embodiment in the group decoding unit 230 may determine whether filtering is performed on a boundary of a lower data processing unit in a large data processing unit. The degree of in-loop filtering may be adjusted for each boundary of the lower data processing unit in the large data processing unit.

[0416] The in-loop filtering unit 3090 considering the large data processing unit according the sixth embodiment in the group decoding unit 230 may perform in-loop filtering in order to reduce the ringing effect within the large data processing unit.

[0417] In the first through sixth embodiments of the group decoding unit 230 described above with reference to FIGS. 30A through 30F, the data of a group may be decoded using a data processing unit other than existing 16×16 , 8×8 , and 4×4 blocks according to the read-out encoding mode. In particular, various processes by a large data processing unit may be applied to an existing image encoding and decoding format by deforming the syntax of the existing image encoding and decoding format in consideration of a large data processing unit.

[VIII Image encoding method and image decoding method based on dynamically-determined data processing unit]

[0418] FIGS. 31A, 31B, 31C, 31D, 31E, and 31F are flowcharts of modified embodiments of the image encoding method 1800 that consider a data processing unit dynamically determined according to an embodiment.

[0419] In operation 1810 of each of FIGS. 31A, 31B, 31C, 31D, 31E, and 31F, image data classified into basic blocks are classified in an extended concept, namely, into groups, subgroups, and basic blocks. Also, in operation 1820, information about a data processing unit and an encoding mode of a group, which are used to encode the group, are determined. The encoding mode of the group may include an encoding mode considering a large data processing unit.

[0420] In operation 1830, frequency transformation and quantization, motion prediction, in-loop filtering, and entropy encoding may be performed to encode the group. A unique process for each operation for encoding the group will now be described in detail with reference to FIGS. 31A through 31F.

[0421] FIG. 31A is a flowchart of a case where frequency transformation and quantization considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0422] In operation 1831, the data of the group is encoded via frequency transformation and quantization based on an encoding mode considering the large data processing unit. To perform the frequency transformation and quantization considering the large data processing unit, at least one selected from the group consisting of the integer DCT and quantization 2032 based on a large block, the DCT and quantization 2034 with respect to a chroma component of the large block, and the low-complexity DCT 2036 based on the large block may be performed.

[0423] FIG. 31B is a flowchart of a case where motion estimation by a processing considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0424] In operation 1833, the data of the group is encoded via motion estimation processing based on the encoding mode considering the large data processing unit. To perform the motion estimation processing considering the large data processing unit, at least one selected from the group consisting of the combination 2052 of a large block for motion estimation and a large block for frequency transformation, the block scanning method and neighbor availability 2054 for motion estimation, and the simplification mode 2056 of a hierarchical structure large block may be performed.

[0425] FIG. 31C is a flowchart of a case where intra prediction considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0426] In operation 1834, the data of the group is encoded via intra prediction based on the encoding mode considering the large data processing unit. To perform the intra prediction considering the large data processing unit, at least one selected from the group consisting of the introduction 2062 of an intra mode based on a large block, the introduction 2064 of a complex plain for intra prediction, and the determination 2066 of a predictable intra mode may be performed.

[0427] FIG. 31D is a flowchart of a case where inter prediction considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0428] In operation 1835, the data of the group is encoded via inter prediction based on the encoding mode considering the large data processing unit. To perform the inter prediction considering the large data processing unit, at least one selected from the group consisting of the introduction 2072 of the prediction mode in which an intra mode and an inter mode are mixed, the operation 2074 of determining the MV of a large block and inducing an MVP, and the introduction 2076 of a new skip mode of the large group and a skip group may be performed.

[0429] FIG. 31E is a flowchart of a case where in-loop filtering considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0430] In operation 1837, the data of the group is encoded via in-loop filtering based on the encoding mode considering the large data processing unit. To perform the in-loop filtering considering the large data processing unit, at least one selected from the group consisting of the determination 2082 of filtering or non-filtering of a boundary within a large block and the degree of the filtering, and the filtering 2084 for reduction of the ringing effect generated in the large block may be performed.

[0431] FIG. 31F is a flowchart of a case where entropy encoding considering a large data processing unit is applied to the group encoding operation 1830 of the image encoding method 1800.

[0432] In operation 1839, the data of the group is encoded via entropy encoding based on the encoding mode considering the large data processing unit. To perform the in-loop filtering considering the large data processing unit, at least one selected from the group consisting of the operation 2092 corresponding to introduction of a CBP for a large block and an CBP scanning method, the introduction 2094 of CABAC or CAVLC both considering the large block, and the operation 2096 corresponding to a method of setting a context for the CBP for the large block may be performed.

[0433] FIGS. 32A, 32B, 32C, 32D, 32E, and 32F are flowcharts of modified embodiments of the image decoding method 1900 that consider a data processing unit dynamically determined according to an embodiment.

[0434] In operation 1910 of each of FIGS. 32A, 32B, 32C, 32D, 32E, and 32F, an image bitstream is received and parsed. The received image bitstream includes data obtained by encoding in units of basic blocks, groups, and subgroups. In operation 1920, a group encoding mode representing a method of encoding a group is read out. The group encoding mode may include a data processing unit of the group, and a motion prediction mode such as an intra mode, an inter mode, a skip mode, etc.

[0435] In operation 1930, entropy decoding, inverse frequency transformation and inverse quantization, motion compensation, and in-loop filtering may be performed to decode the group. A unique process for each operation for decoding the group will now be described in detail with reference to FIGS. 32A through 32F.

[0436] FIG. 32A is a flowchart of a case where entropy decoding considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0437] In operation 1931, the data of the group is decoded via entropy decoding considering the large data processing unit. To perform the entropy decoding considering the large data processing unit, a CBP based on the large data processing unit may be used. In the entropy decoding considering a large data processing unit, CABAC or CAVLC may be applied to the large data processing unit.

[0438] To perform the entropy decoding considering the large data processing unit, a CBP based on the large data

processing unit may be decoded using a context newly set for the CBP.

[0439] FIG. 32B is a flowchart of a case where motion compensation processing considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0440] In operation 1933, the data of a group is decoded via motion prediction or compensation performed according to a process considering the large data processing unit. The data processing unit according to an embodiment may be determined by a combination of a large data processing unit for motion prediction and a data processing unit for frequency transformation. The scanning method for motion prediction for each type of data processing unit and the neighbor availability of a neighboring block, which are determined in consideration of the large data processing unit, may be used.

[0441] In the processing considering a large data processing unit for intra prediction and motion compensation according to an embodiment, the simplified encoding mode using only a square large data processing unit may be used.

[0442] FIG. 32C is a flowchart of a case where intra prediction considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0443] In operation 1934, the data of the group is decoded via intra prediction based on the encoding mode considering the large data processing unit. The intra prediction operation considering the large data processing unit may use an intra mode added for a large data processing unit. A prediction mode using a complex plain may be used into the intra mode considering the large data processing unit. The intra prediction considering the large data processing unit may be performed on only a restricted type of large data processing unit.

[0444] FIG. 32D is a flowchart of a case where motion compensation considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0445] In operation 1935, the data of the group is decoded via motion compensation based on the encoding mode considering the large data processing unit. To perform the motion compensation considering the large data processing unit, a combination mode in which inter prediction and intra prediction can be mixed may be used as a prediction mode for the large data processing unit.

[0446] A MV and an MVP determined for each large data processing unit may be used to perform motion compensation. A new skip mode defined for each large data processing unit may be used, and skip processing may be performed in units of a skip group including data processing units of a plurality of skip modes.

[0447] FIG. 32E is a flowchart of a case where inverse quantization and inverse frequency transformation considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0448] In operation 1937, the data of the group is decoded via inverse quantization and inverse frequency transformation based on the encoding mode considering the large data processing unit.

[0449] To perform the inverse quantization and inverse frequency transformation considering the large data processing unit, integer inverse DCT and inverse quantization based on the large data processing unit may be performed. Integer inverse DCT and inverse quantization for the chroma component of a large data processing unit may be performed. Inverse DCT with low-complexity may be performed to reduce the number of calculations that may increase due to the inverse DCT based on the large data processing unit.

[0450] FIG. 32F is a flowchart of a case where in-loop filtering considering a large data processing unit is applied to the group decoding operation 1930 of the image decoding method 1900.

[0451] In operation 1939, the data of the group is decoded via in-loop filtering considering the large data processing unit.

[0452] To perform the in-loop filtering considering the large data processing unit, it may be determined whether a boundary of a lower data processing unit in a large data processing unit is filtered. The degree of in-loop filtering may be adjusted for each boundary of the lower data processing unit in the large data processing unit. The in-loop filtering may also be performed to reduce the ringing effect within the large data processing unit.

[0453] The embodiments of the present invention can be written as computer programs and can be implemented in general-use digital computers that execute the programs using a computer readable recording medium. Examples of the computer readable recording medium include magnetic storage media (e.g., ROM, floppy disks, hard disks, etc.), and optical recording media (e.g., CD-ROMs, or DVDs).

[0454] While the present invention has been particularly shown and described with reference to exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present invention as defined by the following claims.

Claims

1. An image encoding method comprising:

classifying image data divided into basic blocks, in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group; determining an encoding mode for a predetermined group in order to encode the predetermined group, wherein

the encoding mode represents a mode for encoding data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and encoding the data of the predetermined group according to the determined encoding mode.

- 5 2. The image encoding method of claim 1, further comprising encoding a flag that represents the encoding mode for the predetermined group.
3. The image encoding method of claim 1, wherein the classifying of the image data comprises determining a size of the predetermined group and a shape of the predetermined group, wherein the size represents the number of basic blocks included in the predetermined group and the shape is formed by the basic blocks included in the predetermined group.
- 10 4. The image encoding method of claim 1, wherein the classifying of the image data comprises determining subgroups for the predetermined group based on a level representing the number of subgroups included in the predetermined group and a scan sequence in which the basic blocks are scanned.
- 15 5. The image encoding method of claim 1, wherein the encoding mode comprises at least one of:
 - information about the type of data processing units in which the data of the predetermined group is encoded, wherein the type is one of a group, a subgroup, and a basic block;
 - information about the size of the data processing unit, wherein the size represents the number of basic blocks included in the data processing unit; and
 - information about a prediction mode that represents a mode for determining reference information corresponding to the data processing unit in order to perform motion estimation on the data processing unit.
- 20 6. The image encoding method of claim 1, wherein the determining of the encoding mode for the predetermined group comprises:
 - encoding the predetermined group according to at least one prediction mode in units of different types of data processing units which are groups, subgroups, and basic blocks;
 - determining error rates that are generated by encoding performed on the predetermined group in units of combinations of the different types of data processing units and the at least one prediction mode; and
 - determining a prediction mode for a predetermined data processing unit generating a smallest error rate from among the determined error rates.
- 25 7. The image encoding method of claim 6, wherein the determining of the encoding mode for the predetermined group comprises comparing encoding error rates generated by considering the at least one prediction mode in units of the different types of data processing units with one another and determining, as the encoding mode for the predetermined group, a combination of a data processing unit and an prediction mode, which generates the smallest error rate.
- 30 8. The image encoding method of claim 1, further comprising rearranging frequency-transformed coefficients of each group or each subgroup, in units of a plurality of basic blocks.
- 35 9. The image encoding method of claim 1, further comprising encoding a flag that represents a data processing unit of units in which the data of the predetermined group is frequency-transformed.
- 40 10. The image encoding method of claim 1, wherein in the classifying of the image data, the sizes, shapes, and maximum levels of groups are determined in units of a data unit selected from pictures, sequences, and groups of the image data.
- 45 11. The image encoding method of claim 1, wherein the encoding comprises:
 - performing frequency transformation and quantization for encoding of the input image on the basis of a large data processing unit corresponding to the determined group encoding mode;
 - performing motion estimation for the encoding of the input image, based on the large data processing unit;
 - performing in-loop filtering for the encoding of the input image, based on the large data processing unit; and
 - performing entropy encoding for the encoding of the input image, based on the large data processing unit.
- 50 12. The image encoding method of claim 11, wherein in the encoding, a size of a data processing unit for the frequency

transformation is determined based on a size of a data processing unit for the motion estimation.

13. The image encoding method of claim 11, wherein the performing of the motion estimation comprises performing intra prediction or inter prediction through motion estimation processing based on the large data processing unit, wherein the motion estimation processing based on the large data processing unit is one selected from the group consisting of the group consisting of:

a motion estimation process based on a scanning sequence for each determined data processing unit for sequential motion estimation based on the large data processing unit,
a motion estimation process in which neighbor availability of a neighboring data processing unit located on a right upper end of a current data processing unit is determined for motion estimation with respect to the input image to be performed based on the large data processing unit, and
a motion estimation process using a square-shaped data processing unit from among the data processing units, based on the large data processing unit.

14. The image encoding method of claim 11, wherein the performing of the motion estimation comprises performing intra prediction based on the large data processing unit, wherein the intra prediction based on the large data processing unit comprises at least one selected from the group consisting of:

intra prediction performed in units of the large data processing units in an intra mode of a complex plain type, intra prediction performed in units of predetermined data processing units so that motion estimation can be performed in an intra mode, based on the large data processing unit, and
intra prediction performed in a prediction mode in which an intra mode and an inter mode are mixed, based on the large data processing unit.

15. The image encoding method of claim 11, wherein the performing of the motion estimation comprises performing inter prediction based on the large data processing unit, wherein the inter prediction based on the large data processing unit comprises at least one selected from the group consisting of:

inter prediction in which a motion vector for motion estimation of an inter mode with respect to a current data processing unit is determined,
inter prediction in which a motion vector predictor corresponding to the motion vector is induced, based on the large data processing unit, and
inter prediction in which a skip mode of an extended shape is determined based on the large data processing unit and skip processing is performed in units of a skip group comprising adjacent data processing units of at least one skip mode.

16. The image encoding method of claim 11, wherein the performing of the frequency transformation and quantization comprises performing at least one selected from the group consisting of:

integer discrete cosine transformation based on a large block, based on the large data processing unit, frequency transformation with respect to a chroma component of a current data processing unit, based on the large data processing unit, and
modified frequency transformation for reducing calculation complexity of frequency transformation considering the large data processing unit.

17. The image encoding method of claim 11, wherein the performing of the entropy encoding comprises performing at least one selected from the group consisting of:

a coded block pattern hierarchically determined according to the size of each data processing unit based on the large data processing unit, by using quantized coefficients obtained by frequency-transforming and quantizing a current data processing unit,
scanning of a coded block pattern according to a sequence based on the possibility that a quantized coefficient which is not 0 from among the quantized coefficients of the current data processing unit exists;
Context-based Adaptive Binary Arithmetic Coding (CABAC) or Context-based Adaptive Variable Length Coding (CAVLC) modified based on the large data processing unit, and

a single context determined for the coded block pattern of the large data processing unit of an intra mode or a chroma component, based on the large data processing unit.

18. The image encoding method of claim 11, wherein in the performing of the in-loop filtering,
 5 it is determined based on the large data processing unit whether deblocking filtering is performed on at least one selected from the group consisting of a boundary of a lower data processing unit in the large data processing unit, a boundary of a lower data processing unit in the large data processing unit on which frequency-transformation has been performed in units of a predetermined data processing unit, a boundary of a chroma component of a lower data processing unit in the large data processing unit, and a boundary of a lower data processing unit in the large
 10 data processing unit on which motion estimation has been performed in a mixed mode in which an inter mode and an intra mode are mixed, the degree of deblocking filtering with respect to each boundary is set for a boundary of at least one lower data processing unit in the large data processing unit, and additional filtering for reducing a ringing effect is performed on a current data processing unit having a large size, based on the large data processing unit.
19. An image decoding method comprising:
 receiving and parsing image data encoded in units of at least one data processing unit selected from a basic
 20 block, a group, and a subgroup, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group;
 reading an encoding mode for a predetermined group from the parsed image data, wherein the encoding mode represents a mode for encoding image data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and
 25 decoding the image data of the predetermined group according to the read encoding mode.
20. The image decoding method of claim 19, wherein the reading of the encoding mode for the predetermined group comprises:
 30 reading information about a shape of a group set as a data processing unit in which decoding is performed, information about a size of a group representing the number of basic blocks included in the group, and information about a maximum level associated with a maximum number of subgroups included in the group; and
 determining the group which is the data processing unit by which decoding is performed, by classifying the
 35 parsed image data by using the information about the shape and size of the group.
21. The image decoding method of claim 20, wherein the groups are set in units of a data unit selected from a picture, a sequence, and a group of the image data.
22. The image decoding method of claim 19, wherein the subgroups are set based on a level representing the number
 40 of subgroups included in the group, and a scan sequence in which basic blocks included in the group are scanned.
23. The image decoding method of claim 19, wherein the reading of the encoding mode for the predetermined group comprises:
 45 extracting an encoding mode flag that represents the encoding mode for the predetermined group, from parsed data associated with the predetermined group; and
 reading the encoding mode for the predetermined group on the basis of the encoding mode flag.
24. The image decoding method of claim 19, wherein the encoding mode for the predetermined group comprises at
 50 least one of:
 information about the type of data processing units in which the data of the predetermined group is encoded, wherein the type is one of a group, a subgroup, and a basic block;
 information about the size of the data processing unit, wherein the size represents the number of basic blocks
 55 included in the data processing unit; and
 information about a prediction mode that represents a mode for determining reference information corresponding to the data processing unit in order to perform motion estimation on the data processing unit.

25. The image decoding method of claim 24, wherein the decoding of the data of the predetermined group comprises:
- determining a data processing unit in units in which decoding is to be performed, based on the information about the type and size of the data processing unit;
- 5 determining a prediction mode of the data processing unit on the basis of the information about the prediction mode; and
- decoding the data of the group according to the determined data processing unit and the determined prediction mode.
- 10 26. The image decoding method of claim 24, wherein:
- the predetermined group is encoded according to at least one prediction mode in units of different types of data processing units, namely, groups, subgroups, and basic blocks so as to calculate error rates for the at least one prediction mode;
- 15 a combination of a predetermined data processing unit and a prediction mode, which corresponds to a smallest error rate from among the calculated error rates, is determined; and
- encoding error rates generated by considering the at least one prediction mode in units of the different types of data processing units, are compared with one another so as to determine, as the encoding mode for the predetermined group, a combination of a data processing unit and a prediction mode, which correspond to the
- 20 smallest error rate.
27. The image decoding method of claim 19, wherein the basic blocks are decoded in a zigzag scan sequence.
28. The image decoding method of claim 19, further comprising rearranging frequency transformed coefficients of a plurality of basic blocks, in units of one of groups and subgroups.
- 25 29. The image decoding method of claim 20, further comprising reading, from the parsed data, a flag that represents a data processing unit of units in which the data included in the predetermined group is frequency transformed.
- 30 30. The image decoding method of claim 19, wherein the decoding comprises:
- performing entropy decoding for decoding of data of the group on the basis of a large data processing unit corresponding to the encoding mode for the determined group;
- performing inverse quantization and inverse frequency transformation for the decoding of the data of the group, based on the large data processing unit;
- 35 performing motion compensation for the decoding of the data of the group, based on the large data processing unit; and
- performing in-loop filtering for the decoding of the data of the group, based on the large data processing unit.
- 40 31. The image decoding method of claim 30, wherein in the decoding, a size of a data processing unit for the frequency transformation is determined based on a size of a data processing unit for the motion compensation
32. The image decoding method of claim 30, wherein the performing of the motion prediction uses at least one selected from the group consisting of:
- 45 a scanning sequence determined for each data processing unit for sequential motion estimation determined based on the large data processing unit,
- neighbor availability of a neighboring data processing unit located on a right upper end of a current data processing unit which is determined based on the large data processing unit, and
- 50 a simplification mode in which only data processing units having square block shapes from among the large data processing units are used.
33. The image decoding method of claim 30, wherein the performing of the motion compensation comprises performing intra prediction based on the large data processing unit,
- 55 wherein the performing of the intra prediction comprises performing motion compensation for the decoding of the data of the group by using at least one selected from the group consisting of:
- Intra compensation performed in units of the large data processing units in an intra mode of a complex plain type,

a data processing unit preset so that motion compensation of an intra mode is performed, based on the large data processing unit, and
 a prediction mode in which an intra mode and an inter mode are mixed, based on the large data processing unit.

5 **34.** The image decoding method of claim 30, wherein the performing of the motion compensation comprises performing inter prediction based on the large data processing unit, wherein the performing of the inter prediction comprises performing inter prediction by using at least one selected from the group consisting of:

10 a motion vector determined for motion compensation of an inter mode with respect to a current data processing unit determined based on the large data processing unit,
 a motion vector predictor induced to correspond to the motion vector, based on the large data processing unit, and
 a skip mode of an extended shape determined based on the large data processing unit and a skip group comprising adjacent data processing units of at least one skip mode.

15 **35.** The image decoding method of claim 30, wherein the performing of the inverse quantization and inverse frequency transformation comprises performing at least one selected from the group consisting of:

20 integer inverse discrete cosine transformation based on a large block, based on the large data processing unit,
 inverse frequency transformation with respect to a chroma component of a large data processing unit, based on the large data processing unit, and
 modified inverse frequency transformation for reducing calculation complexity of inverse frequency transformation considering the large data processing unit.

25 **36.** The image decoding method of claim 30, wherein the performing of the entropy decoding comprises using at least one selected from the group consisting of:

30 a hierarchical coded block pattern determined according to the size of each data processing unit on the basis of the large data processing unit;
 a coded block pattern scanned based on the possibility that a quantized coefficient which is not 0 from among the quantized coefficients of the large data processing unit exists;
 entropy decoding with respect to data obtained by entropy encoding according to CABAC or CAVLC considering the large data processing unit; and
 a coded block pattern determined to be a single context for the large data processing unit of an intra mode or
 a chroma component, based on the large data processing unit.

37 **37.** The image decoding method of claim 30, wherein in the performing of the in-loop filtering,
 it is determined based on the large data processing unit whether deblocking filtering is performed on at least one selected from the group consisting of a boundary of a lower data processing unit in a current data processing unit having a large size, a boundary of a lower data processing unit in the current data processing unit on which frequency-transformation has been performed in units of a data processing unit having a predetermined size, a boundary of a chroma component of a lower data processing unit in the current data processing unit, and a boundary of a lower data processing unit in the current data processing unit on which motion estimation has been performed in a mixed mode in which an inter mode and an intra mode are mixed,
 the degree of deblocking filtering with respect to each boundary is set for a boundary of at least one lower data processing unit in the current data processing unit, and
 additional filtering for reducing a ringing effect is performed on a current data processing unit having a large size, based on the large data processing unit.

50 **38.** An image encoding apparatus comprising:

55 a data classifying unit classifying image data divided into basic blocks, in units of groups and subgroups, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group;
 a group encoding mode determination unit determining at least one data processing unit selected from a group, a subgroup, and a basic block and determining an encoding mode for a predetermined group in order to encode the predetermined group, wherein the encoding mode represents a mode for encoding data included in the predetermined group in units of the at least one data processing unit; and

a group encoding unit encoding the data of the predetermined group according to the determined encoding mode.

39. An image decoding apparatus comprising:

- 5
- a receiving unit receiving and parsing image data encoded in units of at least one data processing unit selected from a basic block, a group, and a subgroup, wherein each group comprises at least one basic block and each subgroup comprises at least one basic block and is included in each group;
- a group encoding mode reading unit reading an encoding mode for a predetermined group from the parsed image data, wherein the encoding mode represents a mode for encoding image data included in the predetermined group in units of one data processing unit selected from a group, a subgroup, and a basic block; and
- 10
- a group decoding unit decoding the image data of the predetermined group according to the read encoding mode.

40. The image encoding apparatus of claim 38, wherein the group encoding unit comprises:

- 15
- a frequency transformation and quantization unit for performing frequency transformation and quantization for encoding of the input image on the basis of a large data processing unit corresponding to the determined group encoding mode;
- a motion estimation unit for performing motion estimation for the encoding of the input image, based on the large data processing unit;
- 20
- an in-loop filtering unit for performing in-loop filtering for the encoding of the input image, based on the large data processing unit; and
- an entropy encoding unit for performing entropy encoding for the encoding of the input image, based on the large data processing unit.

25 41. The image decoding apparatus of claim 39, wherein the group decoding unit comprises:

- an entropy decoding unit for performing entropy decoding for decoding of data of the group on the basis of a large data processing unit corresponding to the determined group encoding mode;
- an inverse quantization and inverse frequency transformation unit for performing inverse quantization and inverse frequency transformation for the decoding of the data of the group, based on the large data processing unit;
- 30
- a motion compensation unit for performing motion compensation for the decoding of the data of the group, based on the large data processing unit; and
- an in-loop filtering unit for performing in-loop filtering for the decoding of the data of the group, based on the large data processing unit.
- 35

42. A computer readable recording medium having recorded thereon a program for executing the image encoding method according to claim 1.

40 43. A computer readable recording medium having recorded thereon a program for executing the image decoding method according to claim 19.

45

50

55

FIG. 1

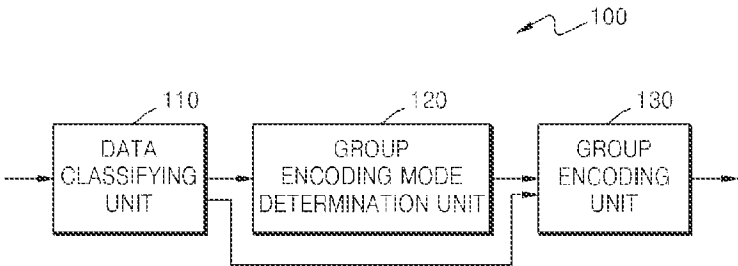


FIG. 2

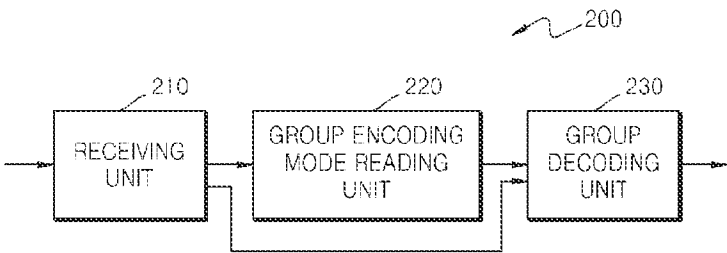
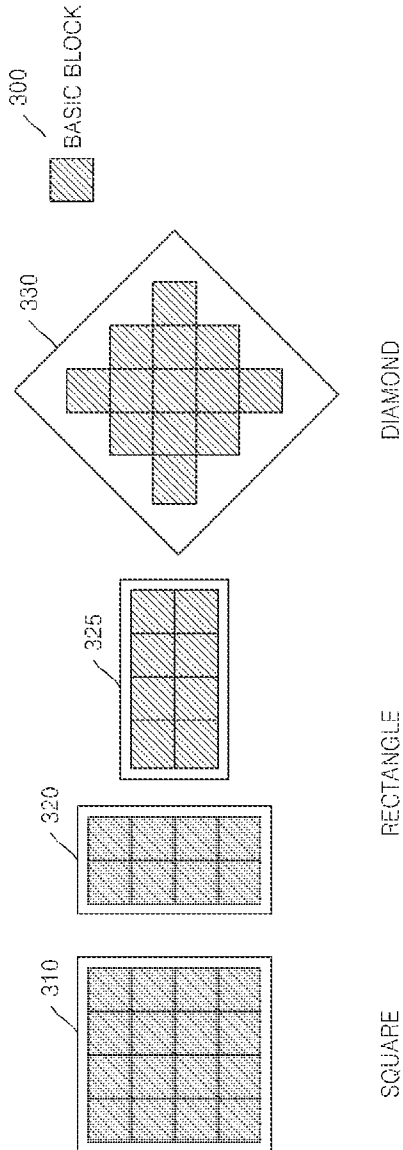


FIG. 3



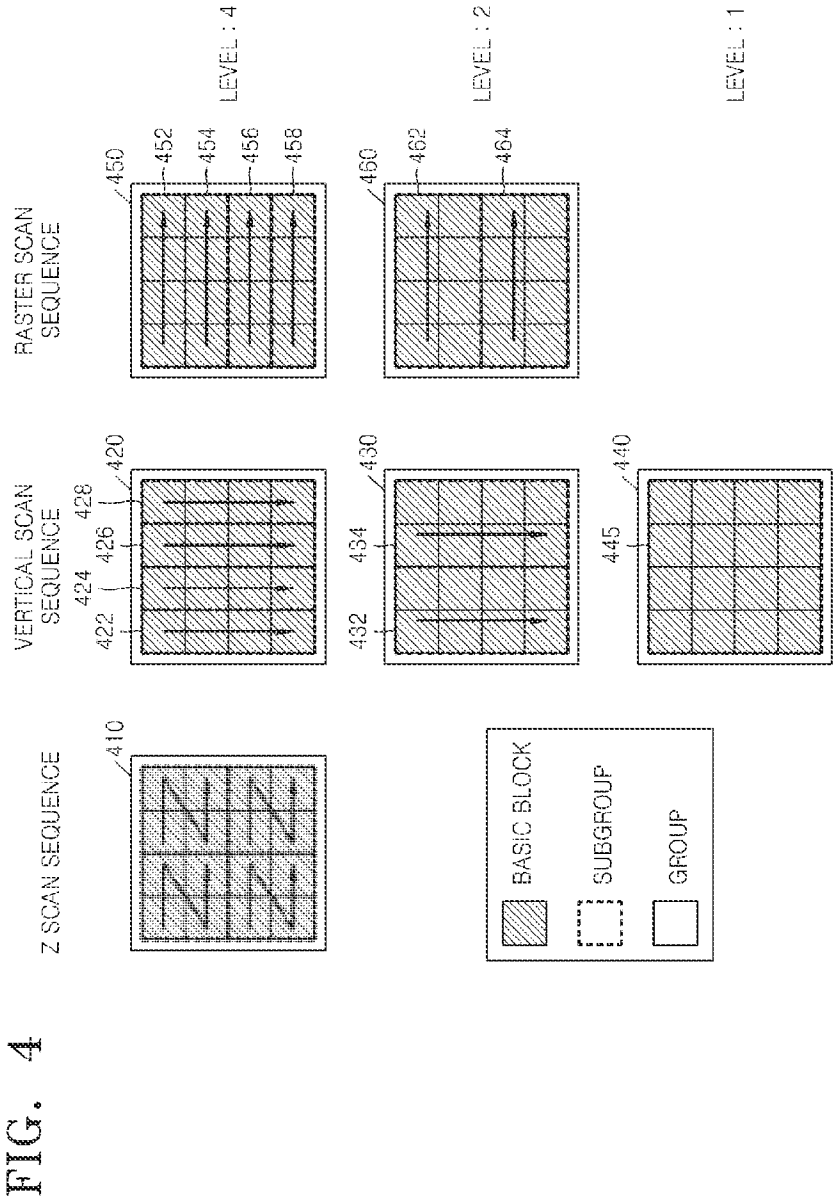


FIG. 5

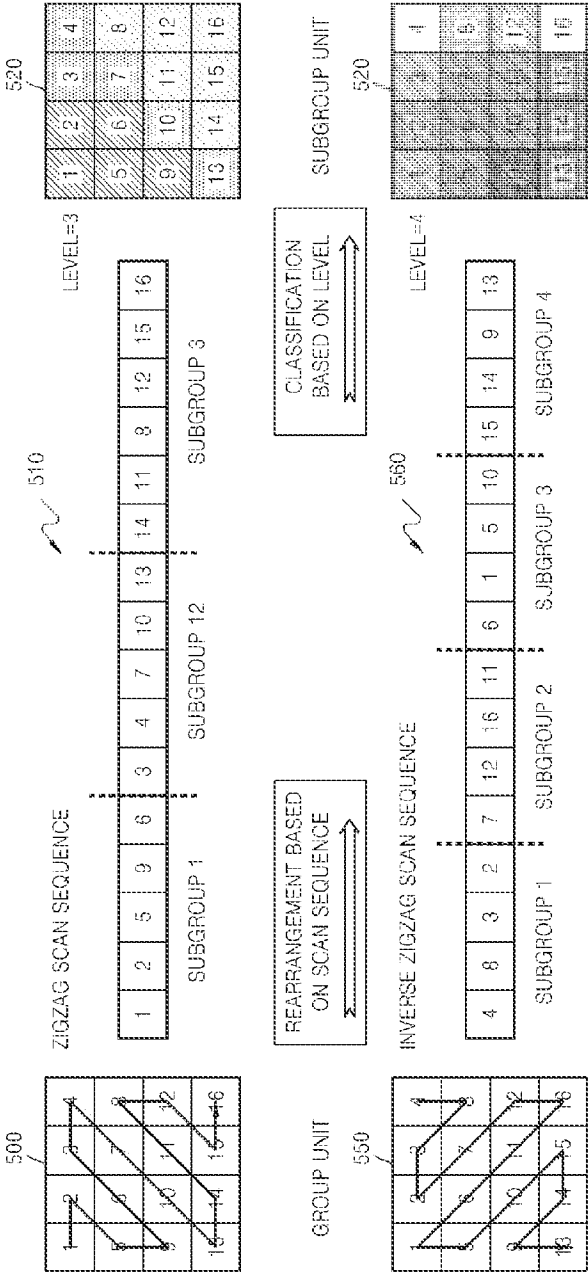


FIG. 6

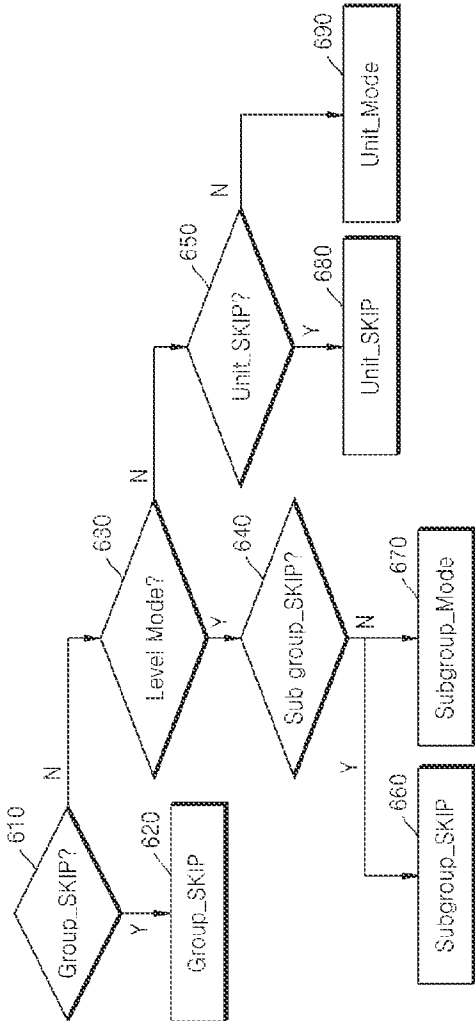


FIG. 7

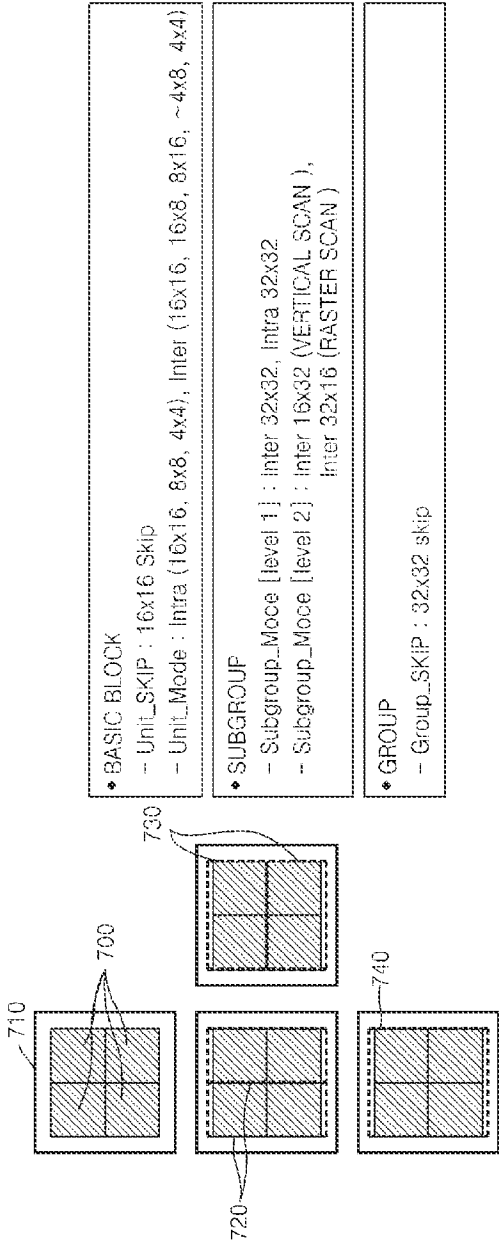


FIG. 8

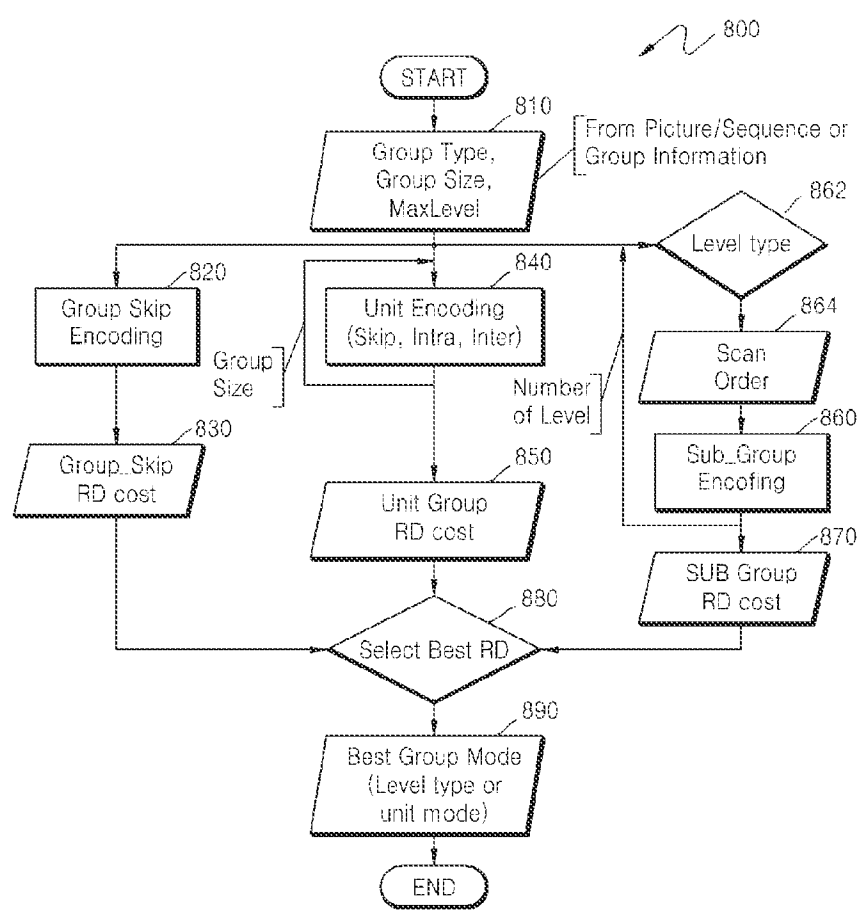
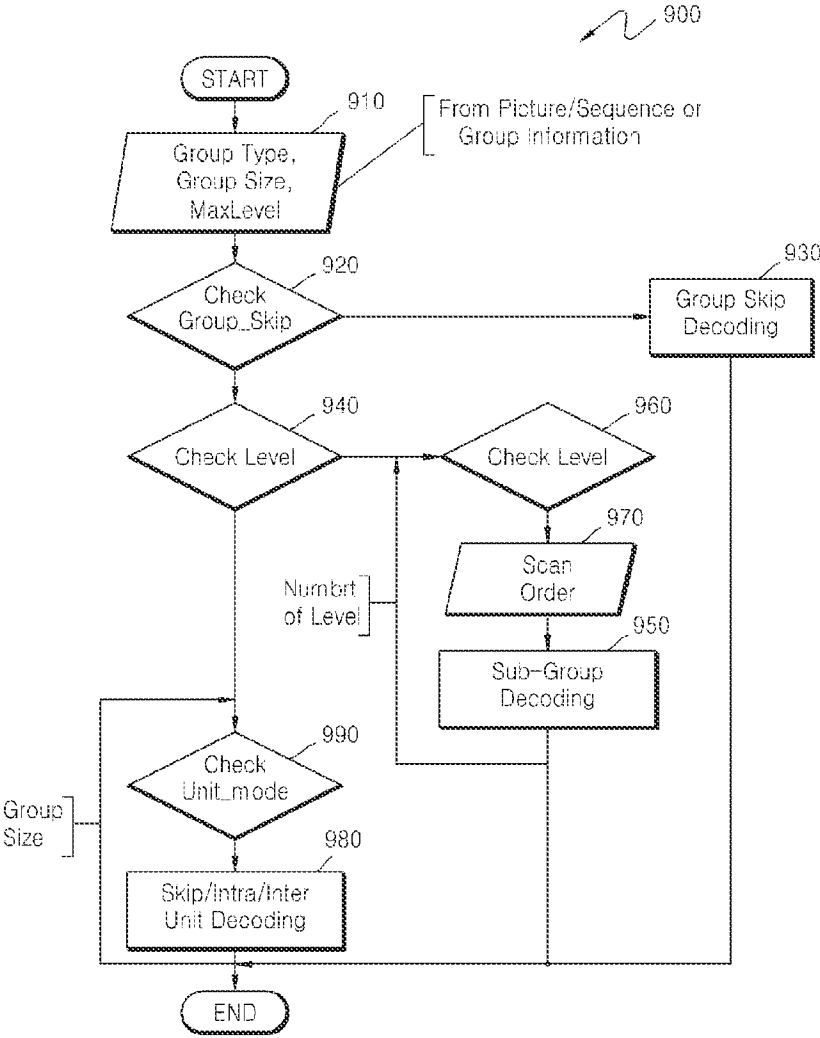


FIG. 9



Oil
Gill

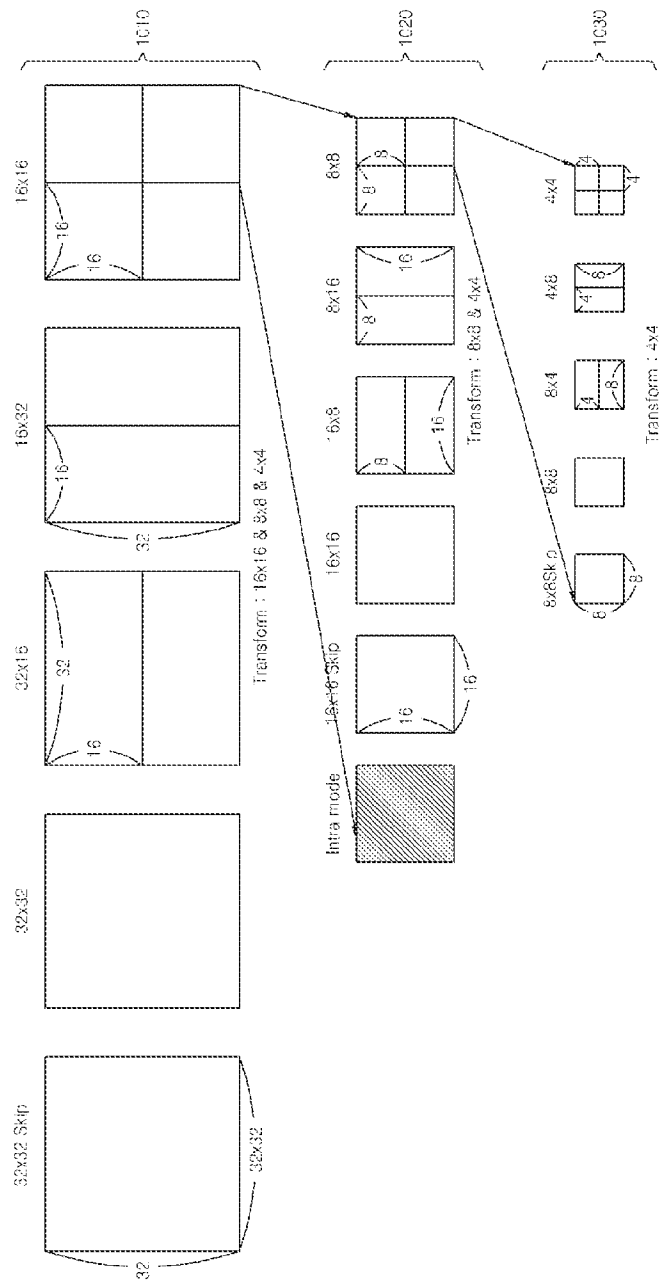


FIG. 11

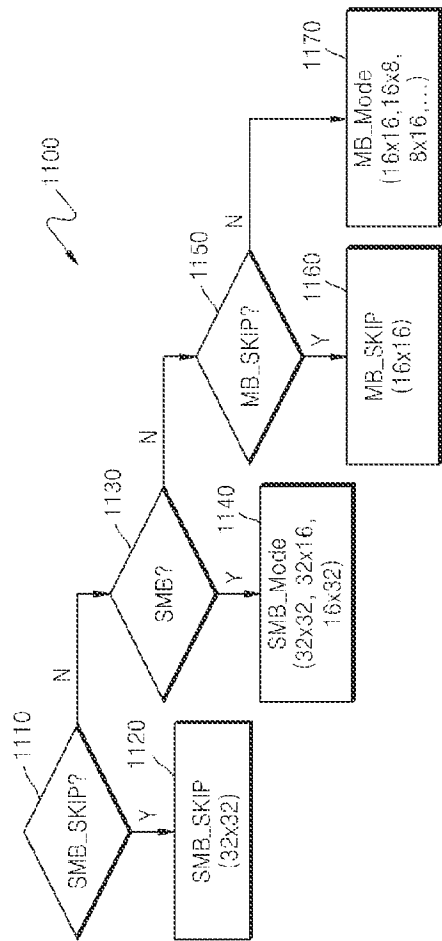


FIG. 12

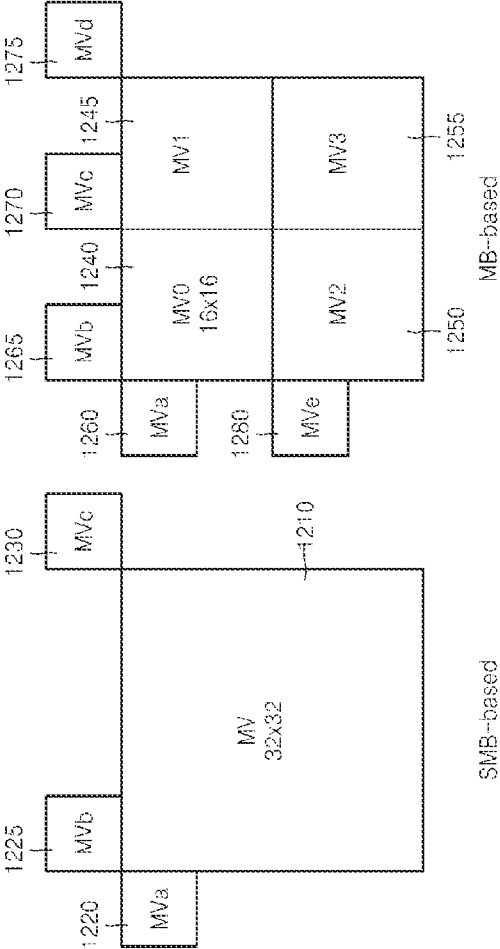


FIG. 13

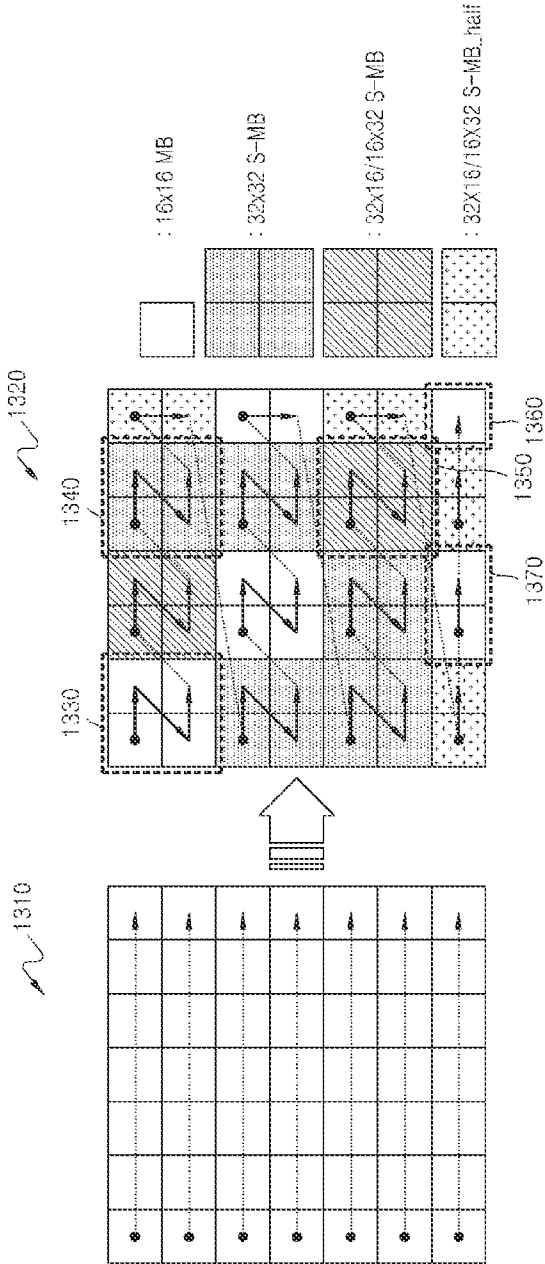


FIG. 14

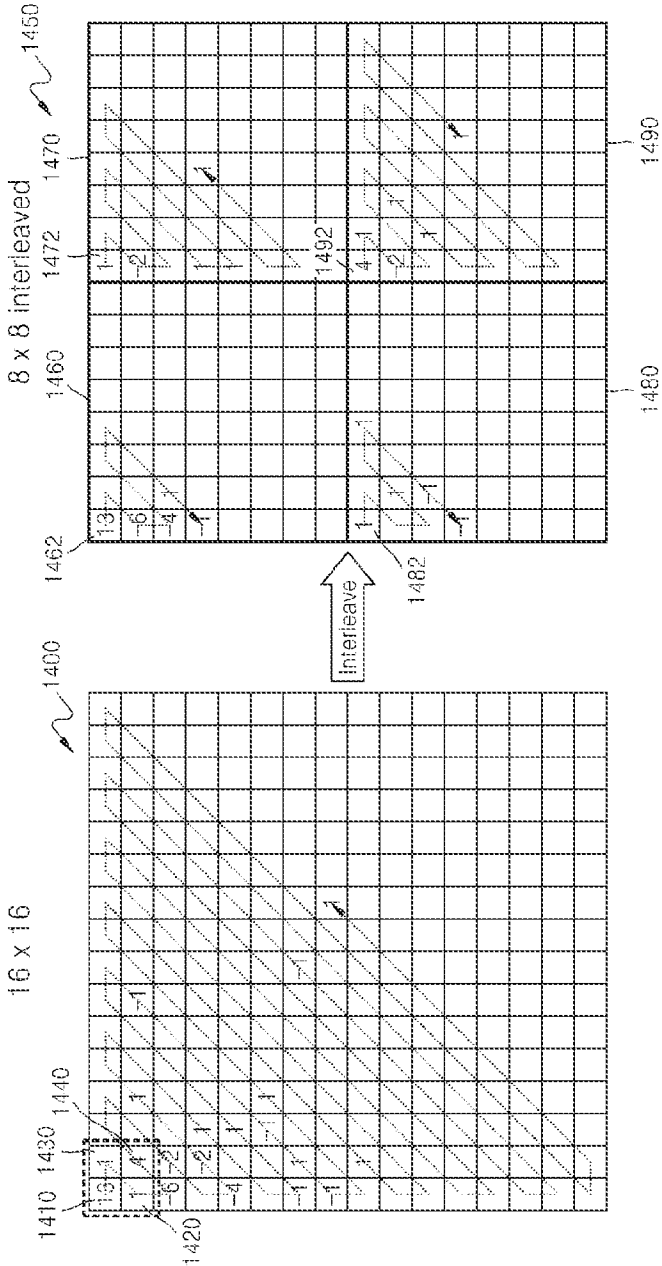


FIG. 15

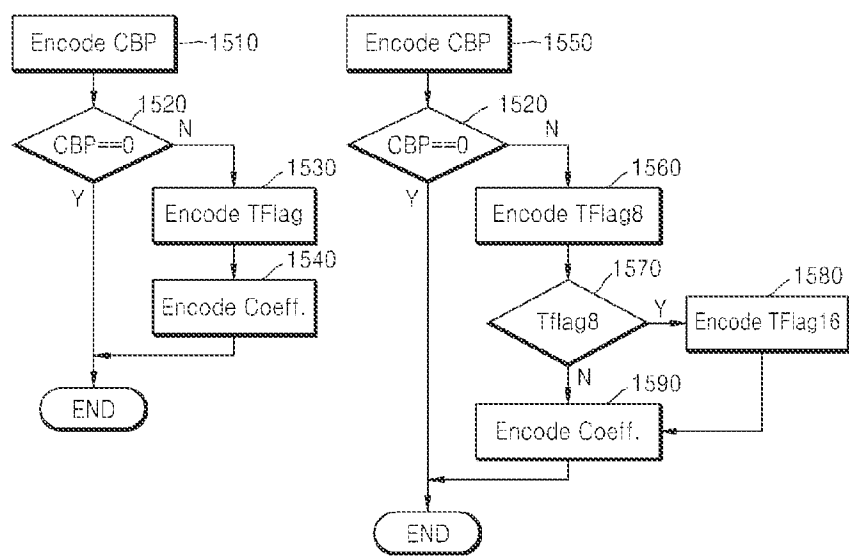


FIG. 16

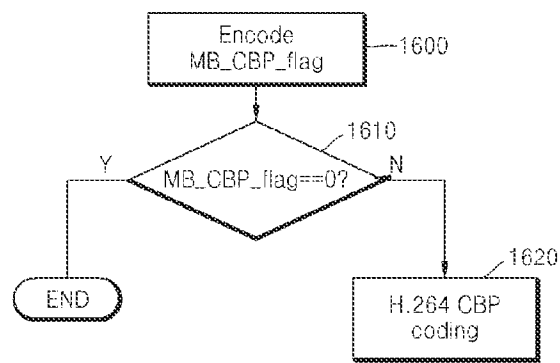


FIG. 17

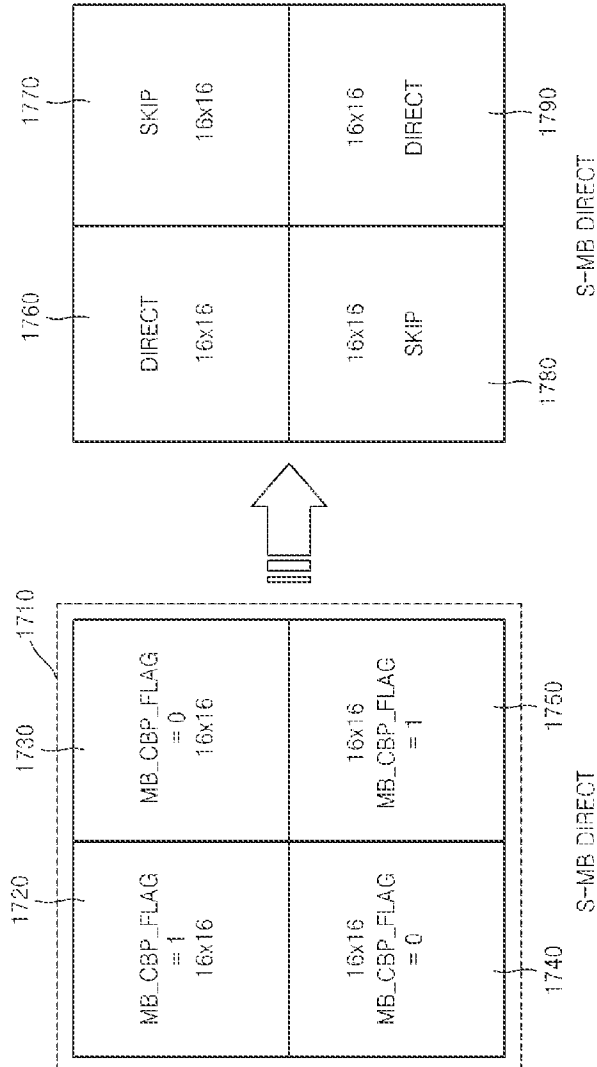


FIG. 18

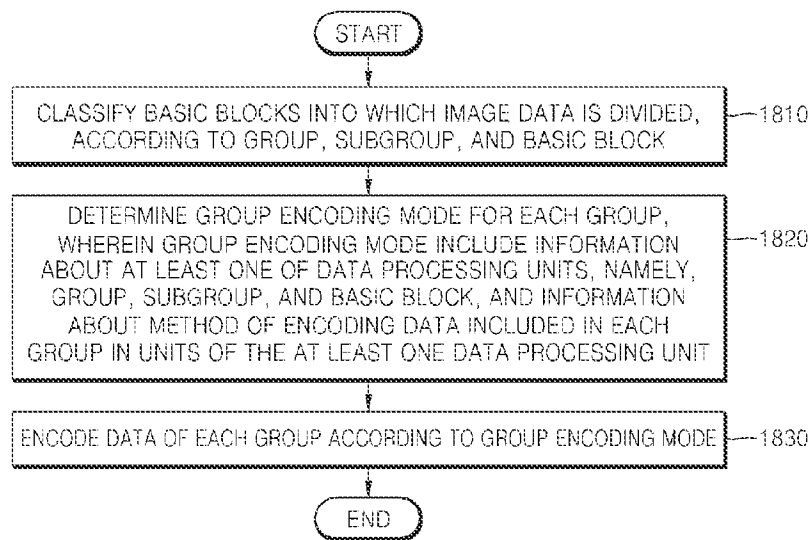


FIG. 19

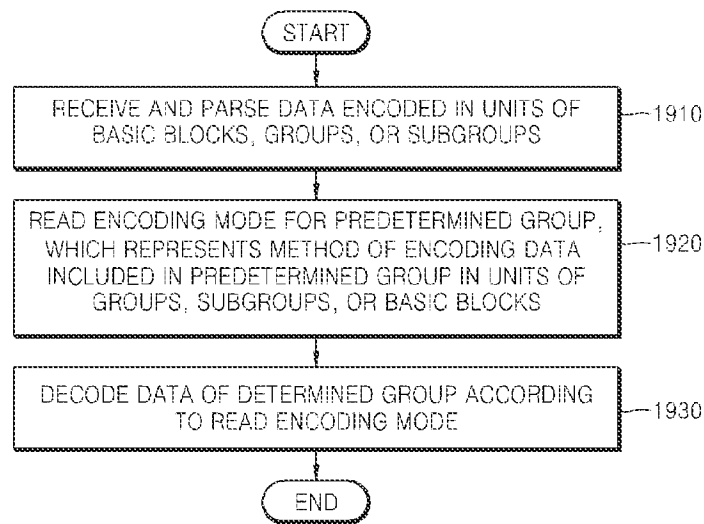


FIG. 20A

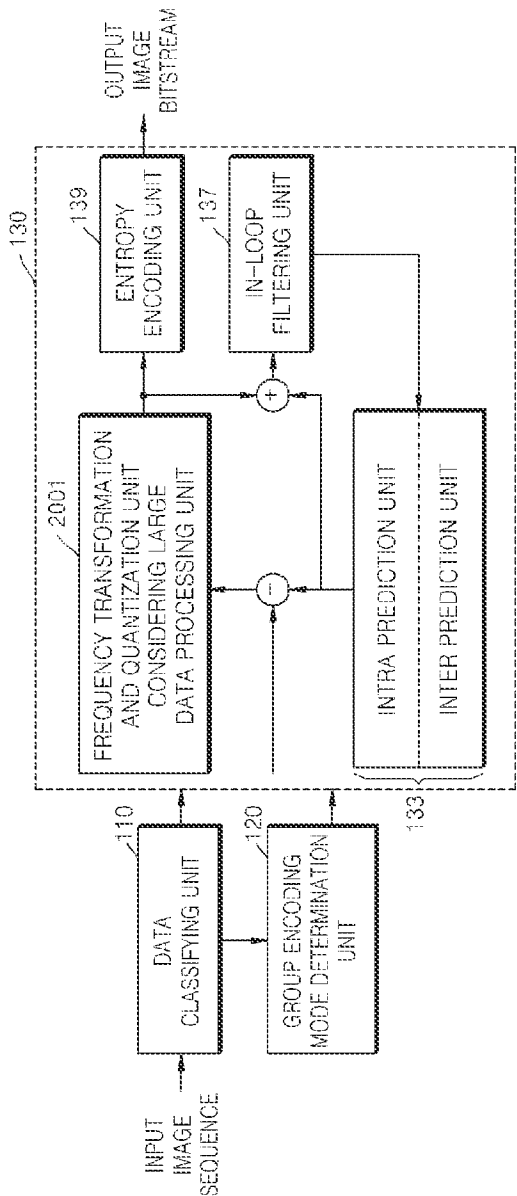


FIG. 20B

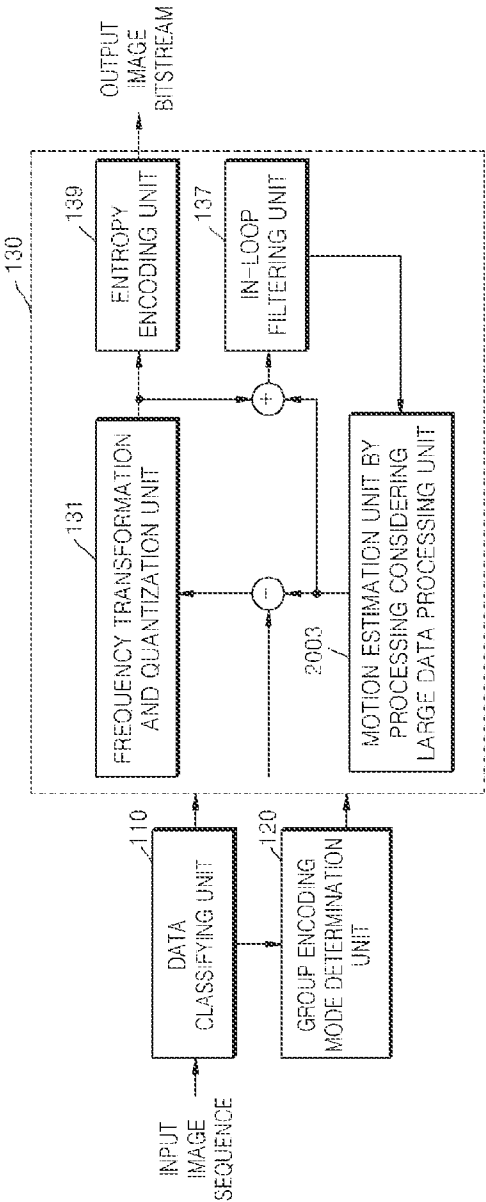


FIG. 20C

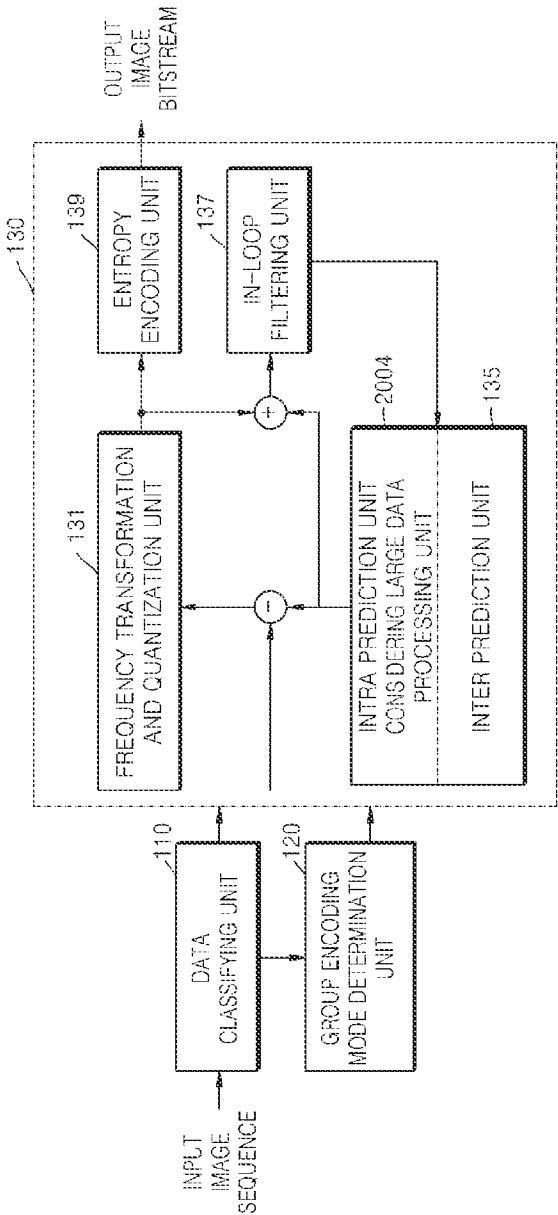


FIG. 20D

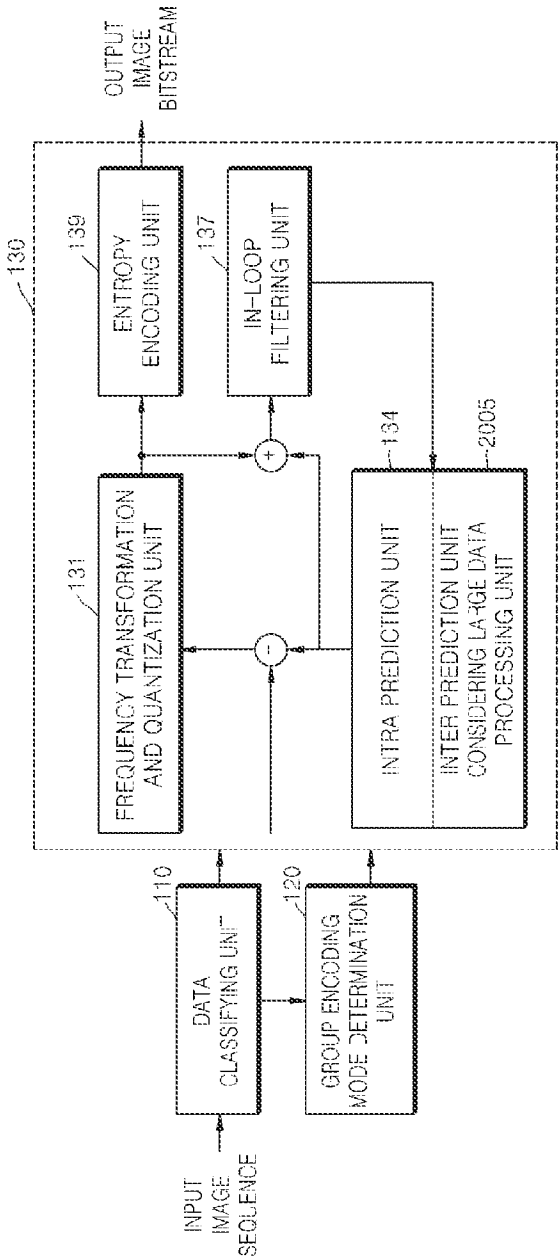


FIG. 20E

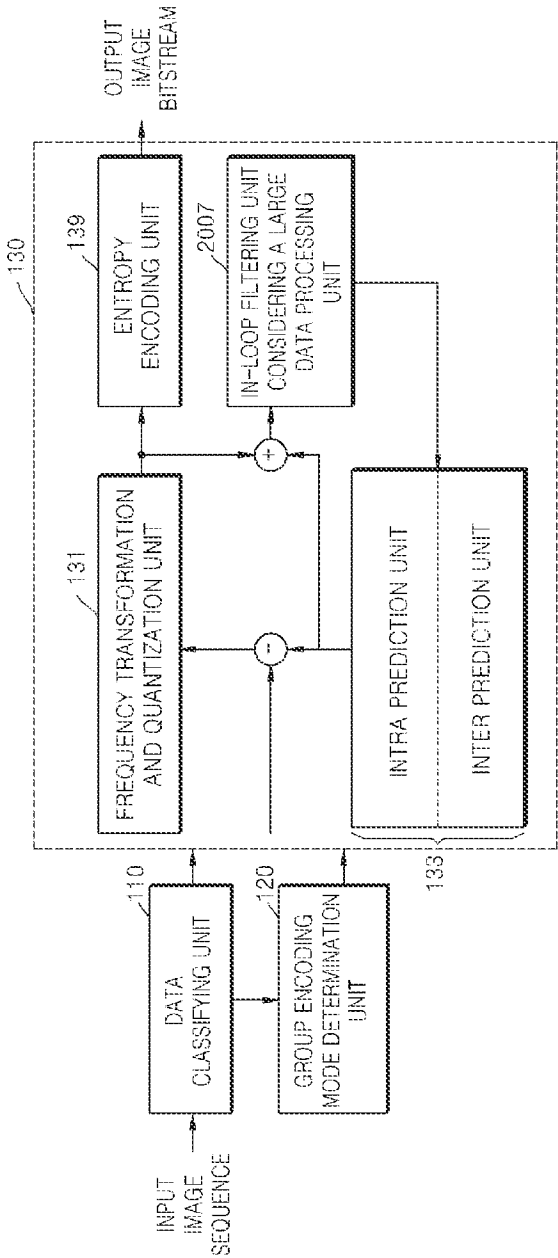


FIG. 20F

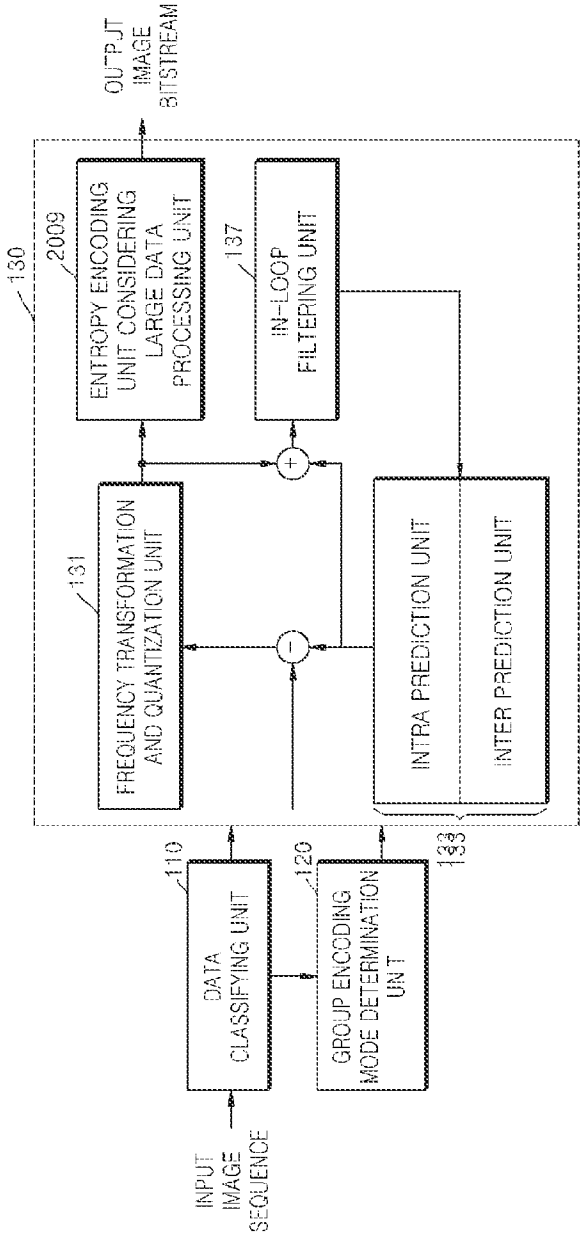


FIG. 21

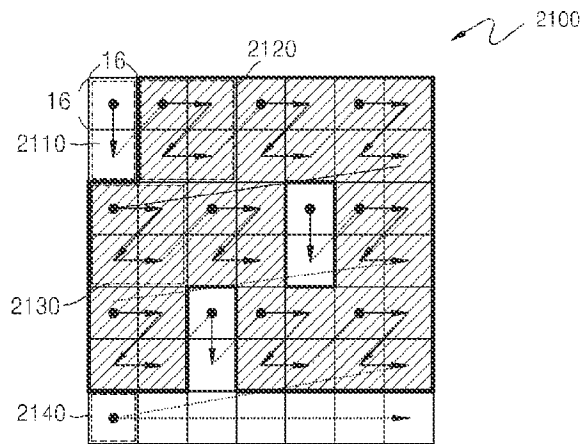
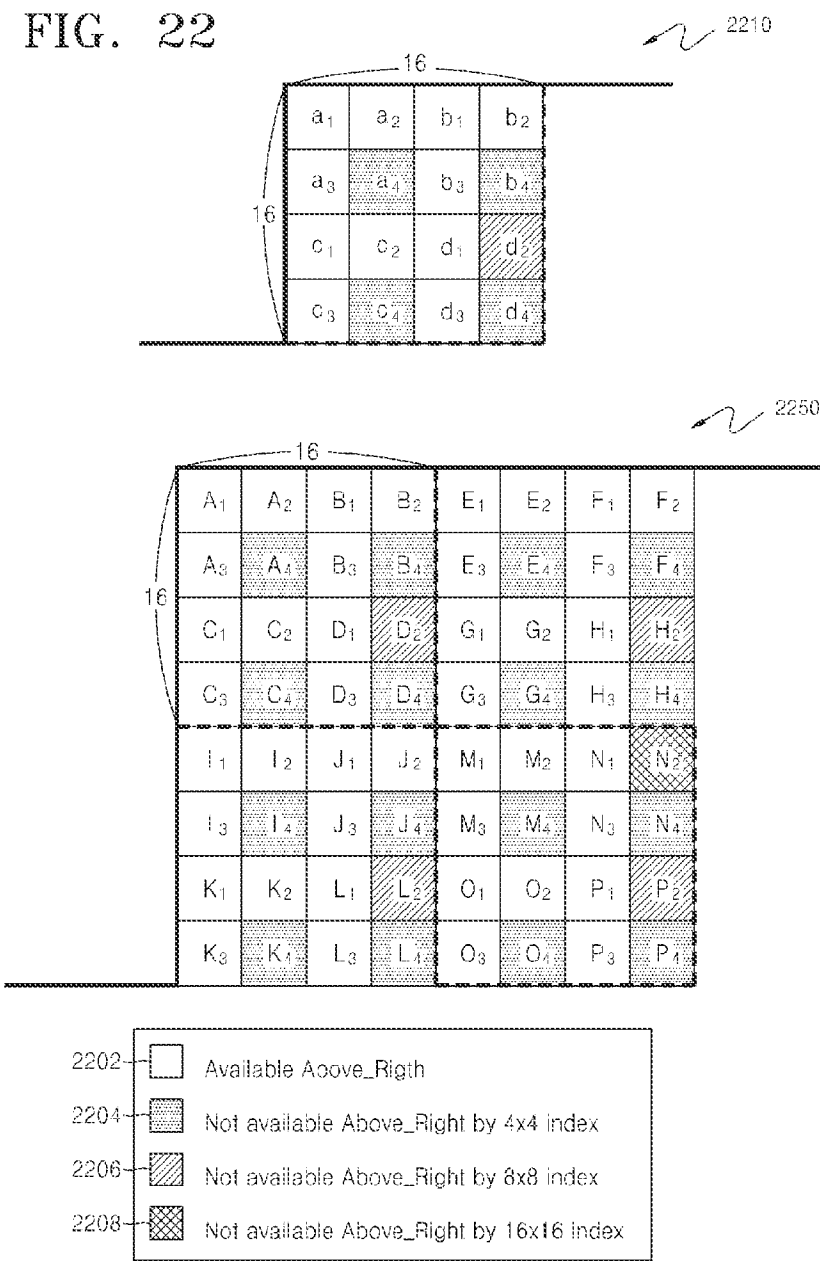


FIG. 22



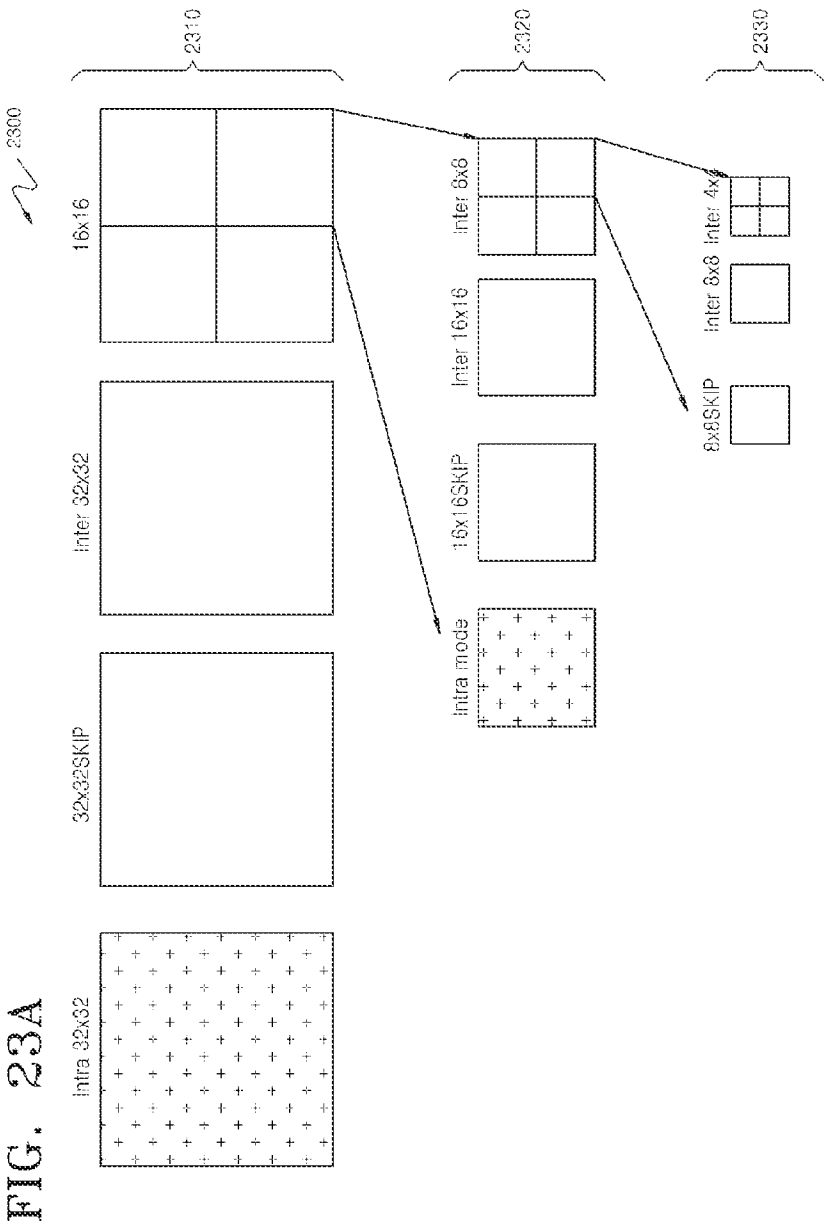


FIG. 23B

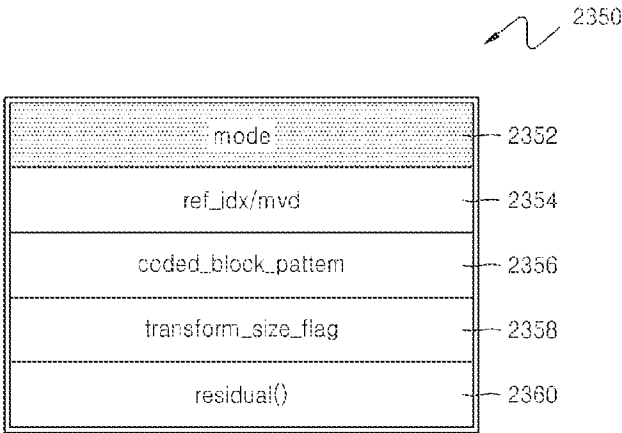


FIG. 24A

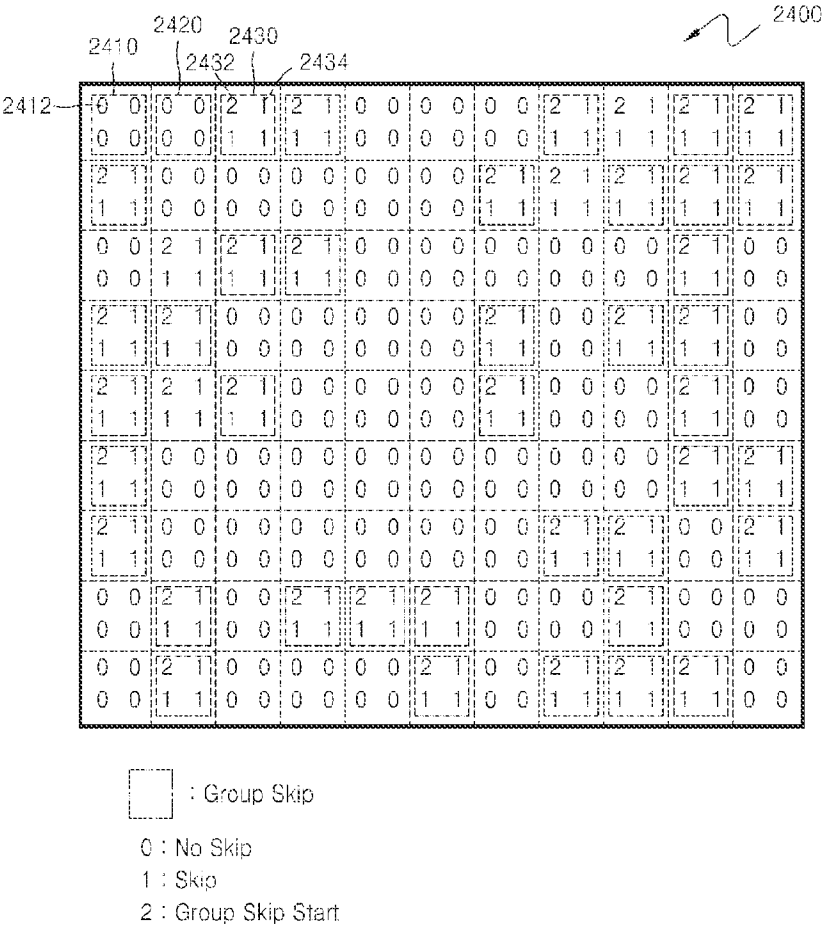


FIG. 24B

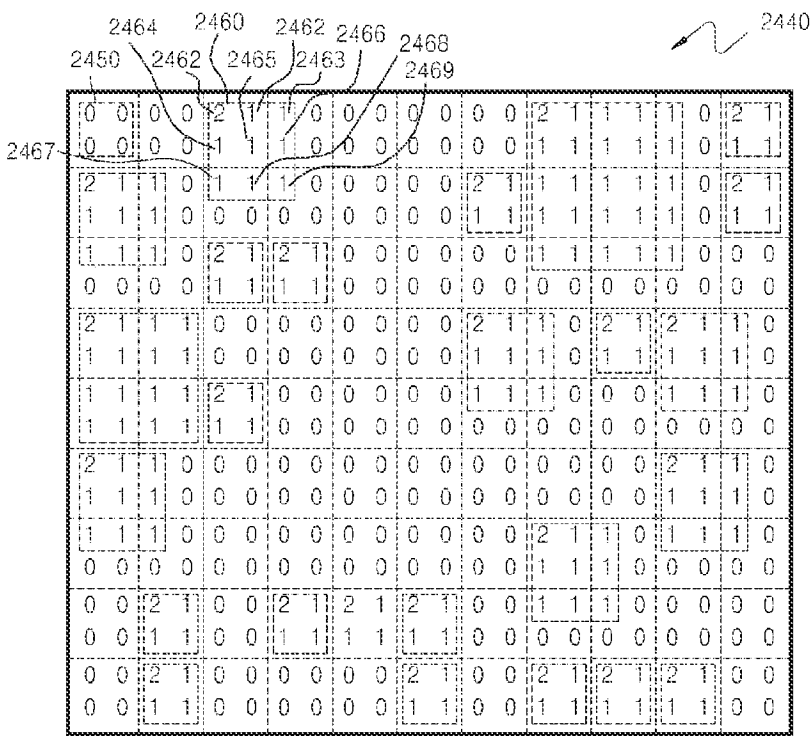


FIG. 24C

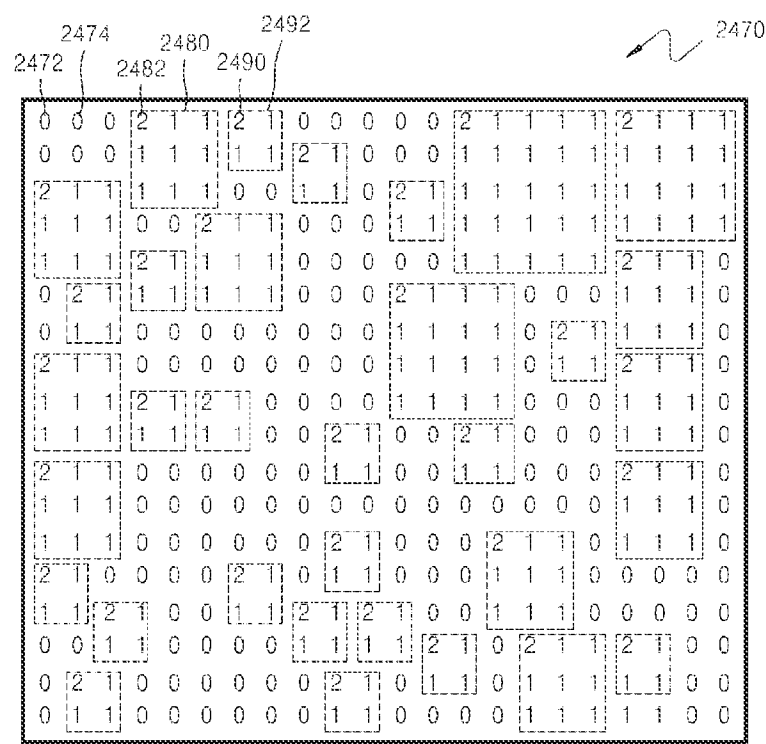


FIG. 25A

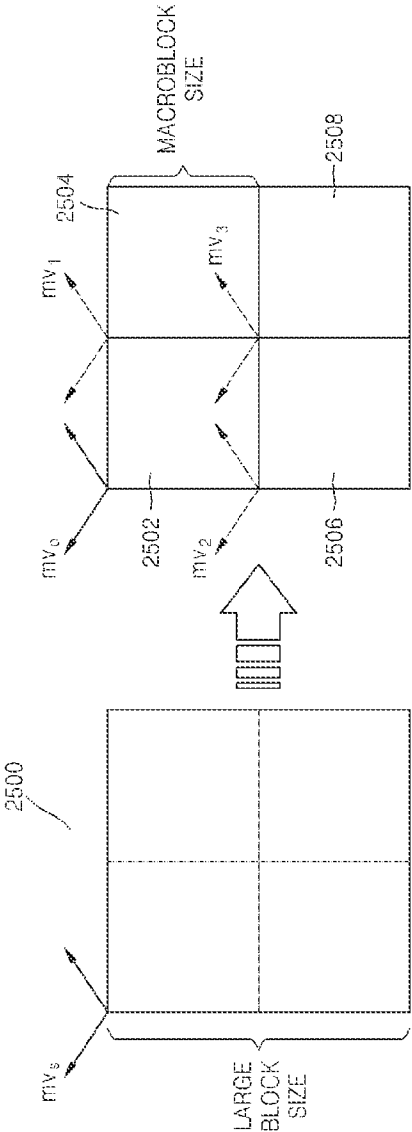


FIG. 25B

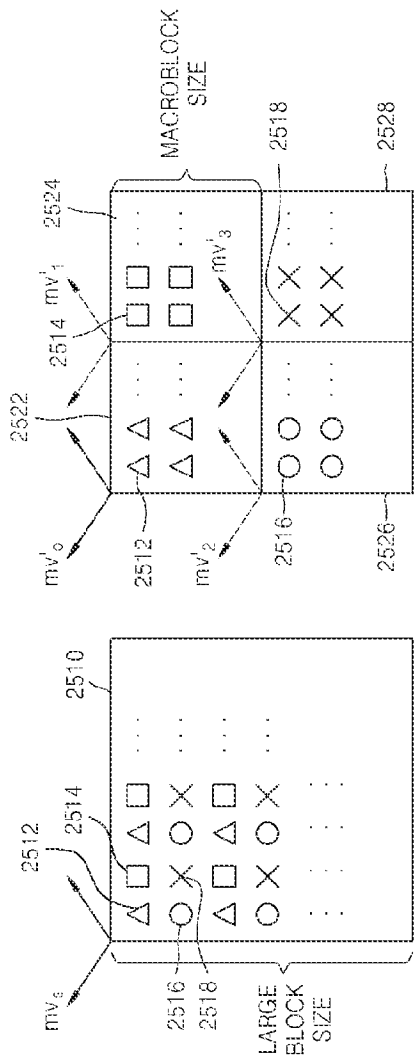


FIG. 26A

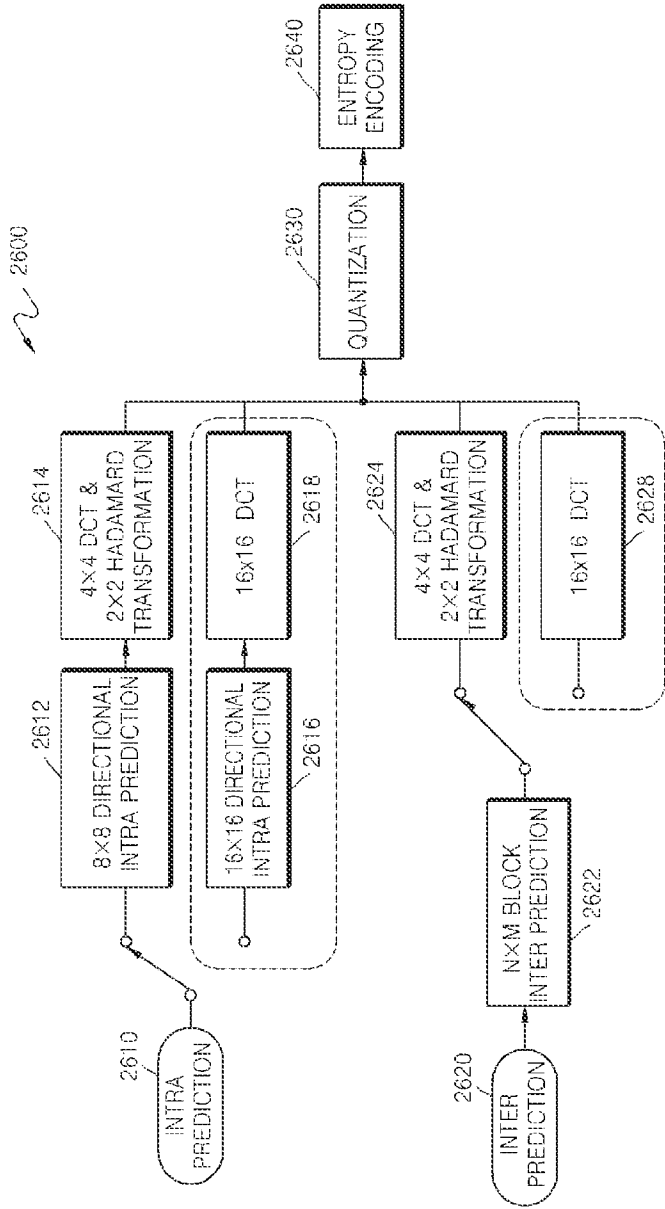


FIG. 26B

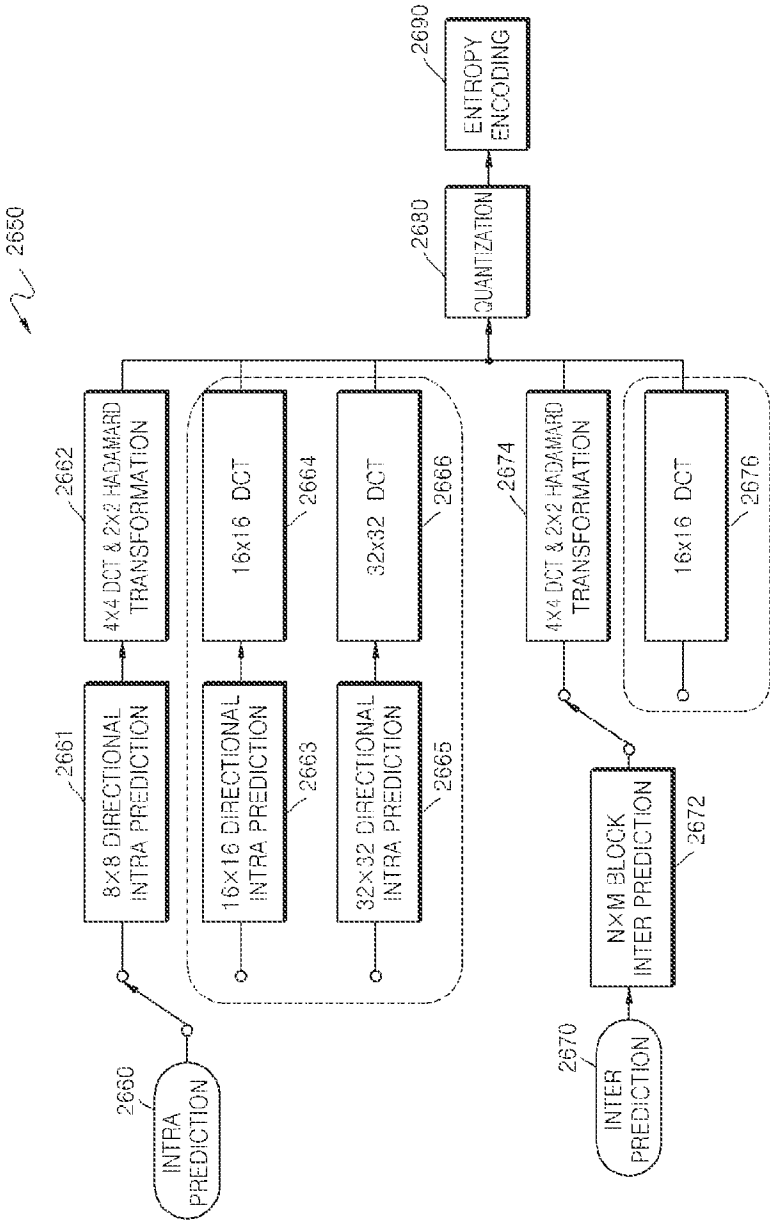


FIG. 27

IMAGE FORMAT	PREDICTION TYPE	SIZE OF DCT BLOCK FOR LUMA COMPONENT	SIZE OF DCT BLOCK FOR CHROMA COMPONENT
YUV 420	INTRA PREDICTION	32x32	16x16
		16x16, 8x8, 4x4	8x8
	INTER PREDICTION	16x16, 8x8, 4x4	8x8
YUV 444	INTRA PREDICTION	32x32	32x32
		16x16	16x16
		8x8, 4x4	8x8
	INTER PREDICTION	16x16	16x16
		8x8, 4x4	8x8

FIG. 28

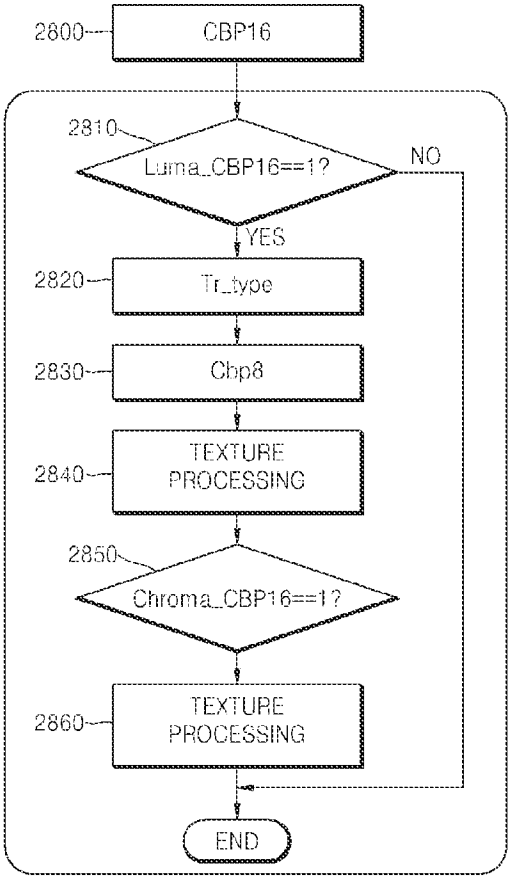


FIG. 29A

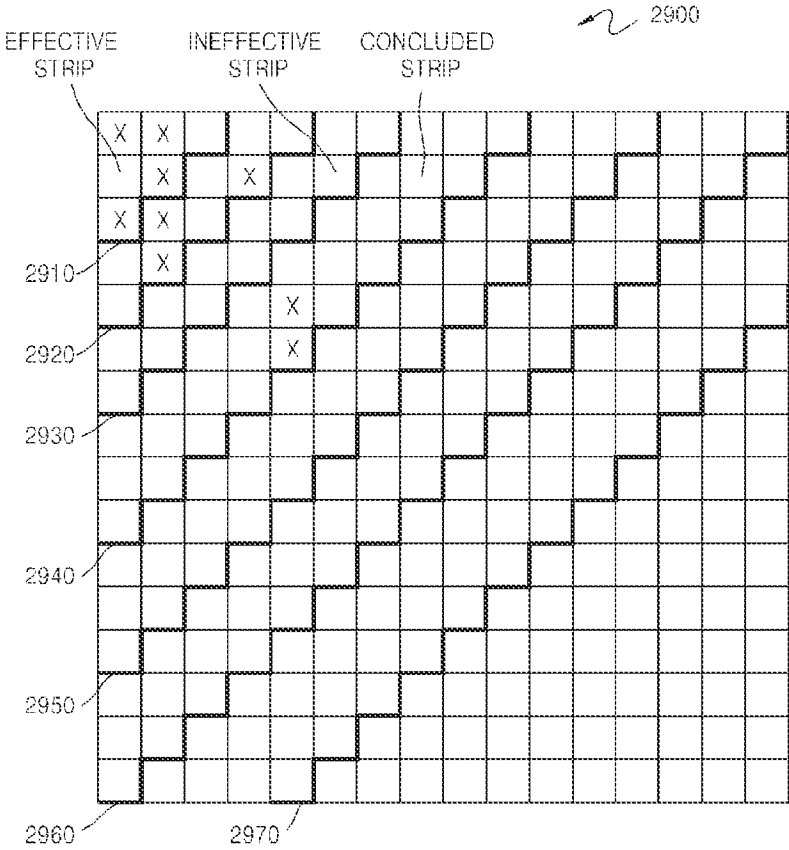


FIG. 29B

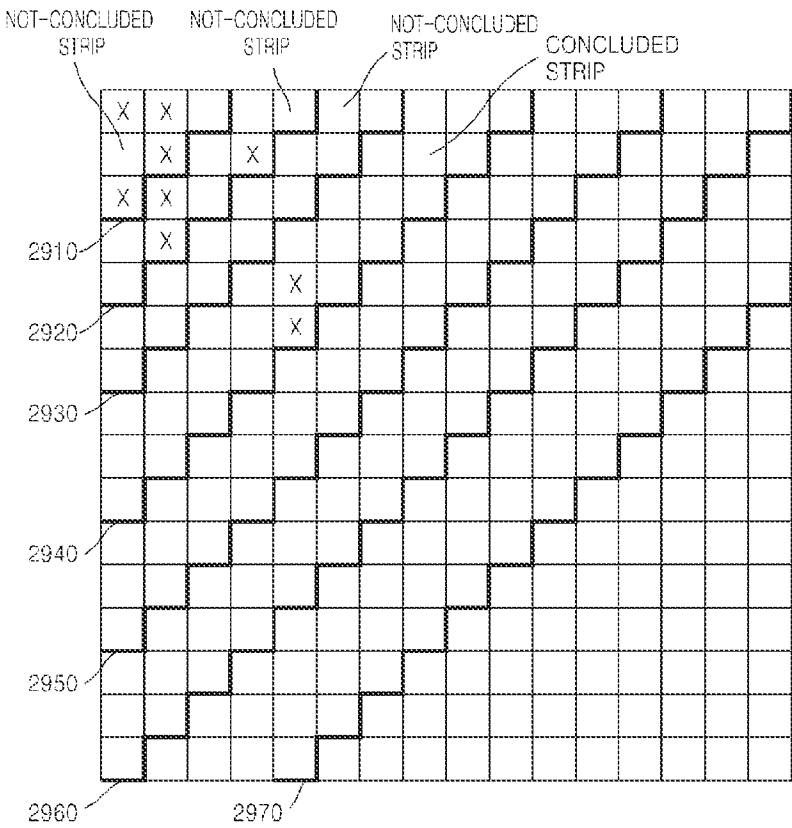


FIG. 30A

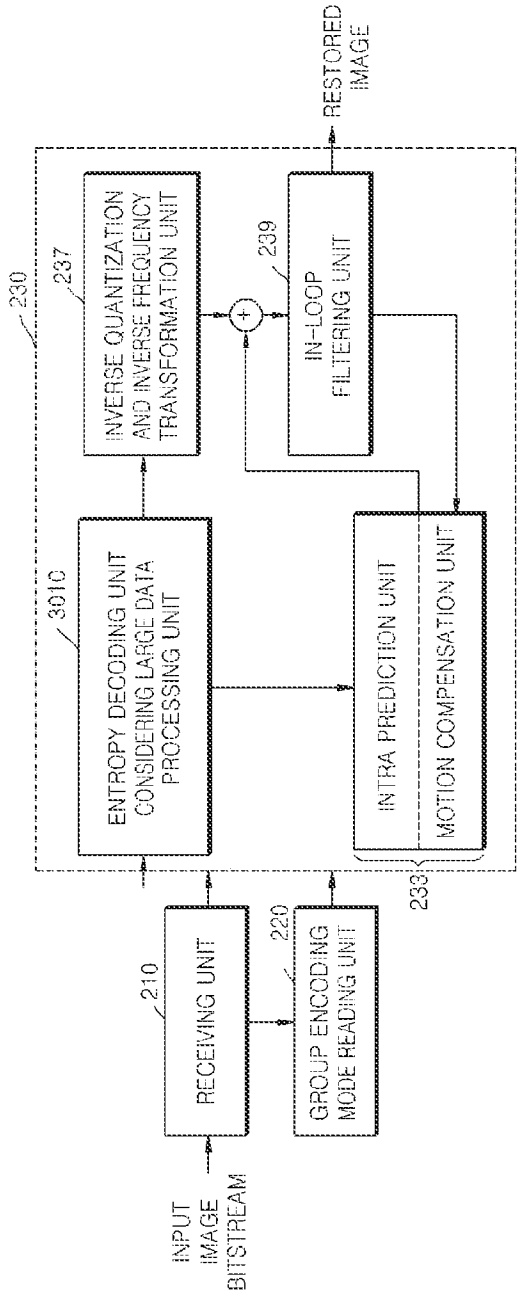


Fig. 30B

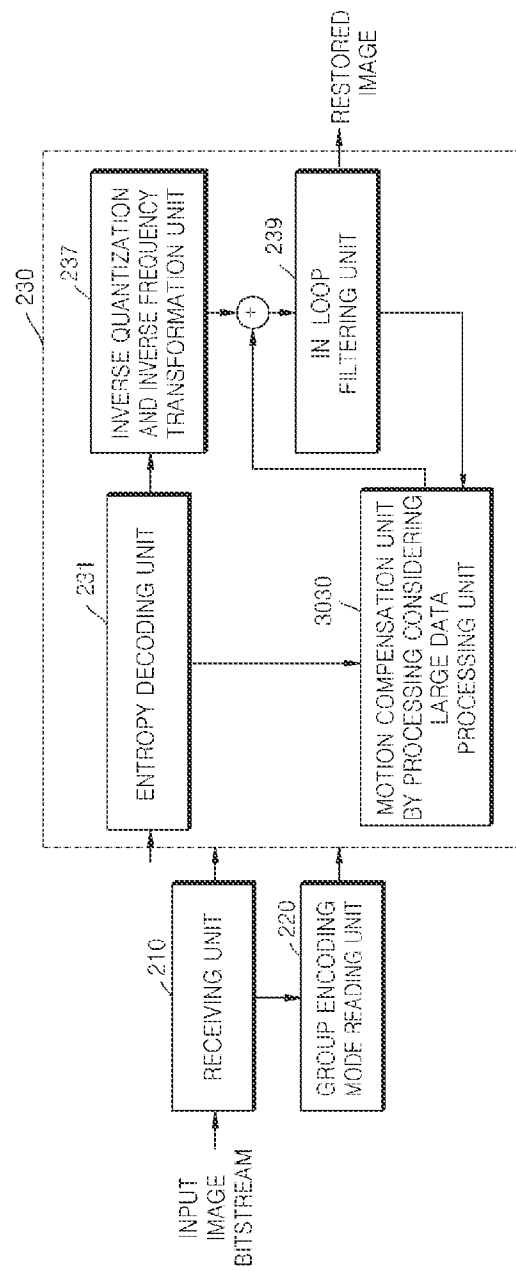


FIG. 30C

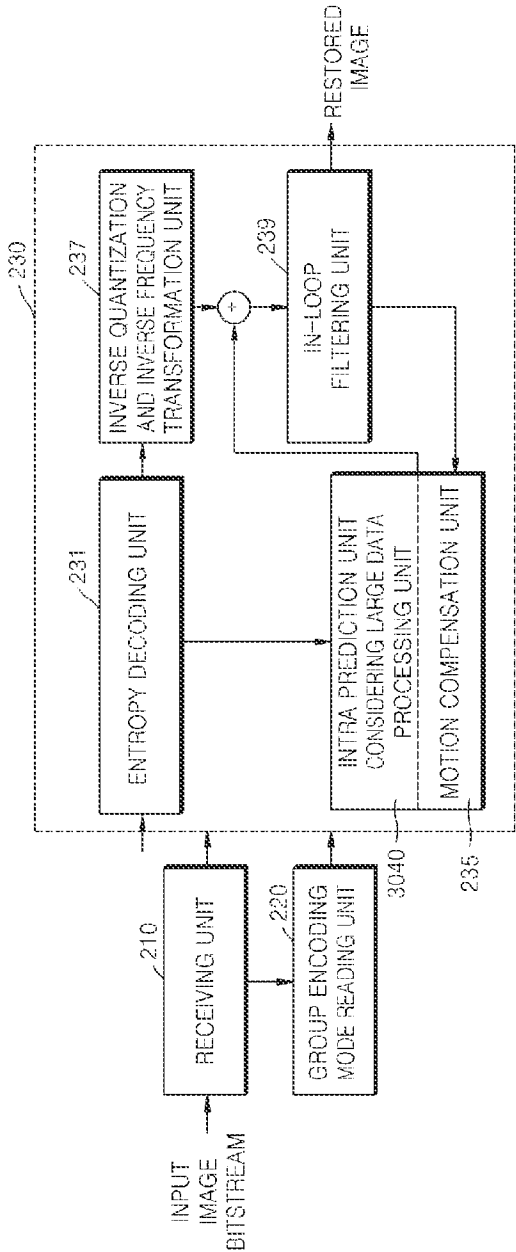


FIG. 30D

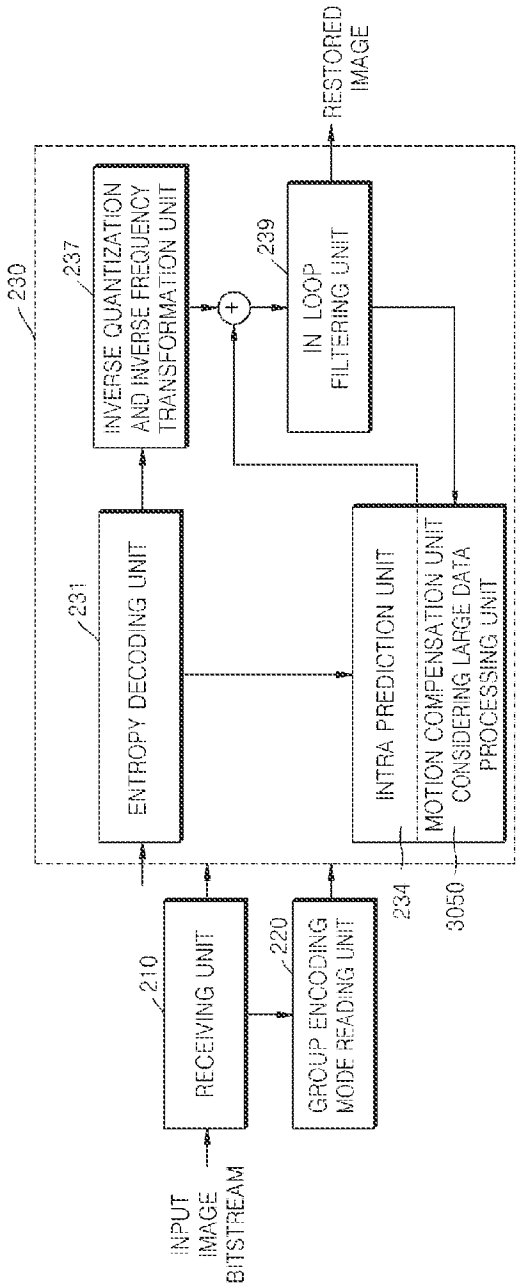


FIG. 30E

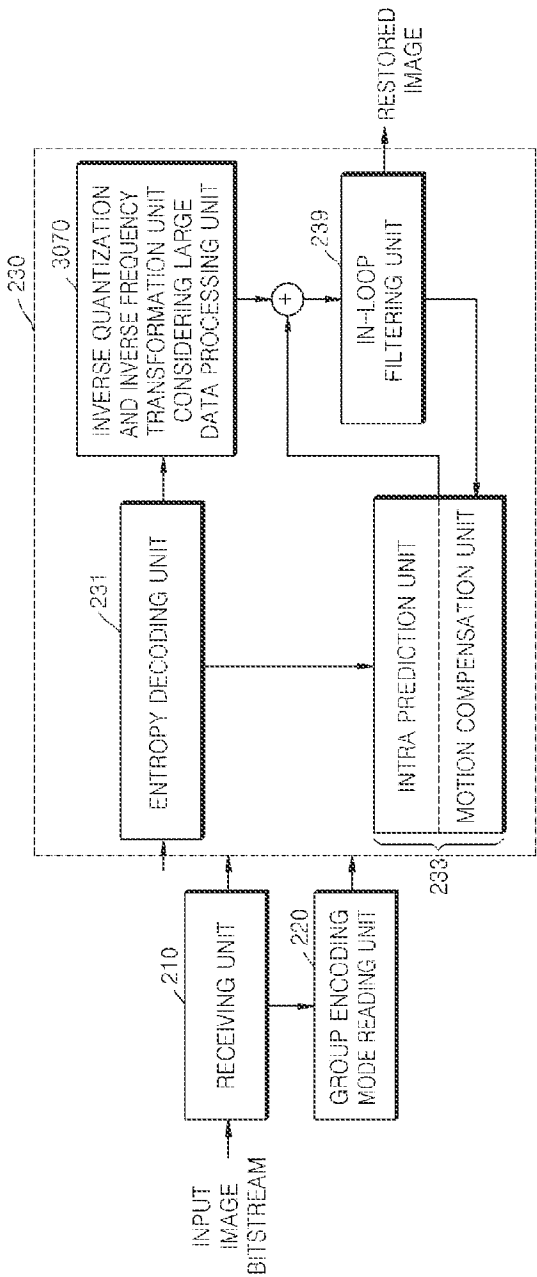


FIG. 30F

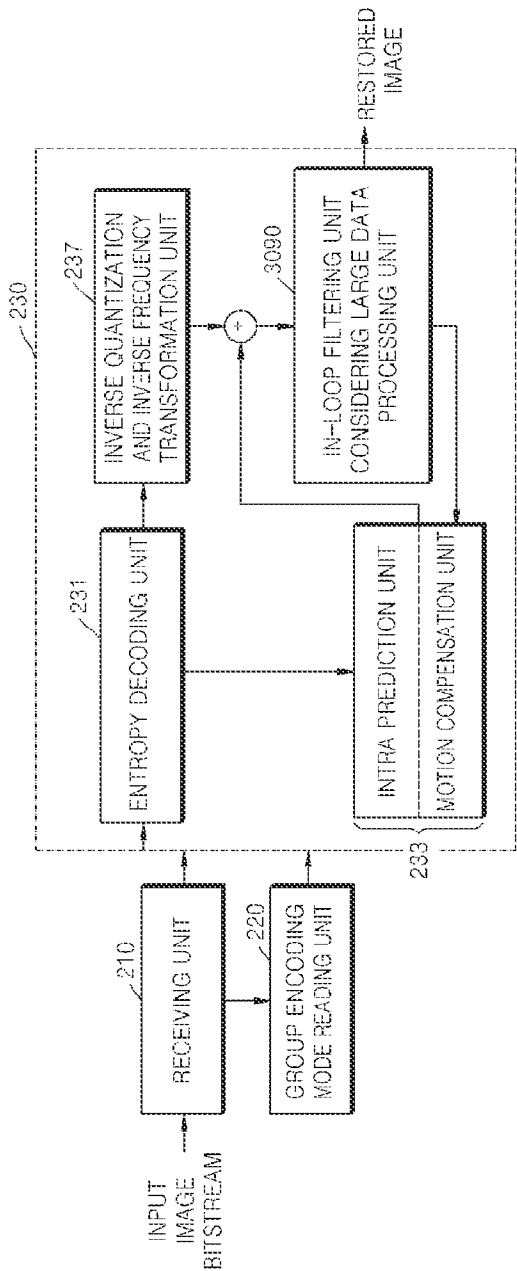


FIG. 31A

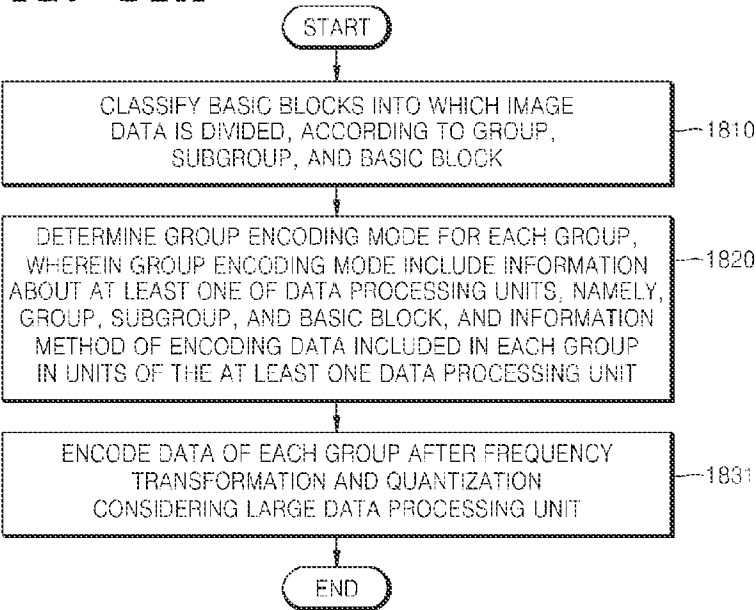


FIG. 31B

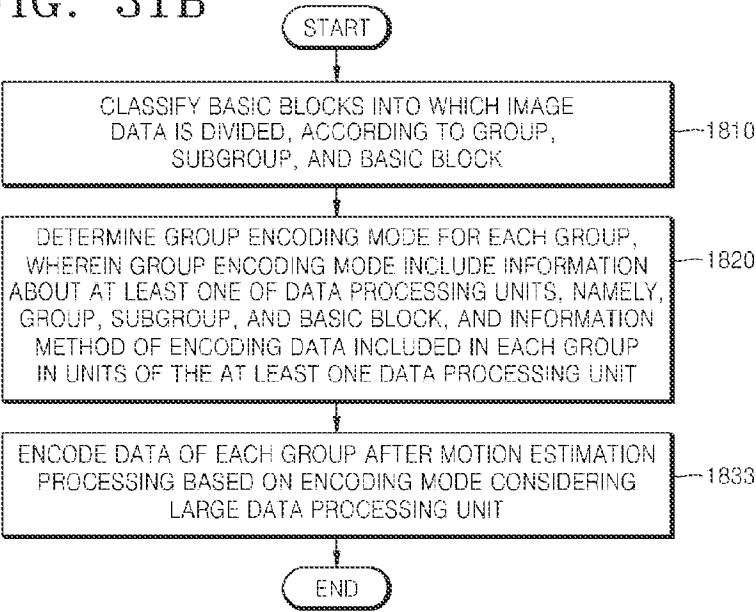


FIG. 31C

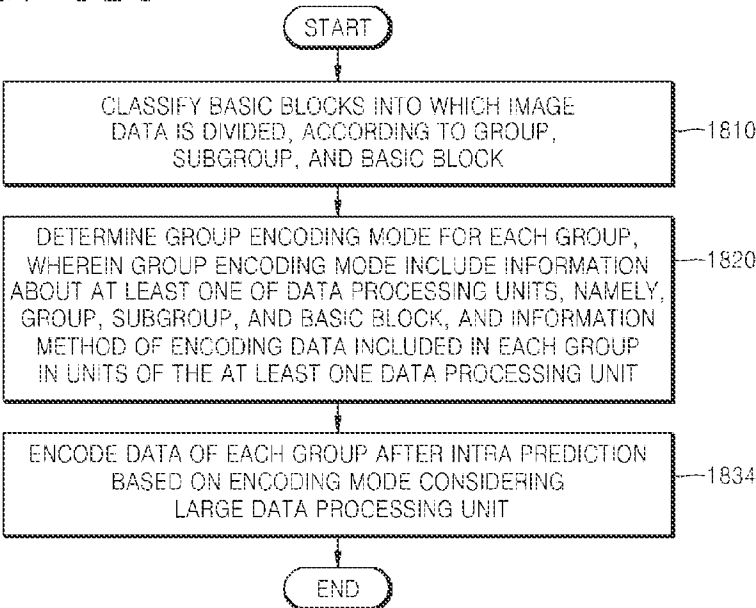


FIG. 31D

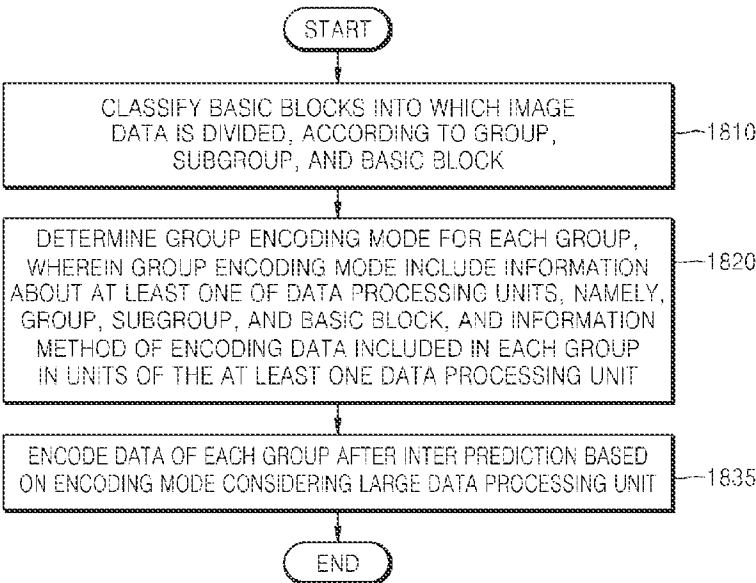


FIG. 31E

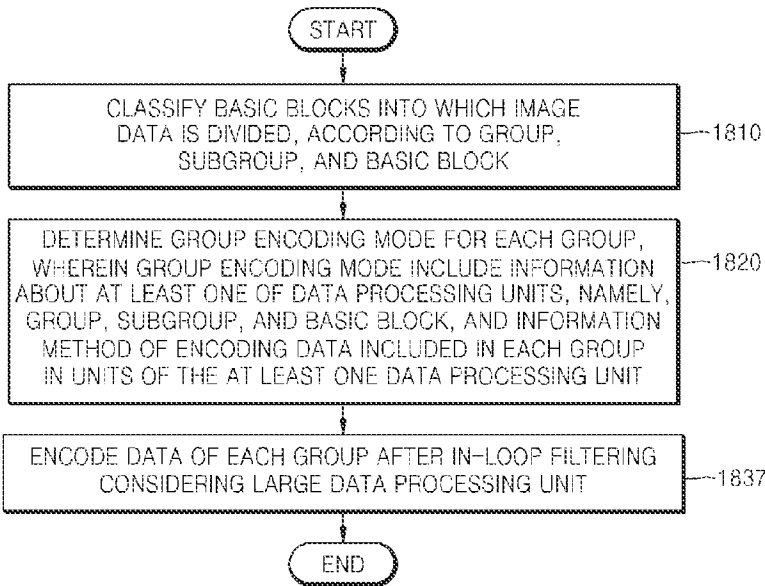


FIG. 31F

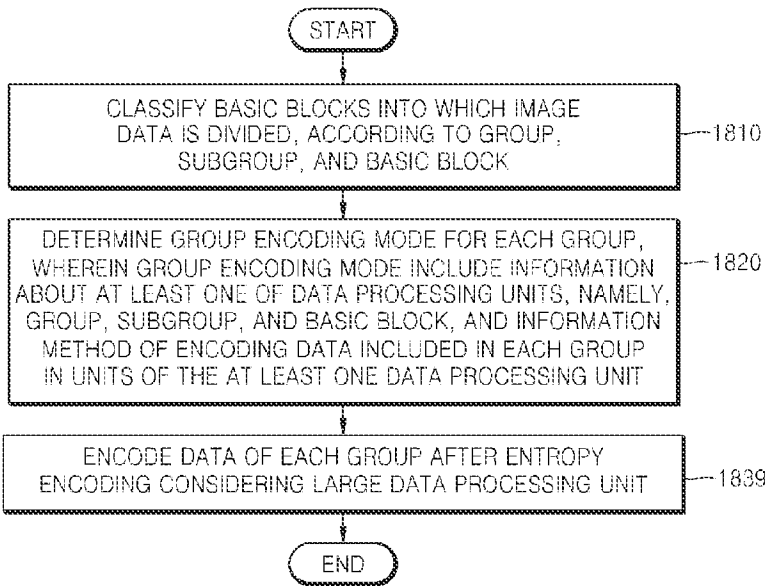


FIG. 32A

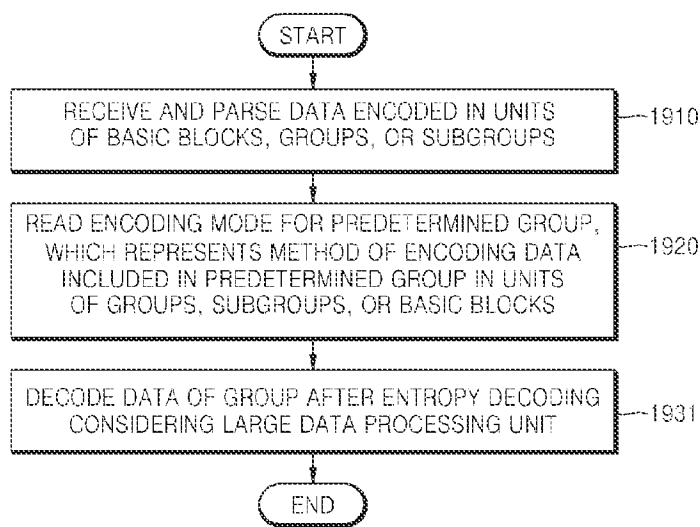


FIG. 32B

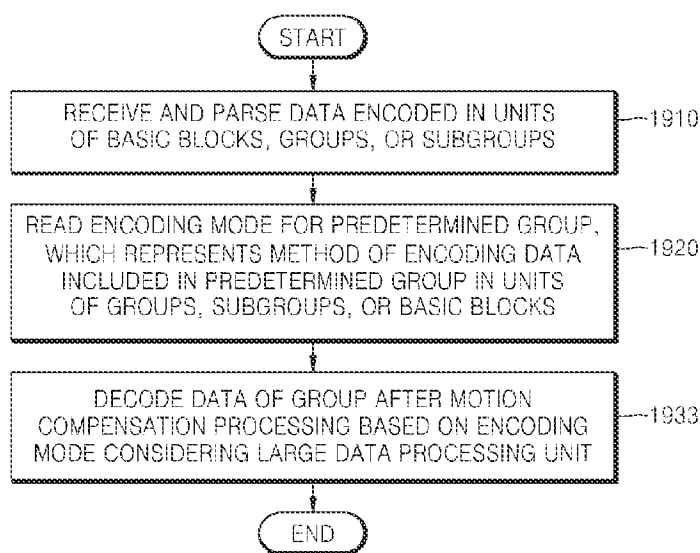


FIG. 32C

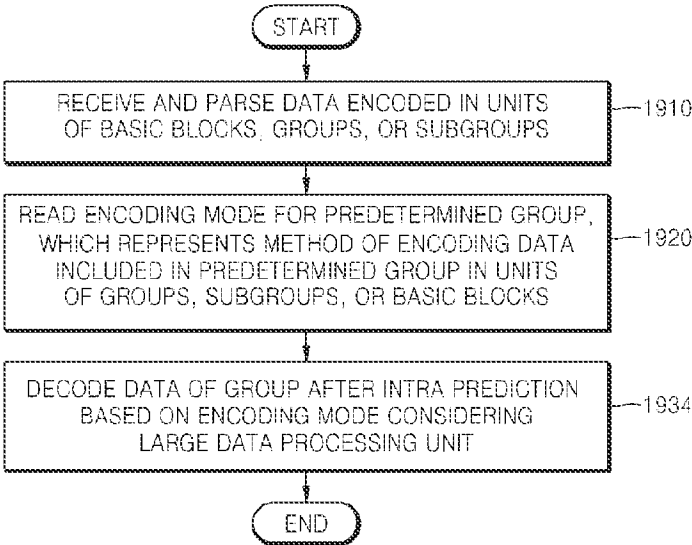


FIG. 32D

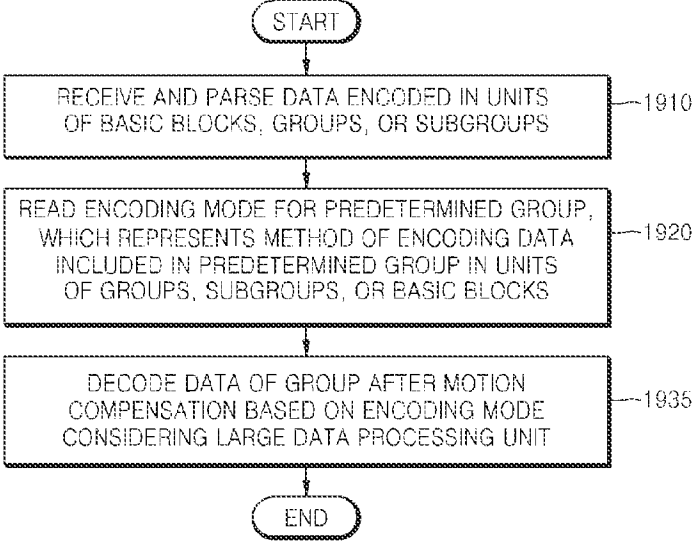


FIG. 32E

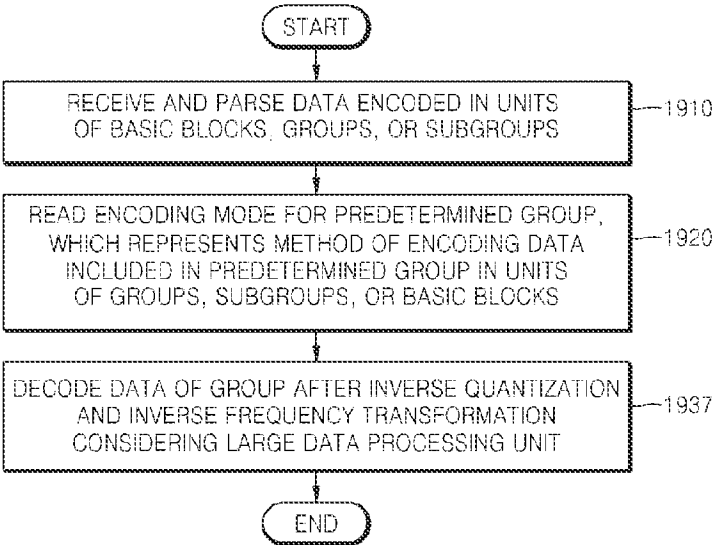
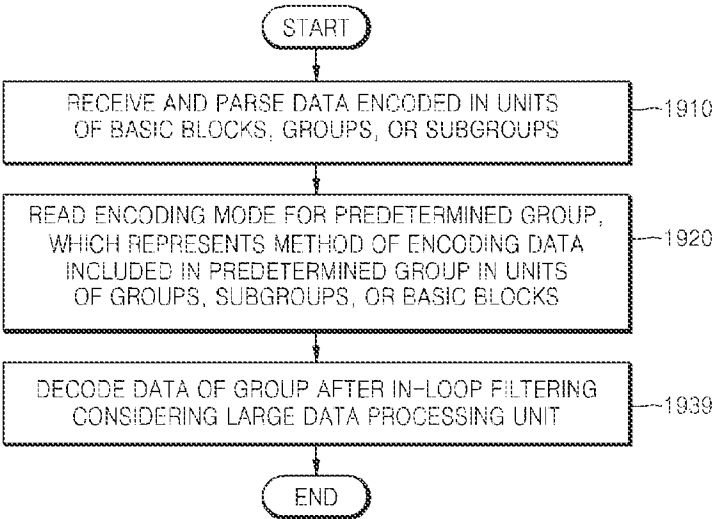


FIG. 32F





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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02

CONFIRMATION NO. 2654

PUBLICATION NOTICE



15800
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

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

Publication No.US-2021-0385460-A1
Publication Date:12/09/2021

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <https://portal.uspto.gov/pair/PublicPair>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Examiner-Initiated Interview Summary	Application No. 17/351,649	Applicant(s) Zheng et al.		
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (First Inventor to File) Status No	Page 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	Title	Type
Tyler W Sullivan	Primary Examiner	Telephonic
Brian Hubbard (Reg. No. 45,873)	Attorney of Record	

Date of Interview: 12 July 2022

Issues Discussed:

Proposed Amendment(s)

Examiner contacted Attorney to discussed a proposed amendment to place Application in condition for Allowance. Attorney and Examiner discussed the proposal of adding the last limitation from the allowed independent claims (regarding the selection of the spatial intra partition type from a group comprising an indication of using a MPM (most probable mode) or not performing intra prediction) in US Application 15/806,893 (now US Patent #11,082,697) and to file a Terminal Disclaimer. Attorney contacted the Examiner on July 18th, 2022 to notify the Examiner the Applicant declines the Examiner's offer.

No agreement regarding patentability of the claims was reached.

Non-statutory Double Patenting

Examiner noted an electronic Terminal Disclaimer would need to be filed to alleviate Obviousness-type Double Patenting against US Patent #11,082,697.

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	
Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04 Please further see: MPEP 713.04 Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b) 37 CFR § 1.2 Business to be transacted in writing	

Applicant recordation instructions:It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	08/18/2022	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	PAPER NUMBER
Suite 1700			2487	
Philadelphia, PA 19103			MAIL DATE	DELIVERY MODE
			08/18/2022	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Response to Amendment

Applicant on June 18th, 2021 filed a Preliminary Amendment.

Applicant cancelled claims 1 – 19.

Applicant added new claims 20 – 39.

The pending claims are 20 – 29.

The Applicant on August 23rd, 2021 filed an Amended Specification. Objections made will be directed towards the August 23rd, 2021 Specification.

Priority

Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, 365(c), or 386(c) is acknowledged (US Provisional Application 61/222,177 filed July 1st, 2009).

Information Disclosure Statement

The information disclosure statements (IDS) submitted on June 18th, 2021 and October 26th, 2021 were filed before the mailing date of the First Action on the Merits (this Office Action). The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the Examiner.

Due to the excessively lengthy Information Disclosure Statement submitted by applicant, the examiner has given only a cursory review of the listed references. In accordance with MPEP 609.04(a), applicant is encouraged to provide a concise explanation of why the information is being submitted and how it is understood to be relevant. Concise explanations (especially those which point out the relevant

pages and lines) are helpful to the Office, particularly where documents are lengthy and complex and applicant is aware of a section that is highly relevant to patentability or where a large number of documents are submitted and applicant is aware that one or more are highly relevant to patentability. Applicant is required to comply with this statement for any non-English language documents. See 37 CFR § 1.56 Duty to Disclose Information Material to Patentability.

Claim Interpretation – Functional Analysis

The following is a quotation of 35 U.S.C. 112(f):

(f) Element in Claim for a Combination. – An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

The following is a quotation of pre-AIA 35 U.S.C. 112, sixth paragraph:

An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

The claims in this application are given their broadest reasonable interpretation using the plain meaning of the claim language in light of the specification as it would be understood by one of ordinary skill in the art. The broadest reasonable interpretation of a claim element (also commonly referred to as a claim limitation) is limited by the description in the specification when 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is invoked.

As explained in MPEP § 2181, subsection I, claim limitations that meet the following three-prong test will be interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph:

- (A) the claim limitation uses the term “means” or “step” or a term used as a substitute for “means” that is a generic placeholder (also called a nonce term or a non-structural term having no specific structural meaning) for performing the claimed function;
- (B) the term “means” or “step” or the generic placeholder is modified by functional language, typically, but not always linked by the transition word “for” (e.g., “means for”) or another linking word or phrase, such as “configured to” or “so that”; and

- (C) the term “means” or “step” or the generic placeholder is not modified by sufficient structure, material, or acts for performing the claimed function.

Use of the word “means” (or “step”) in a claim with functional language creates a rebuttable presumption that the claim limitation is to be treated in accordance with 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph. The presumption that the claim limitation is interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is rebutted when the claim limitation recites sufficient structure, material, or acts to entirely perform the recited function.

Absence of the word “means” (or “step”) in a claim creates a rebuttable presumption that the claim limitation is not to be treated in accordance with 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph. The presumption that the claim limitation is not interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, is rebutted when the claim limitation recites function without reciting sufficient structure, material or acts to entirely perform the recited function.

Claim limitations in this application that use the word “means” (or “step”) are being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, except as otherwise indicated in an Office action. Conversely, claim limitations in this application that do not use the word “means” (or “step”) are not being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, except as otherwise indicated in an Office action.

This application includes one or more claim limitations that use the word “means” or “step” or a generic placeholder, but are nonetheless **not** being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph because the claim limitation(s) recite(s) sufficient structure, materials, or acts to entirely perform the recited function.

Such claim limitation(s) is/are: “processors are configured to” in claims 25 – 27 and 36 – 38.

The Examiner notes one of ordinary skill in the art would understand a “processor” in the context of the claimed invention as connoting sufficient structure.

Because this/these claim limitation(s) is/are **not** being interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, it/they is/are **not** being interpreted to cover only the corresponding structure, material, or acts described in the specification as performing the claimed function, and equivalents thereof.

If applicant intends to have this/these limitation(s) interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, applicant may: (1) amend the claim limitation(s) to remove the structure, materials, or acts that performs the claimed function; or (2) present a sufficient showing that the claim limitation(s) does/do not recite sufficient structure, materials, or acts to perform the claimed function.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP § 2146 *et seq.* for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

Claims 20 – 39 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1 – 6 of U.S. Patent No. #11,082,697.

Although the claims at issue are not identical, they are not patentably distinct from each other because of at least the reasons given in the table below:

Note: In the interest of brevity the Examiner does not cite identical or similar claim limitations in the analysis.

Pending Application	US Patent #11,082,697 Published Claims
Claim 20) See claim 25 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 21) See claim 26 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 22) See claim 27 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 23) See claim 28 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 24) See claim 29 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 25) 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 32x32, and the large block size is greater than a basic coding unit size, [The patented claims render the pending claim as obvious as providing specific examples and sizes	Claim 1) 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,

<p>as those contemplated in the Specification of the pending Application thus the Patented claims render obvious the pending limitation to one of ordinary skill in the art.]</p> <p>wherein the intra prediction is signaled for the large block and the one or more processors are configured to:</p> <p>decode a binary split signaling syntax element specifying that the large block is further split into sub-blocks;</p> <p>partition the large block into sub-blocks in response to the decoded binary split signaling syntax element; and</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “where said binary split signaling syntax element [...] is not further split” and the next “otherwise” limitations) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p> <p>decode 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.</p>	<p>wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:</p> <p>decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p> <p>decode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types,</p> <p>wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 26) The apparatus of claim 25, wherein the one or more processors are configured to</p> <p>decode an intra prediction mode for the basic coding unit sized sub-block.</p> <p>[The Patented claims render obvious associated an intra prediction mode with a sub block and partitioning to sub blocks with their own intra prediction mode (and even the use of signaling if the MPM (most probable mode)) for subblocks further rendering obvious the pending claim to one of ordinary skill in the art.]</p>	<p>Claim 1) [Relevant portions only]</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally</p>

	<p>basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p> <p>decode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types,</p> <p>wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 27) The apparatus of claim 25, wherein the one or more processors are configured to</p> <p>decode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.'</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the "otherwise" limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p>	<p>Claim 1) [Relevant portions only]</p> <p>decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p>
<p>Claim 28) The apparatus of claim 25, wherein the binary split signaling syntax element specifies that the large block is further split into four equally sized sub-blocks.</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the "otherwise" limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p>	<p>Claim 1) [Relevant portions only]</p> <p>decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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</p>

	case where said sub-block is 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and
Claim 29) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit. [One of ordinary skill in the art readily recognizes the single intra partition type for the sub blocks to be partitioned into the basic coding unit as indications of the size specified in using the spatial intra prediction types and the features in the Patented claim renders obvious the same features and semantics / meaning in the pending Application.]	Claim 1) [Relevant portions only] decode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.
Claim 30) The apparatus of claim 25, 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.	Claim 2) The video decoder 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.
Claim 31) The apparatus of claim 25, 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. [Patented claim 1 already recited the use of processors and the picture data received being processed thus the claims are obvious variants to one of ordinary skill in the art.]	Claim 3) The video decoder 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.
Claim 32) See claim 36 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 33) See claim 37 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 34) See claim 38 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 35) See claim 39 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 36) An apparatus comprising:	Claim 4) A video encoder, comprising

<p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size,</p> <p>[The patented claims render the pending claim as obvious as providing specific examples and sizes as those contemplated in the Specification of the pending Application thus the Patented claims render obvious the pending limitation to one of ordinary skill in the art.]</p> <p>wherein the intra prediction is signaled for the large block and the one or more processors are configured to:</p> <p>encode a binary split signaling syntax element specifying that the large block is further split into sub-blocks:</p> <p>partition the large block into sub-blocks in response to the binary split signaling syntax element; and</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “where said binary split signaling syntax element [...] is not further split” and the next “otherwise” limitations) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p> <p>encode 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.</p>	<p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>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,</p> <p>wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 subblock in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and</p> <p>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, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
---	--

<p>Claim 37) The apparatus of claim 36, wherein, the one or more processors are configured to</p> <p>encode an intra prediction mode for the basic coding unit sized sub-block.</p> <p>[The Patented claims render obvious associated an intra prediction mode with a sub block and partitioning to sub blocks with their own intra prediction mode (and even the use of signaling if the MPM (most probable mode)) for subblocks further rendering obvious the pending claim to one of ordinary skill in the art.]</p>	<p>Claim 4) [Relevant Portions only]</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and encode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p> <p>encode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 38) The apparatus of claim 36, wherein, the one or more processors are configured to</p> <p>encode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks.</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “otherwise” limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p>	<p>Claim 4) [Relevant Portions only]</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and encode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax</p>

	element specifies said 32×32 sub-block is not further split; and
Claim 39) The method of claim 36, wherein a spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block. [One of ordinary skill in the art readily recognizes the single intra partition type for the sub blocks to be partitioned into the basic coding unit as indications of the size specified in using the spatial intra prediction types and the features in the Patented claim renders obvious the same features and semantics / meaning in the pending Application.]	Claim 4) [Relevant Portions only] encode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to understand the Patented claims as providing teachings of the specific techniques to perform quadtree partitioning of a large block (e.g. 32 x 32) block rendering obvious the scope of the pending claims. Further, one of ordinary skill in the art would recognize the decoder claims as the inverse of the encoding claims and thus the claims are obvious variants.

Claim Objections

Claims 20 – 39 are objected to because of the following informalities:

Regarding claim 25, the claim does not constructive indicate the differences in processing the large block after the decision / determination of partitioning as if the block isn't partitioned it is not therefore into sub-blocks and thus has no description of how to process the larger blocks with the spatial intra partition types and intra prediction mode. Further, the conditionals stated appear open-ended and thus appear to lack patentable weight.

Regarding claim 20, see claim 25 which is the apparatus performing the steps of the claimed method and thus is similarly Objected.

Regrading claims 32 and 36, see claims 20 and 25 respectively as the decoder performing the inverse steps of the claimed encoder and thus are similarly Objected.

Regarding claims 21 – 24, 26 – 31, 33 – 35, and 37 – 39, the dependent claims do not cure the deficiencies of their respective independent claims and thus are similarly Objected.

Regarding claim 25, the claim does not appear to have no conditions or limitations on the sub-block size thus the claim appears to encompass more than what is disclosed raising scope of enablement issues (e.g. rectangular partitions).

Regarding claim 20, see claim 25 which is the apparatus performing the steps of the claimed method and thus is similarly Objected.

Regrading claims 32 and 36, see claims 20 and 25 respectively as the decoder performing the inverse steps of the claimed encoder and thus are similarly Objected.

Regarding claims 21 – 24, 26 – 31, 33 – 35, and 37 – 39, the dependent claims do not cure the deficiencies of their respective independent claims and thus are similarly Objected.

Regarding claim 21, the claim appears to not further limit independent claim 20 in the case the larger block is not partitioned (no constructive steps / processing claimed how to treat the intra prediction mode in cases of using large blocks).

Regarding claim 26, the claim appears to not further limit independent claim 25 in the case the larger block is not partitioned (no constructive steps / processing claimed how to treat the intra prediction mode in cases of using large blocks).

Regarding claim 33, the claim appears to not further limit independent claim 32 in the case the larger block is not partitioned (no constructive steps / processing claimed how to treat the intra prediction mode in cases of using large blocks).

Regarding claim 37, the claim appears to not further limit independent claim 26 in the case the larger block is not partitioned (no constructive steps / processing claimed how to treat the intra prediction mode in cases of using large blocks).

Regarding claim 22, the claim appears to be missing an Essential Step that the first / previous binary split signaling syntax indicated the current sub-block was partitioned from the large sub-block indicated a partition was to occur. Further, the claim only recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 27, the claim appears to be missing an Essential Step that the first / previous binary split signaling syntax indicated the current sub-block was partitioned from the large sub-block indicated a partition was to occur. Further, the claim only recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 34, the claim appears to be missing an Essential Step that the first / previous binary split signaling syntax indicated the current sub-block was partitioned from the large sub-block

indicated a partition was to occur. Further, the claim only recites half of the processing determination (to perform a second split but lacks and Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 38, the claim appears to be missing an Essential Step that the first / previous binary split signaling syntax indicated the current sub-block was partitioned from the large sub-block indicated a partition was to occur. Further, the claim only recites half of the processing determination (to perform a second split but lacks and Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 22, 27, 34, and 38 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor (or for applications subject to pre-AIA 35 U.S.C. 112, the applicant), regards as the invention.

Claim 22 recites the limitation "a binary split signaling syntax element" in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claim 27 recites the limitation "a binary split signaling syntax element" in lines 1 – 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 34 recites the limitation "a binary split signaling syntax element" in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claim 38 recites the limitation "a binary split signaling syntax element" in lines 1 – 2. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

This application currently names joint inventors. In considering patentability of the claims under pre-AIA 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of pre-AIA 35 U.S.C. 103(c) and potential pre-AIA 35 U.S.C. 102(e), (f) or (g) prior art under pre-AIA 35 U.S.C. 103(a).

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 20, 22 – 25, 27 – 29, 32, 34 – 36, and 38 – 39 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Chen, et al. (US PG PUB 2010/0086029 A1 referred to as "Chen" throughout in which citations will come from US Provisional Applications 61/166,631 (referred to as "Chen '631" throughout) and 61/144,357 (referred to as "Chen '631" throughout)) [The US PG PUB is cited in Applicant's June 18th, 2021 IDS as US PG PUB Item #24 – The Examiner will cite from the US Provisional References where "Chen" refers to the collective of "Chen '631" and "Chen '357"], and further in view of

Moriya, et al. (US PG PUB 2009/0046781 A1 referred to as “Moriya” throughout) [Cited in Applicant’s June 18th, 2021 IDS as US PG PUB Items #20].

Regarding claim 20, see claim 25 which is the apparatus performing the steps of the claimed method.

Regarding claim 22, see claim 27 which is the apparatus performing the steps of the claimed method.

Regarding claim 23, see claim 28 which is the apparatus performing the steps of the claimed method.

Regarding claim 24, see claim 29 which is the apparatus performing the steps of the claimed method.

Regarding claim 32, see claim 36 which is the apparatus performing the steps of the claimed method.

Regarding claim 34, see claim 38 which is the apparatus performing the steps of the claimed method.

Regarding claim 35, see claim 39 which is the apparatus performing the steps of the claimed method.

Regarding claim 25, Chen ‘631 teaches processing large macroblocks with split and merge considerations. Chen ‘357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen ‘631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

at least a memory and one or more processors [Chen '631 Paragraphs 65 and 186 – 188 (microprocessors / DSPs and memory elements used to implement)], 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 [Chen '631 Figures 3 – 5 (decoder includes intra mode processing); Chen '357 Figure 3 as well as Sections 3.1 and 3.2)],

wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size [Chen '631 Figures 4, 5, 10, 15, and 16 (including subfigures) and Paragraphs 20 (16 x 16 coding unit contemplated at least as a basic coding unit size), 27 (32 x 32 or 64 x 64 large macroblocks contemplated as large block sizes), 73 – 75, 96 – 104 (32 x 32 and 64 x 64 large block sizes contemplated as well as a base I_16x16 block size contemplated for intra prediction); Chen '357 Figures 2 – 4 and Section 3.2 (32 x 32 large block illustrated with 16 x 16 blocks illustrated as a base for merging or splitting in Figure 3 in which each subblock is intra coded)],

wherein the intra prediction is signaled for the large block [Chen '357 Figure 3 as well as Sections 3.1 and 3.2.1 (two methods of intra mode signaling for large macroblocks); Chen '631 Figures 4, 5, 10, 12, and 15 as well as Paragraphs 104 – 106 and 111 (intra coding considering larger macroblock sizes to partition to smaller block sizes)] 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 sub-blocks [Chen '631 Figures 3 – 8 (including subfigures – checks to create four equally sized subblocks partitioned from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes considered using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 - 121 (hierarchically partitioning a large macroblock by using tree coding and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 - 159)) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks as is also done in Figure 3 where intra prediction is used and spatial intra partition types are assigned)];

partition the large block into sub-blocks in response to the decoded binary split signaling syntax element [Chen '631 Figures 3 – 8 (including subfigures - check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)]; and

decode 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 [Chen '631 Figure 3 as well as Paragraphs 50 and 102 – 104 (“e.g., one of I_16x16, I_8x8, I_4x4” are the plurality of spatial intra partition types); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 where Moriya in Figures 2 – 3 and 6 illustrates quadtree partitioning of large blocks (e.g. 32x32 blocks) and in Figure 7 (see at least reference characters 29 and 30) as well as Paragraphs 45 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks decoding the encoded information), 63 – 64 (intra block sizes partitioned / determined in the decoder based on encoded information), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) in the decoder using encoded information)].

The motivation to combine Chen '357 with Chen '631 is to combine features in the same field of invention of video coding / decoding [Chen '357 Section 1] in order to code / decode large macroblocks to

improve coding for higher resolution images / HD capabilities [Chen '357 Section 2 where the Examiner observes KSR Rationales (C), (D), or (F) are also applicable].

The motivation to combine Moriya with Chen '357 and Chen '631 is to combine features in the same / similar field of invention of coding and decoding image data [Moriya Paragraphs 1 and 4] in order to improve compression efficient by reducing the amount of coding information (e.g. intra mode used) to transmit as side / header information with the block [Moriya Paragraphs 5 – 9 where the Examiner observes at least KSR Rationales (D) or (F) are also applicable].

This is the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya which will be used throughout the Rejection.

Regarding claim 27, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the one or more processors are configured to [See claim 25 for citations of the claimed "processors"] decode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks [Chen '631 Figures 3 – 8 (including subfigures - check four equally sized blocks partitioning from large block), 12 – 13, and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a

large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 28, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the binary split signaling syntax element specifies that the large block is further split into four equally sized sub-blocks [Chen '631 Figures 4 - 8 (including subfigures - check four equally sized blocks partitioning from large block), 12 – 13, and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate

partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 29, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the spatial intra partition type specifies a size of a partition of the basic coding unit [Chen '631 Paragraphs 50 and 102 – 104 (“e.g., one of $l_{16 \times 16}$, $l_{8 \times 8}$, $l_{4 \times 4}$ ” are the plurality of spatial intra partition types) and 115 – 121 (signaling intra spatial partition information within a basic coding unit (e.g. partitions within a 16×16 block)); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 (see the 32×32 block partitioned into 16×16 blocks in which spatial intra modes indicate partitions / sizes within the 16×16 partitions for intra predication) where Moriya in Figures 2 – 3 and 6 illustrates quadtree partitioning of large blocks (e.g. 32×32 blocks) and in Figure 7 (see at least reference characters 29 and 30) as well as Paragraphs 45 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32×32 or 8×8 and additionally 16×16 and 4×4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)),

48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks decoding the encoded information), 63 – 64 (intra block sizes partitioned / determined in the decoder based on encoded information), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) in the decoder using encoded information)).

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 36, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

at least a memory and one or more processors [Chen '631 Paragraphs 65 and 186 – 188 (microprocessors / DSPs and memory elements used to implement)], 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 [Chen '631 Figure 2, 4, and 5 as well as Paragraphs 100 -1 06 (encoding / decoding large blocks and formation of large blocks); Chen '357 Figure 3 as well as Sections 3.1 and 3.2)],

wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size [Chen '631 Figures 4, 5, 10, 12, 13, 15, and 16 (including subfigures) and Paragraphs 20 (16 x 16 coding unit contemplated at least), 27 (32 x 32 and 64 x 64 macroblocks contemplated), 73 – 75, 96 – 104 (32 x 32 and 64 x 64 initial block sizes contemplated as well as a base 1_16x16 block size / unit contemplated for intra prediction), and 182 – 186 (partitioning

larger blocks and signaling thereof); Chen '357 Figure 2 – 4 and Section 3.2 (64 x 64 blocks illustrated with 16 x 16 blocks illustrated as a base for merging or splitting in Figure 3)],

wherein the intra prediction is signaled for the large block [Chen '357 Figure 3 as well as Sections 3.1 and 3.2.1 (two methods of intra mode signaling for large macroblocks); Chen '631 Figures 4, 5, 10, 12, and 15 as well as Paragraphs 104 – 106 and 111 (intra coding considering larger macroblock sizes to partition to smaller block sizes)] 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 sub-blocks [Chen '631 Figures 4 - 8 (including subfigures - check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 - 121 (hierarchically partitioning a large macroblock by using tree coding and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 - 159)) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 - 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)];

partition the large block into sub-blocks in response to the binary split signaling syntax element [Chen '631 Figures 4 – 8 (including subfigures - check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)]; and

encode 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 [Chen '631 Figure 2 as well as Paragraphs 50 and 102 – 104 (“e.g., one of I_16x16, I_8x8, I_4x4” are the plurality of spatial intra partition types); Chen '357 Sections 3.2 – 3.3 and Figures 2 and 3 where Moriya in Figures 1 – 3 (see at least reference characters 1, 2, 5, 15, and 16) and 6 illustrates quadtree partitioning of large blocks (e.g. 32x32 blocks) as well as Paragraphs 42 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode in the encoded information), 63 – 64 (intra block sizes partitioned / determined in the encoder), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) as generated encoded information)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya where claim 25 is the decoder processing the encoded information generation and in view of at least Chen '631 Paragraphs 73 – 75 (decoding is the inverse / reverse process of encoding as readily recognized by one of ordinary skill in the art) and thus similar motivation exists.

Regarding claim 38, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

wherein, the one or more processors are configured to [See claim 36 for citations of the claimed “processors”] encode a binary split signaling syntax element specifying that the sub-blocks are further split into smaller sub-blocks [Chen '631 Figures 2, 4 – 8 (including subfigures – check four equally sized

blocks partitioning from large block), 12 – 13, and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)].

See claim 36 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 39, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

wherein a spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block [Chen '631 Figure 2 as well as Paragraphs 50 and 102 – 104 (“e.g., one of I_16x16, I_8x8, I_4x4” are the plurality of spatial intra partition types) and 115 – 121 (signaling intra spatial partition information within a basic coding unit (e.g. partitions within a 16x16 block)); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 (see the 32x32 block partitioned into 16x16 blocks in which spatial intra modes indicate partitions / sizes within the 16x16 partitions for intra predication) where Moriya in Figures 1 – 3 and 6 (see at least reference characters 1, 2, 5, 15, and 16) illustrates quadtree partitioning of large blocks (e.g. 32x32

blocks) as well as Paragraphs 42 – 47 (selection of intra prediction block size in the large block in which a plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks), 63 – 64 (intra block sizes partitioned / determined based on encoded information), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) from generated encoded information)].

See claim 36 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Claim 21, 26, 30 – 31, 33, and 37 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Chen (collectively), Moriya, and further in view of Moriya, et al. (US PG PUB 2008/0123947 A1 referred to as "Moriya 47" throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #16 and 17].

Regarding claim 21, see claim 26 which is the apparatus performing the steps of the claimed method.

Regarding claim 33, see claim 37 which is the apparatus performing the steps of the claimed method.

Regarding claim 26, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the one or more processors are configured to [See claim 25 for citations of the claimed "processors"] decode an intra prediction mode for the basic coding unit sized sub-block [Chen '631 Figures 4 and 12 (subfigures included) as well as Paragraphs 102 – 106 (prediction modes a function of the spatial prediction mode used (e.g. a described in Paragraph 103) where the selection of the basic coding unit is rendered obvious (e.g. I_4x4 partition type and its associated intra modes would be a selection of the finite number of possibilities to one of ordinary skill in the art)) and 162 – 165; Chen '357 Figures 2 – 3 and Section 3.2.1; Moriya Figure 7 (see at least reference character 29 and 30 which is to be modified / elaborated by Moriya 47 teaching of the intra prediction modes) as well as Paragraphs 32 (intra prediction performing on a block type), 44 – 46, 55 – 56 (encoding the intra prediction mode); Moriya 47 Figures 3, 4 and 13 (4x4, 8x8 and 16x16 intra prediction modes (such as DC and directional prediction) illustrated) as well as Paragraphs 218 – 237 (signaling mode information for partitions to be predicted with intra prediction), 264 – 281 (discussion of the 8x8 intra prediction modes in contrast to the 16x16 intra prediction modes which is encoded to later be decoded), 300 – 304 (decoded encoded intra prediction mode information)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

The motivation to combine Moriya 47 with Moriya, Chen '357, and Chen '631 is to combine features in the same / related field of invention of "a digital image encoding method, and a [...] decoding method" [Moriya 47 Paragraph 2] to improve efficiency in performing intra prediction [Moriya 47 in at least Paragraphs 279, 295, and 396] and to improve flexibility in selecting intra prediction mode information to

use [Moriya 47 Paragraph 12 where the Examiner observes at least KSR Rationales (C) or (D) are also applicable].

This is the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47 which will be used throughout the Rejection.

Regarding claim 30, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

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 [See claim 25 for citations regarding the “processor” claimed; See claim 25 for citations of the “spatial partition intra partition type” and claim 26 for citations of the “intra prediction mode” and regarding the treatment of table representations see additionally Moriya 47 Figures 3 – 8, 13 (Figures 3 – 4 and 13 show the intra prediction modes), and 26 – 28 (tables used for signaling the intra prediction mode of a current block) as

well as Paragraph 251 – 255 (decoding process in Paragraph 253 is modified to incorporate the table used for encoding in Paragraph 254) and also in Paragraph 254 ("select a code table" implies the tables are known a priori before usage of the table), 375 – 385 (tables used to signal intra prediction modes for processing in a decoder), and 392 – 400 (tables including decoder usage of intra modes signaled with tables)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47.

Regarding claim 31, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

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 [See claim 25 for citations regarding

the "processor" claimed; See claim 30 for citations of the claimed "table" feature of the representation of the plurality of spatial intra partition types (claims 25 and 30) or intra prediction modes (claim 26) and additionally Chen '631 Paragraphs 92 and 177 (levels of syntax elements taught); and Moriya 47 Figures 3 – 8 (see at least reference character 31) 72, and 82 – 86 (signaling including high level syntax elements / organization to insert partition and prediction information) as well as Paragraphs 253 – 255 (signaling intra prediction tables in "a slice, picture, or a sequence" which are "upper data layer" and thus would be recognized by one of ordinary skill in the art as a high level syntax element when afforded the broadest reasonable interpretation in view of Specification Paragraph 40 (using the numbering in the US PG PUB) defining the term), 523 – 533 (signaling and decoding upper data layer / high level syntax elements for intra prediction) and 621 – 622 (high level syntax elements describing blocks encoded to be decoded)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47.

Regarding claim 37, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The combination teaches

wherein, the one or more processors are configured to [See claim 25 for citations of the claimed “processors”] encode an intra prediction mode for the basic coding unit sized sub-block [Chen ‘631 Figures 4 and 12 (subfigures included) as well as Paragraphs 102 – 106 (prediction modes a function of the spatial prediction mode used (e.g. a described in Paragraph 103) where the selection of the basic coding unit is rendered obvious (e.g. I_4x4 partition type and its associated intra modes would be a selection of the finite number of possibilities to one of ordinary skill in the art)) and 162 – 165; Chen ‘357 Figures 2 – 3 and Section 3.2.1; Moriya Figure 1 (see at least reference character 15 and 16 which is to be modified / elaborated by Moriya 47 teaching of the intra prediction modes) as well as Paragraphs 32 (intra prediction performing on a block type), 44 – 46, 55 – 56 (encoding the intra prediction mode); Moriya 47 Figures 3, 4 and 13 (4x4, 8x8 and 16x16 intra prediction modes (such as DC and directional prediction) illustrated) as well as Paragraphs 218 – 237 (signaling mode information for partitions to be predicted with intra prediction), 264 – 281 (discussion of the 8x8 intra prediction modes in contrast to the 16x16 intra prediction modes which is encoded to later be decoded), 300 – 304 (encoded intra prediction mode information)].

See claim 26 for the motivation to combine Chen (Chen ‘631 and Chen ‘357 collectively), Moriya, and Moriya 47 where claim 26 is the decoder processing the encoded information generation and in view of at least Chen ‘631 Paragraphs 73 – 75 (decoding is the inverse / reverse process of encoding as readily recognized by one of ordinary skill in the art) and thus similar motivation exists.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Po, et al. (US PG PUB 2009/0219991 A1 referred to as “Po” throughout) teaches in Figures 3 – 4 and Paragraphs 8 at least the decisions to merge / split blocks with hierarchy of intra prediction modes / splits.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tyler W Sullivan whose telephone number is (571)270-5684. The examiner can normally be reached IFP.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Czekaj can be reached on (571)-272-7327. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: <https://patentcenter.uspto.gov>. Visit <https://www.uspto.gov/patents/apply/patent-center> for more information about Patent Center and <https://www.uspto.gov/patents/docx> for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

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

<i>Notice of References Cited</i>	Application/Control No. 17/351,649		Applicant(s)/Patent Under Reexamination Zheng et al.	
	Examiner Tyler W Sullivan		Art Unit 2487	Page 1 of 1

U.S. PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20090219991-A1	09-2009	Po; Lai-Man	H04N19/147	375/E7.126
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS						
*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)				
	U					
	V					
	W					
	X					

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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.		
/TWS/	54	2007/034918	WO	A1	2007-03-29	Mitsubishi Electric Corporation		
/TWS/	55	2007/081908	WO	A1	2007-07-19	Thomson Licensing		
/TWS/	56	2008/004769	WO	A1	2008-01-10	Samsung Electronics Co., Ltd.		
/TWS/	57	2008/027192	WO	A2	2008-03-06	Thomson Licensing		
/TWS/	58	2008/088140	WO	A1	2008-07-24	Samsung Electronics Co., Ltd.		
	59	2009/028922	WO	A2	2009-03-05	LG Electronics Inc.		
/TWS/	60	2009/051719	WO	A2	2009-04-23	Thomson Licensing		
/TWS/	61	2010/039731	WO	A2	2010-04-08	Qualcomm Incorporated		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/ 	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/ <div></div>	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	/TYLER W. SULLIVAN/	Date Considered	07/18/2022
*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.			

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

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

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

/TYLER W. SULLIVAN/ 07/18/2022

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				Application Number		17/351,649	
				Filing Date		Jun 18, 2021	
				First Named Inventor		Yunfei Zheng	
				Art Unit		Not Yet Assigned	
				Examiner Name		Not Yet Assigned	
Sheet	1	of	1	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
/TWS/	74	2293567	EP	A2	2011-03-09	SAMSUNG ELECTRONICS CO., LTD.		

EXAMINER SIGNATURE			
Examiner Signature	/TYLER W. SULLIVAN/	Date Considered	07/18/2022
*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.			

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

Electronic Acknowledgement Receipt	
EFS ID:	47391244
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-JAN-2023
Filing Date:	18-JUN-2021
Time Stamp:	17:09:26
Application Type:	Utility under 35 USC 111(a)

Payment information:	
Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 640
RAM confirmation Number	E20231HH09504483
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	Extension of Time	ICVW_2009P00073_US02_EOT.pdf	165957	no	2
			b0f5f56c87aee489aed7211f1de733b7a8741e53		
Warnings:					
Information:					
2		ICVW_2009P00073_US02_Response_After_8-18-2022_Non-Final_Office_Action.pdf	187664	yes	8
			41fcd4eb7d497c8a7c706cc2ab27a551aa0b1dec		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Amendment/Request for Reconsideration-After Non-Final Rejection		1	1	
	Claims		2	5	
	Applicant Arguments/Remarks Made in an Amendment		6	8	
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	39105	no	2
			e13f10e58118d5b14a723ba913b2ac6bf98b263e		
Warnings:					
Information:					
Total Files Size (in bytes):			392726		

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.

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.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) ICVW_2009P00073 US02																														
Application Number 17/351,649	Filed June 18, 2021																															
For METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS																																
Art Unit 2487	Examiner Sullivan, Tyler																															
<p>This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application.</p> <p>The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):</p> <table><thead><tr><th></th><th>Fee</th><th>Small Entity Fee</th><th>Micro Entity Fee</th><th></th></tr></thead><tbody><tr><td>☐ One month (37 CFR 1.17(a)(1))</td><td>\$220</td><td>\$110</td><td>\$55</td><td>\$ _____</td></tr><tr><td>☒ Two months (37 CFR 1.17(a)(2))</td><td>\$640</td><td>\$320</td><td>\$160</td><td>\$ 640.00</td></tr><tr><td>☐ Three months (37 CFR 1.17(a)(3))</td><td>\$1,480</td><td>\$740</td><td>\$370</td><td>\$ _____</td></tr><tr><td>☐ Four months (37 CFR 1.17(a)(4))</td><td>\$2,320</td><td>\$1,160</td><td>\$580</td><td>\$ _____</td></tr><tr><td>☐ Five months (37 CFR 1.17(a)(5))</td><td>\$3,160</td><td>\$1,580</td><td>\$790</td><td>\$ _____</td></tr></tbody></table> <p>☐ Applicant asserts small entity status. See 37 CFR 1.27.</p> <p>☐ Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.</p> <p>☐ A check in the amount of the fee is enclosed.</p> <p>☐ Payment by credit card. Form PTO-2038 is attached.</p> <p>☒ The Director has already been authorized to charge fees in this application to a Deposit Account.</p> <p>☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-5519 _____.</p> <p>☒ Payment made via EFS-Web.</p> <p>WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.</p> <p>I am the</p> <p>☐ applicant.</p> <p>☒ attorney or agent of record. Registration number 45,873 _____.</p> <p>☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.</p> <p>/Brian J. Hubbard/ Jan. 18, 2023 Signature Date</p> <p>Brian J. Hubbard 215-558-5740 Typed or printed name Telephone Number</p> <p>NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.</p>				Fee	Small Entity Fee	Micro Entity Fee		☐ One month (37 CFR 1.17(a)(1))	\$220	\$110	\$55	\$ _____	☒ Two months (37 CFR 1.17(a)(2))	\$640	\$320	\$160	\$ 640.00	☐ Three months (37 CFR 1.17(a)(3))	\$1,480	\$740	\$370	\$ _____	☐ Four months (37 CFR 1.17(a)(4))	\$2,320	\$1,160	\$580	\$ _____	☐ Five months (37 CFR 1.17(a)(5))	\$3,160	\$1,580	\$790	\$ _____
	Fee	Small Entity Fee	Micro Entity Fee																													
☐ One month (37 CFR 1.17(a)(1))	\$220	\$110	\$55	\$ _____																												
☒ Two months (37 CFR 1.17(a)(2))	\$640	\$320	\$160	\$ 640.00																												
☐ Three months (37 CFR 1.17(a)(3))	\$1,480	\$740	\$370	\$ _____																												
☐ Four months (37 CFR 1.17(a)(4))	\$2,320	\$1,160	\$580	\$ _____																												
☐ Five months (37 CFR 1.17(a)(5))	\$3,160	\$1,580	\$790	\$ _____																												

☒ * Total of 1 (one) _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). 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 6 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: Mail Stop PCT, 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 disclosure of these records is required by the Freedom of Information Act.
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 Cooperation Treaty.
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 inspection 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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: TYLER SULLIVAN

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

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY PURSUANT TO 37 CFR § 1.111

In response to the Office Action dated August 18, 2022, reconsideration is respectfully requested in view of the amendments and/or remarks as indicated below:

- ☐ **Amendments to the Specification** begin on page of this paper.
- ☒ **Amendments to the Claims** begin on page 2 of this paper.
- ☐ **Amendments to the Drawings** begin on page of this paper and include an attached replacement sheet.
- ☒ **Remarks** begin on page 6 of this paper.
- ☒ The Commissioner is hereby authorized to charge any fee deficiency, charge any additional fees, or credit any overpayment of fees, associated with this application in connection with this filing, or any future filing, submitted to the U.S. Patent and Trademark Office during the pendency of this application, to Deposit Account No. **50-5519**.

Claims:

1 - 19. (Cancelled)

20. (Currently Amended) 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 32x32, 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;

partitioning the large block into the four sub-blocks in response to the decoded binary split signaling syntax element; and

decoding 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.

21. (Previously Presented) The method of claim 20, further comprising decoding an intra prediction mode for the basic coding unit sized sub-block.

22. (Currently Amended) The method of claim 20, further comprising decoding a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks [[are]] is further split into four smaller equally sized sub-blocks.

23. (Cancelled)

24. (Previously Presented) The method of claim 20, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

25. (Currently Amended) 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 32x32, 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;
partition the large block into the four sub-blocks in response to the decoded binary split signaling syntax element; and
decode 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.

26. (Previously Presented) The apparatus of claim 25, wherein the one or more processors are configured to decode an intra prediction mode for the basic coding unit sized sub-block.

27. (Currently Amended) The apparatus of claim 25, 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 [[are]] is further split into four smaller equally sized sub-blocks.

28. (Cancelled)

29. (Previously Presented) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit.

30. (Previously Presented) The apparatus of claim 25, 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.

31. (Previously Presented) The apparatus of claim 25, 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.

32. (Currently Amended) 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 32x32, 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;

partitioning the large block into the four sub-blocks in response to the decoded binary split signaling syntax element; and

encoding 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.

33. (Previously Presented) The method of claim 32, further comprising encoding an intra prediction mode for the basic coding unit sized sub-block.

34. (Currently Amended) The method of claim 32, further comprising encoding a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks [[are]] is further split into four smaller equally sized sub-blocks.

35. (Currently Amended) The method of claim 32, wherein [[a]] the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

36. (Currently Amended) 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 32x32, 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;

partition the large block into the four sub-blocks in response to the binary split signaling syntax element; and

encode 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.

37. (Previously Presented) The apparatus of claim 36, wherein, the one or more processors are configured to encode an intra prediction mode for the basic coding unit sized sub-block.

38. (Currently Amended) The apparatus of claim 36, 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 ~~[[are]]~~ is further split into four smaller equally sized sub-blocks.

39. (Currently Amended) The method of claim 36, wherein ~~[[a]]~~ the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

REMARKS

Claims 20-22, 24-27, and 29-39 are pending in the application. Claims 20, 22, 25, 27, 32, 34-36, 38, and 39 have been amended. Claims 23 and 28 have been cancelled without prejudice or disclaimer. No new matter has been added.

Applicant agrees that the Interview Summary is accurate and thanks Examiner Sullivan both for reaching out to facilitate prosecution and for the courtesy shown to the undersigned in the interview of July 12, 2022.

Regarding the double patenting rejection (*Office Action* page 5 *et seq.*), Applicant is aware of Office policy regarding continuations, and is not averse to filing a terminal disclaimer in favor of the parent application, but merely suggests that as the claims continue to be amended, the analysis will have to be revised.

As to the objections (*Office Action* page 13 *et seq.*), they are moot by amendment, however, if they are to be repeated, the undersigned respectfully suggests the Examiner clarify his position. For example:

Regarding claim 25, the claim does not constructive indicate the differences in processing the large block after the decision / determination of partitioning as if the block isn't partitioned it is not therefore into sub-blocks and thus has no description of how to process the larger blocks with the spatial intra partition types and intra prediction mode. Further, the conditionals stated appear open-ended and thus appear to lack patentable weight.

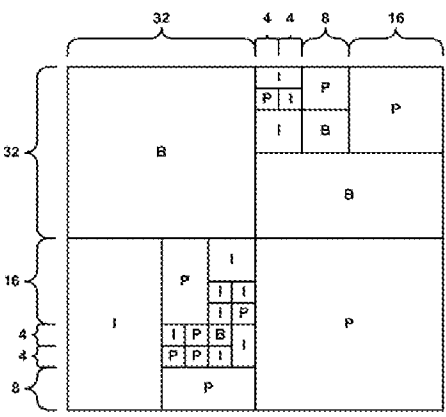
Office Action page 13. One need not claim every potential outcome of a process, and in fact, applicants should write their claims succinctly. For example, for Claim 25, the claim seeks to protect *determining that intra prediction is to be performed for a [specified] large block; decode a binary split signaling syntax element specifying [a defined split]; partition the large block into sub-blocks in response to the decoded binary split signaling syntax element; and decode a single spatial intra partition type for at least one basic coding unit sized sub-block*. In this context, it's hard to understand the allusion to blocks that aren't partitioned¹, allegedly "open ended" conditionals, and how to process unclaimed features. A claim is not a recipe. However, if the claims are somehow internally contradictory, the Examiner is invited to specify.

¹ Nothing in this Remarks section should be construed as a dedication to the public. Applicant reserves the right, in the future, to decide whether to pursue claims directed to what happens when the binary split signaling syntax element specifies that the large block is further not split.

To the extent the obviousness rejections are not moot by amendment, Applicant respectfully traverses. The combination of Chen, Chen, and Moriya do not establish a *prima facie* case of obviousness. For example, the references, whether taken alone or in combination, fail to disclose, teach, or suggest "a *binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks*," as now claimed in independent claims 20, 25, 32, and 36.

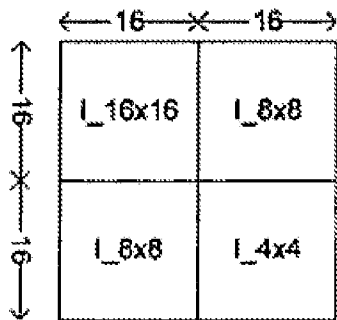
To signal partitioning of a large block, Chen (Chen'631 and Chen'357) discloses using a **macroblock type** that both identifies how the macroblock is partitioned, and the respective coding modes for each of the partitions. See [106] ("*In general, the macroblock type may be used to identify how the macroblock is partitioned, and the respective methods or modes for encoding each of the partitions of the macroblock, as discussed above. Methods for encoding the partitions may include not only intra- and inter-coding, but also particular modes of intra-coding (e.g., I_16x16, I_8x8, I_4x4) or inter-coding (e.g., P_ or B_16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4)*"). Tables 1 and 2 respectively illustrate macroblock type MB_N_type for B and P slices (inter partitioning) for a block of size NxN with M = N/2. Chen fails to disclose such a table for intra partitioning.

Regarding signaling inter partitions, as explained in [107], if the MB64_type of an inter 64x64 MB type indicates that the block is split into 4 partitions, then a MB32_type is signaled for each of the 32x32 partitions. However, if the MB64_type of a 64x64 type for instance indicates that the block is split into 2 partitions, then the disclosed syntax of Chen in Table 1 or 2 only allows to have 2 inter partitions. Therefore, Applicant submits that with the above syntax, with reference to Chen's FIG. 12, it is impossible to signal the partitioning of the 32X32 bottom left block having an Intra 32x16 and a complex hierarchical partitioning for the right part of the block.



The disclosure of Chen '631 on intra partitioning is at [110-111] or in FIG. 3 of Chen'357. Chen teaches to generate intra prediction for 32x32 blocks or 64x64 blocks by introducing mb_types syntax. In a variation, Chen teaches that four intra-coded blocks, each having a size of 16x16, can be grouped together to form a 32x32 intra-coded block as shown on the next page. First, Chen fails disclosing signaling macroblock types for Intra coded slices (I slices), and second, Chen's MB type syntax element is not a

binary split signaling syntax element. So far as the signaling of a recursive quadtree split of a 64X64 block is concerned, Chen discloses signaling an inter (P/B) MB64_type



indicating that the block is split into 4 partitions among respectively 3 or 22 values of inter (P/B) MB64_type, then signaling an inter (P/B) MB32_type indicating that the block is further split into 4 partitions among respectively 3 or 22 values of (P/B) MB32_type. Chen directly indicates how the large block is partitioned (0, 1, 2 or 4 partitions) along with prediction mode which undesirably causes overhead.

Moreover, Chen's intra prediction for 32x32 blocks or 64x64 blocks does not render the claims obvious and comes with significant overhead. Certainly, a MB64_type/MB_type32 syntax element fails to render obvious a binary split signaling syntax element specifying that the large block is further split into four equally sized sub- blocks.

A cpb (Coded Block Pattern) flag indicates, *after partitioning*, prediction, quantization and transform of the residue, whether a partition of a block has nonzero coefficients or not. A 1 means there is at least one nonzero coefficient, otherwise all coefficients are zero. Chen's "cbp" / "CBP64" / "CBP32" flags may be a binary syntax element, but they cannot be a binary *split signaling* syntax element and they do not indicate a quadtree partitioning, let alone "a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks," as claimed.

Moriya does not cure the deficiencies of Chen. Moriya discloses extended macroblock size, such as 32x32, however Moriya does not deviate from using Mb type syntax elements.

The Examiner is cordially invited to contact the undersigned if it will facilitate prosecution.

Date: Jan. 18, 2023

/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

Electronic Patent Application Fee Transmittal					
Application Number:		17351649			
Filing Date:		18-Jun-2021			
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			
Filer:		Brian Hubbard/Cassandra Katz			
Attorney Docket Number:		ICVW_2009P00073 US02			
Filed as Large Entity					
Filing Fees for Utility under 35 USC 111(a)					
Description		Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:					
Pages:					
Claims:					
Miscellaneous-Filing:					
Petition:					
Patent-Appeals-and-Interference:					
Post-Allowance-and-Post-Issuance:					
Extension-of-Time:					

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 2 months with \$0 paid	1252	1	640	640
Miscellaneous:				
Total in USD (\$)				640



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	02/08/2023	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	PAPER NUMBER
Suite 1700			2487	
Philadelphia, PA 19103			MAIL DATE	DELIVERY MODE
			02/08/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Response to Arguments

Applicant amended claims 20 and 25 to incorporate subject matter of claims 23 and 28 respectively.

Applicant amended claims 32 and 36 similarly to claims 20 and 25 respectively, but did not incorporate features of dependent claims and may be considered a scope change.

Applicant amended claims 22, 27, 34 – 35, and 38 – 39 to address 112 Rejections and some Objections.

Applicant canceled claims 23 and 28.

The pending claims are 20 – 22, 24 – 27, and 29 – 39 [Page 6 lines 1 – 4].

Applicant provided their version of the interview held on July 12th, 2022 [Page 6 lines 5 – 7].

Applicant comments on Examiner's Obvious Double Patenting Rejection and appears to hold the Rejection in abeyance [Page 6 lines 8 – 11].

Applicant does NOT comment on Examiner's Functional Analysis, however, in the sole interest to expedite prosecution the Functional Analysis section will be removed as there appears to be agreement in the claims NOT invoking 112(f).

Applicant's arguments filed January 18th, 2023 [Regarding Examiner Claim Objections] have been fully considered but they are not persuasive.

Applicant argues against Objections to claim 25 [Page 6 lines 12 – 26], however, the arguments are not persuasive in particular in view of the PTAB Decision in US Application 15/806,893 (from which this Application depends on) in view of the discussion of Rejection B [Pages 8 – 10 of the Decision] in which conditional limitations of method claims as not having weight if need not be practice [Page 9

Footnote 5 of the Decision]. Additionally in view of MPEP2111.04 II, contingent limitations (which are present even in the apparatus claim) may not necessarily have patentable weight. Thus, the limitations may not carry patentable weight as only half of the conditional is required (the other half is not required and an incomplete test is given leaving open questions of weight to afford the claim limitations).

While the Applicant's points may be understood, the Examiner respectfully disagrees; thus the Objection may be maintained and amended in view of the amended claims.

The Examiner updates the Objections and Rejections in view of the amended claims [No specific arguments made directed towards all the Objections / 112 Rejections made].

Applicant's arguments filed January 18th, 2023 [Page 7 line 1 – Page 8 line 18] have been fully considered but they are not persuasive.

First, the Applicant lists the references against the claims and the feature allegedly not taught by the references [Page 7 lines 1 – 5]. The Examiner notes the disputed features were previously presented in claim 28 thus the citations in claim 28 are incorporated in rebutting arguments made against the amended independent claims.

Second, the Applicant contends Chen '631 in Figure 12 and Paragraphs 106 – 107 as not teaching features of the amended claims [Page 7 lines 6 – 24] and continuing to rebut Chen '631 Paragraphs 110 – 111 and Figure 3 of Chen '357 [Page 7 line 25 – Page 8 line 18]. While the Applicant contends the reference do not teach their syntax elements allegedly claimed, the claims do not differentiate between the syntax elements taught and those claimed. The Applicant in arguing against Chen '631 Paragraphs 106 and Chen '357 Tables 1 – 2 contends a table for intra mode is not shown, but such feature is not claimed [Page 7 lines 6 – 14]. The Applicant in arguing against Chen '631 Figure 12 and Paragraph 107 argues a 32x32 block coded in intra mode is not taught, but in view of the references as a whole, Chen '357 Figure 3 illustrates an intra coded 32x32 block that is further partitioned into 16x16 blocks thus the argument is unpersuasive [Page 7 lines 15 – 24]. Further Applicant's arguments [Page 7 lines 15 – 24] appear to be for inter predicted block, but the Examiner cited features in relation to the

determination intra prediction modes are used (which have different partition types – mainly square partition types such as 4x4, 8x8, and 16x6 as rendered obvious in both Chen references).

Continuing, the Applicant argues against Chen '357 Figure 3 and Chen '631 Paragraphs 110 – 111 [Page 7 line 25 – Page 8 line 12], however the description of Chen '357 Page 4 (last Paragraph) renders obvious signaling the intra mode per macroblock (the claim does not require an intra frame / slice is detected) where the syntax / partition types listed in Chen '357 Figure 3 (illustrative of a 32 x32 intra block partitioned down to 16 x16 blocks or smaller basic coding units) as understood by Chen '631 Paragraphs 103 – 104 and 109 – 111 (describing I_16x16 and smaller basic coding size intra modes used within a larger 32 x 32 block as split signaling syntax elements (even binary as the mode is used or not)). Further, the Applicant contends the coded block pattern split syntax elements [Page 8 lines 13 – 18], however Chen '631 in the cited figures (e.g. Figures 4 – 5, and 14 – 16) and at least Chen '631 Paragraphs 130 – 132 and 183 – 185 renders obvious the binary syntax elements signaling quadtree partitioning and truncating the tree signaled based on when partitions occur (and simplifications as in Paragraphs 118 and 127). Chen '631 Paragraph 127 also addresses the same problem as the Application to reduce overhead signaling (albeit with possibly different syntax elements than those disclosed, but performing the functions of those claimed).

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the binary split syntax element indicates a split type (e.g. quadtree partitioning) and the prediction mode of the block being partitioned [Page 8 lines 13 – 18]) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Third, the Applicant contends Moriya does not render obvious the features claimed [Page 8 lines 19 – 20].

While the Applicant's points may be understood, the Examiner respectfully disagrees; thus the Rejection is maintained and amended in view of the amended claims.

Priority

Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, 365(c), or 386(c) is acknowledged (US Provisional Application 61/222,177 filed July 1st, 2009).

Information Disclosure Statement

The information disclosure statements (IDS) submitted on June 18th, 2021 and October 26th, 2021 were filed before the mailing date of the First Action on the Merits (mailed August 18th, 2022). The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the Examiner.

Due to the excessively lengthy Information Disclosure Statement submitted by applicant, the examiner has given only a cursory review of the listed references. In accordance with MPEP 609.04(a), applicant is encouraged to provide a concise explanation of why the information is being submitted and how it is understood to be relevant. Concise explanations (especially those which point out the relevant pages and lines) are helpful to the Office, particularly where documents are lengthy and complex and applicant is aware of a section that is highly relevant to patentability or where a large number of documents are submitted and applicant is aware that one or more are highly relevant to patentability. Applicant is required to comply with this statement for any non-English language documents. See 37 CFR § 1.56 Duty to Disclose Information Material to Patentability.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the

reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP § 2146 *et seq.* for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

Claims 20 – 39 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1 – 4 of U.S. Patent No. #11,082,697.

Although the claims at issue are not identical, they are not patentably distinct from each other because of at least the reasons given in the table below:

Note: In the interest of brevity the Examiner does not cite identical or similar claim limitations in the analysis.

Pending Application	US Patent #11,082,697 Published Claims
Claim 20) See claim 25 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.

Claim 21) See claim 26 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 22) See claim 27 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
Claim 24) See claim 29 which is the apparatus performing the steps of the claimed method.	Claim 1) The claim is and apparatus performing on a processor method / program steps.
<p>Claim 25) An apparatus, comprising:</p> <p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size,</p> <p>[The patented claims render the pending claim as obvious as providing specific examples and sizes as those contemplated in the Specification of the pending Application thus the Patented claims render obvious the pending limitation to one of ordinary skill in the art.]</p> <p>wherein the intra prediction is signaled for the large block and the one or more processors are configured to:</p> <p>decode a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks;</p> <p>partition the large block into the four sub-blocks in response to the decoded binary split signaling syntax element; and</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “where said binary split signaling syntax element [...] is not further split” and the next “otherwise” limitations) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p> <p>decode a single spatial intra partition type for at least one basic coding unit sized sub-block, the single spatial intra partition type being</p>	<p>Claim 1) A video decoder, comprising</p> <p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>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,</p> <p>wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:</p> <p>decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 subblock in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and</p> <p>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,</p>

<p>determinable from among a plurality of spatial intra partition types.</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “otherwise” limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p>	<p>wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 26) The apparatus of claim 25, wherein the one or more processors are configured to</p> <p>decode an intra prediction mode for the basic coding unit sized sub-block.</p> <p>[The Patented claims render obvious associated an intra prediction mode with a sub block and partitioning to sub blocks with their own intra prediction mode (and even the use of signaling if the MPM (most probable mode)) for subblocks further rendering obvious the pending claim to one of ordinary skill in the art.]</p>	<p>Claim 1) [Relevant portions only]</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p> <p>decode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types,</p> <p>wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 27) The apparatus of claim 25, wherein the one or more processors are configured to</p> <p>decode a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks is further split into four smaller equally sized sub-blocks.</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “otherwise”</p>	<p>Claim 1) [Relevant portions only]</p> <p>decode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p>

limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]	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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and decode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and
Claim 29) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit. [One of ordinary skill in the art readily recognizes the single intra partition type for the sub blocks to be partitioned into the basic coding unit as indications of the size specified in using the spatial intra prediction types and the features in the Patented claim renders obvious the same features and semantics / meaning in the pending Application.]	Claim 1) [Relevant portions only] decode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.
Claim 30) The apparatus of claim 25, 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.	Claim 2) The video decoder 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.
Claim 31) The apparatus of claim 25, 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. [Patented claim 1 already recited the use of processors and the picture data received being processed thus the claims are obvious variants to one of ordinary skill in the art.]	Claim 3) The video decoder 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.
Claim 32) See claim 36 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 33) See claim 37 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 34) See claim 38 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.
Claim 35) See claim 39 which is the apparatus performing the steps of the claimed method.	Claim 4) The claim is and apparatus performing on a processor method / program steps.

<p>Claim 36) An apparatus comprising:</p> <p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size,</p> <p>[The patented claims render the pending claim as obvious as providing specific examples and sizes as those contemplated in the Specification of the pending Application thus the Patented claims render obvious the pending limitation to one of ordinary skill in the art.]</p> <p>wherein the intra prediction is signaled for the large block and the one or more processors are configured to:</p> <p>encode a binary split signaling syntax element specifying that the large block is further split into four equally sized sub-blocks:</p> <p>partition the large block into the four sub-blocks in response to the binary split signaling syntax element; and</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “where said binary split signaling syntax element [...] is not further split” and the next “otherwise” limitations) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p> <p>encode 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.</p>	<p>Claim 4) A video encoder, comprising</p> <p>at least a memory and one or more processors, wherein the one or more processors are configured to:</p> <p>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,</p> <p>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,</p> <p>wherein the intra prediction is signaled for the at least a large block and the one or more processors are configured to:</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 subblock in the case where said binary split signaling syntax element specifies said 32x32 sub-block is not further split; and</p> <p>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, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable</p>
---	--

	intra prediction mode for the basic coding unit size.
<p>Claim 37) The apparatus of claim 36, wherein, the one or more processors are configured to</p> <p>encode an intra prediction mode for the basic coding unit sized sub-block.</p> <p>[The Patented claims render obvious associated an intra prediction mode with a sub block and partitioning to sub blocks with their own intra prediction mode (and even the use of signaling if the MPM (most probable mode)) for subblocks further rendering obvious the pending claim to one of ordinary skill in the art.]</p>	<p>Claim 4) [Relevant Portions only]</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and encode an intra prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and</p> <p>encode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.</p>
<p>Claim 38) The apparatus of claim 36, wherein, the one or more processors are configured to</p> <p>encode a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks is further split into four smaller equally sized sub-blocks.</p> <p>[One of ordinary skill in the art would readily recognize the Patented claim (see the “otherwise” limitation) as having the cases of the binary split signaling syntax elements and the details in splitting a large block into four equal sized subblocks (quadtree partitioning) as the same technique contemplated in the pending Application thus the pending limitation is rendered obvious.]</p>	<p>Claim 4) [Relevant Portions only]</p> <p>encode a binary split signaling syntax element specifying whether the large block is further split into four equally sized sub-blocks;</p> <p>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;</p> <p>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 32×32, a binary split signaling syntax element specifying whether said 32×32 sub-block is further split into four equally basic coding unit sized blocks and encode an intra</p>

	prediction mode for said 32×32 subblock in the case where said binary split signaling syntax element specifies said 32×32 sub-block is not further split; and
Claim 39) The method of claim 36, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block. [One of ordinary skill in the art readily recognizes the single intra partition type for the sub blocks to be partitioned into the basic coding unit as indications of the size specified in using the spatial intra prediction types and the features in the Patented claim renders obvious the same features and semantics / meaning in the pending Application.]	Claim 4) [Relevant Portions only] encode for each sub-block, in the case where said sub-block is 16×16, a single spatial intra partition type, the single spatial intra partition type being determinable from among a plurality of spatial intra partition types, wherein the single spatial intra partition type selected from the plurality of spatial intra partition types is accompanied by at least (1) a single intra prediction type for the basic coding unit size which does not use a most probable intra prediction mode and (2) a single intra prediction type indicating the most probable intra prediction mode for the basic coding unit size.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to understand the Patented claims as providing teachings of the specific techniques to perform quadtree partitioning of a large block (e.g. 32 x 32) block rendering obvious the scope of the pending claims. Further, one of ordinary skill in the art would recognize the decoder claims as the inverse of the encoding claims and thus the claims are obvious variants.

Claim Objections

Claims 20 – 22, 24 – 27, and 29 – 39 are objected to because of the following informalities:

Regarding claim 25, the claim does not constructively indicate the differences in processing the large block after the decision / determination of partitioning as if the block isn't partitioned it is not therefore into sub-blocks and thus has no description of how to process the larger blocks with the spatial intra partition types and intra prediction mode. Further, the conditionals stated appear open-ended and thus appear to lack patentable weight (see at least MPEP2111.04).

Regarding claim 20, see claim 25 which is the apparatus performing the steps of the claimed method and thus is similarly Objected.

Regarding claims 32 and 36, see claims 20 and 25 respectively as the decoder performing the inverse steps of the claimed encoder and thus are similarly Objected.

Regarding claims 21 – 22, 24, 26 – 27, 29 – 31, 33 – 35, and 37 – 39, the dependent claims do not cure the deficiencies of their respective independent claims and thus are similarly Objected.

Regarding claim 22, the claim appears to be missing an Essential Step that only recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 27, the claim appears to be missing an Essential Step that only recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 34, the claim appears to be missing an Essential Step that recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Regarding claim 38, the claim appears to be missing an Essential Step that only recites half of the processing determination (to perform a second split but lacks an Essential Step / element checking if the basic coding unit sized was reached or the minimum block size allowed such as 4 x 4).

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 20, 25, 32, and 36 rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor (or for applications subject to pre-AIA 35 U.S.C. 112, the applicant), regards as the invention.

Claim 20 recites the limitation "the four sub-blocks" in line 8. There is insufficient antecedent basis for this limitation in the claim. The Rejection is made in view of the recitation of "four equally sized sub-blocks" and "four smaller equally sized sub-blocks" in amended dependent claim 22 (the antecedent basis includes the "equally-sized" clause / feature).

Claim 25 recites the limitation "the four sub-blocks" in line 11. There is insufficient antecedent basis for this limitation in the claim. The Rejection is made in view of the recitation of "four equally sized sub-blocks" and "four smaller equally sized sub-blocks" in amended dependent claim 27 (the antecedent basis includes the "equally-sized" clause / feature).

Claim 32 recites the limitation "the four sub-blocks" in line 8. There is insufficient antecedent basis for this limitation in the claim. The Rejection is made in view of the recitation of "four equally sized sub-blocks" and "four smaller equally sized sub-blocks" in amended dependent claim 34 (the antecedent basis includes the "equally-sized" clause / feature).

Claim 36 recites the limitation "the four sub-blocks" in line 11. There is insufficient antecedent basis for this limitation in the claim. The Rejection is made in view of the recitation of "four equally sized sub-blocks" and "four smaller equally sized sub-blocks" in amended dependent claim 38 (the antecedent basis includes the "equally-sized" clause / feature).

Claim Rejections - 35 USC § 103

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

This application currently names joint inventors. In considering patentability of the claims under pre-AIA 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of pre-AIA 35 U.S.C. 103(c) and potential pre-AIA 35 U.S.C. 102(e), (f) or (g) prior art under pre-AIA 35 U.S.C. 103(a).

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 20, 22 – 25, 27 – 29, 32, 34 – 36, and 38 – 39 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Chen, et al. (US PG PUB 2010/0086029 A1 referred to as “Chen” throughout in which citations will come from US Provisional Applications 61/166,631 (referred to as “Chen ‘631” throughout) and 61/144,357 (referred to as “Chen ‘631” throughout)) [The US PG PUB is cited in Applicant’s June 18th, 2021 IDS as US PG PUB Item #24 – The Examiner will cite from the US Provisional References where “Chen” refers to the collective of “Chen ‘631” and “Chen ‘357”], and further in view of Moriya, et al. (US PG PUB 2009/0046781 A1 referred to as “Moriya” throughout) [Cited in Applicant’s June 18th, 2021 IDS as US PG PUB Items #20].

Regarding claim 20, see claim 25 which is the apparatus performing the steps of the claimed method.

Regarding claim 22, see claim 27 which is the apparatus performing the steps of the claimed method.

Regarding claim 24, see claim 29 which is the apparatus performing the steps of the claimed method.

Regarding claim 32, see claim 36 which is the apparatus performing the steps of the claimed method.

Regarding claim 34, see claim 38 which is the apparatus performing the steps of the claimed method.

Regarding claim 35, see claim 39 which is the apparatus performing the steps of the claimed method.

Regarding claim 25, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

at least a memory and one or more processors [Chen '631 Paragraphs 65 and 186 – 188 (microprocessors / DSPs and memory elements used to implement)], 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 [Chen '631 Figures 3 – 5 (decoder includes intra mode processing); Chen '357 Figure 3 as well as Sections 3.1 and 3.2)],

wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size [Chen '631 Figures 4, 5, 10, 15, and 16 (including subfigures) and Paragraphs 20 (16 x 16 coding unit contemplated at least as a basic coding unit size), 27 (32 x 32 or 64 x 64 large macroblocks contemplated as large block sizes), 73 – 75, 96 – 104 (32 x 32 and 64 x 64 large block sizes contemplated as well as a base 16x16 block size contemplated for intra prediction); Chen '357 Figures 2 – 4 and Section 3.2 (32 x 32 large block illustrated with 16 x 16 blocks illustrated as a base for merging or splitting in Figure 3 in which each subblock is intra coded)],

wherein the intra prediction is signaled for the large block [Chen '357 Figure 3 as well as Sections 3.1 and 3.2.1 (two methods of intra mode signaling for large macroblocks); Chen '631 Figures 4, 5, 10, 12, and 15 as well as Paragraphs 104 – 106 and 111 (intra coding considering larger macroblock sizes to partition to smaller block sizes)] 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 [Chen '631 Figures 3 – 8 (including subfigures – checks to create four equally sized subblocks partitioned from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes considered using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 159)) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks as is also done in Figure 3 where intra prediction is used and spatial intra partition types are assigned)];

partition the large block into the four sub-blocks in response to the decoded binary split signaling syntax element [Chen '631 Figures 3 – 8 (including subfigures – check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large

macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)]; and

decode 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 [Chen '631 Figure 3 as well as Paragraphs 50 and 102 – 104 (“e.g., one of I_16x16, I_8x8, I_4x4” are the plurality of spatial intra partition types); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 where Moriya in Figures 2 – 3 and 6 illustrates quadtree partitioning of large blocks (e.g. 32x32 blocks) and in Figure 7 (see at least reference characters 29 and 30) as well as Paragraphs 45 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks decoding the encoded information), 63 – 64 (intra block sizes partitioned / determined in the decoder based on encoded information), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) in the decoder using encoded information)].

The motivation to combine Chen '357 with Chen '631 is to combine features in the same field of invention of video coding / decoding [Chen '357 Section 1] in order to code / decode large macroblocks to improve coding for higher resolution images / HD capabilities [Chen '357 Section 2 where the Examiner observes KSR Rationales (C), (D), or (F) are also applicable].

The motivation to combine Moriya with Chen '357 and Chen '631 is to combine features in the same / similar field of invention of coding and decoding image data [Moriya Paragraphs 1 and 4] in order to improve compression efficient by reducing the amount of coding information (e.g. intra mode used) to transmit as side / header information with the block [Moriya Paragraphs 5 – 9 where the Examiner observes at least KSR Rationales (D) or (F) are also applicable].

This is the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya which will be used throughout the Rejection.

Regarding claim 27, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the one or more processors are configured to [See claim 25 for citations of the claimed "processors"] decode further a binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks is further split into four smaller equally sized sub-blocks [Chen '631 Figures 3 – 8 (including subfigures - check four equally sized blocks partitioning from large block), 12 – 13, and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks) and then further partitions into 16 x 16 blocks or smaller (e.g. 8 x 8 blocks))) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the "cbp" / "CBP64" / "CBP32" flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see "mb64_type" and "mb32_type" syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 29, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the spatial intra partition type specifies a size of a partition of the basic coding unit [Chen '631 Paragraphs 50 and 102 – 104 (“e.g., one of $I_{16 \times 16}$, $I_{8 \times 8}$, $I_{4 \times 4}$ ” are the plurality of spatial intra partition types) and 115 – 121 (signaling intra spatial partition information within a basic coding unit (e.g. partitions within a 16×16 block)); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 (see the 32×32 block partitioned into 16×16 blocks in which spatial intra modes indicate partitions / sizes within the 16×16 partitions for intra predication) where Moriya in Figures 2 – 3 and 6 illustrates quadtree partitioning of large blocks (e.g. 32×32 blocks) and in Figure 7 (see at least reference characters 29 and 30) as well as Paragraphs 45 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32×32 or 8×8 and additionally 16×16 and 4×4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks decoding the encoded information), 63 – 64 (intra block sizes partitioned / determined in the decoder based on encoded information), and 74 – 81 (partitioning large intra blocks to

be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) in the decoder using encoded information)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 36, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

at least a memory and one or more processors [Chen '631 Paragraphs 65 and 186 – 188 (microprocessors / DSPs and memory elements used to implement)], 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 [Chen '631 Figure 2, 4, and 5 as well as Paragraphs 100 -1 06 (encoding / decoding large blocks and formation of large blocks); Chen '357 Figure 3 as well as Sections 3.1 and 3.2)],

wherein the large block has a large block size equal to or larger than 32x32, and the large block size is greater than a basic coding unit size [Chen '631 Figures 4, 5, 10, 12, 13, 15, and 16 (including subfigures) and Paragraphs 20 (16 x 16 coding unit contemplated at least), 27 (32 x 32 and 64 x 64 macroblocks contemplated), 73 – 75, 96 – 104 (32 x 32 and 64 x 64 initial block sizes contemplated as well as a base 16x16 block size / unit contemplated for intra prediction), and 182 – 186 (partitioning larger blocks and signaling thereof); Chen '357 Figure 2 – 4 and Section 3.2 (64 x 64 blocks illustrated with 16 x 16 blocks illustrated as a base for merging or splitting in Figure 3)],

wherein the intra prediction is signaled for the large block [Chen '357 Figure 3 as well as Sections 3.1 and 3.2.1 (two methods of intra mode signaling for large macroblocks); Chen '631 Figures 4, 5, 10, 12, and 15 as well as Paragraphs 104 – 106 and 111 (intra coding considering larger macroblock sizes to partition to smaller block sizes)] 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 [Chen '631 Figures 4 - 8 (including subfigures - check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 - 121 (hierarchically partitioning a large macroblock by using tree coding and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 - 159)) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 - 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)];

partition the large block into the four sub-blocks in response to the binary split signaling syntax element [Chen '631 Figures 4 – 8 (including subfigures - check four equally sized blocks partitioning from large block) and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks (e.g. 64 x 64 into 4 32 x 32 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)]; and

encode 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

[Chen '631 Figure 2 as well as Paragraphs 50 and 102 – 104 (“e.g., one of I_16x16, I_8x8, I_4x4” are the plurality of spatial intra partition types); Chen '357 Sections 3.2 – 3.3 and Figures 2 and 3 where Moriya in Figures 1 – 3 (see at least reference characters 1, 2, 5, 15, and 16) and 6 illustrates quadtree partitioning of large blocks (e.g. 32x32 blocks) as well as Paragraphs 42 – 47 (selection of intra prediction block size in the large block to be decoded in which a plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode in the encoded information), 63 – 64 (intra block sizes partitioned / determined in the encoder), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) as generated encoded information)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya where claim 25 is the decoder processing the encoded information generation and in view of at least Chen '631 Paragraphs 73 – 75 (decoding is the inverse / reverse process of encoding as readily recognized by one of ordinary skill in the art) and thus similar motivation exists.

Regarding claim 38, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

wherein, the one or more processors are configured to [See claim 36 for citations of the claimed “processors”] encode a further binary split signaling syntax element specifying that at least one of the four equally sized sub-blocks is further split into four smaller equally sized sub-blocks [Chen '631 Figures 2, 4 – 8 (including subfigures – check four equally sized blocks partitioning from large block), 12 – 13, and 15 – 16 (partitioning of a large macroblock and signaling of the partitioning in to four equally sized blocks

(e.g. 64 x 64 into 4 32 x 32 blocks) and then further partitions into 16 x 16 blocks or smaller (e.g. 8 x 8 blocks)) as well as Chen '631 Paragraphs 73 – 75 (large sizes split using Quad-tree partitioning), 97 – 106 (partitioning of large block with intra blocks partitioned taught), 108, 110 – 112, 115 – 121 (hierarchically partitioning a large macroblock by using tree coding (e.g. the use of the “cbp” / “CBP64” / “CBP32” flags are binary split signaling syntax element) and illustrating a binary bit string to transmit and further a signal to indicate partitioning (see Paragraphs 121, 127 – 131 and 158 – 163 (e.g. see “mb64_type” and “mb32_type” syntax elements))) and 184 – 186 (binary signals of splitting using a tree structure signifying the hierarchy of the blocks); Chen '357 Sections 3.2 – 3.3 and Figures 2 – 4 (splitting hierarchically large macroblock into 4 equally sized smaller blocks)].

See claim 36 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Regarding claim 39, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted. The combination teaches

wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block [Chen '631 Figure 2 as well as Paragraphs 50 and 102 – 104 (“e.g., one of 16x16, 18x8, 16x4” are the plurality of spatial intra partition types) and 115 – 121 (signaling intra spatial partition information within a basic coding unit (e.g. partitions within a 16x16 block)); Chen '357 Sections 3.2 - 3.3 and Figures 2 and 3 (see the 32x32 block partitioned into 16x16 blocks in which spatial intra modes indicate partitions / sizes within the 16x16 partitions for intra predication) where Moriya in Figures 1 – 3 and 6 (see at least reference characters 1, 2, 5, 15, and 16) illustrates quadtree partitioning of large blocks (e.g. 32x32 blocks) as well as Paragraphs 42 – 47 (selection of intra prediction block size in the large block in which a

plurality of sizes is given (e.g. 32 x 32 or 8 x 8 and additionally 16 x 16 and 4 x 4 intra prediction partitions / modes to use and available for selection to further render obvious the teachings of Chen collectively)), 48 – 51 (large block size to use can be changed), 58 – 61 (partitioning blocks based on prediction mode included large blocks), 63 – 64 (intra block sizes partitioned / determined based on encoded information), and 74 – 81 (partitioning large intra blocks to be processed (e.g. splitting 32 x 32 blocks into 16 x 16 blocks) from generated encoded information)].

See claim 36 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

Claim 21, 26, 30 – 31, 33, and 37 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Chen (collectively), Moriya, and further in view of Moriya, et al. (US PG PUB 2008/0123947 A1 referred to as "Moriya 47" throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #16 and 17].

Regarding claim 21, see claim 26 which is the apparatus performing the steps of the claimed method.

Regarding claim 33, see claim 37 which is the apparatus performing the steps of the claimed method.

Regarding claim 26, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

wherein the one or more processors are configured to [See claim 25 for citations of the claimed "processors"] decode an intra prediction mode for the basic coding unit sized sub-block [Chen '631 Figures 4 and 12 (subfigures included) as well as Paragraphs 102 – 106 (prediction modes a function of the spatial prediction mode used (e.g. a described in Paragraph 103) where the selection of the basic coding unit is rendered obvious (e.g. 4x4 partition type and its associated intra modes would be a selection of the finite number of possibilities to one of ordinary skill in the art)) and 162 – 165; Chen '357 Figures 2 – 3 and Section 3.2.1; Moriya Figure 7 (see at least reference character 29 and 30 which is to be modified / elaborated by Moriya 47 teaching of the intra prediction modes) as well as Paragraphs 32 (intra prediction performing on a block type), 44 – 46, 55 – 56 (encoding the intra prediction mode); Moriya 47 Figures 3, 4 and 13 (4x4, 8x8 and 16x16 intra prediction modes (such as DC and directional prediction) illustrated) as well as Paragraphs 218 – 237 (signaling mode information for partitions to be predicted with intra prediction), 264 – 281 (discussion of the 8x8 intra prediction modes in contrast to the 16x16 intra prediction modes which is encoded to later be decoded), 300 – 304 (decoded encoded intra prediction mode information)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively) and Moriya.

The motivation to combine Moriya 47 with Moriya, Chen '357, and Chen '631 is to combine features in the same / related field of invention of "a digital image encoding method, and a [...] decoding method" [Moriya 47 Paragraph 2] to improve efficiency in performing intra prediction [Moriya 47 in at least Paragraphs 279, 295, and 396] and to improve flexibility in selecting intra prediction mode information to

use [Moriya 47 Paragraph 12 where the Examiner observes at least KSR Rationales (C) or (D) are also applicable].

This is the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47 which will be used throughout the Rejection.

Regarding claim 30, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

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 [See claim 25 for citations regarding the “processor” claimed; See claim 25 for citations of the “spatial partition intra partition type” and claim 26 for citations of the “intra prediction mode” and regarding the treatment of table representations see additionally Moriya 47 Figures 3 – 8, 13 (Figures 3 – 4 and 13 show the intra prediction modes), and 26 – 28 (tables used for signaling the intra prediction mode of a current block) as

well as Paragraph 251 – 255 (decoding process in Paragraph 253 is modified to incorporate the table used for encoding in Paragraph 254) and also in Paragraph 254 ("select a code table" implies the tables are known a priori before usage of the table), 375 – 385 (tables used to signal intra prediction modes for processing in a decoder), and 392 – 400 (tables including decoder usage of intra modes signaled with tables)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47.

Regarding claim 31, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The Examiner observes Chen '631 in at least Paragraphs 73 – 75 renders obvious the decoder as performing the inverse processing of an encoder thus encoder citations may be used to render obvious features of a decoder as would be readily appreciated by one of ordinary skill in the art. The combination teaches

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 [See claim 25 for citations regarding

the "processor" claimed; See claim 30 for citations of the claimed "table" feature of the representation of the plurality of spatial intra partition types (claims 25 and 30) or intra prediction modes (claim 26) and additionally Chen '631 Paragraphs 92 and 177 (levels of syntax elements taught); and Moriya 47 Figures 3 – 8 (see at least reference character 31) 72, and 82 – 86 (signaling including high level syntax elements / organization to insert partition and prediction information) as well as Paragraphs 253 – 255 (signaling intra prediction tables in "a slice, picture, or a sequence" which are "upper data layer" and thus would be recognized by one of ordinary skill in the art as a high level syntax element when afforded the broadest reasonable interpretation in view of Specification Paragraph 40 (using the numbering in the US PG PUB) defining the term), 523 – 533 (signaling and decoding upper data layer / high level syntax elements for intra prediction) and 621 – 622 (high level syntax elements describing blocks encoded to be decoded)].

See claim 25 for the motivation to combine Chen (Chen '631 and Chen '357 collectively), Moriya, and Moriya 47.

Regarding claim 37, Chen '631 teaches processing large macroblocks with split and merge considerations. Chen '357 teaches the use of intra prediction for large blocks and further suggests design decision considerations for large macroblock sizes as well as suggesting spatial intra prediction types used. Moriya expands and further clarifies the teachings of Chen regarding the use of spatial intra partition types in view of processing large macroblocks. Moriya 47 teaches intra prediction modes to use (e.g. directional intra prediction or the use of a DC mode or using the MPM (most probable mode) to predict a block as a function of the intra spatial partition type of the block and further teaches the use of tables and organizational techniques to signal partition types for prediction modes and the intra modes to use as well.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Chen (collectively) with the spatial intra partition types as taught by Moriya in partitioning large blocks / macroblocks that are being intra predicted and the intra prediction modes as a function of intra partitions used and the table organization methods to signal intra prediction information as taught by Moriya 47. The combination teaches

wherein, the one or more processors are configured to [See claim 25 for citations of the claimed “processors”] encode an intra prediction mode for the basic coding unit sized sub-block [Chen ‘631 Figures 4 and 12 (subfigures included) as well as Paragraphs 102 – 106 (prediction modes a function of the spatial prediction mode used (e.g. a described in Paragraph 103) where the selection of the basic coding unit is rendered obvious (e.g. I_4x4 partition type and its associated intra modes would be a selection of the finite number of possibilities to one of ordinary skill in the art)) and 162 – 165; Chen ‘357 Figures 2 – 3 and Section 3.2.1; Moriya Figure 1 (see at least reference character 15 and 16 which is to be modified / elaborated by Moriya 47 teaching of the intra prediction modes) as well as Paragraphs 32 (intra prediction performing on a block type), 44 – 46, 55 – 56 (encoding the intra prediction mode); Moriya 47 Figures 3, 4 and 13 (4x4, 8x8 and 16x16 intra prediction modes (such as DC and directional prediction) illustrated) as well as Paragraphs 218 – 237 (signaling mode information for partitions to be predicted with intra prediction), 264 – 281 (discussion of the 8x8 intra prediction modes in contrast to the 16x16 intra prediction modes which is encoded to later be decoded), 300 – 304 (encoded intra prediction mode information)].

See claim 26 for the motivation to combine Chen (Chen ‘631 and Chen ‘357 collectively), Moriya, and Moriya 47 where claim 26 is the decoder processing the encoded information generation and in view of at least Chen ‘631 Paragraphs 73 – 75 (decoding is the inverse / reverse process of encoding as readily recognized by one of ordinary skill in the art) and thus similar motivation exists.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Po, et al. (US PG PUB 2009/0219991 A1 referred to as “Po” throughout) teaches in Figures 3 – 4 and Paragraphs 8 at least the decisions to merge / split blocks with hierarchy of intra prediction modes / splits.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tyler W Sullivan whose telephone number is (571)270-5684. The examiner can normally be reached IFP.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Czekaj can be reached on (571)-272-7327. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: <https://patentcenter.uspto.gov>. Visit <https://www.uspto.gov/patents/apply/patent-center> for more information about Patent Center and <https://www.uspto.gov/patents/docx> for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

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

<i>Notice of References Cited</i>	Application/Control No. 17/351,649		Applicant(s)/Patent Under Reexamination Zheng et al.	
	Examiner Tyler W Sullivan		Art Unit 2487	Page 1 of 1

U.S. PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20090219991-A1	09-2009	Po; Lai-Man	H04N19/147	375/E7.126
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS						
*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)				
	U					
	V					
	W					
	X					

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Electronic Acknowledgement Receipt	
EFS ID:	48194607
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:	22-JUN-2023
Filing Date:	18-JUN-2021
Time Stamp:	17:30:30
Application Type:	Utility under 35 USC 111(a)

Payment information:	
Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 2200
RAM confirmation Number	E20236LH31426008
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 1.21 (Miscellaneous fees and charges) 37 CFR 1.20 (Post Issuance fees)	

37 CFR 1.19 (Document supply fees)					
37 CFR 1.17 (Patent application and reexamination processing fees)					
37 CFR 1.16 (National application filing, search, and examination fees)					
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	ICVW_2009P00073US02_RCE.pdf	697716	no	3
			7bdfc934ac6d7cb0bba3604620d3d70a8ef78929		
Warnings:					
Information:					
2		ICVW_2009P00073US02_Response_to_Final_Office_Action_dated_2_8_2023.pdf	66443	yes	7
			fe7680c822a529cc48b6221cfbd00fc4e2630f70		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Amendment after Notice of Allowance (Rule 312)		1	1	
	Claims		2	6	
	Applicant Arguments/Remarks Made in an Amendment		7	7	
Warnings:					
Information:					
3	Extension of Time	ICVW_2009P00073US02_EOT.pdf	165954	no	2
			0ea09e5b2f332d80e641ca71874520f39f9f3f4d		
Warnings:					
Information:					
4	Internet Communications Authorization	ICVW_2009P00073US02_PTO-SB-439.pdf	186467	no	2
			0c7f64582715e9e5729c7820d405393a41375fe7		
Warnings:					
Information:					
5	Fee Worksheet (SB06)	fee-info.pdf	43502	no	2
			741e097399fad3bb926c91533274a2fff4a79d23		

Warnings:	
Information:	
Total Files Size (in bytes):	1160082
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>	

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL (Submitted Only via EFS-Web)							
Application Number	17351649	Filing Date	2021-06-18	Docket Number (if applicable)	ICVW_2009P00073 US02	Art Unit	2487
First Named Inventor	Yunfei Zheng			Examiner Name	TYLER SULLIVAN		
This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application. Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV							
SUBMISSION REQUIRED UNDER 37 CFR 1.114							
Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).							
<div>☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.<div>☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on <div></div>☐ Other <div></div></div></div> <div>☒ Enclosed<div>☒ Amendment/Reply☐ Information Disclosure Statement (IDS)☐ Affidavit(s)/ Declaration(s)☒ Other <div>Authorization for Internet Communications in a Patent Application (PTO/SB/439) Petition for Extension of Time</div></div></div>							
MISCELLANEOUS							
<div>☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months <div></div><div>(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)</div>☐ Other <div></div></div>							
FEES							
☒ The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed. The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No <div>505519</div>							
SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED							
<div>☒ Patent Practitioner Signature Applicant Signature</div>							

Signature of Registered U.S. Patent Practitioner			
Signature	/Brian J. Hubbard/	Date (YYYY-MM-DD)	2023-06-22
Name	Brian J. Hubbard	Registration Number	45873

This collection of information is required by 37 CFR 1.114. 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 12 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

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 Cooperation Treaty.
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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: TYLER SULLIVAN

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

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY PURSUANT TO 37 CFR § 1.114

In response to the Final Office Action dated February 8, 2023, reconsideration is respectfully requested in view of the amendments and/or remarks as indicated below:

- ☐ **Amendments to the Specification** begin on page of this paper.
- ☒ **Amendments to the Claims** begin on page 2 of this paper.
- ☐ **Amendments to the Drawings** begin on page of this paper and include an attached replacement sheet.
- ☒ **Remarks** begin on page 7 of this paper.
- ☒ The Commissioner is hereby authorized to charge any fee deficiency, charge any additional fees, or credit any overpayment of fees, associated with this application in connection with this filing, or any future filing, submitted to the U.S. Patent and Trademark Office during the pendency of this application, to Deposit Account No. **50-5519**.

Claims:

1 - 19. (Cancelled)

20. (Currently Amended) 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 32x32, 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;

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.

21. (Previously Presented) The method of claim 20, further comprising decoding an intra prediction mode for the basic coding unit sized sub-block.

22. (Currently Amended) The method of claim 20, 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.

23. (Cancelled)

24. (Previously Presented) The method of claim 20, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

25. (Currently Amended) 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 32x32, 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;

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.

26. (Previously Presented) The apparatus of claim 25, wherein the one or more processors are configured to decode an intra prediction mode for the basic coding unit sized sub-block.

27. (Currently Amended) The apparatus of claim 25, 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.

28. (Cancelled)

29. (Previously Presented) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit.

30. (Previously Presented) The apparatus of claim 25, 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.

31. (Previously Presented) The apparatus of claim 25, 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.

32. (Currently Amended) 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 32x32, 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;

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.

33. (Previously Presented) The method of claim 32, further comprising encoding an intra prediction mode for the basic coding unit sized sub-block.

34. (Currently Amended) The method of claim 32, 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.

35. (Previously Presented) The method of claim 32, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

36. (Currently Amended) 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 32x32, 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;

 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.

37. (Previously Presented) The apparatus of claim 36, wherein, the one or more processors are configured to encode an intra prediction mode for the basic coding unit sized sub-block.

38. (Currently Amended) The apparatus of claim 36, 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.

39. (Previously Presented) The method of claim 36, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

40. (New) The method of claim 20, further comprising decoding an intra prediction mode for the large block that is not further split into four equally sized sub-blocks.

41. (New) The apparatus of claim 25, wherein the one or more processors are configured to decode an intra prediction mode for the large block that is not further split into four equally sized sub-blocks.

42. (New) The method of claim 32, further comprising encoding an intra prediction mode for the large block that is not further split into four equally sized sub-blocks.

43. (New) The apparatus of claim 36, wherein, the one or more processors are configured to encode an intra prediction mode for the large block that is not further split into four equally sized sub-blocks.

REMARKS

Claims 20-22, 24-27, 29-39, and 40-43 are pending in the application. Claims 20, 22, 25, 27, 32, 34, 36, and 38 have been amended. No claims have been cancelled. New claims 40-43 have been added. No new matter has been added.

Applicant thanks the Examiner for the careful listing of the issues. All objections and rejections are moot by virtue of amendment to the independent claims. Applicant is under no obligation to argue moot rejections, and due to prosecution history estoppel it is typically not a best practice. The undersigned would like to point out that Applicant is not technically asking that the double patenting rejection be held in abeyance, but merely pointing to the reality that each time the claims are amended, the Office is required to repeat the rationale for the double patenting rejection.

As to quoting footnote 5 of the PTAB Decision, Applicant would like to point out that the phrasing "in the case where" was heavily highlighted by the Board (see pages 8-9) and does not seem applicable in the current set of claims. The undersigned will submit an interview request in order to better understand the position of the Examiner if the current amendments do not overcome the objection and hopefully identify a path to move closer to allowance.

The Examiner is cordially invited to contact the undersigned if it will facilitate prosecution.

Date: June 22, 2023

/Brian J. Hubbard/

Brian J. Hubbard

Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
215-558-5740

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.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) ICVW_2009P00073 US02																														
Application Number 17/351,649	Filed June 18, 2021																															
For METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS																																
Art Unit 2487	Examiner Sullivan, Tyler																															
<p>This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above-identified application.</p> <p>The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):</p> <table><thead><tr><th></th><th>Fee</th><th>Small Entity Fee</th><th>Micro Entity Fee</th><th></th></tr></thead><tbody><tr><td>☐ One month (37 CFR 1.17(a)(1))</td><td>\$220</td><td>\$110</td><td>\$55</td><td>\$ _____</td></tr><tr><td>☒ Two months (37 CFR 1.17(a)(2))</td><td>\$640</td><td>\$320</td><td>\$160</td><td>\$ 640.00</td></tr><tr><td>☐ Three months (37 CFR 1.17(a)(3))</td><td>\$1,480</td><td>\$740</td><td>\$370</td><td>\$ _____</td></tr><tr><td>☐ Four months (37 CFR 1.17(a)(4))</td><td>\$2,320</td><td>\$1,160</td><td>\$580</td><td>\$ _____</td></tr><tr><td>☐ Five months (37 CFR 1.17(a)(5))</td><td>\$3,160</td><td>\$1,580</td><td>\$790</td><td>\$ _____</td></tr></tbody></table> <p>☐ Applicant asserts small entity status. See 37 CFR 1.27.</p> <p>☐ Applicant certifies micro entity status. See 37 CFR 1.29. Form PTO/SB/15A or B or equivalent must either be enclosed or have been submitted previously.</p> <p>☐ A check in the amount of the fee is enclosed.</p> <p>☐ Payment by credit card. Form PTO-2038 is attached.</p> <p>☒ The Director has already been authorized to charge fees in this application to a Deposit Account.</p> <p>☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-5519 _____.</p> <p>☒ Payment made via EFS-Web.</p> <p>WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.</p> <p>I am the</p> <p>☐ applicant.</p> <p>☒ attorney or agent of record. Registration number 45,873 _____.</p> <p>☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.</p> <p>/Brian J. Hubbard/ _____ June 22, 2023 Signature Date</p> <p>Brian J. Hubbard _____ 215-558-5740 Typed or printed name Telephone Number</p> <p>NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.</p>				Fee	Small Entity Fee	Micro Entity Fee		☐ One month (37 CFR 1.17(a)(1))	\$220	\$110	\$55	\$ _____	☒ Two months (37 CFR 1.17(a)(2))	\$640	\$320	\$160	\$ 640.00	☐ Three months (37 CFR 1.17(a)(3))	\$1,480	\$740	\$370	\$ _____	☐ Four months (37 CFR 1.17(a)(4))	\$2,320	\$1,160	\$580	\$ _____	☐ Five months (37 CFR 1.17(a)(5))	\$3,160	\$1,580	\$790	\$ _____
	Fee	Small Entity Fee	Micro Entity Fee																													
☐ One month (37 CFR 1.17(a)(1))	\$220	\$110	\$55	\$ _____																												
☒ Two months (37 CFR 1.17(a)(2))	\$640	\$320	\$160	\$ 640.00																												
☐ Three months (37 CFR 1.17(a)(3))	\$1,480	\$740	\$370	\$ _____																												
☐ Four months (37 CFR 1.17(a)(4))	\$2,320	\$1,160	\$580	\$ _____																												
☐ Five months (37 CFR 1.17(a)(5))	\$3,160	\$1,580	\$790	\$ _____																												

☒ * Total of 1 (one) _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). 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 6 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: Mail Stop PCT, 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 disclosure of these records is required by the Freedom of Information Act.
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 Cooperation Treaty.
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 inspection 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.

PTO/SB/439 (11-15)

AUTHORIZATION FOR INTERNET COMMUNICATIONS IN A PATENT APPLICATION OR REQUEST TO WITHDRAW AUTHORIZATION FOR INTERNET COMMUNICATIONS	Application No.	17/351,649
	Filing Date	June 18, 2021
	First Named Inventor	Yunfei Zheng
	Art Unit	2487
	Examiner Name	SULLIVAN, TYLER
	Practitioner Docket No.	ICVW_2009P00073 US02

To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. To authorize permission for Internet Communications.

☒ Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with the undersigned and practitioners in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. I understand that a copy of these communications will be made of record in the application file. (MPEP 502.03)

II. To withdraw authorization for Internet Communications.

☐ The authorization given on _____, to the USPTO to communicate with the undersigned and any practitioner in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via Internet communications such as video conferencing, instant messaging, or electronic mail is hereby withdrawn. I understand that the withdrawal is effective when approved rather than when received.

I am the

☐ applicant.

☒ attorney or agent of record. Registration number 45873.

☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.

/Brian J. Hubbard/

Signature

June 22, 2023

Date

Brian J. Hubbard

Typed or printed name

215-558-5740

Telephone Number

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Juristic entities must be represented by a patent practitioner (see 37 CFR 1.31, which is applicable to any paper filed on or after September 16, 2012, that is presented on behalf of a juristic entity, regardless of application filing date). Submit multiple forms if more than one signature is required, see below*.

☒ * Total of 1 (one) forms are submitted.

Page 1184 of 1673

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 disclosure of these records is required by the Freedom of Information Act.
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 Cooperation Treaty.
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 inspection 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.

Electronic Patent Application Fee Transmittal				
Application Number:		17351649		
Filing Date:		18-Jun-2021		
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		
Filer:		Brian Hubbard/Cassandra Katz		
Attorney Docket Number:		ICVW_2009P00073 US02		
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
CLAIMS IN EXCESS OF 20	1202	2	100	200
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Extension - 2 months with \$0 paid	1252	1	640	640
Miscellaneous:				
RCE- 1ST REQUEST	1801	1	1360	1360
Total in USD (\$)				2200

Examiner-Initiated Interview Summary	Application No. 17/351,649	Applicant(s) Zheng et al.		
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (First Inventor to File) Status No	Page 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	Title	Type
Tyler W Sullivan	Primary Examiner	Telephonic
Brian Hubbard (Reg. No. 45,873)	Attorney of Record	

Date of Interview: 25 August 2023

Issues Discussed:

Proposed Amendment(s)

Examiner contacted the Attorney on August 23rd, 2023 to indicate newly presented claims 40 - 43 would be allowable should they be incorporated into their respective independent claims and an eTD is filed to address the ODP Rejection. Examiner requested a response by mid-day August 25th, 2023.

Attorney e-mailed the Examiner with a proposed amendment which incorporated new dependent claims 40 - 43 into their respective independent claims and filed an eTD which was approved. Attorney and Examiner spoke about the amendment which is authorized for entry. Examiner was further authorized to correct formalities / 112 issues / dependencies should any be present in the amendment to enter. Attorney and Examiner further discussed timing related to filing an IDS for consideration.

Agreement was reached regarding patentability of the claims.

☒ Attachment

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	
Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04 Please further see: MPEP 713.04 Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b) 37 CFR § 1.2 Business to be transacted in writing	

Applicant recordation instructions:It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

From: [Brian Hubbard](#)
To: [Sullivan, Tyler](#)
Cc: [Cassandra Katz](#)
Subject: Serial No. 17/351649; ICVV_2009P00073 US02
Date: Friday, August 25, 2023 11:34:57 AM
Attachments: [image001.png](#)
[2009P00073_US02_Examiners_Amendment.docx](#)

CAUTION: This email has originated from a source outside of USPTO. **PLEASE CONSIDER THE SOURCE** before responding, clicking on links, or opening attachments.

Examiner Sullivan,

Thank you for your call earlier this week. On June 22, 2023, we authorized internet communications, so a reply by you to this email is proper.

We propose the attached Examiner's amendment. Claims 20-22, 24-27, and 29-39 will be pending. Claims 40-43 are cancelled and their limitations added to their respective independent claims.

Please acknowledge receipt.

Best,
Brian
/Brian J. Hubbard/
Reg. No. 45,873

Brian J. Hubbard
CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd, Suite 1700
Philadelphia, PA 19103

T: 215.600.2373 | M: 989.430.6827 | BHubbard@condoroccia.com



Claims:

1 - 19. (Cancelled)

20. (Currently Amended) 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 32x32, 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.

21. (Previously Presented) The method of claim 20, further comprising decoding an intra prediction mode for the basic coding unit sized sub-block.

22. (Previously Presented) The method of claim 20, 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.

23. (Cancelled)

24. (Previously Presented) The method of claim 20, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

25. (Currently Amended) 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 32x32, 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.

26. (Previously Presented) The apparatus of claim 25, wherein the one or more processors are configured to decode an intra prediction mode for the basic coding unit sized sub-block.

27. (Previously Presented) The apparatus of claim 25, 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.

28. (Cancelled)

29. (Previously Presented) The apparatus of claim 25, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit.

30. (Previously Presented) The apparatus of claim 25, 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.

31. (Previously Presented) The apparatus of claim 25, 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.

32. (Currently Amended) 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 32x32, 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.

33. (Previously Presented) The method of claim 32, further comprising encoding an intra prediction mode for the basic coding unit sized sub-block.

34. (Previously Presented) The method of claim 32, 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.

35. (Previously Presented) The method of claim 32, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

36. (Currently Amended) 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 32x32, 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.

37. (Previously Presented) The apparatus of claim 36, wherein, the one or more processors are configured to encode an intra prediction mode for the basic coding unit sized sub-block.

38. (Previously Presented) The apparatus of claim 36, 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.

39. (Previously Presented) The method of claim 36, wherein the spatial intra partition type specifies a size of a partition of the basic coding unit sized sub-block.

40. - 43. (Cancelled)

Electronic Acknowledgement Receipt	
EFS ID:	48491261
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/Melissa Inzaris
Filer Authorized By:	Brian Hubbard
Attorney Docket Number:	ICVW_2009P00073 US02
Receipt Date:	25-AUG-2023
Filing Date:	18-JUN-2021
Time Stamp:	11:12:51
Application Type:	Utility under 35 USC 111(a)

Payment information:	
Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 170
RAM confirmation Number	E20238OB12499138
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Terminal Disclaimer-Filed (Electronic)	eTerminal-Disclaimer.pdf	41465	no	2
			25d83a058e7807c34c66d7c615995cd044e01c4b		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	38305	no	2
			a09aaffbxcdf55d6fa938bccd7f21d61bb93f56ad		
Warnings:					
Information:					
Total Files Size (in bytes):			79770		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce
Electronic Petition Request	TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	17351649	
Filing Date	18-Jun-2021	
First Named Inventor	Yunfei Zheng	
Attorney Docket Number	ICVW_2009P00073 US02	
Title of Invention	METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS	
☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action		
☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
InterDigital VC Holdings, Inc.	100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>11082697</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none">- expires for failure to pay a maintenance fee;- is held unenforceable;- is found invalid by a court of competent jurisdiction;- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;- has all claims canceled by a reexamination certificate;- is reissued; or- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer. <p>☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>		

<div>☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.</div>	
<div>Applicant claims the following fee status:</div> <div><div>☐ Small Entity</div><div>☐ Micro Entity</div><div>☒ Regular Undiscounted</div></div> <div>I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.</div>	
<div>THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES</div> <div>I certify, in accordance with 37 CFR 1.4(d)(4) that I am:</div> <div><div>☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application</div><div>Registration Number <u>45873</u></div><div>☐ A sole inventor</div><div>☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application</div><div>☐ A joint inventor; all of whom are signing this request</div></div>	
Signature	/Brian J. Hubbard/
Name	Brian J. Hubbard

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal				
Application Number:		17351649		
Filing Date:		18-Jun-2021		
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		
Filer:		Brian Hubbard/Melissa Inzaris		
Attorney Docket Number:		ICVW_2009P00073 US02		
Filed as Large Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	1814	1	170	170
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				170

Doc Code: DISQ.E.FILE
Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 17351649

Filing Date: 18-Jun-2021

Applicant/Patent under Reexamination: Zheng

Electronic Terminal Disclaimer filed on August 25, 2023

☒ APPROVED

This patent is subject to a terminal disclaimer

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	48501431
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:	28-AUG-2023
Filing Date:	18-JUN-2021
Time Stamp:	16:28:23
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Miscellaneous Incoming Letter	ICVW_2009P00073_US02_SIDS _Transmittal.pdf	91761	no	1
			a84f66133ff079dfec98082eb858d9ec7dc242c3		

Warnings:

Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	ICVW_2009P00073_US02_SIDS_1449.pdf	194233	no	1
			4942bd09189396db46d9cef3f2697aeda27c7586		
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	75_EP2081388A1.pdf	1298914	no	19
			06d8bad8a15bb2ecaf5324205290aa31bb59ad03		
Warnings:					
Information:					
4	Foreign Reference	76_KR20090065401A.pdf	1082534	no	25
			cbdfabc06e907b4e29f8ddd5300192f039e3deae		
Warnings:					
Information:					
5	Foreign Reference	77_WO2008083633A1.pdf	3649179	no	30
			9a70bfb491b173463a59c252a458f4a468dd8a89		
Warnings:					
Information:					
6	Non Patent Literature	78_VAISEY_Image_Compression_with_Variable_Block_Size_Segmentation.pdf	2249035	no	21
			3e86ba38aa2498e72a401c7fb45086c536ee643c		
Warnings:					
Information:					
7	Non Patent Literature	81_Chen_Video_Coding_Using_Extended_Block_Sizes_October2008_3pp.pdf	55175	no	3
			b8a7d39987ebd88040ee3c34a750268c3d494213		
Warnings:					
Information:					
8	Non Patent Literature	83_Wien_Intra_Coding_using_Variable_Block_Sizes_VCEG-o31_December2001_12pp.pdf	432954	no	12
			37012fb2fcd3747eaaac3a5fb6226ad525f97b90d		
Warnings:					
Information:					

9	Non Patent Literature	84_English_Translation_KR20090065401A.pdf	12071938	no	23
			9a5f2cb5dc3e5f183a5f18b9ee58bb21d80e0cec		
Warnings:					
Information:					
10	Non Patent Literature	79_VAISEY_Variable_block-size_image_coding_1987_4pp.pdf	6469175	no	4
			cabd11dd963a48f6d4b1b5649482cf3420716819		
Warnings:					
Information:					
11	Non Patent Literature	82_Chen_Video_Coding_Extended_Block_Sizes_ITU-T_COM16_C123E_4pp.pdf	4232298	no	4
			4bb62c80f0976f0d6477bfec112897c2213a5a57		
Warnings:					
Information:					
12	Non Patent Literature	80_Samet_The_Quadtree_and_Related_Hierarchical_Data_Structures_June1984_74pp_.pdf	20205375	no	74
			0816b560c0fa839ae65bed42d6f0657176bf74b89		
Warnings:					
Information:					
Total Files Size (in bytes):			52032571		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: Tyler Sullivan

Title: 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 75-84 listed on the attached Form PTO-1449 are enclosed herewith.

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.	English Language Counterpart	Cite No.
KR 200960065401 A	76	PatDocs English Translation of KR20090065401 A	84
WO 2008-083633 A1	77	EP 2081388 A1	75

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: August 28, 2023

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
215-558-5740

/Brian J. Hubbard/

Brian J. Hubbard
Registration No. 45,873

Substitute for 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	17/351,649
				Filing Date	June 18, 2021
				First Named Inventor	Yunfei Zheng
				Art Unit	2487
				Examiner Name	Tyler Sullivan
Sheet	1	of	1	Attorney Docket Number	ICVW_2009P00073 US02

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number	Country Code	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)	T
	75	2081388	EP	A1	2009-07-22	HUAWEI TECH CO LTD		
	76	20090065401	KR	A	2009-06-22	ICU RES & IND COOP GROUP		x
	77	2008-083633	WO	A1	2008-07-17	HUAWEI TECH CO LTD		

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
	78	Vaisey, et al., "Image Compression with Variable Block Size Segmentation", IEEE, Transactions on Signal Processing, Vol 40, No. 8, August 1992, 3 pages	
	79	Vaisey, et al., "Variable Block-Size Image Coding", Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, IEEE, 1987, 4 pages	
	80	Samet, et al., "The Quadtree and Related Hierarchical Data Structures", Computer Surveys, Vol 16, No. 2, June 1984, 74 pages	
	81	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, 36 th Meeting: San Diego, USA, October 8-10, 2008, 3 pages	
	82	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	
	83	Wen, et al., "Intra Coding using Variable Block Sizes", ITU-Telecommunications Standardization Sector, STUDY GROUP 16 Question 6, Video Coding Experts Group (VCEG), VCEG-O31, 15 th Meeting, Pattaya, Thailand, December 4-6, 2001, 12 pages	
	84	PatDocs English Translation of KR 20090065401 A	

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.			

Other Prior Art

According to the information contained in form PTO-1449 or PTO-892, there are one or more other prior art/non-patent literature documents missing from the original file history record obtained from the United States Patent and Trademark Office. Upon your request we will attempt to obtain these documents from alternative resources. Please note that additional charges will apply for this service.



(11) EP 2 081 388 A1

(12) **EUROPEAN PATENT APPLICATION**
published in accordance with Art. 153(4) EPC

(43) Date of publication: 22.07.2009 Bulletin 2009/30	(51) Int Cl.: H04N 7/32 (2006.01)
(21) Application number: 08700767.0	(86) International application number: PCT/CN2008/070092
(22) Date of filing: 11.01.2008	(87) International publication number: WO 2008/083633 (17.07.2008 Gazette 2008/29)

<p>(84) Designated Contracting States: AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR</p> <p>(30) Priority: 11.01.2007 CN 200710007801</p> <p>(71) Applicant: Huawei Technologies Co., Ltd. Shenzhen, Guangdong Province 518129 (CN)</p>	<p>(72) Inventor: XIONG, Lianhuan Guangdong 518129 (CN)</p> <p>(74) Representative: Körber, Martin Hans Mitscherlich & Partner Patent- und Rechtsanwälte Sonnenstraße 33 80331 München (DE)</p>
---	---

(54) **A METHOD AND A DEVICE FOR INTRA FRAME PREDICTION ENCODING/DECODING**

(57) Methods and devices of intra prediction encoding and decoding are described, so that the encoding performance of intra coding is better. A prediction encoding is performed on the same frame by using at least two intra prediction encoding modes, and information indicating the intra prediction encoding mode of each block is written into the encoded result, so as to encode different blocks by using the most appropriate modes. By using at least two intra prediction encoding modes, relevant indication information is written into the encoded result, so as to correctly perform the decoding. During decoding, the information of intra prediction encoding mode of each block of the same frame is firstly read, and then the block is decoded by using this mode. Each block is organized in a form of hierarchical tree, so as to obtain the optimal performance, and reduce the amount of the information indicating the intra prediction encoding mode of each block in the encoded result.

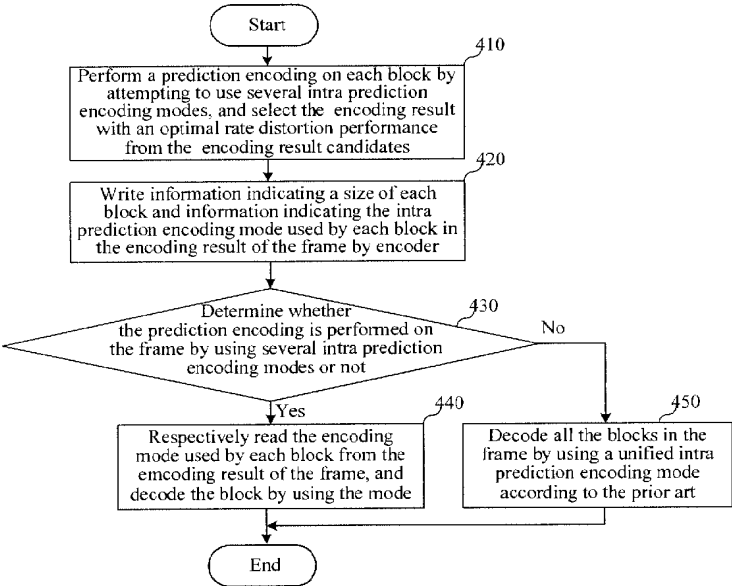


FIG. 4

Printed by Jouve, 75001 PARIS (FR)

EP 2 081 388 A1

Description

FIELD OF THE TECHNOLOGY

[0001] The present invention relates to a video encoding and decoding field, and more particularly to methods and devices of intra prediction encoding and decoding.

BACKGROUND

[0002] A video has a large amount of information, and during video encoding, if an encoder independently performs intra encoding without depending on the previous encoded frame, the decoder may directly decode the frame without knowing the previous decoded frame. However, in this method, compression efficiency of the intra encoding is quite low, and the amount of the encoded information is extremely large. In order to improve the encoding compression efficiency, during the intra encoding, the prediction is performed on the current block by using the peripheral information; or, for an inter prediction frame, particularly a motion compensated frame, the intra coding of the current frame may be performed by using the inter-encoded blocks as the prediction reference of the current encoded frame. If one frame only uses the intra encoding, the frame may serve as a starting point of a video sequence, for example, an instantaneous decoding refresh (IDR) frame in H.264/advanced video coding (AVC) standard.

[0003] The H.264/AVC intra prediction encoding technique is briefly introduced as below. The main objective of the H.264/AVC intra encoding is reducing the space redundancy of the frame. In order to improve the efficiency of the H.264/AVC intra frame encoding, the space dependency of the neighboring macroblocks is fully utilized of the frame, because the neighboring macroblocks usually have the similar characteristics. Therefore, when a macroblock is encoded, the prediction is performed according to the neighboring macroblocks, and then the difference between the predicted value and the actual value is encoded. Referring to FIG 1, it is a schematic view of the method of H.264/AVC intra prediction encoding in the prior art. It may be known from FIG 1 that compared with the technique of directly encoding the frame, this method may greatly reduce the bit rate.

[0004] The H.264/AVC supports the intra prediction on the basis of 4×4 , 8×8 , and 16×16 blocks, and provides various prediction modes including one DC prediction mode and several directional prediction modes. Through this intra prediction, the space redundancy of the picture is greatly reduced, and the encoding compression efficiency is improved.

[0005] In addition to the method of H.264/AVC intra prediction encoding, in the prior art, a method of H.264/AVC displacement intra prediction (DIP) encoding is further provided. In this method, the inter prediction with the variable size of the block in the H.264/AVC is applied to the H.264/AVC intra prediction. The current block may refer to the previous encoded area by using a displacement vector labeling method. In other words, when the current block is being encoded, firstly a block similar to the current block is searched in the encoded area and chosen as a candidate block, and the candidate block is indicated through the displacement vector labeling method, which is used to identify the position difference between the current block and the candidate block. Because the candidate block is decoded before the current block, the decoder can decode the current block according to the decoded information of the candidate block, the displacement vector, and other information. Referring to FIG 2, it is a schematic view of a method of DIP encoding in the prior art. The method as shown in FIG 2 is usually applied to the macroblock level.

[0006] In the prior art, a method of intra prediction encoding by using template matching is further provided. Referring to FIG 3, it is a schematic view of the method of template matching intra prediction encoding in the prior art. In the method, self-similar area of prediction picture area are exploited for prediction by using the template matching, in which the prediction picture area is also referred to as an object area. Particularly, the neighboring area of the prediction picture area serves as a template, the area having pixels most similar to those of the template is iteratively searched in the previous encoded area, the most similar area is referred to as a candidate neighboring area, and the previous encoded area is also referred to as the searching area. The decoder may found the candidate neighboring area from the searching area, and because the pixels of candidate neighboring area are similar to those of the neighboring area of the object area, the decoder may obtain the prediction of the object area without any other additional header information. The template matching technique may serve as an additional mode of the H.264/AVC, and is applicable to the intra 4×4 block or the intra 8×8 prediction block.

[0007] During the process of implementing the present invention, inventors find that the prior art at least has the problems as follows.

[0008] The H.264/AVC intra prediction encoding, the DIP encoding, and the template matching intra prediction encoding have respective features, and are respectively applicable to a part of the scenarios. Different intra prediction techniques are applicable to the different local areas in a video picture. The H.264/AVC intra prediction technique is applicable to an area where the current block and neighboring pixels have a similar direction of extend texture structure. The DIP technique is applicable to an area which has the repeated structure. The template matching intra prediction technique is applicable to an area where the reference neighboring area has similar structures. It may be known that the encoding

performance obtained by using one intra prediction encoding method is not optimal.

SUMMARY

- 5 [0009] Various embodiments of the present invention provide methods and devices of intra prediction encoding and decoding which make coding performance better.
- [0010] In order to solve the above technical problem, one embodiment of the present invention provides a method of intra prediction encoding, which includes the steps as follows. A prediction encoding is performed on blocks in a frame by using at least two intra prediction encoding modes. Information indicating the intra prediction encoding mode of each block of the frame is written into an encoded result of the frame. In the encoded result, each block uses only one intra prediction encoding mode, and at least two encoding modes are used in the encoded result of the frame.
- 10 [0011] One embodiment of the present invention provides a method of intra prediction decoding, which includes the steps as follows. Information indicating an intra prediction encoding mode of each block of a frame is read from an encoded result of the frame, and the block is decoded by using the mode.
- 15 [0012] One embodiment of the present invention provides a device of intra prediction encoding, which includes a unit adapted to perform a prediction encoding on blocks in a frame by using at least two intra prediction encoding modes; and a unit adapted to write information indicating the intra prediction encoding mode of each block of the frame into an encoded result of the frame. In the encoded result, each block of the frame uses only one intra prediction encoding mode, and at least two encoding modes are used in the encoded result of the frame.
- 20 [0013] One embodiment of the present invention provides a device of intra prediction decoding, which includes at least two decoders, each decoder adapted to decode according to one intra prediction encoding mode; and a first reading unit adapted to read information indicating an intra prediction encoding mode of each block in a frame from an encoded result of the frame, and indicate the decoder corresponding to the intra prediction encoding mode to decode the block.
- [0014] As compared with the prior art, the present invention has beneficial effects as follows.
- 25 [0015] The prediction encoding is performed on the same frame by using at least two intra prediction encoding modes, and the information indicating the intra prediction encoding mode used by each block is written into the encoded result, so that different blocks may be encoded by using the most appropriate modes. As compared with the technical solution in the prior art that the intra encoding is performed on all the blocks of the frame by using only one mode, the present invention obtains the better coding performance.
- 30 [0016] If some frames in one media stream use at least two intra prediction encoding modes, but some frames in the media stream use only one intra prediction encoding mode, information indicating that at least two intra prediction encoding modes are used should be written into the encoded result of the former one, so as to correctly perform the decoding process.
- [0017] Under a situation that at least two intra prediction encoding modes are used, during decoding process, the intra prediction encoding mode used by each block of the same frame is firstly read, and then the block is decoded by using this mode, so each block is correctly decoded.
- 35 [0018] Information indicating the size of each block is written into the encoded result, so that each block of the same frame is not limited to a fixed dividing pattern, and the better coding performance by using different intra prediction encoding modes is obtained.
- 40 [0019] The prediction encoding is performed on each block by attempting to use several intra prediction encoding modes, the results by using several intra prediction encoding modes are obtained as candidates, and then an optimal candidate is selected as a final encoding result according to a preset performance measurement, for example, a rate distortion performance, so that the final encoding result has an optimal performance.
- [0020] The blocks of the frame are organized in a form of hierarchical tree, so as to obtain the optimal performance, and reduce the amount of information indicating the intra prediction encoding mode used by each block in the encoded result.
- 45 [0021] The technical solution of the present invention may be applied in a space domain frame or a frequency domain frame, and the frequency domain includes the frequency domains generated by various transforms, such as a discrete cosine transform (DCT), a discrete Fourier transformation (DFT), and a wavelet transform. Therefore, embodiments of the present invention have a wide application field.
- 50

BRIEF DESCRIPTION OF THE DRAWINGS

- [0022] FIG 1 is a schematic view of a method of H.264/AVC intra prediction encoding in the prior art;
- 55 [0023] FIG 2 is a schematic view of a method of DIP encoding in the prior art;
- [0024] FIG 3 is a schematic view of a method of intra prediction encoding by using template matching in the prior art;
- [0025] FIG 4 is a flow chart of a method of intra prediction encoding and decoding according to a first embodiment of the present invention;

[0026] FIG 5 is a schematic view of dividing a macroblock in the method of intra prediction encoding and decoding according to the first embodiment of the present invention;

[0027] FIG 6 is a flow chart of a decoding example in the method of intra prediction encoding and decoding according to the first embodiment of the present invention;

[0028] FIG 7 is a schematic view of a rate distortion performance gain brought by the method of intra prediction encoding and decoding according to the first embodiment of the present invention;

[0029] FIG 8 is a structural view of a device of intra prediction encoding according to a second embodiment of the present invention; and

[0030] FIG 9 is a structural view of a device of intra prediction decoding according to the second embodiment of the present invention.

DETAILED DESCRIPTION

[0031] In order to have a further understanding of the objective, technical solution and advantage of the present invention, a detailed description of the present invention is given as follows with accompanying drawings.

[0032] In each embodiment of the present invention, a prediction encoding is performed on the same frame by using at least two intra prediction encoding modes, and information indicating the intra prediction encoding mode used by each block is written into an encoded result, so as to encode different blocks by using the most appropriate modes. As compared with the technical solution in the prior art that the intra encoding is performed on all the blocks of the frame by using only one mode, the technical solution of the present invention obtains a better coding performance.

[0033] A first embodiment of the present invention is described as follows, and referring to FIG 4, the first embodiment of the present invention relates to a method of intra prediction encoding and decoding.

[0034] In Step 410, an encoder performs a prediction encoding on each block of a frame by attempting to use all available intra prediction encoding modes to obtain corresponding encoding result candidates respectively, and the candidate with the optimal performance is selected from the candidates to serve as a final encoding result of the block. The manner is equivalent to encoding the different blocks by using the most appropriate modes. As compared with the technical solution in the prior art that the intra encoding is performed on all the blocks of the frame by using only one mode, this embodiment obtains the better encoding performance. Here, the performance measurement may be a rate distortion performance. The candidates of intra prediction encoding mode may be an H.264/AVC directional prediction mode, a DIP mode and a template matching prediction mode etc.

[0035] Particularly, the encoded block may be a basic encoding unit, that is, macroblock, and may be sub-blocks divided from the macroblock. After several modes of prediction encoding are performed on each block, and the encoding result with the optimal performance measurement is selected as the final encoding result, according to the intra prediction encoding mode corresponding to the final encoding result of each block, the blocks are organized in a form of hierarchical tree, so as to obtain the optimal performance and reduce the amount of information indicating the intra prediction encoding mode used by each block in the encoded result. For example, for a 16×16 macroblock, one 16×6 macroblock may be divided into four 8×8 sub-blocks, and one 8×8 sub-block may also be further divided into four 4×4 sub-blocks. The encoder respectively encodes the 16×16 macroblock, each 8×8 sub-block in the macroblock, and each 4×4 sub-block in the macroblock by using several prediction modes, so as to select the encoding mode combination with the optimal rate distortion performance from the candidate encoding result combinations to serve as the final encoding result of the macroblock, thereby organizing the macroblock in the form of hierarchical tree. FIG 5 is schematic view of dividing a macroblock in the method of intra prediction encoding and decoding according to the first embodiment of the present invention. The example of the encoding result combinations of the 16×16 macroblock is shown in FIG 5, the final encoding result of the 8×8 sub-block on the top left corner is the encoding result using the DIP mode, the final encoding result of the 8×8 sub-block on the bottom left corner is the encoding result using the H.264/AVC directional prediction mode, the final encoding result of the 8×8 sub-block on the bottom right corner is the encoding result using the template matching intra prediction mode, the final encoding result of the 4×4 sub-block on the top right corner is the encoding result using the DIP mode, and the optimal encoding results of the remaining three 4×4 sub-blocks are all the encoding results of the H.264/AVC directional prediction mode.

[0036] Definitely, here the macroblock is not always the 16×16 macroblock, and may be a macroblock of other formats, for example, a 32×32 macroblock. Similarly, the sub-block is not always organized in a form of quartered hierarchical level, and may also be organized in other forms, for example, a form of dimidiate hierarchical level. In the form of dimidiate hierarchical level, one 16×16 macroblock includes two 8×16 sub-blocks or two 16×8 sub-blocks.

[0037] In Step 410, the details of the prediction encoding modes may be obtained with reference to the prior art. For example, for the H.264/AVC directional prediction encoding mode, please refer to ITU-T Rec. H.264, Advanced Video Coding for Generic Audiovisual Services, 2005; for the DIP encoding mode, please refer to S.-L. Yu and C. Chrysafis, New Intra Prediction using Intra-Macroblock Motion Compensation, JVT meeting Fairfax, doc JVT-C151, May 2002; and for the template matching intra prediction encoding mode, please refer to T.K. Tan, C.S. Boon, and Y Suzuki, Intra

Prediction by Template Matching, ICIP 2006.

[0038] It should be noted that when the DIP is performed, a displacement vector is encoded by using the method similar to the inter frame motion compensation and by adopting a motion vector encoding method in an H.264/AVC standard.

[0039] In addition, in this embodiment, a scanning sequence of the sub-blocks of the macroblock is the same as the scanning sequence of the sub-blocks of the H.264/AVC P_8x8 macroblock type.

[0040] Next, in Step 420, the information indicating the size of each block and the information indicating the intra prediction encoding mode used by each block are written into the encoded result of the frame. If each block of a frame includes the information indicating the size of the block, the blocks of the frame are not limited to a fixed division pattern, thereby facilitating the obtaining of the better encoding performance by using different intra prediction encoding modes.

[0041] Particularly, if the prediction encoding is performed on each block of the frame by using at least two different intra prediction encoding modes, and each mode is applied to one or more blocks of the frame, information indicating that the prediction encoding is performed on the frame by using several intra prediction encoding modes is written in the encoded result of the frame, so that even the technical solution of the present invention may coexist with the technical solution in the prior art that the intra frame encoding is performed on all the blocks of the frame by using only one mode, and the decoding end may perform the correct decoding by parsing the encoded result.

[0042] If a block, for example, a macroblock or a sub-block, corresponding to a root node of one branch of the hierarchical tree, uses only one intra prediction encoding mode, that is, the optimal encoding method of all the sub-blocks in the macroblock or the sub-block is the same intra prediction encoding mode, or the macroblock or the sub-block is the block corresponding to a leaf node of one branch, the information indicating the size of the block and the information indicating the intra prediction encoding mode used by the block are written into the encoded result of the frame; otherwise, each sub-branch of the branch is processed.

[0043] It should be noted that for the DIP encoding mode, the displacement vector is required to be encoded.

[0044] A storage structure of the frame indicated by a syntax table is shown in Table 1 as follows. The type of macroblocks, on which the prediction encoding is performed by using several frame prediction encoding modes, is indicated by I_EXT, and NumMbPart(I_EXT)=1.

[0045] In the Table 1, ext_intra_pred_flag indicates whether the prediction encoding is performed on the frame by using several intra prediction encoding modes or not, and ext_intra_pred_flag being equal to 1 indicates that the prediction encoding is performed by using several intra prediction encoding modes.

[0046] ext_16x16_pred_flag being equal to 1 indicates that the macroblock using the intra prediction is 16x16.

[0047] ext_16x16_pred_flag being equal to 0 indicates that the macroblock is further divided into 8x8.

[0048] ext_16x16_pred_mode indicates the intra prediction encoding mode of the 16x16 macroblock.

[0049] ext_8x8_pred_flag[] being equal to 1 indicates that the prediction is performed according to the 8x8 division, and ext_8x8_pred_flag[] being equal to 0 indicates that the sub-block is further divided into 4x4 sub-blocks.

[0050] ext_8x8_pred_mode[] indicates the intra prediction encoding mode of each 8x8 sub-block.

[0051] ext_4x4_pred_mode[][] indicates the intra prediction encoding mode of each 4x4 sub-block.

[0052] dvd[][] indicates the displacement vector in the DIP encoding mode.

Table 1

	mb_pred(mb_type) {	C	Descriptor
5	ext_intra_pred_flag	2	u(1) ae(v)
	if(ext_intra_pred_flag) {		
	ext_16x16_pred_flag	2	u(1) ae(v)
	if(ext_16x16_pred_flag)		
10	ext_16x16_pred_mode	2	ue(v) ae(v)
	if(ext_16x16_pred_mode == DIP)		
	for(compIdx = 0; compIdx < 2; compIdx++)		
	dvd[0][0][compIdx]	2	ue(v) ae(v)
	else {		
15	for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
	ext_8x8_pred_flag[luma8x8BlkIdx]	2	u(1) ae(v)
	if(ext_8x8_pred_flag[luma8x8BlkIdx]) {		
	ext_8x8_pred_mode[luma8x8BlkIdx]	2	ue(v) ae(v)
20	if(ext_8x8_pred_mode[luma8x8BlkIdx] == DIP)		
	for(compIdx = 0; compIdx < 2; compIdx++)		
	dvd[luma8x8BlkIdx][0][compIdx]	2	ue(v) ae(v)
	} else {		
25	noSubMbPartSizeLessThan8x8Flag = 0		
	for(luma4x4BlkIdx=0; luma4x4BlkIdx<4; luma4x4BlkIdx++		
	) {		
	ext_4x4_pred_mode[luma8x8BlkIdx][luma4x4BlkIdx]	2	ue(v) ae(v)
30	if(ext_4x4_pred_mode[luma8x8BlkIdx][luma4x4BlkIdx]		
	==		
	DIP)		
	for(compIdx = 0; compIdx < 2; compIdx++)		
35	dvd[luma8x8BlkIdx][luma4x4BlkIdx][compIdx]	2	ue(v) ae(v)
	}		
	}		
	}		
40	}		
	} else {		
	...		

45 [0053] In addition, ae(v) is a syntax element encoded by a context-based adaptive binary arithmetic code.

[0054] u(n) is n bits of unsigned integers. In the syntax table, if n is 'v', a bit number thereof is determined by other syntax element values. A parsing process is defined by the return value of function read_bits(n), and the return value is indicated by a high-to-low order binary value.

[0055] ue(v) is a unsigned integer syntax element encoded by using an exponential-Golomb code.

50 [0056] Next, in Step 430, the decoder firstly determines whether the prediction encoding is performed on the frame by using several intra prediction encoding modes or not. If information indicating that the prediction encoding is performed on the frame by using at least two intra prediction encoding modes is read from the encoded result of the frame, the procedure proceeds to Step 440; otherwise, the procedure proceeds to Step 450, in which all the blocks of the frame are decoded by using a unified intra prediction encoding mode according to the prior art.

55 [0057] In Step 440, the decoder reads the encoded result of the frame, and searches for the information indicating the size of the block in the encoded result one by one, and performs the decoding according to the information indicating the size of the block and the information indicating the intra prediction encoding mode used by the block. The blocks of the frame are organized in the form of hierarchical tree. If the information indicating the size of the block is read from

the encoded result of the frame, and the block includes the macroblocks or the sub-blocks, it indicates that the block with the size adopts the same intra prediction encoding mode. The decoder further reads the information indicating the intra prediction encoding mode used by the block, and decodes the block with the size by using this mode.

[0058] An example is given according to the encoding structure of Table 1 as follows, and referring to FIG 6, it is a flow chart of a decoding example in the method of intra prediction encoding and decoding according to the first embodiment of the present invention. According to an `ext_intra_pred_flag` flag bit obtained by parsing, the decoder firstly determines whether the prediction encoding is performed on the current encoded stream by using several intra prediction encoding modes; and if the prediction encoding is not performed on the current encoded stream by using several intra prediction encoding modes, all the blocks of the frame are decoded by using the unified intra frame prediction encoding mode according to the prior art, for example, according to the H.264/AVC standard; and if the prediction encoding is performed on the current encoded stream by using several intra prediction encoding modes, the current macroblock division situation of the intra frame is further determined according to the `ext_16x16_pred_flag` flag bit.

[0059] If the `ext_16x16_pred_flag` is equal to 1, it indicates that the macroblock is divided into 16×16 , the encoding mode of the current block is obtained by parsing the `ext_16x16_pred_mode`, and the 16×16 macroblock is decoded according to the decoding mode corresponding to the encoding mode. If the encoding mode of the current block is the DIP encoding mode, the displacement vector `dvd` is parsed from the encoded stream, and then the block is decoded by using the corresponding decoding method. If the `ext_16x16_pred_flag` is equal to 0, it indicates that the 16×16 macroblock is divided into 8×8 blocks, and thus each 8×8 block of the macroblock is further parsed.

[0060] For each 8×8 block, according to the `ext_8x8_pred_flag`, if the `ext_8x8_pred_flag` is equal to 1, the prediction encoding mode `ext_8x8_pred_mode` of the 8×8 block in the encoded stream is parsed, and the corresponding decoding method is selected according to the encoding mode, so as to perform the decoding. If the current block encoding mode is the DIP encoding mode, the displacement vector `dvd` of each 8×8 block is parsed from the encoded stream, and then the decoding is performed by using the corresponding decoding method. If the `ext_8x8_pred_flag` is equal to 0, it indicates that the current block is further divided into 4×4 sub-blocks, and the prediction encoding mode `ext_4x4_pred_mode` of each 4×4 block is parsed.

[0061] For each 4×4 block, the prediction encoding mode `ext_4x4_pred_mode` is parsed; the corresponding method of decoding is selected according to the encoding mode, so as to perform the decoding. If the encoding mode of the current block is the DIP encoding mode, the displacement vector `dvd` of each 4×4 block is parsed from the encoded stream, and then the decoding is performed by using the corresponding method of decoding.

[0062] FIG 7 shows the rate distortion performance gain brought by the method of intra prediction encoding and decoding according to the embodiment of the present invention.

[0063] It should be noted that the frame in this implementation may be the frame of a space domain or a frequency domain, the frequency domain includes the frequency domains generated by various transforms, for example, a DCT, a DFT, and a wavelet transform etc. Therefore, the present invention has a wide application field.

[0064] A second embodiment of the present invention relates to a device of intra prediction encoding, and referring to FIG 8, it is a structural view of the device of intra prediction encoding according to the second embodiment of the present invention. The device of intra prediction encoding of this embodiment includes an encoding unit adapted to perform a prediction encoding on a frame by using at least two intra prediction encoding modes, and each mode is applied to at least one block of the frame. The prediction encoding is performed by using several intra prediction encoding modes, so as to encode the different blocks by using the most appropriate modes. As compared with the technical solution in the prior art that the intra encoding is performed on all the blocks of the frame by using only one mode, the encoding device of this embodiment obtains the better encoding performance. The encoding unit further includes at least two encoders, each of the encoders adapted to perform the prediction encoding on each block by using one intra prediction encoding mode, and a selector adapted to select a encoding result with the optimal rate distortion performance from the encoding result candidates output from the encoders to serve as a final encoding result of the block, so that the final encoding result has the optimal performance. The device of intra prediction encoding further includes a writing unit adapted to write information indicating the intra prediction encoding mode used by each block in the frame and information indicating the size of each block of the frame into the encoded result of the frame. By writing the information indicating the size of each block into the encoded result, the blocks of a frame may be not limited to a fixed dividing pattern, thus obtaining a better encoding performance by using the different intra prediction encoding modes. If some frames in one media stream use at least two intra prediction encoding modes, but some other frames in the media stream use only one intra prediction encoding mode, when the prediction encoding is performed on one frame by using at least two intra prediction encoding modes, the writing unit writes the information indicating that the prediction encoding is performed on the frame by using at least two intra prediction encoding modes into the encoded result of the frame, so as to correctly perform the decoding. The frame may be a space domain frame or a frequency domain frame, and the performance measurement may be the rate distortion performance, and the intra prediction encoding modes at least include a directional prediction mode, a DIP mode, and a template matching prediction mode.

[0065] In addition, the writing unit may organize each block in a form of hierarchical tree, so as to obtain the optimal

performance, and reduce the amount of the information indicating the intra prediction encoding mode used by each block in the encoded result.

[0066] A third embodiment of the present invention relates to a device of intra prediction decoding, and referring to FIG 9, it is a structural view of a device of intra prediction decoding according to the third embodiment of the present invention. The device of intra prediction decoding of this embodiment includes a decoding unit, a first reading unit, and a second reading unit.

[0067] The decoding unit includes at least two decoders, each decoder adapted to decode according to one intra prediction encoding mode.

[0068] The first reading unit is adapted to read information indicating an intra prediction encoding mode used by each block of a frame from an encoded result of the frame, and indicate the decoder corresponding to the intra prediction encoding mode to decode the block.

[0069] The second reading unit is adapted to attempt to read information indicating that prediction encoding is performed on the frame by using at least two intra prediction encoding modes from the encoded result of the frame, and indicate the first reading unit and each decoder to respectively decode each block of the frame, if the information indicating that the prediction encoding is performed on the frame by using the at least two intra prediction encoding modes is read; otherwise, indicate one of the decoders to decode all the blocks of the frame by using a unified intra prediction encoding mode.

[0070] Similarly, the frame is a frame of a space domain or a frequency domain, and the frequency domain includes the frequency domains generated by various transforms, for example, a DCT, a DFT, and a wavelet transform etc. Therefore, the present invention has a wide application field.

[0071] The intra prediction encoding modes at least include a directional prediction mode, a DIP mode, and a template matching prediction mode.

[0072] It should be noted that the units in the embodiments of the two devices are logic units, and during practical application, the units may have different physical realizing forms.

[0073] To sum up, in each embodiment of the present invention, the prediction encoding is performed on the same frame by using at least two intra prediction encoding modes, and the information indicating the intra prediction encoding mode used by each block is written into the encoded result, so as to encode the different blocks by using the most appropriate modes. As compared with the technical solution in the prior art that the intra encoding is performed on all the blocks of the frame by using only one mode, the present invention obtains the better encoding performance.

[0074] If some frames in one media stream use at least two intra prediction encoding modes, but some other frames in the media stream use only one intra prediction encoding mode, the information indicating that at least two intra prediction encoding modes are used is written into the encoded result of the former one, so as to correctly perform the decoding.

[0075] Under a situation that at least two intra prediction encoding modes are used, during decoding, the intra prediction encoding mode used by each block of the same frame is firstly read, and then the block is decoded by using the mode, so each block is correctly decoded.

[0076] The information indicating the size of each block is written into the encoded result, so that blocks of the same frame are not limited to the fixed dividing pattern, thereby obtaining better encoding performance by using the different intra prediction encoding modes.

[0077] The prediction encoding is performed on each block by attempting to use several intra prediction encoding modes, several encoding result candidates are obtained, and then according to a preset performance measurement, for example, the rate distortion performance, the optimal encoding result candidate is selected as the final encoding result, so that the final encoding result has the optimal performance.

[0078] Each block is organized in the form of hierarchical tree, so as to obtain the optimal performance, and reduce the amount of the information indicating the intra prediction encoding mode used by each block in the encoding result.

[0079] The technical solution of the present invention may be applied to the frame of a space domain or the frame of a frequency domain, and the frequency domain includes the frequency domains generated by various transforms, for example, the DCT, the DFT, and the wavelet transform etc. Therefore, the present invention has a wide application field.

[0080] It will be apparent to those skilled in the art that various modifications and variations can be made to the structure of the present invention without departing from the scope of the invention. In view of the foregoing, it is intended that the present invention cover modifications and variations of this invention provided they fall within the scope of the following claims and their equivalents.

Claims

1. A method of intra prediction encoding, **characterized by** comprising the steps of:

performing a prediction encoding on blocks in a frame by using at least two intra prediction encoding modes; and writing information indicating the intra prediction encoding mode of each block of the frame to an encoded result of the frame, wherein each block of the frame was encoded by using the alternative of said at least two intra prediction encoding modes .

- 5
2. The method of intra prediction encoding according to claim 1, **characterized in that** the method further comprises the step of:

10 writing information indicating that the prediction encoding is performed by using at least two intra prediction encoding modes into the encoded result of the frame.

3. The method of intra prediction encoding according to claim 1, **characterized in that** the method further comprises the step of:

15 writing information indicating the size of each block into the encoded result of the frame.

4. The method of intra prediction encoding according to claim 1, **characterized in that** the step of performing the prediction encoding on blocks in a frame by using at least two intra prediction encoding modes comprises the steps of:

20 respectively performing the prediction encoding on each block of the frame by attempting to use all the available intra prediction encoding modes;
obtaining at least two encoding result candidates of the block;
selecting a encoding result with an optimal performance measurement from the encoding result candidates as a final encoding result of the block.

- 25
5. The method of intra prediction encoding according to claim 4, **characterized in that** the performance measurement is a rate distortion performance.

- 30
6. The method of intra prediction encoding according to any one of claims 1 to 5, **characterized in that** the intra prediction encoding modes comprise:

a directional prediction mode, a displacement intra prediction mode, and a template matching prediction mode.

- 35
7. The method of intra prediction encoding according to any one of claims 1 to 5, **characterized in that** the frame is a space domain frame or a frequency domain frame.

8. The method of intra prediction encoding according to any one of claims 1 to 5, **characterized in that** the blocks are organized in a form of hierarchical tree; and
40 if the block corresponding to a root node of one branch of the hierarchical tree is encoded by using only one intra prediction encoding mode, the information indicating the size of the block and the information indicating the intra prediction encoding mode of the block are written into the encoded result of the frame; otherwise, each sub-branch of the branch is further processed.

- 45
9. A method of intra prediction decoding, **characterized by** comprising the steps of:

50 reading information indicating an intra prediction encoding mode of each block of a frame from an encoded result of the frame, wherein each block of the frame was encoded by using the alternative of at least two intra prediction encoding modes;
decoding each block by using the decoding mode corresponding to the intra prediction encoding mode indicated by the information.

10. The method of intra frame prediction decoding according to claim 9, **characterized in that** the method further comprises the steps of:

55 if information indicating that a prediction encoding is performed on the frame by using at least two intra prediction encoding modes is read from the encoded result of the frame, respectively decoding each block of the frame by using the decoding mode corresponding to the intra prediction encoding mode; otherwise, decoding all the blocks of the frame by using a unified intra prediction decoding mode.

11. The method of intra prediction decoding according to claim 9, **characterized in that** the blocks are organized in a form of a hierarchical tree; and
if information indicating the size of the block is read from the encoded result of the frame, information indicating the intra prediction encoding mode of the block is further read, and the block with the size is decoded by using the decoding mode corresponding to the intra prediction encoding mode.

12. The method of intra prediction decoding according to claim 9, **characterized in that** the intra prediction encoding modes comprise:

a directional prediction mode, a displacement intra prediction mode, and a template matching prediction mode.

13. The method of intra prediction decoding according to claim 9, **characterized in that** the frame is a space domain frame or a frequency domain frame.

14. A device of intra frame prediction encoding, **characterized by** comprising:

a unit adapted to perform a prediction encoding on blocks in a frame by using at least two intra prediction encoding modes; and

a unit adapted to write information indicating the intra prediction encoding mode of each block of the frame into an encoded result of the frame, wherein each block of the frame was encoded by using the alternative of said at least two intra prediction encoding modes.

15. The device of intra prediction encoding according to claim 14, **characterized in that** the device of intra frame prediction encoding further comprise: a unit adapted to write information indicating that the prediction encoding is performed on the frame by using the at least two intra prediction encoding modes into the encoded result of the frame.

16. The device of intra prediction encoding according to claim 14, **characterized in that** the device of intra frame prediction encoding further comprise: a unit adapted to write information indicating the size of each block into the encoded result of the frame.

17. The device of intra prediction encoding according to claim 14, **characterized in that** the unit adapted to perform the prediction encoding on blocks in a frame by using at least two intra prediction encoding modes further comprises:

at least two encoders, each encoder adapted to perform the prediction encoding on each block by using one intra prediction encoding mode; and

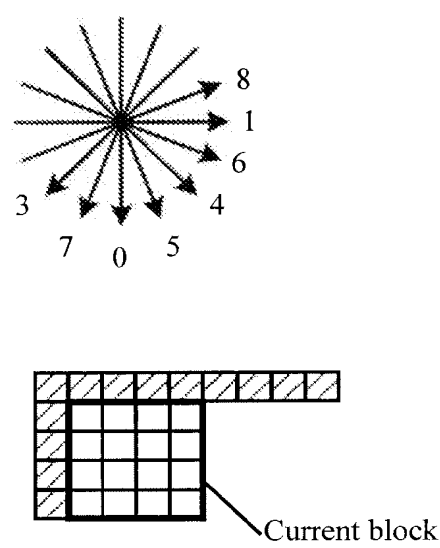
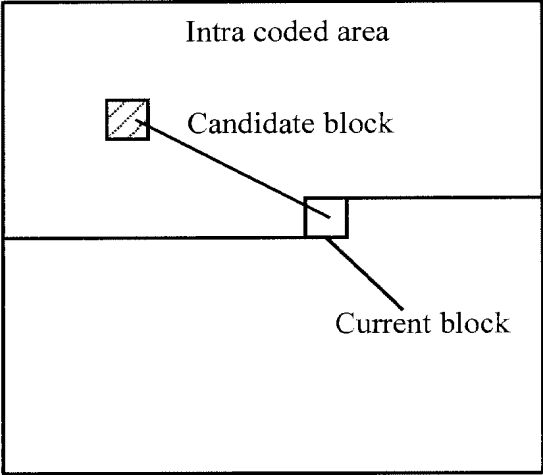
a selector adapted to select a encoding result with an optimal rate distortion performance from the encoding result candidates output from the encoders as a final encoding result of the block.

18. A device of intra frame prediction decoding, **characterized by** comprising:

at least two decoders, each decoder adapted to decode according to one intra prediction encoding mode; and a first reading unit adapted to read information indicating an intra prediction encoding mode of each block in a frame from an encoded result of the frame, and indicate the decoder corresponding to the intra prediction encoding mode to decode the block.

19. The device of intra prediction decoding according to claim 18, **characterized in that** the device of intra frame prediction decoding further comprise:

a second reading unit adapted to attempt to read information indicating that a prediction encoding is performed on the frame by using at least two intra prediction encoding modes from the encoded result of the frame; if the information indicating that the prediction encoding is performed on the frame by using at least two intra prediction encoding modes is read, indicate the first reading unit and the decoders to respectively decode each block of the frame; otherwise, indicate one of the decoders to decode all the blocks of the frame by using a unified intra prediction decoding mode.



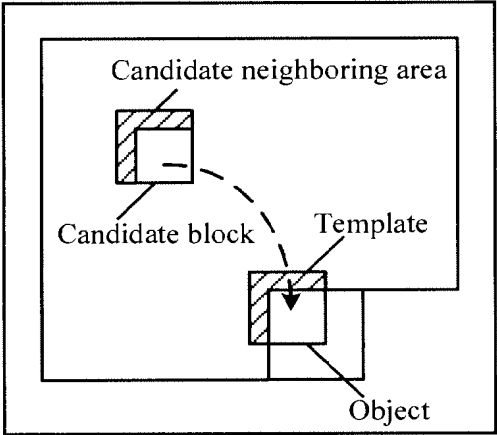


FIG. 3

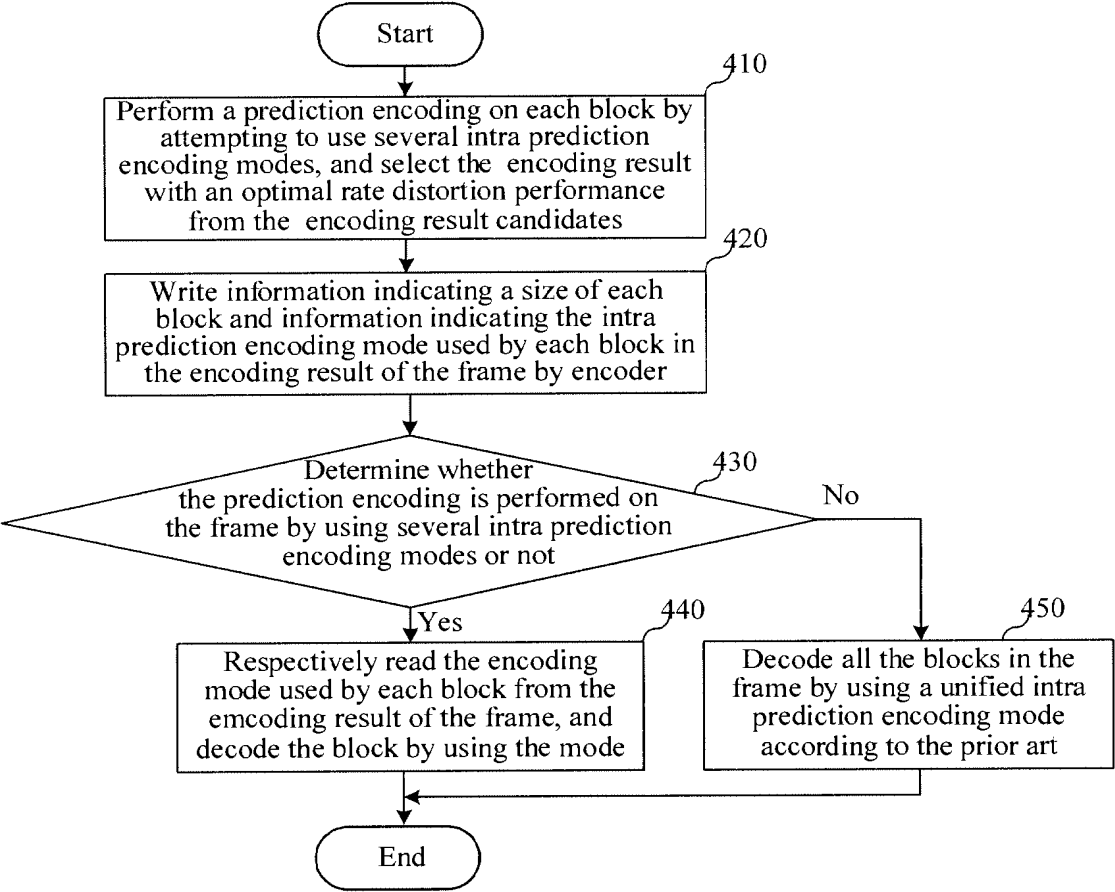


FIG. 4

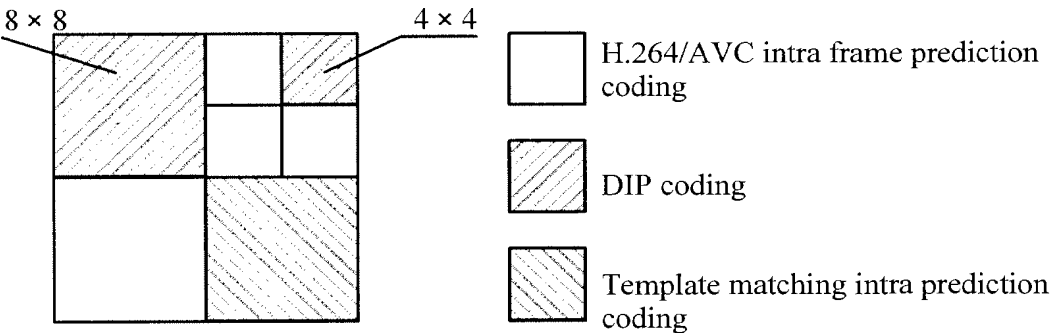


FIG. 5

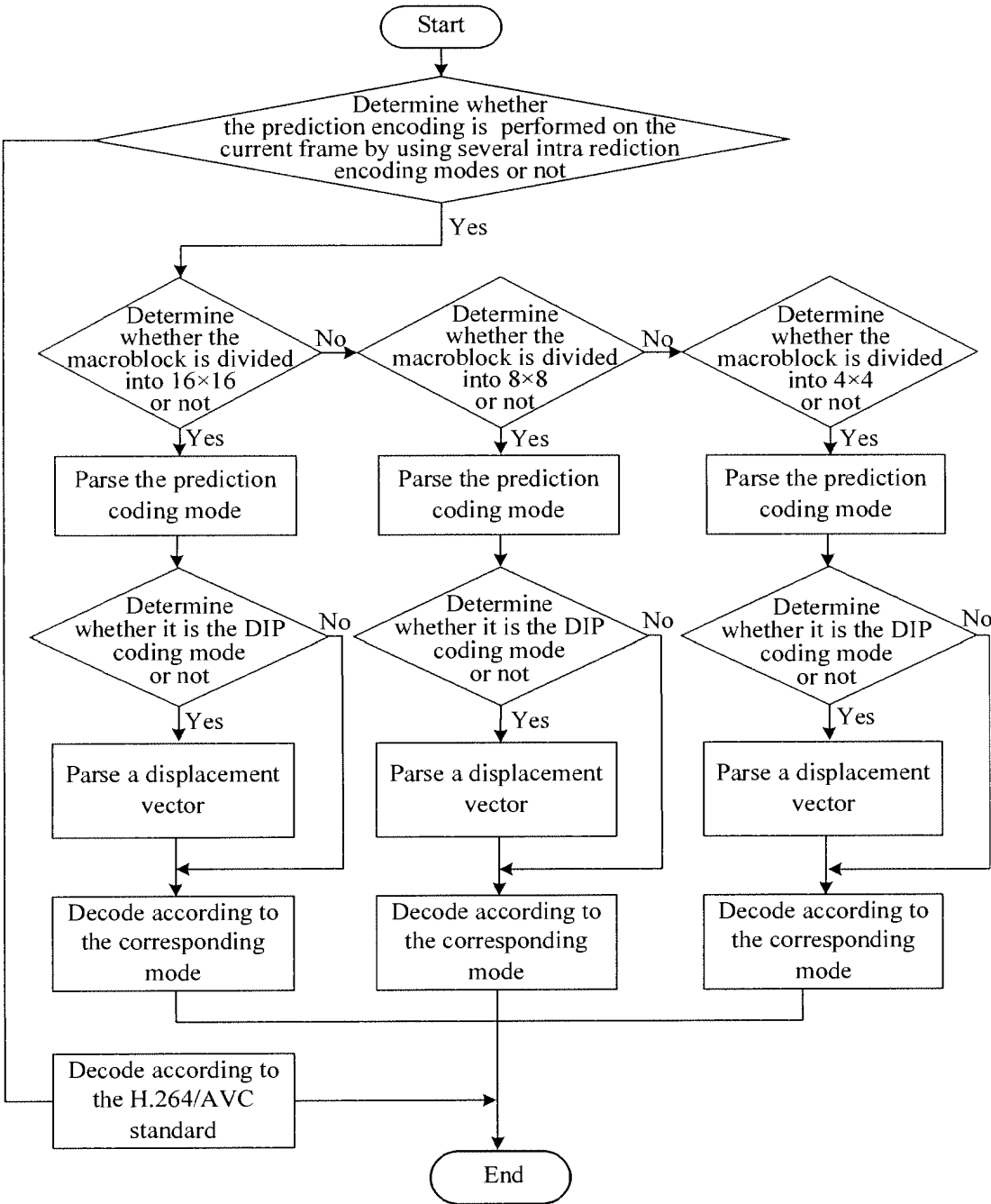


FIG. 6

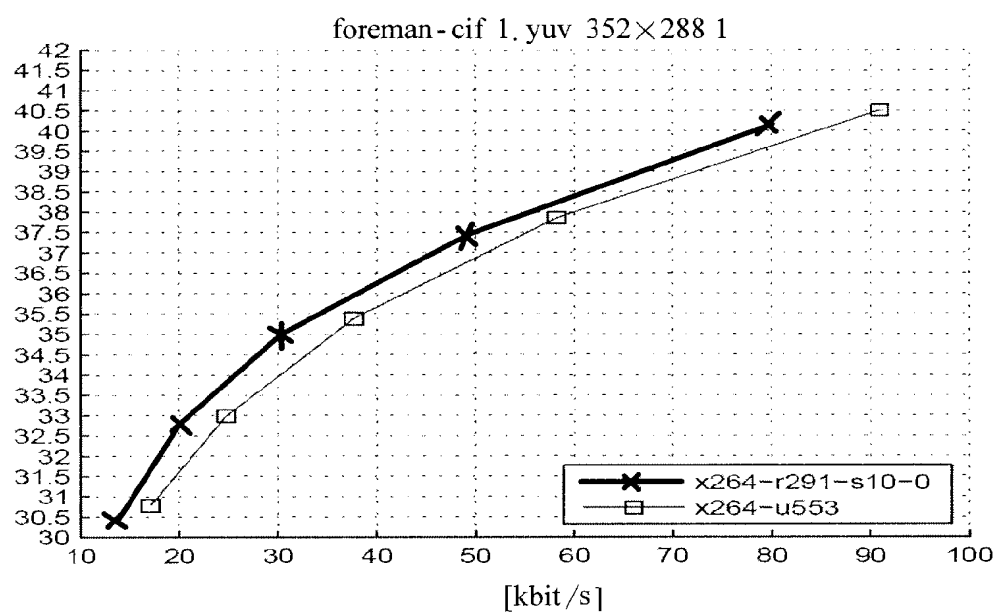


FIG. 7

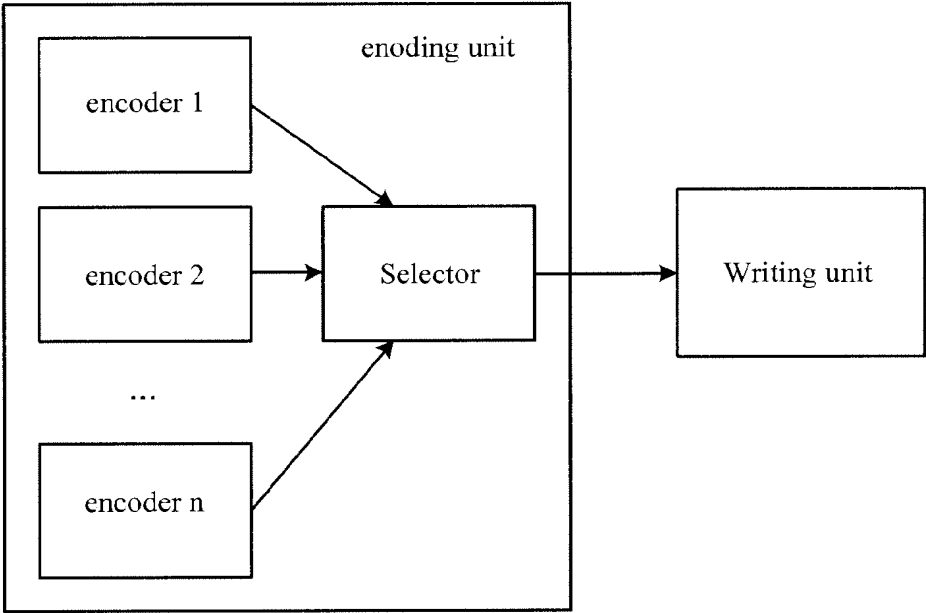


FIG. 8

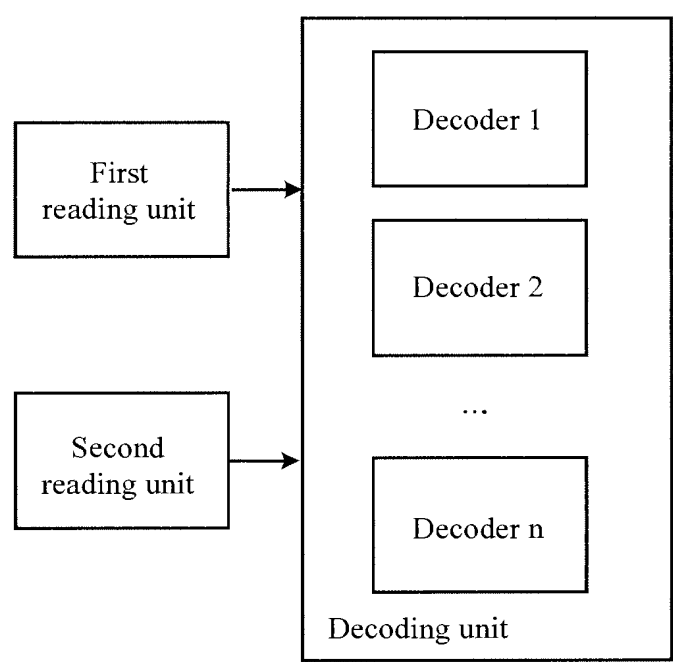


FIG. 9

INTERNATIONAL SEARCH REPORT		International application No. PCT/CN2008/070092
A. CLASSIFICATION OF SUBJECT MATTER		
H04N 7/32 (2006.01) i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
IPC: H04N 7/-		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
WPI; EPODOC; PAJ: FRAME, INTRA, PREDICTION, ENCODE, DECODE, VIDEO, MACRO BLOCK, PIXEL, ADAPT+, SELECT, H.264, AVC, HIBRID		
CNPAT:		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN1794816A (NEC CORP) 28 June 2006 (28.06.2006) description page 8 line 16-line 28, figure 3	1-7, 9-19
Y		8
Y	JP1080186A (FUJITSU LTD) 27 March 1989 (27.03.1989) description page 5 top right corner line 20 – page 7 top left corner line 2, figure 5	8
A	CN1476253A (UNIV QINGHUA) 18 February 2004 (18.02.2004) the whole document	1-19
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
<div>Special categories of cited documents:</div> <div><div><div>"A"</div>document defining the general state of the art which is not considered to be of particular relevance</div><div><div>"E"</div>earlier application or patent but published on or after the international filing date</div><div><div>"L"</div>document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)</div><div><div>"O"</div>document referring to an oral disclosure, use, exhibition or other means</div><div><div>"P"</div>document published prior to the international filing date but later than the priority date claimed</div></div> <div><div>"T"</div>later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</div> <div><div>"X"</div>document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</div> <div><div>"Y"</div>document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</div> <div><div>"&"</div>document member of the same patent family</div>		

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2008/070092

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN1794816A	28.06.2006	EP1675404A2	28.06.2006
		US2006133489A1	22.06.2006
		JP2006180195A	06.07.2006
		KR20060071894A	27.06.2006
JP1080186A	27.03.1989	JP2716703B2	18.02.1998
CN1476253A	18.02.2004	None	

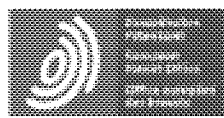
Form PCT/ISA/210 (patent family annex) (April 2007)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- **S.-L. Yu ; C. Chrysafis.** New Intra Prediction using Intra-Macroblock Motion Compensation. *JVT meeting Fairfax, doc JVT-C151*, May 2002 **[0037]**
- **T.K. Tan ; C.S. Boon ; Y Suzuki.** Intra Prediction by Template Matching. *ICIP*, 2006 **[0037]**



Espacenet

Bibliographic data: KR20090065401 (A) — 2009-06-22

METHOD OF INTER-FRAME/INTRA BLOCK PREDICTIVE CODING IN VIDEO CODING

Inventor(s): KIM MUN CHURL [KR]; PARK CHANG SEOB [KR]; HAHM SANG JIN [KR]; PARK KEUN SOO [KR] ± (KIM, MUN CHURL, ; PARK, CHANG SEOB, ; HAHM, SANG JIN, ; PARK, KEUN SOO)

Applicant(s): ICU RES & IND COOP GROUP [KR]; KOREAN BROADCASTING SYSTEM [KR] ± (ICU RESEARCH AND INDUSTRIAL COOPERATION GROUP, ; KOREAN BROADCASTING SYSTEM)

Classification: - **international:***H04N19/102; H04N19/51*
- **cooperative:** H04N19/119 (KR); H04N19/124 (KR); H04N19/13 (KR); H04N19/176 (KR); H04N19/51 (KR)

Application number: KR20080012337 20080211 Global Dossier

Priority number(s): KR20070132435 20071217

Also published as: KR100951301 (B1)

Abstract of KR20090065401 (A)

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).

	(19) 대한민국특허청(KR)	(11) 공개번호	10-2009-0065401
	(12) 공개특허공보(A)	(43) 공개일자	2009년06월22일

(51) Int. Cl.	(71) 출원인
HO4N 7/32 (2006.01) HO4N 7/24 (2006.01)	한국정보통신대학교 산학협력단
(21) 출원번호	대전 유성구 문지동 103-6
(22) 출원일자	한국방송공사
심사청구일자	서울 영등포구 여의도동 18번지
(30) 우선권주장	(72) 발명자
1020070132435 2007년12월17일 대한민국(KR)	김문철
	대전 서구 둔산동 한마루아파트 110동 802호
	박창섭
	서울 서초구 반포본동 1230 반포주공아파트 97동 211호
	(뒷면에 계속)
	(74) 대리인
	최태창

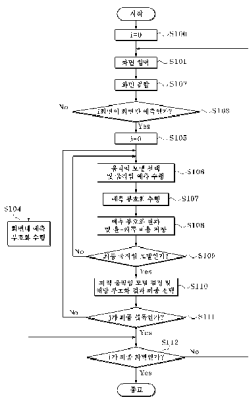
전체 청구항 수 : 총 18 항

(54) 비디오 부호화에서의 화면간/화면내 예측 부호화 방법

(57) 요약

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 움직인 모델 기반의 픽셀 와핑(warping)을 이용하여 예측 정확도를 크게 향상시킴으로써 발생 잔차 신호의 양을 줄여 부호화 효율을 더욱 개선할 수 있는 효과가 있다.

도 5



(72) 발명자
함상진
서울 영등포구 문래3동 벽산 메가트리움 103동
1803호

박근수
서울 양천구 목6동 한신아파트 113동 508호

특허청구의 범위

청구항 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.

청구항 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.

청구항 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.

청구항 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.

청구항 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.

청구항 6

제4 항에 있어서,
상기 움직임 모델은 파라미터로 모형화 된 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

청구항 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.

청구항 8

제4 항에 있어서,
상기 움직임 모델을 적용한 예측 부호화 과정은,
움직임 모델 파라미터를 추정하고, 상기 추정된 움직임 모델 파라미터를 이용하여 참조 화면으로부터 움직임 예

측을 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.

청구항 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 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 방법.

청구항 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 부호화 하고자 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.

청구항 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.

청구항 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.

청구항 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.

청구항 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.

청구항 15

비디오 부호화에서의 화면간 예측 부호화 방법에 있어서,
부호화 블록 단위를 16×16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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.

청구항 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 윌-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.

청구항 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.

청구항 18

제1 항 내지 제17 항 중 어느 한 항의 방법을 컴퓨터로 실행시킬 수 있는 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

명세서

발명의 상세한 설명

기술분야

<1> 본 발명은 비디오 부호화에서 슈퍼 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.

배경 기술

- <2> 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 많은 진전을 이루었다.
- <3> 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 윌-왜곡 비용의 관점에서 비용 최소화를 발생시키는 블록을 최적의 블록모드로 하여 부호화한다.
- <4> 이렇게 함으로써 미세한 움직임이나 복잡한 영상의 움직임 예측을 보다 효과적으로 수행하여 발생하는 잔차 신호를 크게 줄임으로써 압축 효율을 크게 향상할 수 있었다.
- <5> 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.
- <6> 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.
- <7> 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.
- <8> 결론적으로, 움직임 예측 부호화 시에 다양한 블록의 크기를 사용하여 움직임 예측 부호화를 수행하고, 이 중 하기의 수학적 식 1과 같이 윌-왜곡 비용을 가장 적게 발생시키는 블록이 최적의 부호화 최종 블록모드로 결정된다.

수학적 식 1

<9>
$$MODE = \underset{Mode}{\operatorname{argmin}} \{ D(c, r, Mode | QP) + \lambda_{MODE} \cdot R(c, r, Mode | QP) \}$$

- <10> 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 윌-왜곡 최적화를 수행하는 모든 블록 모드(또는 크기)를 의미한다.
- <11> 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.
- <12> 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 한계를 가지는 문제점이 있다.
- <13> 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.
- <14> 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 위치

를 고정시킨 채 줌 렌즈의 초점거리를 변화시켜 피사체에 접근하여 가는 것처럼 촬영, Zoom-out: 카메라의 위치를 고정시킨 채 줌 렌즈의 초점거리를 변화시켜 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.

<15> 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 수학적 식 2와 수학적 식 3은 투영된 공간에서의 x축과 y축으로의 움직임을 수식으로 표현한다.

수학적 식 2

$$u=-\frac{f}{Z}(T_x-xT_z)+\frac{xy}{f}\cdot R_x-f\left(1+\frac{x^2}{f^2}\right)\cdot R_y+y\cdot R_z+f\cdot \tan^{-1}\left(\frac{x}{f}\right)\left(1+\frac{x^2}{f^2}\right)\cdot R_{zoom}$$

<16>

수학적 식 3

$$v=-\frac{f}{Z}(T_y-yT_z)-\frac{xy}{f}\cdot R_y+f\left(1+\frac{y^2}{f^2}\right)\cdot R_x-x\cdot R_z+f\cdot \tan^{-1}\left(\frac{y}{f}\right)\left(1+\frac{y^2}{f^2}\right)\cdot R_{zoom}$$

<17>

<18> 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.

<19> 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.

발명의 내용

해결 하고자하는 과제

<20> 본 발명은 진술한 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.

<21> 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 수행함으로써 기대되는 잔차 신호(Residual Signals)의 발생량을 줄여 부호화 효율을 향상시키는 방법을 제공하는데 있다.

과제 해결수단

<22> 진술한 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.

<23> 여기서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들을 통한 잔차 신호에 대하여 변환 및 양자화를 수행한 후, 그 결과를 엔트로피 부호화하는 것이 바람직하다.

<24> 바람직하게, 상기 수퍼 매크로 블록의 크기는 16×16 매크로 블록의 크기보다 큰 가로 및 세로 길이가 2의 거듭

제공으로 정의되는 정사각형이며, 이를 분할한 하위 블록은 정사각형 및 직사각형을 포함하는 2의 거듭제곱으로 정의되는 블록으로 이루어질 수 있다.

<25> 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.

<26> 여기서, 상기 부호화 단위 블록은 매크로 블록 또는 16×16 매크로 블록보다 큰 크기의 수퍼 매크로 블록인 것이 바람직하다.

<27> 바람직하게, 상기 움직임 모델은 파라미터로 모형화 된 것이다.

<28> 바람직하게, 상기 움직임 모델은 병진 움직임 모델, 의사(擬似)(Affine) 움직임 모델, 이중선형(Bilinear) 움직임 모델 또는 투시(Perspective) 움직임 모델 중 적어도 하나를 포함할 수 있다.

<29> 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.

<30> 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.

<31> 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 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 방법을 제공하는 것이다.

<32> 상기 단계(d')에서, 상기 움직임 모델 최종 결정은 율-왜곡 비용을 고려하여 결정함이 바람직하다.

<33> 바람직하게, 상기 단계(a) 또는 단계(a') 이전에 부호화를 위한 화면 순서 및 블록 순서를 초기화하는 단계를 더 포함할 수 있다.

<34> 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.

<35> 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.

<36> 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.

<37> 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.

- <38>여기서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 윌-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화함이 바람직하다.
- <39>본 발명의 제7 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 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.
- <40>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 윌-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.
- <41>상기 단계(f")에서, 상기 움직임 모델을 적용하여 움직임 예측 수행 시 움직임 모델 파라미터를 추정하는 단계를 더 포함함이 바람직하다.
- <42>본 발명의 제9 측면은, 상술한 비디오 부호화에서의 화면간 예측 부호화 방법을 실행시키기 위한 프로그램을 기록한 기록매체를 제공한다.

효 과

- <43>이상에서 설명한 바와 같은 본 발명의 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 따르면, 부호화하고자 하는 블록의 크기를 기존의 16×16 매크로 블록 크기이상으로 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.
- <44>또한, 본 발명에 따르면, 다양한 움직임 모델을 사용하여 보다 정확한 움직임 예측을 수행함으로써 발생하는 잔차 신호의 양을 줄여 부호화 효율을 증대시킬 수 있을 뿐만 아니라 움직임 모델에 의한 참조 영상의 왜곡(Warping)을 통해 정화소로부터 1/2-화소, 1/4-화소, 1/8-화소의 서브-펠(Sub-Pel) 단계로 움직임 보상을 수행함으로써 소요되는 움직임 예측의 많은 연산량의 감소를 얻을 수 있는 이점이 있다.

발명의 실시를 위한 구체적인 내용

- <45>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.
- <46>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 윌-왜곡 비용을 최소화하는 블록을 최종 선택하여 예측 부호화

를 수행한다.

- <47> 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.
- <48> 도 4는 본 발명의 일 실시예에 적용된 16×16 매크로 블록 크기 이상의 슈퍼 매크로 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.
- <49> 즉, 32×32 슈퍼 매크로 블록 및 그 하위 분할 블록, 64×64 슈퍼 매크로 블록 및 그 하위 분할 블록, 128×128 슈퍼 매크로 블록 및 그 하위 분할 블록 등의 형태로 블록 크기를 확장할 수 있다.
- <50> 또한, 부호화기의 복잡도를 고려하여 블록 크기의 확장을 제한하거나 그 하위 블록의 블록 분할 타입의 수를 제한할 수 있다. 실제 부호화 시에 전술한 수학적 1에서와 같이 율-왜곡 비용을 최소화하는 블록 모드(Mode)를 선택하여 부호화할 수 있다.
- <51> 뿐만 아니라 크기가 증가된 슈퍼 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.
- <52> 또한, 화면간 예측 부호화에 부호화하고자 하는 블록에 대해 움직임 예측 모델을 선택적으로 적용하여 부호화 효율을 더욱 증가시킬 수 있다.
- <53> 이하에는 움직임 예측 모델에 대하여 구체적으로 설명하기로 한다.
- <54> 움직임 예측 모델을 보다 상세히 설명하기 위해 먼저, f_c 와 f_r 를 각각 부호화하고자 하는 현재 영상과 참조 영상으로 한다. (x,y) 를 f_c 의 부호화하고자 하는 현재 블록의 한 픽셀 좌표값을 나타낸다. 또한, 부호화하고자 하는 현재 프레임의 예측을 $\hat{f}_c$ 라고 하면 참조 영상으로부터 하기의 수학적 4와 같이 정의할 수 있다.

수학적 4

- <55>
$$\hat{f}_c(x,y)=f_r(u,v)$$
- <56> 여기서, (u,v) 는 참조 영상의 좌표값을 나타낸다. 상기의 수학적 4는 움직임 보상이 참조 영상의 좌표값을 현재 영상의 좌표값에 대응시키는 기하학적 변환(Geometric Transformation)을 의미하며 하기의 수학적 5와 같이 일 반화하여 표현할 수 있다.

수학적 5

- <57>
$$u=g_x(u,v), \quad v=g_y(u,v)$$
- <58> 여기서, g_x 와 g_y 는 공간 변환 함수를 나타낸다. 이러한 공간 변환을 텍스처 매핑(Texture Mapping) 또는 이미지 매핑(Image Mapping)으로 불린다.
- <59> 움직임 예측에 일반적으로 사용되는 블록 정합 방법인 2차원 병진 움직임 모형은 하기의 수학적 6과 같이 표현 된다.

수학식 6

$$\begin{aligned} u &= g_x(u, v) = x + a_1 = x + d_x \\ v &= g_y(u, v) = y + a_2 = y + d_y \end{aligned}$$

<60>

<61> 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.

<62> 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.

수학식 7

$$\begin{aligned} u &= g_x(u, v) = a_1x + a_2y + a_3 \\ v &= g_y(u, v) = a_4x + a_5y + a_6 \end{aligned}$$

<63>

수학식 8

$$\begin{aligned} u &= g_x(u, v) = a_1xy + a_2x + a_3y + a_4 \\ v &= g_y(u, v) = a_5xy + a_6x + a_7y + a_8 \end{aligned}$$

<64>

수학식 9

$$\begin{aligned} u &= g_x(u, v) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} \\ v &= g_y(u, v) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} \end{aligned}$$

<65>

<66> 여기서, a_i 는 움직임 모델 파라미터를 나타낸다. 움직임 모델 기반의 예측 부호화를 수행할 경우, 상기 모델들 중 가장 유효-왜곡 비용을 최소화하는 모델을 적용하여 부호화함으로써 부호화 효율을 증가시킬 수 있다.

<67> 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.

<68> 한편, 본 발명의 일 실시예에서는 비디오 부호화에서의 화면간 예측 부호화 방법 위주로 설명하였지만, 이에 국한하지 않으며, 비디오 부호화에서의 화면내 예측 부호화에 적용시킬 수도 있다.

<69> 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.

<70> 이때, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 유효-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화

함이 바람직하다.

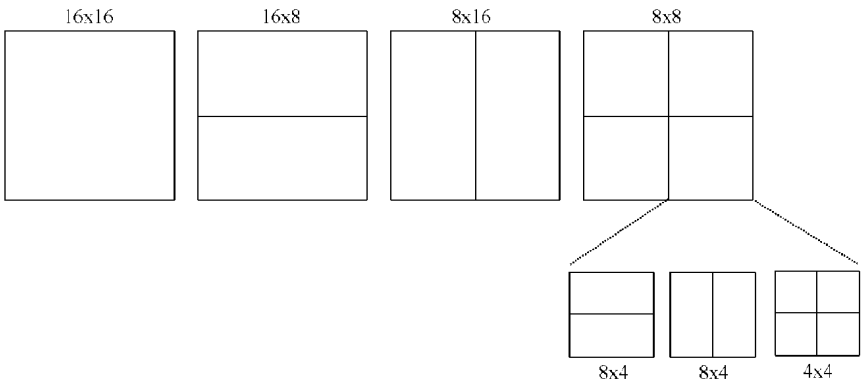
- <71>도 5는 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간 예측 부호화 방법을 설명하기 위한 전체적인 흐름도로서, 비디오 부호화에서 선택적 움직임 모델 기반의 화면간 예측 부호화 방법을 나타낸 것이다.
- <72>도 5를 참조하면, 먼저, 부호화를 위한 화면(i) 순서를 초기화(i=0)한 후(S100), 부호화를 위해 순서에 맞게 화면(i)을 입력한다(S101).
- <73>그런 다음, 상기 단계S101에서 입력된 화면(i)을 부호화 단위 블록(매크로 블록 또는 수퍼 매크로 블록)으로 분할한 후(S102), 현재의 화면(i)이 화면간 예측부호화를 수행할 것인지를 판단한다(S103).
- <74>단약, 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).
- <75>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 윌-왜곡 비용값을 저장한다(S108).
- <76>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.
- <77>그렇지 않고, 상기 단계S109에서의 판단 결과, 사용 움직임 모델이 최종 움직임 모델이면 최적의 움직임 모델을 결정하고 해당 부호화 결과를 최종 선택한다(S110).
- <78>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.
- <79>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.
- <80>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.
- <81>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.
- <82>또한, 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터 통신망으로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 읽을 수 있는 코드로서 저장되고 실행될 수 있다.
- <83>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.

도면의 간단한 설명

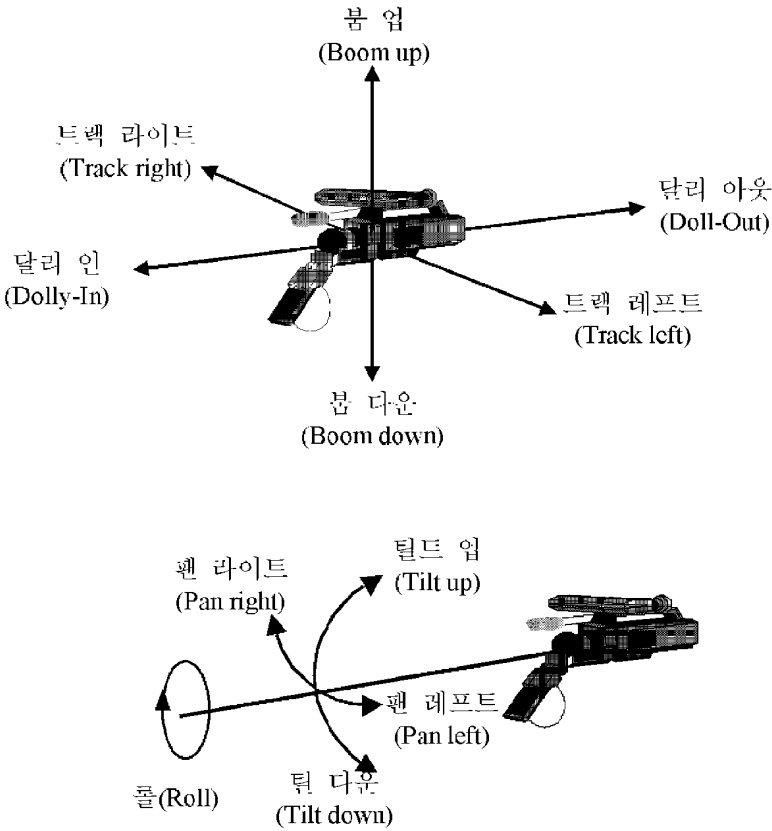
- <84>도 1은 종래 기술의 H.264|AVC 부호화기에서 화면간 예측 부호화를 위한 16×16 매크로 블록 단위 블록의 분할 블록 타입을 나타낸 도면.
- <85>도 2는 비디오 영상에서의 카메라의 움직임에 의한 움직임 생성 타입을 설명하기 위한 도면.
- <86>도 3은 2차원 영상 공간에 투영된 움직임 모델을 설명하기 위한 개념도.
- <87>도 4는 본 발명의 일 실시예에 적용된 16×16 매크로 블록 크기 이상의 수퍼 매트릭스 블록 및 블록 분할 타입의 일 예를 나타낸 도면.
- <88>도 5는 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간 예측 부호화 방법을 설명하기 위한 전체적인 흐름도.

도면

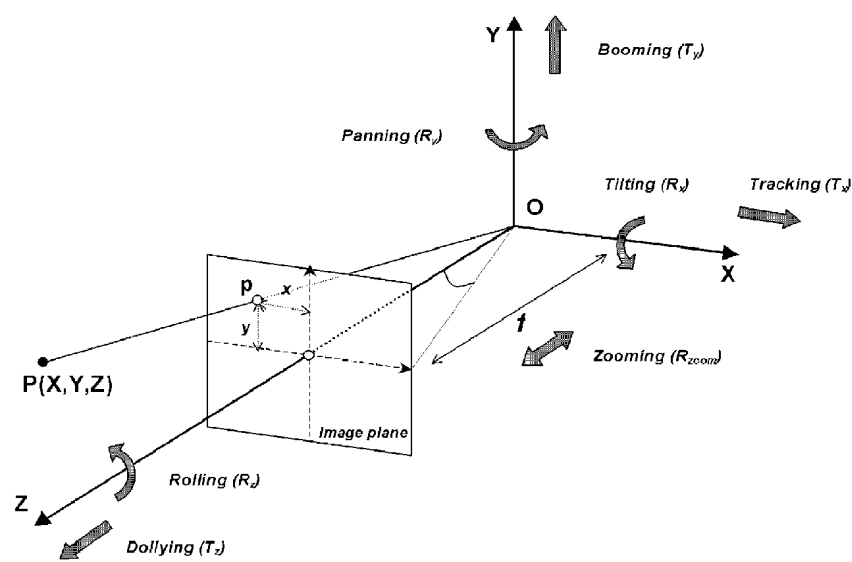
도면1



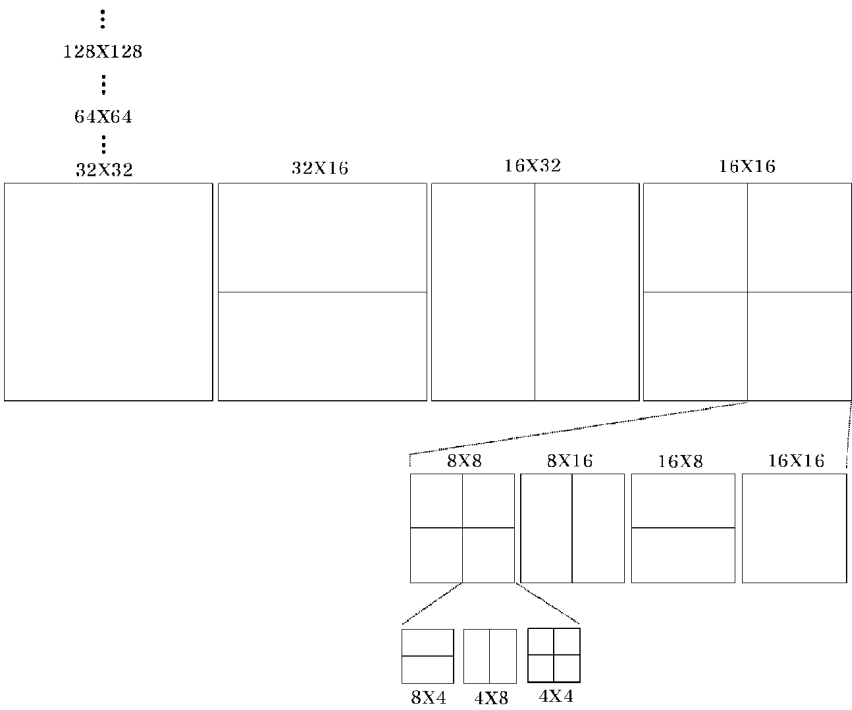
도면2



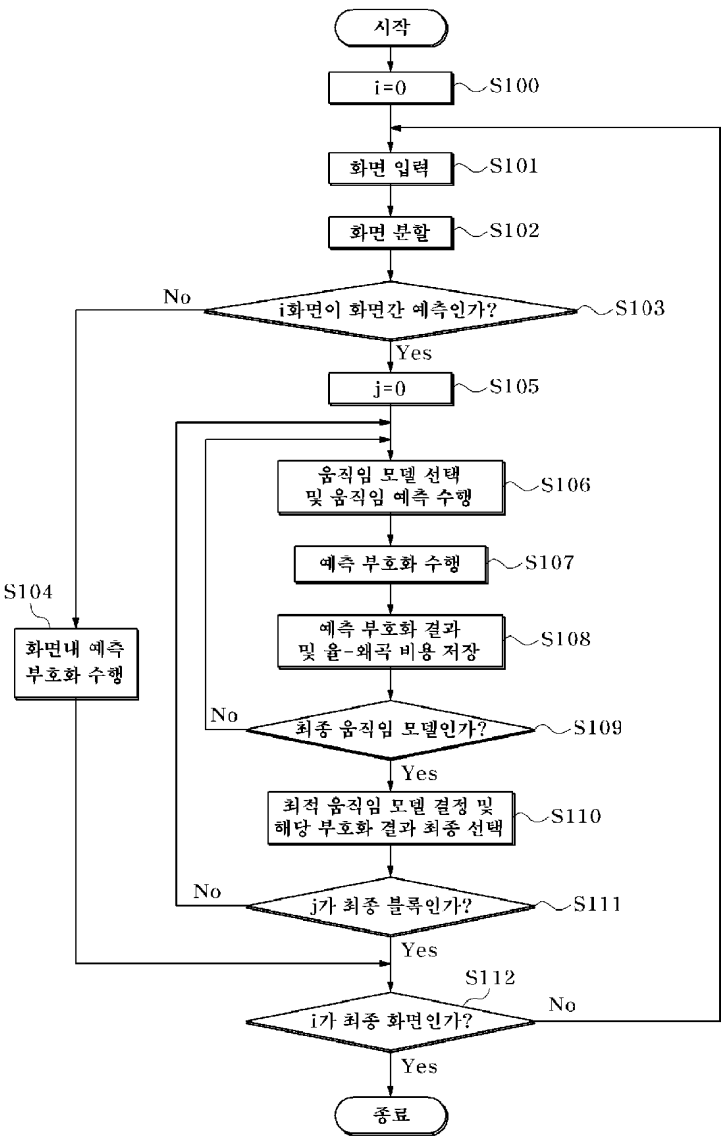
도면3



도면4



도면5

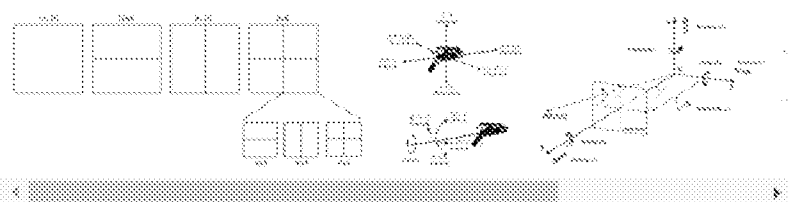


Method of inter-frame/intra block predictive coding in video coding

Abstract

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 blocks 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).

Images (5)



Classifications

H04N19/51 Motion estimation or motion compensation

View 4 more classifications

KR20090065401A
South Korea

Download PDFFind Prior ArtSimilar

Other languages: Korean

Inventor: 김문철, 박정선, 함상진, 박근수

Worldwide applications

2008 KR

Application KR1020080012337A events ⓘ

Priority claimed from KR1020070132438

2008-02-11 Application filed by 한국정보통신대학교 산학협약단, 한국형송공사

2009-06-22 Publication of KR20090065401A

2010-04-02 Application granted

2010-04-02 Publication of KR100951301B1

Info: Patent citations (3), Cited by (9), Legal events, Similar documents, Priority and Related Applications

External links: Espacenet, Global Dossier, Discuss

Claims (18)

Hide Dependent >
translated from Korean

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. 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. 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. 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. 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. The method of claim 4, wherein And the motion model is modeled as a parameter. 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. 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. 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. 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. 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. The method of claim 9 or 10, And initiating a picture order and a block order for encoding before step (a) or step (a'). 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. 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. 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. 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: The method of claim 16, And estimating a motion model parameter when performing motion prediction by applying the motion model in step (f "). A computer-readable recording medium having recorded thereon a program capable of executing the method of any one of claims 1 to 17.

Description translated from Korean

Inter-screen / In-picture predictive coding method in video coding {METHOD OF INTER-FRAME / INTRA BLOCK PREDICTIVE CODING IN VIDEO CODING}

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.

Generally, ISO / IEC 14496-10 (MPEG-4 Advanced Video Coding) or H.264 and ISO / IEC 14496-10 Amendment 3 (MPEG-4), 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.

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.

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.

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.

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.

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.

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.

$$MODE = \arg \min_{Mode} \{ D(c, r, Mode | QP) + \lambda_{MODE} \cdot R(c, r, Mode | QP) \}$$

$$MODE = \arg \min_{Mode} \{ D(c, r, Mode | QP) + \lambda_{MODE} \cdot R(c, r, Mode | QP) \}$$

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.

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.

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.

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.

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.

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.

$$u = -\frac{f}{Z}(T_x - xT_z) + \frac{xy}{f} \cdot R_x - f \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_y + y \cdot R_z + f \cdot \tan^{-1}\left(\frac{x}{f}\right) \left(1 + \frac{x^2}{f^2}\right) \cdot R_{zoom}$$

$$u = -\frac{f}{Z}(T_x - xT_z) + \frac{xy}{f} \cdot R_x - f \cdot \left(1 + \frac{x^2}{f^2}\right) \cdot R_y + y \cdot R_z + f \cdot \tan^{-1}\left(\frac{x}{f}\right) \left(1 + \frac{x^2}{f^2}\right) \cdot R_{zoom}$$

$$v = -\frac{f}{Z}(T_y - yT_z) - \frac{xy}{f} \cdot R_y + f \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_x - x \cdot R_z + f \cdot \tan^{-1}\left(\frac{y}{f}\right) \left(1 + \frac{y^2}{f^2}\right) \cdot R_{zoom}$$

$$v = -\frac{f}{Z}(T_y - yT_z) - \frac{xy}{f} \cdot R_y + f \cdot \left(1 + \frac{y^2}{f^2}\right) \cdot R_x - x \cdot R_z + f \cdot \tan^{-1}\left(\frac{y}{f}\right) \left(1 + \frac{y^2}{f^2}\right) \cdot R_{zoom}$$

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.

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.

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.

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.

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 spitting 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.

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.

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.

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.

Herein, the coding unit block is preferably a macroblock or a super macroblock having a size larger than that of a 16 × 16 macroblock.

Preferably, the motion model is modeled with parameters.

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.

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.

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.

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.

In the step (d '), the final determination of the motion model is preferably made in consideration of the rate-distortion cost.

Preferably, the method may further include initializing a screen order and a block order for encoding before step (a) or step (a ').

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.

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.

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.

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.

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.

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.

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 i "; (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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Hereinafter, the motion prediction model will be described in detail.

라고 하면 참조 영상으로부터 하기의 수학적 식과 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

f_c

in this case, it can be defined as shown in Equation 4 below from the reference image.

$$\hat{f_c}(x,y)=f_r(u,v)$$
$$\hat{f_c}(x,y)=f_r(u,v)$$

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.

$$u = g_x(u,v), \quad v = g_y(u,v)$$
$$u = g_x(u,v), \quad v = g_y(u,v)$$

Where g_x and g_y represent spatial transform functions. This spatial transformation is called texture mapping or image mapping.

A 2D translational motion model, which is a block matching method generally used for motion prediction, is expressed by Equation 6 below.

$$u=g_x(u,v)=x+a_1=x+d_x$$
$$v=g_y(u,v)=y+a_2=y+d_y$$
$$u=g_x(u,v)=x+a_1=x+d_x$$
$$v=g_y(u,v)=y+a_2=y+d_y$$

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.

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.

$$u = g_x(u,v) = a_1x + a_2y + a_3$$
$$v = g_y(u,v) = a_4x + a_5y + a_6$$
$$u = g_x(u,v) = a_1x + a_2y + a_3$$
$$v = g_y(u,v) = a_4x + a_5y + a_6$$

$$u=g_x(u,v)=a_1xy+a_2x+a_3y+a_4$$
$$v=g_y(u,v)=a_5xy+a_6x+a_7y+a_8$$
$$u=g_x(u,v)=a_1xy+a_2x+a_3y+a_4$$
$$v=g_y(u,v)=a_5xy+a_6x+a_7y+a_8$$

$$u = g_x(u,v) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1}$$
$$v = g_y(u,v) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1}$$

$$u = g_x(u, v) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1}$$

$$v = g_y(u, v) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1}$$

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.

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.

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.

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.

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.

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.

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).

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).

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).

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).

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.

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).

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.

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.

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.

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).

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.

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.

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.

FIG. 2 is a diagram for describing a motion generation type due to camera movement in a video image. FIG.

3 is a conceptual diagram illustrating a motion model projected on a two-dimensional image space.

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.

5 is an overall flowchart illustrating a method of inter prediction encoding in video encoding according to an embodiment of the present invention.

Patent Citations (3)

..

Publication number	Priority date	Publication date	Assignee	Title
Family To Family Citations				
KR100775787B1 *	2005-08-03	2007-11-13	경희대학교 산학협력단	A hybrid scalable encoding Apparatus and method of moving pictures using spatial-temporal specific each area
KR101498044B1 *	2007-01-10	2015-03-05	삼성전자주식회사	Method for generating small image of compressed image adaptively encoded macro block and apparatus thereof
KR101366242B1 *	2007-03-29	2014-02-20	삼성전자주식회사	Method for encoding and decoding motion model parameter, and method and apparatus for video encoding and decoding using motion model parameter

* Cited by examiner, † Cited by third party

Cited By (9)

Publication number	Priority date	Publication date	Assignee	Title
KR20110126567A *	2011-08-31	2011-11-23	한국과학기술원	Method for encoding partitioned block in video encoding, method for decoding partitioned block in video decoding and recording medium implementing the same
US20110292999A1 *	2009-06-30	2011-12-01	Electronics And Telecommunications Research Institute	Super macro block based intra coding method and apparatus
CN102484709A *	2009-07-01	2012-05-30	汤姆泰特许可公司	Methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders
CN102907100A *	2010-05-17	2013-01-30	Sk电信有限公司	Image coding/decoding device using coding block in which intra block and inter block are mixed, and method thereof
CN103039073A *	2010-06-07	2013-04-10	数码士有限公司	Method for encoding/decoding high-resolution image and device for performing same
CN103141105A *	2010-09-27	2013-06-05	Lg电子株式会社	Method for partitioning block and decoding device
US9485512B2	2009-02-23	2016-11-01	Korea Advanced Institute Of Science And Technology	Video encoding method for encoding division block, video decoding method for decoding division block, and recording medium for implementing the same
US10419752B2	2009-12-16	2019-09-17	Electronics And Telecommunications Research Institute	Adaptive image encoding device and method
Family To Family Citations				
KR101395838B1	2013-01-24	2014-05-19	전자부품연구원	Video stabilization method and system by selection one of various motion models

* Cited by examiner, † Cited by third party, ‡ Family to family citation

Similar Documents

Publication	Publication Date	Title
KR100951301B1	2010-04-02	Method of inter-frame/intra block predictive coding in video coding
JP4509104B2	2010-07-21	Fast mode decision algorithm for intra prediction in advanced video coding
JP5234586B2	2013-07-10	Video encoding method and decoding method, apparatus thereof, program thereof, and storage medium storing program
KR100688383B1	2007-03-02	Motion estimation and compensation for panorama image
JP4752631B2	2011-08-17	Image coding apparatus and image coding method
JP6005087B2	2016-10-12	Image decoding apparatus, image decoding method, image encoding apparatus, image encoding method, and data structure of encoded data
KR101821867B1	2018-01-24	Image decoding apparatus, image decoding method, image encoding apparatus, image encoding method and recording medium
JP5234587B2	2013-07-10	Video encoding method and decoding method, apparatus thereof, program thereof, and storage medium storing program
JP2010016454A	2010-01-21	Image encoding apparatus and method, image decoding apparatus and method, and program
WO2010082463A1	2010-07-22	Image encoding apparatus and image encoding method, image decoding apparatus and image decoding method
KR100924048B1	2009-10-27	A fast mode decision method of intra block prediction in video coding

KR101396754B1	2014-05-28	Method and apparatus for compressing video using template matching and motion prediction
KR100790757B1	2008-01-03	Adaptive fast motion estimation method for h.264 video encoder
JP2007184846A	2007-07-19	Moving image coding apparatus

Priority And Related Applications

Applications Claiming Priority (2)

Application	Filing date	Title
KR1020070132435	2007-12-17	
KR20070132435	2007-12-17	

Legal Events

Date	Code	Title	Description
2008-02-11	A201	Request for examination	
2009-07-07	N231	Notification of change of applicant	
2009-09-24	E902	Notification of reason for refusal	
2010-03-22	E701	Decision to grant or registration of patent right	
2010-03-29	GRNT	Written decision to grant	
2013-03-04	FPAY	Annual fee payment	Payment date: 20130304 Year of fee payment: 4
2014-03-03	FPAY	Annual fee payment	Payment date: 20140303 Year of fee payment: 5
2015-07-16	FPAY	Annual fee payment	Payment date: 20150716 Year of fee payment: 6
2016-02-25	FPAY	Annual fee payment	Payment date: 20160225 Year of fee payment: 7
2017-02-24	FPAY	Annual fee payment	Payment date: 20170224 Year of fee payment: 8
2018-02-26	FPAY	Annual fee payment	Payment date: 20180226 Year of fee payment: 9
2019-03-04	FPAY	Annual fee payment	Payment date: 20190304 Year of fee payment: 10
2020-03-02	FPAY	Annual fee payment	Payment date: 20200302 Year of fee payment: 11

Concepts

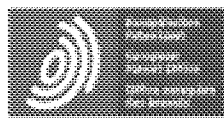
machine-extracted

[Download](#)
[Filter table](#)

Name	Image	Sections	Count	Query match
method		claims,description	7	0.000
transforming		claims,description	4	0.000
corresponding		claims,description	3	0.000

⌵ favourable	claims	1	0.000
⌵ initiatory	claims	1	0.000
⌵ decreasing	abstract	1	0.000
Show all concepts from the description section			

Data provided by IFI CLAIMS Patent Services



Espacenet

Bibliographic data: WO2008083633 (A1) — 2008-07-17

A METHOD AND A DEVICE FOR INTRA FRAME PREDICTION
ENCODING/DECODING

Inventor(s): XIONG LIANHUAN [CN] ± (XIONG, LIANHUAN)

Applicant(s): HUAWEI TECH CO LTD [CN]; XIONG LIANHUAN [CN] ± (HUAWEI TECHNOLOGIES CO., LTD, ; XIONG, LIANHUAN)

Classification: - international: H04N7/32
- cooperative: H04N19/11 (EP, KR, US); H04N19/147 (EP, US); H04N19/176 (EP, US); H04N19/51 (KR); H04N19/61 (EP, US)

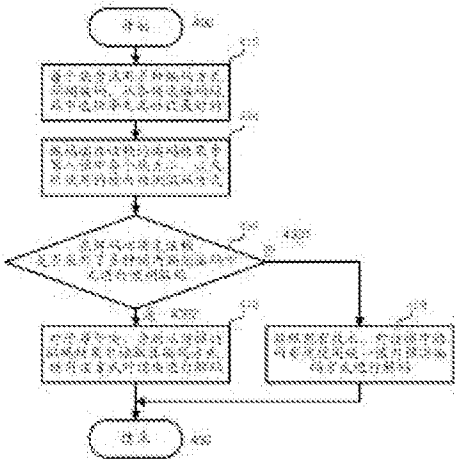
Application number: WO2008CN70092 20080111 Global Dossier

Priority number(s): CN20071007801 20070111

Also published as: CN101222641 (A) CN101222641 (B) EP2081388 (A1) EP2081388 (A4) JP2010515310 (A) KR101048735 (B1) KR20090088863 (A) US2009196350 (A1) less

Abstract of WO2008083633 (A1)

A method and a device for intra frame prediction encoding /decoding, which can optimize the encoding result of the intra frame encoding. Prediction encoding by at least two kinds of intra frame prediction methods in a same frame, and writing the intra frame prediction encoding method information of using to encoded result in the frame, so encoding by the most appropriate method for the different blocks. In the case of using at least two kinds of intra frame prediction methods, needing to write in correlational indicative information, in order to decoding in accurately. When decoding, reading intra frame prediction encoding method of blocks for every block in the same frame, and decoding the blocks by using the method. By organizing every blocks in hierarchy tree form, it could acquiring the optimization encoding capability while reducing the amount of information of information of intra frame prediction method of the every blocks in the encoding result.



400 START
410 CARRY BLOCKS ATTEMPTING TO PREDICTION ENCODING BY USING BLOCKS ACCORDING
420 SET ESTIMATES BY METHOD OF DETERMINING RATE FRAMES, TERMINATE
ENCODING METHOD
430 IN ENCODING POINT, WITH THE EVERY BLOCKS 430N OF THE FRAME AND INTRA-FRAME
PREDICTION (BLOCKS) METHOD IN ENCODING RESULT OF THE FRAME
430Y IN DECISION POINT, DETERMINING PREDICTION ENCODING WITH THE ADAPTIVE RESULT
INTRA-FRAME PREDICTION ENCODING METHOD 430Y OR NOT
440 NO
450 END
460 YES
460 FOR EVERY BLOCK, PREDICTION ENCODING METHOD RESPECTUALLY FROM ENCODING
RESULT OF THE FRAME, DECIDING THE BLOCK BY THE METHOD
450 IN THE END, DECIDING ALL BLOCKS OF THE FRAME BY USING INTRA-FRAME
PREDICTION ENCODING METHOD
450 END

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2008 年 7 月 17 日 (17.07.2008)



PCT



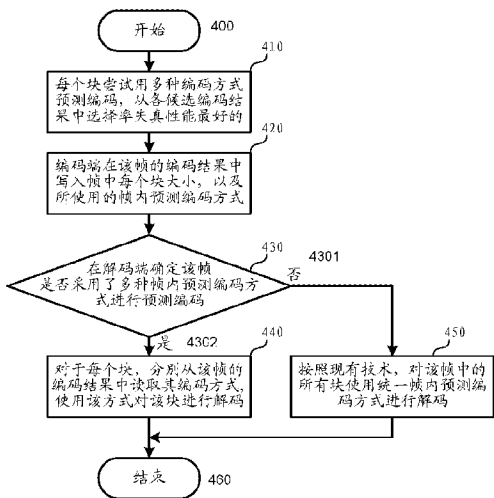
(10) 国际公布号
WO 2008/083633 A1

- (51) 国际专利分类号: [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN).
H04N 7/32 (2006.01)
- (21) 国际申请号: PCT/CN2008/070092 (72) 发明人; 及
(22) 国际申请日: 2008 年 1 月 11 日 (11.01.2008) (75) 发明人/申请人 (仅对美国): 熊联欢(XIONG, Lian-huan) [CN/CN]; 中国广东省深圳市龙岗区坂田华为基地总部办公楼, Guangdong 518129 (CN).
- (25) 申请语言: 中文 (74) 代理人: 北京同立钧成知识产权代理有限公司 (LEADER PATENT & TRADEMARK FIRM); 中国北京市朝阳区北辰西路 69 号峻峰华亭 A 座 902, Beijing 100029 (CN).
- (26) 公布语言: 中文
- (30) 优先权: 200710007801.0
2007 年 1 月 11 日 (11.01.2007) CN (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,
- (71) 申请人 (对除美国外的所有指定国): 华为技术有限公司(HUAWEI TECHNOLOGIES CO., LTD.)

[见续页]

(54) Title: A METHOD AND A DEVICE FOR INTRA FRAME PREDICTION ENCODING/DECODING

(54) 发明名称: 帧内预测编解码方法及其装置



400 START
410 EVERY BLOCKS ATTEMPTING TO PREDICTION ENCODING BY USING MULTI ENCODING METHODS, SELECTING THE BEST METHOD OF DISTORTION RATE FROM ALTERNATE ENCODING RESULTS
420 IN ENCODING PORT, WRITING EVERY BLOCKS SIZE OF THE FRAME AND INTRA FRAME PREDICTION ENCODING METHOD IN ENCODING RESULT OF THE FRAME
430 IN DECODING PORT, DETERMINING PREDICTION ENCODING WHETHER ADOPTING MULTI INTRA FRAME PREDICTION ENCODING METHODS OR NOT
4301 NO
4302 YES
440 FOR EVERY BLOCK, READING ENCODING METHOD RESPECTIVELY FROM ENCODING RESULT OF THE FRAME, DECODING THE BLOCK BY THE METHOD
450 IN THE ART, DECODING ALL BLOCKS OF THE FRAME BY USING UNIFORM INTRA FRAME PREDICTION ENCODING METHOD
460 END

(57) Abstract: A method and a device for intra frame prediction encoding /decoding, which can optimize the encoding result of the intra frame encoding. Prediction encoding by at least two kinds of intra frame prediction methods in a same frame, and writing the intra frame prediction encoding method information of using to encoded result in the frame, so encoding by the most appropriate method for the different blocks. In the case of using at least two kinds of intra frame prediction methods, needing to write in correlational indicative information, in order to decoding in accurately. When decoding, reading intra frame prediction encoding method of blocks for every block in the same frame, and decoding the blocks by using the method. By organizing every blocks in hierarchy tree form, it could acquiring the optimization encoding capability while reducing the amount of information of information of intra frame prediction method of the every blocks in the encoding result.

[见续页]

WO 2008/083633 A1



GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧洲 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧洲 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。
本国际公布:
— 包括国际检索报告。

(57) 摘要:

本发明涉及视频编解码领域,公开了一种帧内预测编解码方法及其装置,使得帧内编码的编码结果性能更优。本发明中,对同一个帧使用至少两种帧内预测编码方式进行预测编码,并将每个块所对应的帧内预测编码方式信息写入编码结果中,从而可以对不同的块使用最适合的方式进行编码。在使用了至少两种帧内预测编码方式的情况下,需要向编码结果中写入相关指示信息,以便可以正确地进行解码。解码时,对同一帧内每个块先读取该块所用的帧内预测编码方式再使用该方式对该块进行解码。以分级树的形式对各个块进行组织,可以在获得最优编码性能的同时减少编码结果中表示每个块帧内预测编码方式的信息的信息量。

帧内预测编解码方法及其装置

技术领域

本发明涉及视频编解码领域，特别涉及帧内预测编解码方法及装置。

背景技术

由于视频的信息量较大，在视频编码中，如果编码端在进行帧内编码时不依赖于本帧前面已编码帧的信息，独立编码，则解码端可以在不知道前面帧的解码信息的情况下直接对本帧进行解码。然而采用该方法，帧内编码压缩率很低，编码信息量极大。为了提高编码的压缩率，在进行帧内编码时，可以使用周围的信息来对编码块进行预测；或者，对帧间预测的帧，尤其是运动补偿的帧，也可以使用帧间编码块作为当前编码帧的预测参考进行帧内编码。如果一帧仅仅使用了帧内编码，那它可以作为一个视频序列的起始点，例如：标准 H.264/先进视频编码（Advanced Video Coding，以下简称：AVC）中的即时解码刷新图像。

下面对 H.264/AVC 的帧内预测编码技术进行简单介绍。H.264/AVC 帧内编码的主要目的是缩减图像的空间冗余。为了提高 H.264/AVC 帧内编码的效率，在给定帧中充分利用相邻宏块的空间相关性，相邻的宏块通常含有相似的属性。因此，在对一给定宏块编码时，首先可以根据周围的宏块进行预测，然后对预测值与实际值的差值进行编码，如图 1 所示，为现有技术中 H.264/AVC 帧内预测编码方法示意图，由图 1 可知，该方法相对于直接对该帧编码而言，可以大大减小码率。

H.264/AVC 支持 4×4、8×8、和 16×16 像素宏块的帧内预测，提供了包括 1 种直流预测模式和各种方向预测模式在内的多种预测模式。通过帧内预测，大大缩减了图像的空间冗余，提高了编码的压缩率。

除了 H.264/AVC 的帧内预测编码方法，现有技术还提出了一种针对

H.264/AVC 的移位帧内预测 (displacement intra prediction) 编码方法。该方法将 H.264/AVC 中块大小可变的帧间预测思想移植到 H.264/AVC 的帧内预测上。条带内当前块可以通过移位矢量标记的方法来参考先前的编码区域。也就是说, 在对当前块进行编码时, 可以先在已编码的区域中寻找与当前块相似的块作为候选块, 通过位移矢量标记的方法指示该候选块, 标识当前块与该候选块之间的区别, 由于该候选块在当前块之前被解码, 从而解码端可以根据该候选块的解码信息以及位移矢量等信息获得当前块, 如图 2 所示, 为现有技术中移位帧内预测编码方法示意图。图 2 所示的方法通常应用于宏块级。

现有技术还提出一种使用模板匹配的帧内预测编码方法。如图 3 所示, 为现有技术中模板匹配帧内预测编码方法示意图, 在该方法中, 将预测图像区域中自相似的区域通过模板匹配来进行帧内预测, 其中的预测图像区域也称为对象区域,。具体地说, 即将预测图像区域的相邻区域作为模板, 在已编码的一个区域中迭代寻找与该模板的像素最相似的区域, 将该相似区域称为候选邻区, 其中已编码的一个区域也称为搜索区域。在解码端, 可以从搜索区域找到该候选邻区, 由于该候选邻区的象素类似与对象区域的相邻区域, 因此不需要其他附加的头信息, 解码端就可以得到该对象区域的预测。该模板匹配方式可作为 H.264/AVC 的一个附加模式, 适用于帧内 4x4 块或帧内 8x8 的预测块。

在实现本发明的过程中, 发明人发现现有技术至少存在以下问题:

无论 H.264/AVC 帧内预测编码、移位帧内预测编码、还是模板匹配帧内预测编码, 它们均有各自的特点, 各自适用于部分场景, 不同的帧内预测技术适用于一副编码图象不同的局部区域。H.264/AVC 中的帧内预测技术适用于当编码块具有和其边缘像素相同方向的伸展结构的纹理图象区域; 移位帧内预测技术适用于当图象内具有重复的结构区域; 模版匹配技术适用于所参考的临近区域内具有相似结构时的模式预测。可见只使用一种帧内预测编

码方法得到的编码结果并不是最优的，但当前并没有一种技术能够结合这三种方法的特点，灵活组合编码。

发明内容

本发明实施例所要解决的技术问题是：提供一种帧内预测编解码方法及其装置，使得帧内编码的编码结果性能更优。

为解决上述技术问题，本发明实施例提供的一种帧内预测编码方法，包括以下步骤：

对同一个帧使用至少两种帧内预测编码方式进行预测编码；

在该帧的编码结果中写入表示该帧中每个块所使用的帧内预测编码方式的信息，其中，在该编码结果中每个块只使用一种帧内预测编码方式，至少有两种编码方式应用在该帧的编码结果中。

本发明实施例提供的一种帧内预测解码方法，包括以下步骤：

对于一个帧中的每个块，分别从该帧的编码结果中读取该块所使用的帧内预测编码方式的信息，并使用该方式对该块进行解码。

本发明实施例提供的一种帧内预测编码装置，包括：

对同一个帧使用至少两种帧内预测编码方式进行预测编码的单元；

在该帧的编码结果中写入表示该帧中每个块所使用的帧内预测编码方式的信息的单元，其中，在该编码结果中每个块只使用一种帧内预测编码方式，至少有两种编码方式应用在该帧的编码结果中。

本发明实施例提供的一种帧内预测解码装置，包括：

至少两个解码器，每个解码器用于对一种帧内预测编码方式进行解码；

第一读取单元，用于对同一帧内的每个块从帧的编码结果中读取该块所使用的帧内预测编码方式的信息，并指示该帧内预测编码方式所对应的所述解码器对该块进行解码。

与现有技术相比，本发明实施例具有以下主要区别及有益效果：

对同一个帧使用至少两种帧内预测编码方式进行预测编码，并将每个块所对应的帧内预测编码方式信息写入编码结果中，从而可以对不同的块使用最适合的方式进行编码，相对于现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案，本发明实施例所得到的编码结果性能更优。

如果同一媒体流中有些帧使用了至少两种帧内预测编码方式，有些帧只用了一种帧内预测编码方式，则需要向前者的编码结果中写入表示使用了至少两种帧内预测编码方式的信息，以便可以正确地进行解码。

对于使用了至少两种帧内预测编码方式的情况，解码时，对同一帧内每个块先读取该块所用的帧内预测编码方式再使用该方式对该块进行解码，从而使每个块都得到正确的解码。

可以在编码结果中写入表示每个块大小的信息，从而同一帧中的各个块可以不限于固定的划分方式，有利于使用不同的帧内预测编码方式获得更优的编码性能。

可以对每个块尝试使用各种帧内预测编码方式进行预测编码，获得多种候选编码结果，再根据预定的性能指标，例如：率失真性能，选出最佳的候选编码结果作为最终的编码结果，从而使最终的编码结果具有最佳的性能。

以分级树的形式对各个块进行组织，可以在获得最优编码性能的同时减少编码结果中表示每个块帧内预测编码方式的信息的信息量。

本发明实施例的技术方案可以应用于空间域的帧或频域的帧，频域包括各种变换所产生的频域，例如：离散余弦变换（Discrete Cosine Transform，以下简称：DCT）、离散傅立叶变换（Discrete Fourier Transformation，以下简称：DFT）、小波变换等，所以本发明实施例的应用领域较广。

附图说明

图 1 为现有技术中 H.264/AVC 帧内预测编码方法示意图；

图 2 为现有技术中移位帧内预测编码方法示意图；

- 图 3 为现有技术中模板匹配帧内预测编码方法示意图；
- 图 4 为根据本发明第一实施例的帧内预测编解码方法流程图；
- 图 5 为根据本发明第一实施例的帧内预测编解码方法中宏块分割示意图；
- 图 6 为根据本发明第一实施例的帧内预测编解码方法中的解码实例流程图；
- 图 7 为根据本发明第一实施例的帧内预测编解码方法带来的率失真性能增益示意图；
- 图 8 为根据本发明第二实施例的帧内预测编码装置结构图；
- 图 9 为根据本发明第二实施例的帧内预测解码装置结构图。

具体实施方式

为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明作进一步地详细描述。

在本发明各实施例中，对同一个帧使用至少两种帧内预测编码方式进行预测编码，并将每个块所对应的帧内预测编码方式信息写入编码结果中，从而可以对不同的块使用最适合的方式进行编码，相对于现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案，本发明的技术方案所得到的编码结果性能更优。

下面对本发明第一实施例进行说明，本发明第一实施例涉及一种帧内预测编解码方法，如图 4 所示。

在步骤 410 中，编码端对同一个帧中的每个块分别尝试使用所有可用的帧内预测编码方式进行预测编码，分别得到对应的候选编码结果，从各候选编码结果中选择性能指标最佳的那个作为该块最终的编码结果。该方式相当于对不同的块使用最适合的方式进行编码，相对于现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案，本实施例所得到的编码结果性能更

优。这里的性能指标可以是率失真性能。所使用的帧内预测编码方式可以是 H.264/AVC 的方向预测方式、移位帧内预测方式、模板匹配预测方式等。

具体地说,进行编码的块可以是编码基本单位——宏块,也可以是宏块分成的子块。对每个块进行多种方式的预测编码并从中选择性能指标最佳的一个作为最终编码结果后,根据每个块的最终的编码结果对应的帧内预测编码方式,以分级树的形式进行组织。从而使得在获得最优编码性能的同时减少编码结果中表示每个块帧内预测编码方式的信息的信息量。以 16×16 的宏块为例,一个 16×16 的宏块可以四分为 4 个 8×8 的子块,一个 8×8 的子块也可以进一步四分为 4 个 4×4 的子块。编码端对这个 16×16 的宏块、该宏块内各个 8×8 子块、该宏块内各 4×4 子块分别进行多种预测方式的编码,从各候选编码结果的组合中选择率失真性能最佳的编码方式组合作为该宏块最终的编码结果,以分级树的形式进行组织。图 5 为根据本发明第一实施例的帧内预测编解码方法中宏块分割示意图。一个 16×16 宏块的编码结果例子如图 5 所示,其左上角的 8×8 子块的最终编码结果是移位帧内预测方式的编码结果;左下角的 8×8 子块的最终编码结果是 H.264/AVC 的方向预测方式的编码结果;右下角的 8×8 子块的最终编码结果为模板匹配预测方式的编码结果;而右上角的 4×4 子块的最终编码结果为移位帧内预测方式的编码结果,其余 3 个 4×4 子块的最佳编码结果均为 H.264/AVC 的方向预测方式的编码结果。

当然,这里的宏块也不一定是 16×16 的宏块,还可以是其它格式如 32×32 的宏块。同样,子块也不一定以四分级形式进行组织,也可以有其它组织形式,如二分级形式。在二分级形式中,一个 16×16 的宏块可以包含 2 个 8×16 的子块或 2 个 16×8 的子块。

该步骤 410 中具体的预测编码方式可以参考现有技术。例如: H.264/AVC 的方向预测编码方式可以参考 ITU-T 的“通用视听业务的高级视频编码”(ITU-T Rec. H.264, Advanced video coding for generic audiovisual services, 2005), 移位帧内预测编码方式可以参考“使用帧内宏块运动补偿的新帧内预

测”(S.-L. Yu and C. Chrysafis, New Intra Prediction using Intra-Macroblock Motion Compensation, JVT meeting Fairfax, doc JVT-C151, May 2002), 模板匹配预测编码方式可以参考“通过模板匹配的帧内预测”(T.K. Tan, C.S. Boon, and Y. Suzuki, Intra Prediction by Template Matching, ICIP 2006)。

需要说明的是,在进行移位帧内预测时,使用和帧间预测运动补偿相似的方法,并采用 H.264/AVC 标准中的运动矢量编码方法对移位矢量编码。

另外,本实施例中宏块子块的扫描顺序和 H.264/AVC P_8x8 宏块类型的子块扫描顺序相同。

接着进入步骤 420,在该帧的编码结果中写入表示该帧中每个块大小的信息,以及表示每个块所使用的帧内预测编码方式的信息。由于包含表示块大小的信息,同一帧中的各个块可以不限于固定的划分方式,有利于使用不同的帧内预测编码方式获得更优的编码性能。

具体地说,如果该帧中各块使用了至少两种不同的帧内预测编码方式进行预测编码,每种方式应用于该帧内的一个或多个块,则在该帧的编码结果中写入表示该帧使用了多种帧内预测编码方式进行预测编码的信息,从而即使本发明技术方案可以与现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案共存,解码端也能分辨得出,进而进行正确的解码。

如果分级树的一个分支的根节点所对应的块,该块可以是宏块或子块,只使用了一种帧内预测编码方式,即:该宏块或子块中所有子块的最佳编码方式为同一种帧内预测编码方式,或该宏块或子块为一个分支的叶节点所对应的块,则向该帧的编码结果中写入表示该块大小的信息和表示该块所使用的帧内预测编码方式的信息,否则进一步对该分支的各下属分支进行处理。

需要说明的是,对于移位帧内预测编码方式,还需要记录其移位矢量。

通过语法表表示出来的该帧的存储结构如下表 1 所示。其中,使用了多种帧内预测编码方式进行预测编码的宏块类型用 I_EXT 表示,并且 NumMbPart(I_EXT) = 1。

表 1 中, ext_intra_pred_flag 表示该帧是否使用了多种帧内预测编码方式进行预测编码, ext_intra_pred_flag 等于 1 表示使用了多种帧内预测编码方式进行预测编码;

ext_16x16_pred_flag 等于 1 表示该帧内预测类型的宏块分割为 16×16;
ext_16x16_pred_flag 等于 0 表示宏块被进一步分割为 8×8;
ext_16x16_pred_mode 表示 16×16 分割的宏块的帧内预测编码方式;
ext_8x8_pred_flag[] 等于 1 表示按照 8×8 分割进行预测, ext_8x8_pred_flag[] 等于 0 表示子块被进一步分割为 4×4 的子块;
ext_8x8_pred_mode[] 表示每个 8×8 分割的子块的帧内预测编码方式;
ext_4x4_pred_mode[][] 表示每个 4×4 分割的子块的帧内预测编码方式;
dvd[][][] 表示在移位帧内预测编码方式 (displacement intra prediction, DIP) 时的移位矢量。

表 1

mb_pred(mb_type) {	C	Descriptor
ext_intra_pred_flag	2	u(1) ae(v)
if(ext_intra_pred_flag) {		
ext_16x16_pred_flag	2	u(1) ae(v)
if(ext_16x16_pred_flag)		
ext_16x16_pred_mode	2	ue(v) ae(v)
if(ext_16x16_pred_mode == DIP)		
for(compIdx = 0; compIdx < 2; compIdx++)		
dvd[0][0][compIdx]	2	ue(v) ae(v)
else {		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
ext_8x8_pred_flag[luma8x8BlkIdx]	2	u(1) ae(v)
if(ext_8x8_pred_flag[luma8x8BlkIdx]) {		
ext_8x8_pred_mode[luma8x8BlkIdx]	2	ue(v) ae(v)
if(ext_8x8_pred_mode[luma8x8BlkIdx] == DIP)		
for(compIdx = 0; compIdx < 2; compIdx++)		
dvd[luma8x8BlkIdx][0][compIdx]	2	ue(v) ae(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 0		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<4;		
luma4x4BlkIdx++) {		
ext_4x4_pred_mode[luma8x8BlkIdx][luma4x4BlkIdx]	2	ue(v) ae(v)
if(ext_4x4_pred_mode[luma8x8BlkIdx][luma4x4BlkIdx]		
==		
DIP)		
for(compIdx = 0; compIdx < 2; compIdx++)		
dvd[luma8x8BlkIdx][luma4x4BlkIdx][compIdx]	2	ue(v) ae(v)
}		
}		
}		
}		
}		
} else {		
...		

另外，ae(v):用基于上下文的自适应二元算术码编码的语法元素；

u(n): n 位无符号整数。在语法表中，如果 n 是'v'，其比特数由其他语法元素值确定。解析过程由函数 read_bits(n)的返回值规定，该返回值用高位在前的二进制表示。

ue(v): 无符号整数语法元素，用指数哥伦布码编码。

接着进入步骤 430，在解码端，首先确定该帧是否采用了多种帧内预测编码方式进行预测编码，如果从该帧的编码结果中读取到表示该帧使用了至

少两种帧内预测编码方式进行预测编码的信息，则进入步骤 440，否则进入步骤 450，按照现有技术，对该帧中的所有块使用统一帧内预测编码方式进行解码。

在步骤 440 中，解码端读取该帧的编码结果，逐一寻找编码结果中表示块大小的信息，根据表示块大小的信息和表示该块所使用的帧内预测编码方式的信息进行解码。由于该帧内包含的块是以分级树的形式组织的，如果从该帧的编码结果中读取到表示块大小的信息，其中的块包括宏块或子块，则表明该大小的块中采用的是同一种帧内预测编码方式，进一步读取表示该块所使用的帧内预测编码方式的信息，并使用该方式对该大小的块进行解码。

下面根据表 1 的编码结构进行举例，如图 6 所示，为根据本发明第一实施例的帧内预测编解码方法中的解码实例流程图，解码端首先根据解析得到的 `ext_intra_pred_flag` 标志位来判断当前码流是否采用了多种帧内预测编码方式进行预测编码，如果不是，则照现有技术，对该帧中的所有块使用统一帧内预测编码方式进行解码，例如：按照 H.264/AVC 标准进行解码；如果是，则继续根据 `ext_16x16_pred_flag` 标志位判断当前的帧内宏块分割情况。

如果 `ext_16x16_pred_flag` 等于 1 表明宏块分割为 16×16，解析 `ext_16x16_pred_mode` 得到当前块的编码方式，根据该编码方式对应的解码方式对该 16×16 的宏块进行解码，如果当前块的编码方式为移位帧内预测编码方式，则还需要从码流中解析出移位矢量 `dvd`，然后采用对应的解码方法进行解码；如果 `ext_16x16_pred_flag` 等于 0，则表明该 16×16 的宏块被分为 8×8 的块，进一步解析该宏块的每个 8×8 块。

对于每个 8×8 块，根据其 `ext_8x8_pred_flag`，如果 `ext_8x8_pred_flag` 等于 1 则解析码流中的 8×8 块的预测编码方式 `ext_8x8_pred_mode`，根据编码方式选择对应的解码方法进行解码，如果当前块编码方式为移位帧内预测编码方式，则还需要从码流中解析出每个 8×8 块的移位矢量 `dvd`，然后采用对应的解码方法进行解码；如果 `ext_8x8_pred_flag` 等于 0 则表明当前块进一步被

分割为 4×4 子块，解析每个 4×4 块的预测编码方式 ext_4x4_pred_mode。

对于每个 4×4 块，解析其预测编码方式 ext_4x4_pred_mode，根据编码方式选择对应的解码方法进行解码，如果当前块的编码方式为移位帧内预测编码方式，则还需要从码流中解析出每个 4×4 块的移位矢量 dvd，然后采用对应的解码方法进行解码。

图 7 表明了使用本实施例的帧内预测编解码方法所带来的率失真性能增益。

值得一提的是，本实施方式中所说的帧可以是空间域或频域的帧，频域包括各种变换所产生的频域，例如：DCT、DFT、小波变换等，所以本实施例的应用领域较广。

本发明第二实施方式涉及一种帧内预测编码装置，如图 8 所示，为根据本发明第二实施例的帧内预测编码装置结构图，该实施例的帧内预测编码装置包括：编码单元，用于对同一个帧使用至少两种帧内预测编码方式进行预测编码，其中每种方式应用于该帧内至少一个块。使用多种帧内预测编码方式进行预测编码使得可以对不同的块使用最适合的方式进行编码，相对于现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案，本编码装置所得到的编码结果性能更优。该编码单元进一步包含：至少两个编码器，每个编码器用于使用一种帧内预测编码方式对每个块进行预测编码；一个选择器，用于从各编码器输出的选编码结果中选择率失真性能最佳的那个作为该块最终的编码结果，从而使最终的编码结果具有最佳的性能。该帧内预测编码装置还包含写入单元，用于在该帧的编码结果中写入表示该帧中每个块所使用的帧内预测编码方式的信息和每个块大小的信息。通过在编码结果中写入表示每个块大小的信息，使得同一帧中的各个块可以不限于固定的划分方式，有利于使用不同的帧内预测编码方式获得更优的编码性能。如果同一媒体流中有些帧使用了至少两种帧内预测编码方式，有些帧只用了一种帧内预测编码方式，则该写入单元在一个帧中使用了至少两种帧内预测编码方式进

行预测编码时，在该帧的编码结果中写入表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，以便可以正确地进行解码。其中，上述的帧可以是空间域或频域的帧，性能指标可以是率失真性能，帧内预测编码方式至少包含：方向预测方式、移位帧内预测方式、和模板匹配预测方式。

另外，该写入单元还可以以分级树的形式对各个块进行组织，从而可以在获得最优编码性能的同时减少编码结果中表示每个块帧内预测编码方式的信息的信息量。

本发明第三实施方式涉及一种帧内预测解码装置，如图 9 所示，为根据本发明第二实施例的帧内预测解码装置结构图，该实施例的帧内预测解码装置包括：

解码单元，其中包含至少两个解码器，每个解码器用于对一种帧内预测编码方式进行解码；

第一读取单元，用于对同一帧内的每个块从帧的编码结果中读取该块所使用的帧内预测编码方式的信息，并指示该帧内预测编码方式所对应的解码器对该块进行解码；

第二读取单元，用于从帧的编码结果中尝试读取表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，如果读取到表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，则指示第一读取单元和各解码器对该帧中的每个块分别进行解码，否则指示解码器中的一个对该帧中的所有块使用统一的帧内预测编码方式进行解码。

其中的帧同样可以是空间域或频域的帧，频域包括各种变换所产生的频域，例如：DCT、DFT、小波变换等，所以本实施方式的应用领域较广。

帧内预测编码方式至少包括：方向预测方式、移位帧内预测方式、和模板匹配预测方式。

需要说明的是，上述两个装置实施方式中的单元均是逻辑单元，在具体应用时，可以有不同的物理实现形式。

综上所述，在本发明的各实施例中，对同一个帧使用至少两种帧内预测编码方式进行预测编码，并将每个块所对应的帧内预测编码方式信息写入编码结果中，从而可以对不同的块使用最适合的方式进行编码，相对于现有技术中只用一种方式对帧内所有块进行帧内编码的技术方案，本发明的技术方案所得到的编码结果性能更优。

如果同一媒体流中有些帧使用了至少两种帧内预测编码方式，有些帧只用了一种帧内预测编码方式，则需要向前者的编码结果中写入表示使用了至少两种帧内预测编码方式的信息，以便可以正确地进行解码。

对于使用了至少两种帧内预测编码方式的情况，解码时，对同一帧内每个块先读取该块所用的帧内预测编码方式再使用该方式对该块进行解码，从而使每个块都得到正确的解码。

可以在编码结果中写入表示每个块大小的信息，从而同一帧中的各个块可以不限于固定的划分方式，有利于使用不同的帧内预测编码方式获得更优的编码性能。

可以对每个块尝试使用各种帧内预测编码方式进行预测编码，获得多种候选编码结果，再根据预定的性能指标，例如：率失真性能，选出最佳的候选编码结果作为最终的编码结果，从而使最终的编码结果具有最佳的性能。

以分级树的形式对各个块进行组织，可以在获得最优编码性能的同时减少编码结果中表示每个块帧内预测编码方式的信息的信息量。

本发明的技术方案可以应用于空间域的帧或频域的帧，频域包括各种变换所产生的频域，例如：DCT、DFT、小波变换等，所以本发明的应用领域较广。

最后所应说明的是，以上实施例仅用以说明本发明的技术方案而非限制，尽管参照较佳实施例对本发明进行了详细说明，本领域的普通技术人员应当理解，可以对本发明的技术方案进行修改或者等同替换，而不脱离本发明技术方案的精神和范围。

权 利 要 求

1、一种帧内预测编码方法，其特征在于，包括以下步骤：

对同一个帧使用至少两种帧内预测编码方式进行预测编码；

在该帧的编码结果中写入表示该帧中每个块所使用的帧内预测编码方式的信息，其中，在该编码结果中每个块只使用一种帧内预测编码方式，至少有两种编码方式应用在该帧的编码结果中。

2、根据权利要求1所述的帧内预测编码方法，其特征在于，还包括以下步骤：

在所述帧的编码结果中写入表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息。

3、根据权利要求1所述的帧内预测编码方法，其特征在于，还向所述帧的编码结果中写入表示每个块大小的信息。

4、根据权利要求1所述的帧内预测编码方法，其特征在于，通过以下方式对同一个帧使用至少两种帧内预测编码方式进行预测编码：

对所述帧中的每个块分别尝试使用所有可用的帧内预测编码方式进行预测编码，得到该块的至少两种候选编码结果，从各候选编码结果中选择性能指标最佳的那个作为该块最终的编码结果。

5、根据权利要求4所述的帧内预测编码方法，其特征在于，所述性能指标为率失真性能。

6、根据权利要求1至5中任意一项所述的帧内预测编码方法，其特征在于，所述帧内预测编码方式包括：

方向预测方式、移位帧内预测方式、和模板匹配预测方式。

7、根据权利要求1至5中任意一项所述的帧内预测编码方法，其特征在于，所述帧为空间域或频域的帧。

8、根据权利要求1至5中任意一项所述的帧内预测编码方法，其特征在于，所述块以分级树的形式组织；

如果分级树的一个分支的根节点所对应的块只使用了一种帧内预测编码方式，则向该帧的编码结果中写入表示该块大小的信息和表示该块所使用的帧内预测编码方式的信息，否则进一步对该分支的各下属分支进行处理。

9、一种帧内预测解码方法，其特征在于，包括以下步骤：

对于一个帧中的每个块，分别从该帧的编码结果中读取该块所使用的帧内预测编码方式的信息，并使用该方式对该块进行解码。

10、根据权利要求 9 所述的帧内预测解码方法，其特征在于，还包括以下步骤：

在对所述帧进行解码之前，如果从该帧的编码结果中读取到表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，则对该帧中的每个块分别使用该块所对应的帧内预测编码方式进行解码，否则对该帧中的所有块使用统一帧内预测编码方式进行解码。

11、根据权利要求 9 所述的帧内预测解码方法，其特征在于，所述块以分级树的形式组织；

如果从所述帧的编码结果中读取到表示块大小的信息，则进一步读取表示该块所使用的帧内预测编码方式的信息，并使用该方式对该大小的块进行解码。

12、根据权利要求 9 所述的帧内预测解码方法，其特征在于，所述帧内预测编码方式包括：

方向预测方式、移位帧内预测方式、和模板匹配预测方式。

13、根据权利要求 9 所述的帧内预测解码方法，其特征在于，所述帧为空间域或频域的帧。

14、一种帧内预测编码装置，其特征在于，包括：

对同一个帧使用至少两种帧内预测编码方式进行预测编码的单元；

在该帧的编码结果中写入表示该帧中每个块所使用的帧内预测编码方式的信息的单元，其中，在该编码结果中每个块只使用一种帧内预测编码方式，

至少有两种编码方式应用在该帧的编码结果中。

15、根据权利要求 14 所述的帧内预测编码装置，其特征在于，还包括，在该帧的编码结果中写入表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息的单元。

16、根据权利要求 14 所述的帧内预测编码装置，其特征在于，还包括向所述帧的编码结果中写入表示每个块大小的信息的单元。

17、根据权利要求 14 所述的帧内预测编码装置，其特征在于，所述对同一个帧使用至少两种帧内预测编码方式进行预测编码的单元进一步包括：

至少两个编码器，每个编码器用于使用一种帧内预测编码方式对每个块进行预测编码；

一个选择器，用于从各所述编码器输出的选编码结果中选择率失真性能最佳的那个作为该块最终的编码结果。

18、一种帧内预测解码装置，其特征在于，包括：

至少两个解码器，每个解码器用于对一种帧内预测编码方式进行解码；

第一读取单元，用于对同一帧内的每个块从帧的编码结果中读取该块所使用的帧内预测编码方式的信息，并指示该帧内预测编码方式所对应的所述解码器对该块进行解码。

19、根据权利要求 18 所述的帧内预测解码装置，其特征在于，还包括：

第二读取单元，用于从所述帧的编码结果中尝试读取表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，如果读取到表示该帧使用了至少两种帧内预测编码方式进行预测编码的信息，则指示所述第一读取单元和各所述解码器对该帧中的每个块分别进行解码，否则指示各所述解码器中的一个对该帧中的所有块使用统一的帧内预测编码方式进行解码。

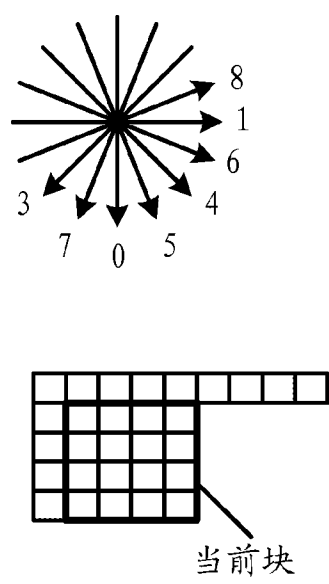


图 1

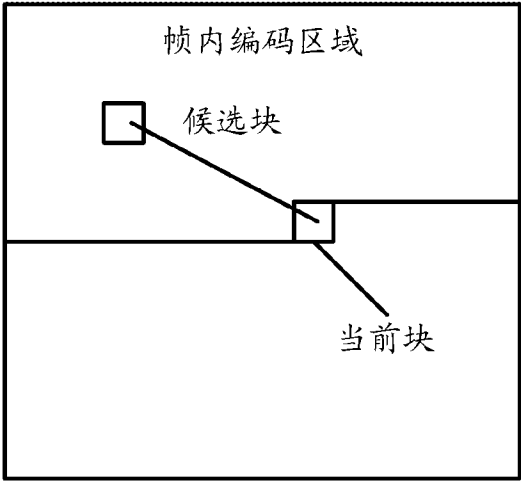


图 2

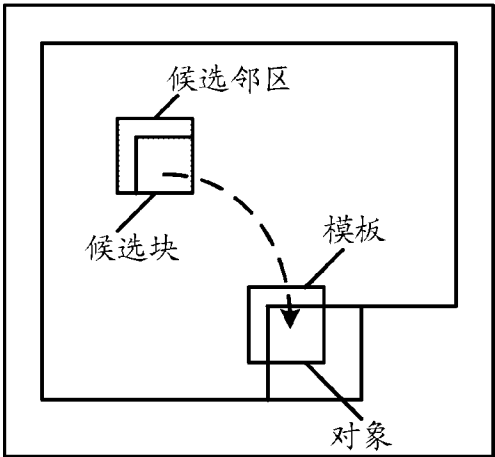


图 3

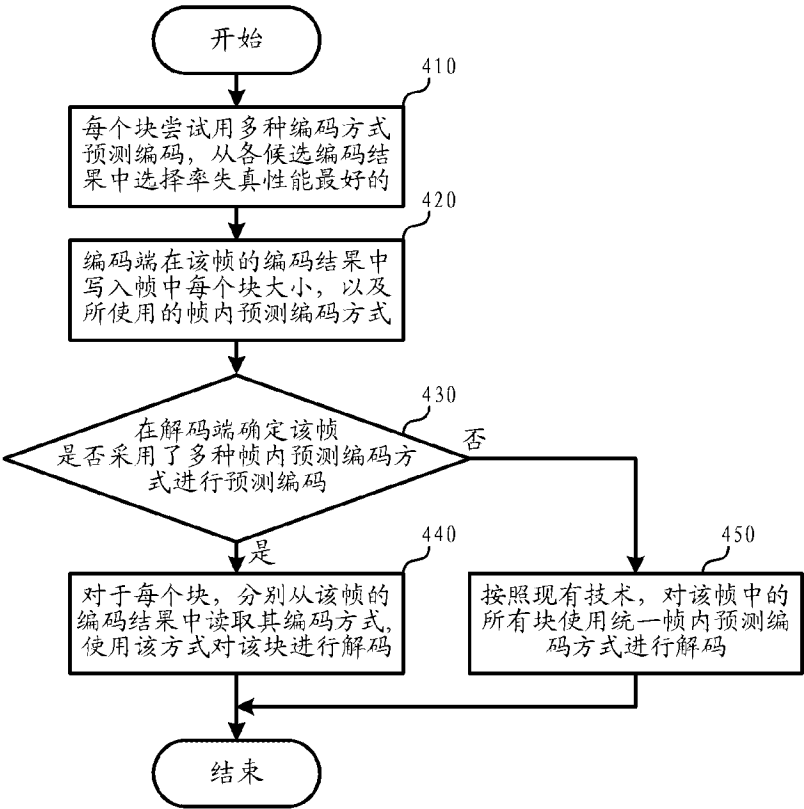


图 4

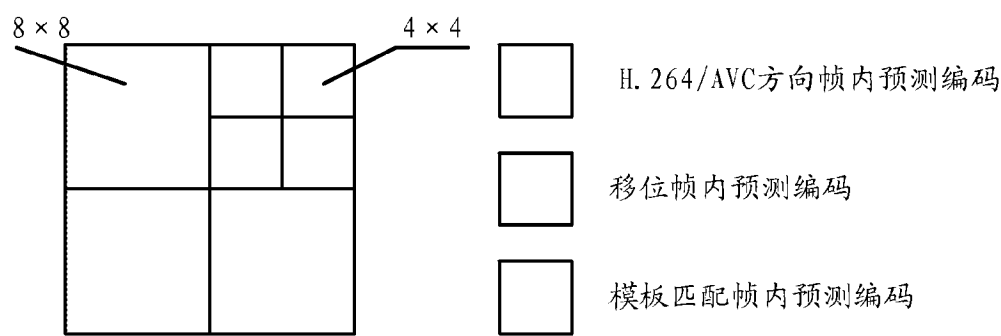


图 5

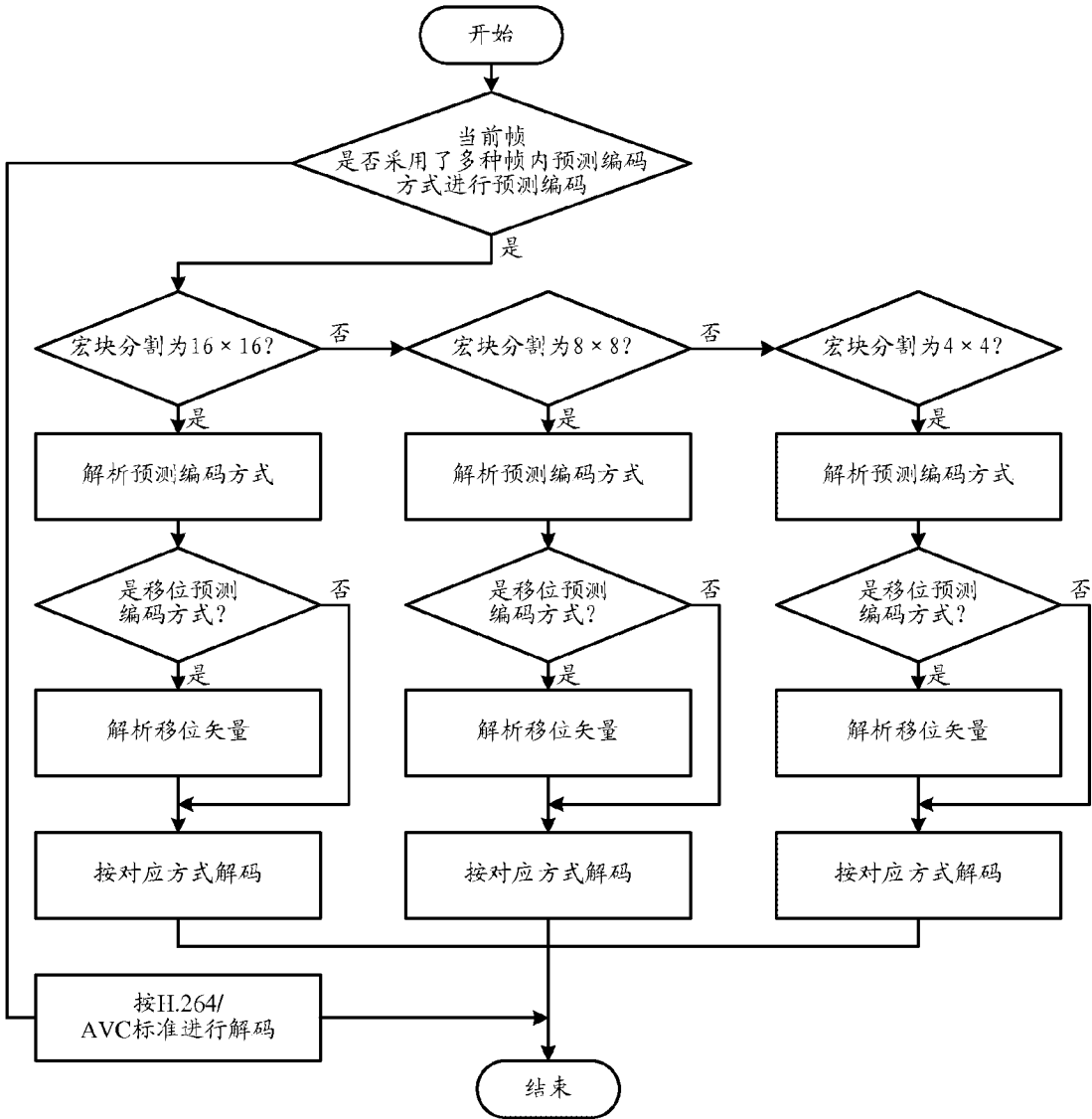


图 6

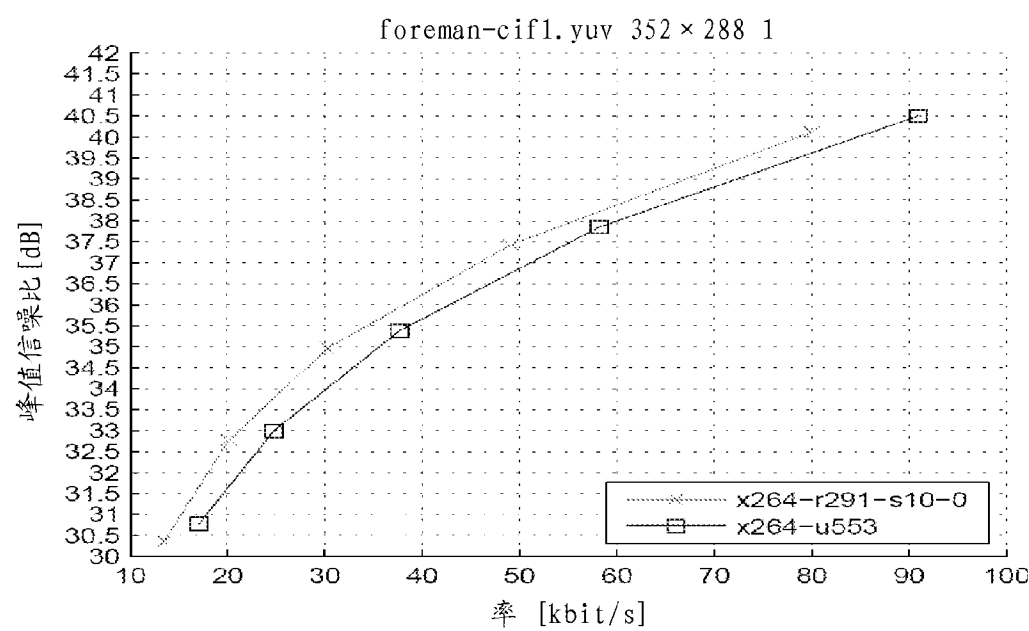


图 7

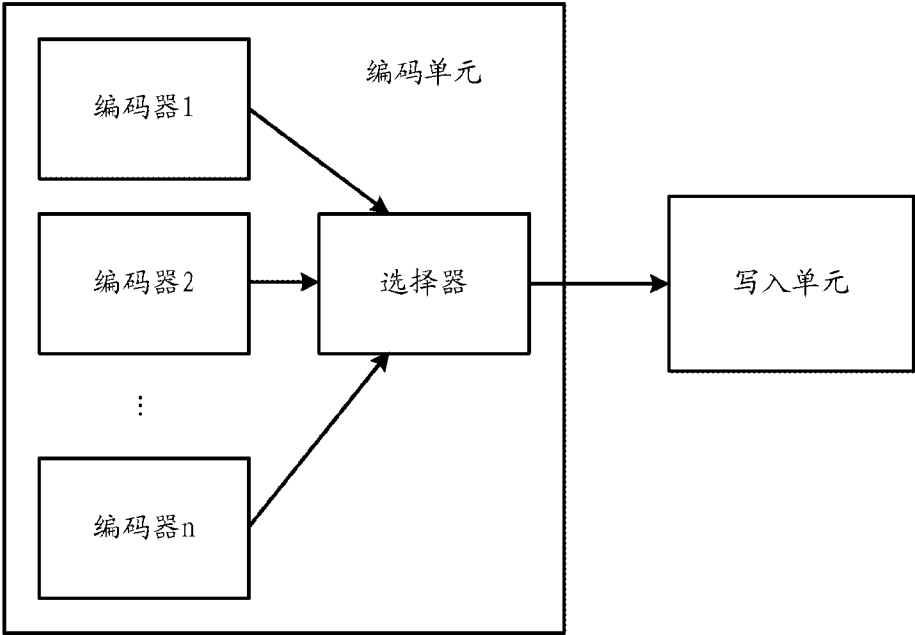


图 8

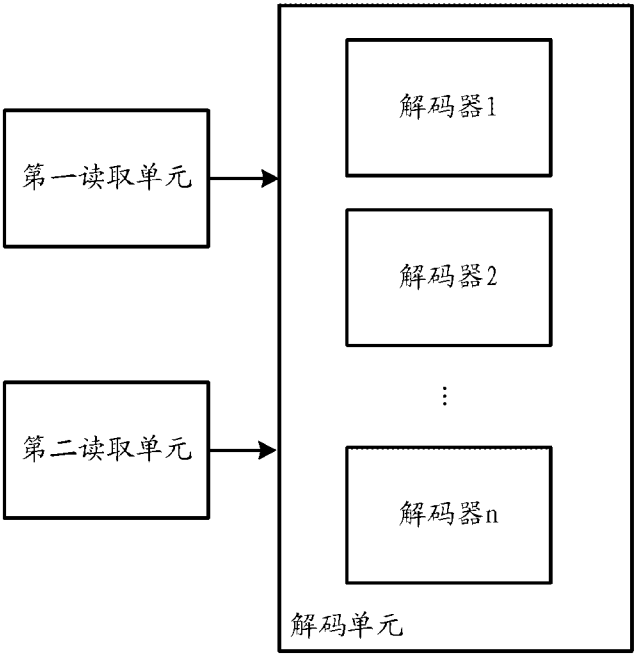


图 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2008/070092

A. CLASSIFICATION OF SUBJECT MATTER		
H04N 7/32 (2006.01) i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
IPC: H04N 7/-		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
WPI; EPODOC; PAJ: FRAME, INTRA, PREDICTION, ENCODE, DECODE, VIDEO, MACRO BLOCK, PIXEL, ADAPT+, SELECT, H.264, AVC, HIBRID		
CNPAT:		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN1794816A (NEC CORP) 28 June 2006 (28.06.2006) description page 8 line 16-line 28, figure 3	1-7, 9-19
Y		8
Y	JP1080186A (FUJITSU LTD) 27 March 1989 (27.03.1989) description page 5 top right corner line 20 -- page 7 top left corner line 2, figure 5	8
A	CN1476253A (UNIV QINGHUA) 18 February 2004 (18.02.2004) the whole document	1-19
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search 09 April 2008 (09.04.2008)		Date of mailing of the international search report 24 Apr. 2008 (24.04.2008)
Name and mailing address of the ISA/CN The State Intellectual Property Office, the P.R.China 6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088 Facsimile No. 86-10-62019451		Authorized officer WU Shuang Telephone No. (86-10)62411507

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2008/070092

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN1794816A	28.06.2006	EP1675404A2	28.06.2006
		US2006133489A1	22.06.2006
		JP2006180195A	06.07.2006
		KR20060071894A	27.06.2006
JP1080186A	27.03.1989	JP2716703B2	18.02.1998
CN1476253A	18.02.2004	None	

国际检索报告		国际申请号 PCT/CN2008/070092
A. 主题的分类		
H04N 7/32 (2006.01) i		
按照国际专利分类表(IPC)或者同时按照国家分类和 IPC 两种分类		
B. 检索领域		
检索的最低限度文献(标明分类系统和分类号)		
IPC: IPC: H04N 7/-		
包含在检索领域中的除最低限度文献以外的检索文献		
在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))		
WPI; EPODOC; PAJ: FRAME, INTRA, PREDICTION, ENCODE, DECODE, VIDEO, MACRO BLOCK, PIXEL, ADAPT+, SELECT, H.264, AVC, HIBRID		
CNPAT: 帧内, 预测, 编码, 解码, 视频, 宏块, 像素, 自适应, 选择, H.264, AVC, 混合		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	CN1794816A (日本电气株式会社) 28.6 月 2006 (28.06.2006) 说明书第 8 页第 16 行—第 28 行, 附图 3	1—7, 9—19
Y		8
Y	JP1080186A (富士通株式会社) 27.3 月 1989 (27.03.1989) 说明书第 5 页右上栏第 20 行—第 7 页左上栏第 2 行, 附图 5	8
A	CN1476253A (清华大学) 18.2 月 2004 (18.02.2004) 全文	1—19
☐ 其余文件在 C 栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件		“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件
国际检索实际完成的日期 9.4 月 2008 (9.04.2008)		国际检索报告邮寄日期 24.4 月 2008 (24.04.2008)
中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路 6 号 100088 传真号: (86-10)62019451		受权官员 吴爽 电话号码: (86-10) 62411507

国际检索报告 关于同族专利的信息		国际申请号 PCT/CN2008/070092	
检索报告中引用的 专利文件	公布日期	同族专利	公布日期
CN1794816A	28.06.2006	EP1675404A2	28.06.2006
		US2006133489A1	22.06.2006
		JP2006180195A	06.07.2006
		KR20060071894A	27.06.2006
JP1080186A	27.03.1989	JP2716703B2	18.02.1998
CN1476253A	18.02.2004	无	

PCT/ISA/210 表(同族专利附件) (2007 年 4 月)

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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

3940

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

Image Compression with Variable Block Size Segmentation

Jacques Vaissey, 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 0.45 and 0.7 b/p depending on the nature of the original image, and satisfactory results have been obtained at 0.25 b/p.

1. 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 8×8 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 SRI W3 Program.

J. Vaissey 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 920, 071

1053-587X/92\$04.00 © 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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

VAUSEY AND GERSHIO: IMAGE COMPRESSION

7941



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 quadrees 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 quadrees 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, Borgelet *et al.* [18] used quadrees in lossless image coding, and both Strobach [19] and Nasrabadi *et al.* [20] used them in the coding of image sequences. Buhler and Fertier [21], [22] have independently introduced a quadree 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 quadree 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 quadree segmentation of the input image, $\hat{Q}$, 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 four 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。
取扱いにあたっては、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_0 , which is then applied to an interpolation procedure that generates an interpolated image, Ω_1 , 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_2 .

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_2 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 $\hat{\Omega}$.

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. Quadrees, 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

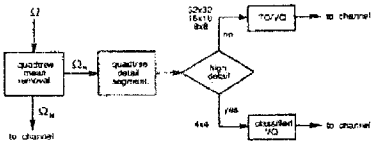
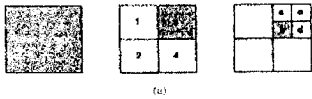


Fig. 2. Operations performed by the encoder.



(a)

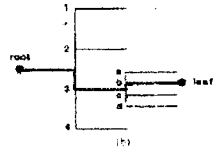


Fig. 3. The structure of the quadtree. (a) The relationship to image regions. (b) The relationship in a tree.

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 n th 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_1 , 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。
取扱いにあたっては、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_k and Ω_l 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_l 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_l . 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.

4. 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_k (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 value 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 than 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 64×64 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_d) 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 uncoded 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。

2044

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

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 removed at node "a" 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_Q . 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_Q ; 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 orders 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_p that is easier to code than when using Ω_Q 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 black, or black, image plane blocked in the same manner as Ω_Q ; this acts as a foundation upon which a "tent" is constructed. Tent-

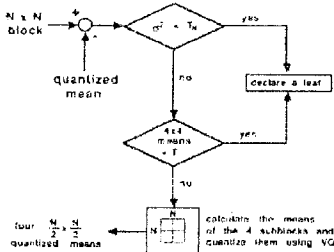


Fig. 5. The flowchart for the mean segmentation.

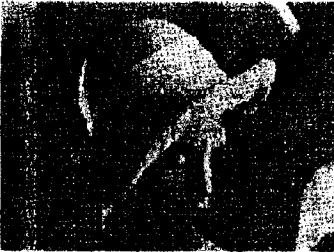


Fig. 6. The mean image for Lenna.

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_Q ; 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。

VASIV AND GERSH: IMAGE COMPRESSION

2945

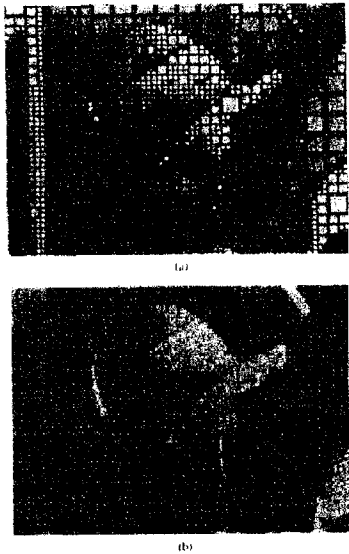


Fig. 7. Interpolating the mean image: (a) The pole image for Lenna; (b) The interpolated image for Lenna.

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_i , (where the i indicates that the pixel is located at the intersection of the i th row and j th column of the image). Our technique, motivated by that described in [26], is to look in each angular direction, θ , away from f_i 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_i is then determined by

$$f_i = \int_0^{2\pi} \frac{p(\theta)}{d_i(\theta)} d\theta \cdot \frac{1}{\int_0^{2\pi} \frac{d\theta}{d_i(\theta)}} \quad (1)$$

where $d_i(\theta)$ is the Euclidean distance between the centers of f_i and the pixel belonging to the closest pole pixel in

the direction θ . Other distance measures could be used; however, $d_i(\theta)$ 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_i into sections whose angular extent is $\pi/4$ rad. The center of the k th section occurs at the angle

$$\theta_k = \frac{k\pi}{4}, \quad k = 0, 1, \dots, 7. \quad (2)$$

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 $d_k^*(\theta_k)$. With these definitions, the interpolated pixel amplitude is given by

$$f_i = \sum_{k=0}^7 \frac{p^*(\theta_k)}{d_k^*(\theta_k)} \cdot \frac{1}{\sum_{k=0}^7 \frac{1}{d_k^*(\theta_k)}}. \quad (3)$$

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(a)). 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。

2046

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 48, NO. 8, AUGUST 1997

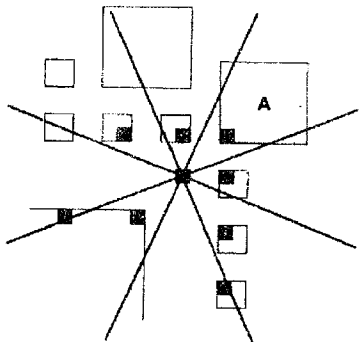


Fig. 8. Interpolating the pole image. The black pixel is to be interpolated, and the medium-gray pixels are the closest gray pixels in each direction.

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 1×4 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 $\hat{\Omega}$. 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 encoders,

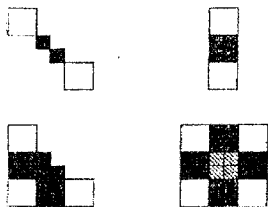


Fig. 9. Hardening boundary pixels. Each of the 4 possible cases displayed. The filled in regions are shown by the small squares.

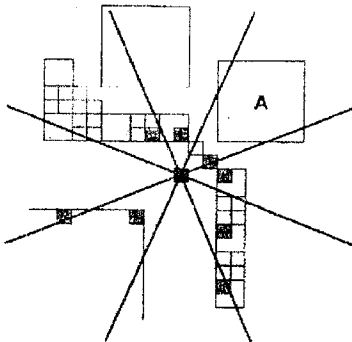


Fig. 10. An example of hardening. The boundary has been hardened and pole A no longer affects the interpolation.

TABLE I
The Choice of Pole Size

Parameter	Pole Sizes (Pixels)	
	Case 1	Case 2
Region size	12×12	24×24
	16×16	12×12
	8×8	8×8
	4×4	4×4
Mean PSNR (dB)	24.15	24.26
Satout PSNR (dB)	23.70	25.35
Colet PSNR (dB)	26.35	26.02

while case 2 is a reasonable alternative. The size of the poles also affects the appearance of $\hat{\Omega}$, since the amount of smoothing increases as the pole size decreases. Images with higher amounts of smoothing tend to look more nat-

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。
取扱いにあたっては、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_2 . 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_2 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 highlyly 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 Noll [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_2 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_3 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 [31]). 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. Bubler [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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

2048

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

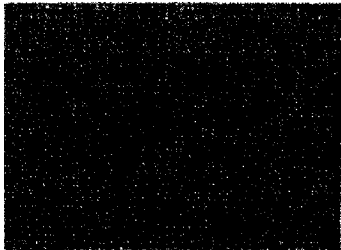


Fig. 11. The residual image for Lena.

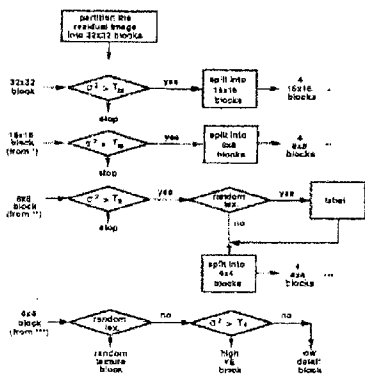


Fig. 12. The decision for the VE segmentation.

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_k , 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 conserv-

atively 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 Prasad [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 coarse 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。

VARIABLE-LENGTH IMAGE COMPRESSION

7349

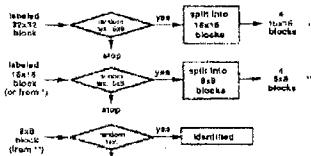


Fig. 13 Coding the random texture representation.

cidatus 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 3×8, four nonoverlapping blocks and one in the center that overlaps each of the four others equally. The variance of the *i*th block is denoted by $\sigma_{i,16}^2$, and the average of the five variances by $\sigma_{a,16}^2$. The block of interest is passed onto the next test as a random texture candidate if

$$u_{acc} > 200 \tag{4}$$

and

$$\frac{(\sigma_{i,16}^2 + \sigma_{a,16}^2)}{\sigma_{a,16}^2} < T_{tex}, \quad i = 1, 2, 3, 4, 5. \tag{5}$$

Through experimentation, we found that a value of $T_{tex} = 0.65$ 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

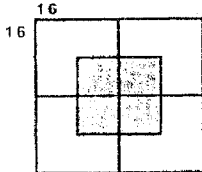


Fig. 14 The texture neighborhood.

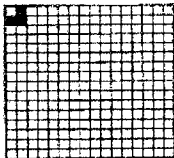


Fig. 15 The DCT exclusion regions for random texture identification.

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(c), 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。
取扱いにあたっては、著作権侵害とされないよう十分にご注意ください。

299

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

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_d 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-b 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 to 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

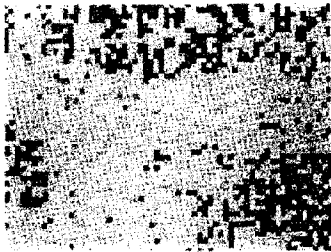


Fig. 16. The random texture regions of bridge

inherently superior in small regions where complexity is not such an 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-Loeve 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.

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) [26], 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。

VAISEY AND GUNSHO: IMAGE COMPRESSION

2351

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 [33].

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, munson, artistle, 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 coded 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



Fig. 17. The training images.

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_0 ensures that the dc coefficient will be fairly small, it is not 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.

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 $\{i_1, i_2, \dots, i_N\}$, 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 been 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-

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。

2002

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 49, NO. 8, AUGUST 2002

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. Or 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 re-

quires 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 $\hat{Q}$ from the coded image $\hat{Q}'$. 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 $\hat{Q}'$ 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 $\hat{Q}'$ 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 $\hat{Q}'$ 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 the 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 $\hat{Q}'$ according to their row number i and column number j . If we describe the pixels in $\hat{Q}'$ by f , those in the smoothed image after the k th stage g_k , and call the set of pixels to be smoothed in the k th stage Ψ_k , then the smoothing operation can be described as follows. For the first stage we have

$$g_1(i, j) = \begin{cases} f(i, j) & (i, j) \notin \Psi_1 \\ sf(i, j) & (i, j) \in \Psi_1 \end{cases} \quad (6)$$

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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

VAISEY AND GERSHO, IMAGE COMPRESSION

2053

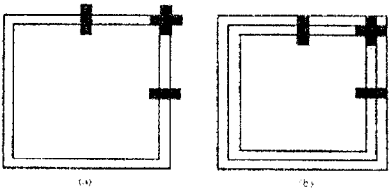


Fig. 18. The two stages of smoothing: (a) First stage; (b) Second stage.

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

$$s(i, j) = \frac{f(i, j) + f(i - 1, j) + f(i + 1, j) + f(i, j - 1) + f(i, j + 1)}{4} \quad (9)$$

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)

$$\text{PSNR} = 10 \log \left[255^2 / \frac{1}{K} \sum_{i=1}^K (c_i - c_i')^2 \right] \quad \text{dB} \quad (10)$$

where a_i and c_i are the i th pixels from Ω and Ω' (or Ω_1), 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_1 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_1 , which in turn allows us to form Ω_2 . 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

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。
取扱いにあたっては、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_0 is computed in the early stages of the quantizer and many problems are easily repaired in the subsequent stages. Recall that T_0 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_0 and Ω_1 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_1 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 2256 for the 32×32 , 16×16 , and 8×8 blocks, respectively. We also adjusted T_1 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_1 . 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			42	50
8×8		90	47	65
4×4	35	53	76	102

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	246	215	85
8×8	1 107	1 641	1 546
4×4	4 344	6 125	8 824

TABLE IV
COMPARING THE MEAN AND INTERPOLATED IMAGES

Image Name	Bit Rate (b/p)	PSNR (dB)	
		Mean Image	Interp. Image
Lenna	0.079	24.59	25.44
Barbara	0.098	22.38	22.73
Bridge	0.127	21.67	22.25
F16-jet	0.075	25.71	24.32

TABLE V
BIT COUNTS FOR A 64x64 SUB-INTERPOLATION

Parameter	Image Name		
	Lenna	Barbara	Bridge
Mean tree info. (bits)	3 256	3 948	4 716
Total mean info. (bits)	79 377	22 475	33 264
Classification info. (bits)	4 970/4 500	7 370/7 331	12 693
Texture tree info. (bits)	491/498	595/605	7 692
VE tree info. (bits)	2 900/3 036	7 708/3 836	5 196

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+3	2+3
2	7+15	4+6	4+10
3	16+28	7+15	
4	19+28	16+28	
5	46+68		
6	67+91		

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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

VAISEY AND GERSHO, IMAGE COMPRESSION

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

Vector #	Block Size					
	32 × 32		16 × 16		8 × 8	
	LR	HR	LR	HR	LR	HR
0	4	4	4	8	5	1
1	32	128	32	128	8	32
2	32	128	32	128	16	96
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 Sabier-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 lev-

TABLE VIII
THE MEAN AND GAIN QUANTIZERS
FOR THE HR CODER. LK
QUANTIZES THE HIGH-VE REGIONS

Output Levels	
Mean	Gain
2	65.9
5	37.9
6	6.2
11	169.4
15	133.9
21	159.1
26	197.8
41	355.9

els 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_0) and thus only the positive output levels are shown.

B. *The VE Segmentation*

Once Ω_0 has been subtracted, a VE segmentation on Ω_R is performed. As described in the body of the paper, the segmentation thresholds T_{15} and T_4 , and T_1 are set equal to 60, 60, and 255 for both coders, while (as explained in Section VI-A2) T_{12} 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_{66} , 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 REGIONS IN EACH TEST AS DETERMINED BY THE
VE SEGMENTATION

Size of Region	Image Name		
	Lenna	Barbara	Bridge
32×32	80 483	413 204	4
16×16	449 320	702 285	65
8×8	577 595	185 989	173
$4 \times 4_{\text{L}}$	5809 1503	2655 4066	4785
$4 \times 4_{\text{H}}$	2287 1292	2413 4322	5475
$4 \times 4_{\text{M}}$	266	330	3208

TABLE X
THE PSNR'S FOR TEST IMAGES

Parameter	Size of Region	Image Name					
		Lenna		Barbara		Bridge	
		Low Rate	High Rate	Low Rate	High Rate	Low Rate	High Rate
PSNR	32×32	33.73	34.99	33.64	35.13	N/A	N/A
by class	16×16	33.21	33.76	33.90	34.82	32.06	32.32
3B1	8×8	32.90	33.35	33.48	33.96	32.20	32.73
	$4 \times 4_{\text{L}}$	25.39	27.68	22.33	23.91	24.27	25.90
	$4 \times 4_{\text{H}}$	29.13	30.79	28.87	29.90	28.76	29.60
	$4 \times 4_{\text{M}}$	23.46	23.46	22.48	22.48	22.72	22.72
PSNR ₁ [dB]		30.26	31.36	28.50	29.82	25.60	26.45
PSNR ₂ [dB]		30.51	31.79	28.65	29.83	26.83	28.20
PSNR ₃ [dB]		30.47	31.62	28.57	29.89	25.73	26.46
PSNR ₄ [dB]		30.80	32.08	28.70	28.11	26.94	28.23
rate [b/p]		0.277	0.383	0.389	0.534	0.501	0.650



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. 10(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。

VAITEY AND GERSHGO, IMAGE COMPRESSION

205



(a)



(b)



(c)

Fig. 21. The coded versions of Bridge: (a) The HR case—0.650 b/p; (b) The LR case—0.513 b/p.



(a)

Fig. 22. The coded versions of Barbara: (a) The HR case—0.534 b/p; (b) The LR case—0.369 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 an other 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 decreased 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 0.097 b/p in the HR case and 0.026 b/p in the LR case. The PSNR drops by 0.5 and 1.1 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]

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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

1058

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

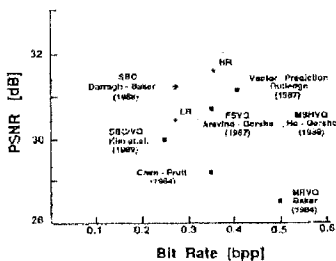


Fig. 23. A performance comparison (Lena).

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 Lena 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 Lena. Actually, we have used the green band from the original USC color image of Lena, 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 LR and LR results for Lena 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 [11], 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 30.93 (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



(a)



(b)

Fig. 24. The results of Ho and Aravind. (a) Ho—0.36 b/p; (b) Aravind—0.35 b/p.

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。
取扱いにあたっては、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 asynchronous and color images," *IEEE Trans. Commun.*, vol. COM-25, no. 11, pp. 1283-1291, 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. 117-126, Mar. 1982.

[4] N. Nishabadi and R. Kline, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, pp. 857-871, Aug. 1988.

[5] J. Woods and S. O'Neill, "Subband coding of images," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1278-1286, Oct. 1986.

[6] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. COM-36, pp. 1165-1173, Nov. 1988.

[7] E. Rucko, T. Lookabaugh, P. Chou, and R. Gray, "Variable rate vector quantization for medical image compression," *IEEE Trans. Med. Imaging*, vol. 9, pp. 290-298, Sep. 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 structures and vector quantization," in *Proc. EUSIPCO-88* (Nice, France), Sept. 1988, pp. 1133-1136.

[10] R. Baker, "Vector quantization of digital images," Ph.D. dissertation, Engin. Syst. Lab., Stanford University, June 1984.

[11] K. Argyrakis and A. Gersho, "Image compression based on vector quantization with finite memory," *Dis. Acc.*, vol. 26, no. 7, pp. 570-580, July 1987.

[12] JPEG, *Draft Technical Specification*, revision 8, Aug. 14, 1990.

[13] N. Nishabadi, "Use of vector quantizers in image coding," in *Proc. ICASSP-85*, Mar. 1985, pp. 125-128.

[14] C. Chen, "Adaptive transform coding via quadtree-based variable block-size 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. 293-297.

[16] R. Lencowski and M. Kani, "Adaptive split and merge for image analysis and coding," in *Proc. APIC Conf. on Visual Comm. Image Processing* (Cannes, France), Dec. 1985, pp. 2-9.

[17] P. Fardelle and A. Jain, "Quadtree-based two-stage image coding," in *Proc. SPIE Conf.*, 1986, pp. 492-1-492-5.

[18] C. Boncelet, J. Corbio, and A. Mosen, "Joint size compression of medical X-ray images," *Proc. SPIE Int. Soc. Opt. Eng.*, pp. 269-276, 1988.

[19] P. Strohach, "Image coding based on quadtree structured recursive least-squares approximation," in *Proc. ICASSP-89*, 1989, pp. 1961-1964.

[20] N. M. Nishabadi, S. L. Liu, and Y. Peng, "Interframe hierarchical vector quantization," *Opt. Eng.*, vol. 28, pp. 17-25, July 1989.

[21] V. Rother and M. Reutter, "Hierarchical picture coding using quadtree decomposition," in *Proc. 4th Int. Conf. Comp. Opt. Electronics*, Appl. Sci. Eng. (The Hague, The Netherlands), Apr. 1987.

[22] V. Buijter, "Etude d'une methode hiérarchique pour le codage à faible débit des images numériques," INRS Information, Rep. 84-04, May 1986.

[23] H. Sahet, "The quad tree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 168-260, June 1983.

[24] M. Kunt, A. Konomopoulos, and M. Kacner, "Second generation image coding techniques," *Proc. IEEE*, vol. 73, no. 4, pp. 549-574, Apr. 1985.

[25] M. Kunt, M. Bédard, and R. Leonard, "Recent results in high compression image coding," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 11, pp. 1336-1336, Nov. 1987.

[26] J. Clark, M. Paley, and P. Lawlor, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-23, no. 4, Oct. 1975.

[27] S. Dudge, T. Brockham, D. Chao, and R. Christensen, "Vector quantization of color digital images within a human visual model," in *Proc. ICASSP-88*, Apr. 1988, pp. 316-319.

[28] R. Noll, "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. Prasad, "Adaptive quantization of picture signals using spatial masking," *Proc. IEEE*, vol. 65, pp. 536-548, Apr. 1977.

[30] J. Olzak and J. Thomas, "Seizure spatial patterns," in *Handbook of Perception and Human Performance*, J. P. Thomas, Ed., New York: Wiley, 1986, p. 2.1.

[31] W. Chen and W. Pratt, "Scene adaptive coding," *IEEE Trans. Commun.*, vol. COM-32, pp. 225-232, Mar. 1984.

[32] J. I. Gilbert, "Use of activity classes in adaptive transform image coding," *IEEE Trans. Commun.*, vol. COM-33, pp. 758-760, July 1985.

[33] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[34] H. Dehn and H. Ellis, "Modeling and segmentation of noisy and textured images using filter-regularity fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 59-55, Jan. 1987.

[35] Y. Stetsko and R. Chellappa, "Structural and textural segmentation for MAP texture segmentation," in *Proc. ICASSP-88*, 1988, pp. 1120-1122.

[36] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. Commun.*, vol. COM-36, pp. 80-95, Jan. 1988.

[37] K. Zeger, J. Vaisey, and A. Gersho, "Locally optimal vector quantizer design via stochastic relaxation," *IEEE Trans. Signal Processing*, vol. 30, no. 2, pp. 310-322, Feb. 1982.

[38] J. Vaisey and A. Gersho, "Simulated annealing and lookbook de sign," in *Proc. ICASSP-88* (New York, New York), Apr. 1988, pp. 1176-1179.

[39] T. Otsuchi, Y. Takasama, and H. Okada, "DCT-VQ coding scheme using categorization with adaptive code book," in *Proc. Picture Coding Symp.*, 1989, June 1989, pp. 161-162.

[40] M. Gratz, L. Engelhardt, and R. Mclane, "Coding of arbitrarily shaped

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。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

1360

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 40, NO. 8, AUGUST 1992

image segments based on a generalized orthogonal transformation," *Signal Processing*, no. 1, pp. 143-180, 1988.

[141] M. Sabin and R. Gray, "Product-coding vector quantizers for waveform and voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 474-488, June 1984.

[142] J. Darragh and R. Baker, "Fixed distortion, variable rate subband coding of images," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 1001, pp. 979-990, Dec. 1988.

[143] C. S. Kim, M. J. E. Smith, and R. M. Mersereau, "An improved SBC-VQ scheme for color image coding," in *Proc. ICASSP 89*, 1989, pp. 1941-1944.

[144] C. Rutledge, "Variable block vector DPCM: Vector predictive coding of color images," in *Proc. ICU '87*, June 1987, pp. 5.2.1-5.2.5.

[145] R. Atiyeh, "Image compression based on orthogonal vector quantization," Ph.D. dissertation, Univ. California, Santa Barbara, Jan. 1988.

[146] Y. Hu and A. Gersho, "Variable-rate multistage vector quantization for image coding," in *Proc. ICASSP'88*, Apr. 1988, pp. 1156-1159.

[147] H. Lee and D. Liss, "A gain-type vector quantizer for image coding," in *Proc. ICASSP 88*, 1988, pp. 111-114.

Jacques Valney (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'64-SM'78-F'82) received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree from Cornell University, Ithaca, N.Y., in 1963.

He was at BELL Laboratories from 1963 to 1982. 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 two-dimensional works. He holds patents on speech coding, quantization, adaptive quantization, 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 1989, and as a Member of the Communication Theory Technical Committee and the Signal Processing and Communication's 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 1989, he was awarded the Guillermo-Cauer Once-Award from the Circuits and Systems Society. He received the Donald McClennan Meritorious Service Award from the IEEE Communications Society in 1983, and in 1984 he was awarded an IEEE Centennial Medal.

VARIABLE BLOCK-SIZE IMAGE CODING†

D. Jacques Vaisey and Allen Gersho

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

ABSTRACT

Current algorithms for the efficient coding of images generally employ one of two techniques: Vector Quantization or Transform Coding. Both of these methods are based on block coding, and the traditional algorithms require the partitioning of the original image into a number of, usually square, regions of uniform size. This paper explores techniques where the size of these regions, or blocks, is varied according to the local detail of the image. By isolating regions of differing detail, using a quad-tree based segmentation algorithm to control the size of the blocks, the low detail areas can be encoded at substantially lower rates than is otherwise possible. Satisfactory quality coding results are achieved at rates between 0.35 and 0.4 *bpp*; the actual rate depends on the image being encoded.

INTRODUCTION

Most natural images can be segmented into regions of high and low detail. Since high detail regions are intrinsically less compressible than regions of low detail, it is very attractive to implement a variable-rate image coding scheme that varies the number of bits used per unit area according to the local detail. Given a certain number of bits to describe an image, there will be some total amount of perceivable degradation distributed over the reconstructed image. The objective of a variable-rate scheme is to try and spread this perceived distortion uniformly over the entire coded image, resulting in a very "thin layer" of degradation per unit area. In contrast, fixed-rate schemes encode high detail regions with much greater distortion than low detail regions, implying a waste of bits. This difference is especially true at low rates where we may see large and striking degradations in some areas but no degradation at all in others. A variable-rate procedure is absolutely essential in order to achieve satisfactory quality compression at very low rates.

Traditional coding schemes such as Transform Coding (TC) and Vector Quantization (VQ) break an image into blocks prior to the actual encoding. It is possible to classify these blocks according to detail; however, the block-size in VQ is usually too small to take advantage of the fact that large areas of the image may consist of low detail. The block-size in TC is usually large enough to be useful for this purpose, however, when the block size is kept constant, as is the case with most adaptive TC schemes, it is impossible to do an adequate job of isolating the high detail regions. The solution to this problem is to segment an image into blocks of varying size. Large blocks are used in large regions of low detail, and small blocks when the detail gets high.

In our work, we use a quad-tree approach to structure a partition of the image so that regions of different size can be addressed

using a small amount of side-information. In this paper, each region is described simultaneously by two quad-trees: the *detail tree* corresponding to local detail and the *mean tree* describing the local average of the pixel amplitudes in a subblock. The encoding algorithm works by first removing the mean from each terminal node, or leaf, of the mean tree. The residual image is then encoded using either VQ or a hybrid TC-VQ scheme, according to the block's position in the detail tree.

This paper initiates a major departure from the approach taken in previous work on image coding with Vector Quantization. Virtually all prior studies, for example [1], have used a fixed VQ block size, usually based on memoryless coding of each block. The use of interblock memory was used in a limited way by Baker and Gray [2] in differential encoding of the means of successive blocks and extensive use of interblock memory has recently been included in VQ based image coding using finite-state techniques [3]. In all of this work the basic block size was fixed and had rather small values, usually 4×4 or 5×5.

QUAD-TREES

A quad-tree is a data structure that can be used to efficiently address regions of different size in an image. This technique can thus be applied to problems where we want to segment images into regions that are homogeneous in some property. A paper by Samet [4] gives a good tutorial on the nature and uses of quad-trees; however, only the basic ideas will be needed here. Also, Farrelle and Jain [5] have recently introduced an approach to image coding that uses the quad-tree as an adjunct to an interpolation technique. Other methods of image segmentation such as Region Growing [6] do better jobs of isolating regions of constant pixel statistics; however, in these techniques the shape of the regions is determined solely by the image being examined and this implies that an excessive number of bits are needed to describe the shape and location of each region. This large amount of overhead is unacceptable in applications where we are trying to code an image efficiently since there are not enough bits left over to encode detailed regions with good fidelity. With structures such as quad-trees, a small overhead rate is achieved by decomposing the data into subblocks whose size, shape and location are predetermined.

The quad-tree is a tree structure in which each node, unless it is a leaf, generates four children. Each node corresponds to a sub-block of the image that is determined, both in size and location, by its position in the tree. Each child represents a quarter the area of its parent. When using this structure to segment an image we start at some initial node level, and travel through the tree. At each node a test is performed to see if the subblock represented by the node is homogeneous in the property of interest. If the test is positive

† This work was supported by the National Science Foundation.

25.1.1

CH2396-0/87/0000-1051 \$1.00 © 1987 IEEE 1051

Authorized licensed use limited to: InterDigital. Downloaded on August 25, 2023 at 07:37:33 UTC from IEEE Xplore. Restrictions apply.

then the node becomes a leaf, otherwise the children of the node are generated and examined in turn. An illustration of a variable sized partition based on a quad-tree is shown in Fig. 1, where in this case the objective is to isolate regions containing edges.

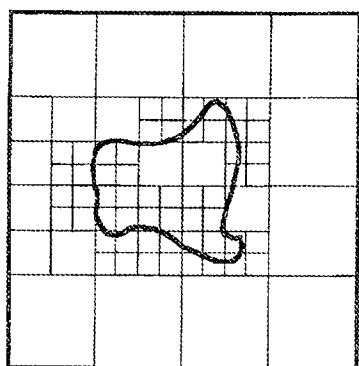


Fig. 1: A Sample Quad-Tree Segmentation

MEAN SUBTRACTION

The behavior of the local average pixel amplitude in different regions of an image is very important to compression algorithms. If an image is partitioned into a regular array of uniform sized cells, and if each region contains only a relatively small number of pixels, then the pixel means between neighboring regions will in general be highly correlated. This high correlation makes it possible to encode the means at a low bit rate as in [2]. We thus use a product-code structure to divide the encoding into two parts: an encoding of the mean and an encoding of the residual image after the mean has been removed.

The relative advantage in performing a mean removal (MR) operation depends on the method that will be used to code the residual image. This gain is fairly small in TC schemes where the region size is large and the correlation between region means of adjacent blocks is low. On the other hand, space-domain VQ uses small sized regions, avoiding astronomic codebook sizes, and the correlation between block means can be efficiently exploited. Since image blocks can have an enormous variety of pattern distributions, large codebooks are needed if all patterns are to be represented accurately. However, many of these vectors will have ac-components that are almost identical. A product-code structure thus allows us either to increase the range of patterns that can be described with a given codebook size or to significantly reduce the size of the codebook needed for a given pattern-matching capability.

An algorithm for subtracting the mean can be developed using the quad-tree approach discussed previously. The idea is to develop a variable-rate scheme where more bits are used to encode the local mean in the areas of the image where the mean evolves quickly. The first step in the development is to choose the starting block-size, which specifies the root of the quad-tree. We chose to start with a partition of the image into 32×32 blocks, since we found that the local mean within in a larger block is almost always inhomogeneous.

The MR algorithm starts by calculating the pixel mean for each 32×32 block. The corresponding quantized mean for each block is then subtracted from each pixel in that block to form the *residual image*. At this stage a decision to stop or descend further into the tree is made. If this decision is positive, then four children

are generated from the appropriate parent block. Means are then calculated, quantized and subtracted to form the next level of residuals. The procedure continues until we get to the 4×4 level which is the deepest level allowed in our tree. The information that must be stored or transmitted as a result of this procedure consists of the quantized means and decision-data which specifies the complete tree structure.

The decision to proceed to the next level of the tree is made by examining the ramifications of declaring the current node a leaf. It is desirable to terminate the current tree branch when all of the means that would otherwise be subtracted at the next levels are quite small, implying that most of the mean information has already been removed. Consequently, we establish a threshold value for each level, and the decision to terminate is made if the absolute values of the means at the next levels are all below their thresholds.

A separate quantizer is used at each level in the tree in order to quantize the means according to their individual statistics. Uniform quantization, with a suitable amplitude range, is used throughout, since the histograms of the mean values indicate a fairly uniform distribution. When choosing the ranges of the quantizers, it must be kept in mind that large errors are easily noticeable in coded images. It is thus important to minimize the larger quantization errors, even at the expense of an increase in the average MSE. The bit assignments and ranges used to generate the results reported in this paper are shown in Table 1.

blk size	# of bits	max value	min value	threshold
32×32	4	255	0	NA
16×16	3	79	-80	32
8×8	2	89	-90	32
4×4	3	99	-100	40

Table 1: Properties of the Mean Quantizers

The thresholds must be chosen judiciously since they have a large effect on the bit rate needed for the mean information.

DETAIL SEGMENTATION

The detail segmentation algorithm operates on the residual image after the means have been processed. A key issue here is the method employed to assess the amount of detail in a subblock. After experimenting with several sophisticated detail indicators, we found that the variance of the mean-removed block is a simple and adequate indicator of detail. In effect, the mean removal acts as a high pass filter and, roughly speaking, the variance measures the energy level in the high pass filter output as the indicator of detail. The segmentation proceeds in much the same way as in the MR algorithm, except that the decisions to terminate or to go further into the tree are based upon a calculation of the block variance. Also, since this is the last stage of processing, it is not necessary to look further down the tree before making the decision. After examining many different kinds of blocks and their variances the decision threshold was set to 80.

Once the residual image is segmented it must be coded. However, in order to increase efficiency, it is desirable to vary the type of encoding algorithm for the different block sizes. For the three largest sizes we chose to use a hybrid of TC and VQ, while a pure space-domain VQ algorithm was selected for the 4x4 sub-blocks. VQ is an appropriate technique for the small blocks since these blocks tend to have high detail content and the space-domain allows us to give higher weight to the matching of perceptually important patterns. However, direct spatial domain VQ is a poor choice for the larger blocks due to the exponential growth in com-

putation for VQ encoding with the vector dimension. On the other hand, TC is well suited for encoding larger sized blocks; this is especially true in our case since these blocks all have a low detail character and there is no need for special perceptual weighting.

THE TRANSFORM CODER

As mentioned previously, the 32×32 , the 16×16 and the 8×8 blocks are both encoded using a TC algorithm. In this algorithm the TC coefficients of each block are grouped into vectors which are then individually coded using VQ. This grouping is done by requiring that all components of each vector have approximately the same perceptual importance. The codebook designs were then done using the Generalized Lloyd clustering algorithm [7] with the initial codebooks chosen from a uniform sampling of the training sets. We decided to use VQ since it is an effective way to exploit the correlation between the various TC coefficients and also allows us to use fractional numbers of bits per coefficient. The number of bits used per block was 34, 24 and 11 for the 32×32 , 16×16 and 8×8 cases respectively; a diagram showing the decomposition into vectors for the 16×16 case is shown in Fig. 2, where the coefficients that are not encoded are not displayed. The total number of bits used to encode these vectors is quite low and as a result some degree of "blocking" appears in the coded image. However, this effect can be easily removed by selectively smoothing the low detail regions of the reconstructed image.

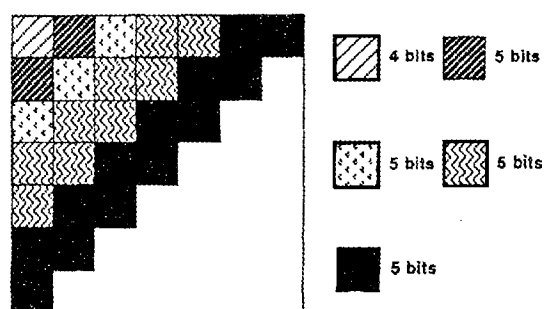


Fig. 2: The Decomposition of the 16×16 TC Coefficients into Vectors

THE SPACE-DOMAIN VECTOR QUANTIZER

PP As mentioned earlier, the high detail 4×4 vectors are encoded using space-domain VQ. The version of VQ that is used here is an unclassified memoryless procedure [8]; however, improvements have been made through the use of both the MR algorithm and a new technique, described below, that allows us to improve the quality of the high detail subblocks.

One of the main problems in VQ is the design of universal codebooks that can be effective in encoding virtually any test image. The reason that these codebooks are difficult to design is that the properties of different images vary greatly, especially in the types of edges that are present. Perceptual quality is heavily dependent on how well edges are reproduced and a good coder must weight any loss in edge integrity higher than other kinds of degradations.

In standard VQ the codebooks are designed using the Generalized Lloyd clustering algorithm based on the MSE criterion. Codevectors are formed by taking centroids of clusters of training vectors with the final result being heavily dependent on the composition of the training set and the choice of the initial codebook. The training set is usually chosen from a set of actual images that represent all of the types of images that we would like to encode.

We selected a training set of containing 1024 32×32 vectors taken from four different images. These images consist of: a building, a satellite photograph of an airbase, a woman's face and a landscape. Many more training images would be needed if we did not employ the MR algorithm.

The standard way to choose the initial codebook is to simply do a random, or perhaps uniform, sampling of the training set. However, this technique tends to weight edge and non-edge vectors similarly since both will be represented according to their proportions in the original image. Substantially better perceptual quality can be obtained with codebooks that are designed using initial codebooks having a higher percentage of edge vectors than is found in the training set [8]. In our work we force half of the vectors in the initial codebook to contain edges; this is opposed to the naturally occurring level of 15-20%.

It may still be difficult to code many subblocks of some images even though the codebook has been carefully designed and contains a healthy proportion of edges. The reason for this problem is that the image to be encoded may contain high detail subblocks 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 high detail, and that they are perceptually most critical when they contain edge segments. A solution can be found to this problem by generating and sending side-information that partially describes some specific attributes of the edge areas that are unique to the image being encoded. As long as the bit rate of this information is kept low we will still be able to achieve a high compression factor. The VQ codebook provides a natural way to generate this side information. To our knowledge, no other technique has an equivalent ability to represent the high detail data as side-information.

The essential idea is to generate a small supplemental codebook after an initial pass through the image to be encoded. This codebook is then attached to the main codebook during the second pass, where the actual encoding takes place, and its specification sent to the receiver as part of the data stream. One version of this approach was studied by Goldberg and Sun [9], where the codebook of a image sequence encoder is temporally updated to account for changing statistics. We form a training set for this supplemental codebook by isolating the highest detail 4×4 vectors through a block variance threshold of 1400. However, before doing the codebook design, we remove those vectors not classified as edges from the training set. This elimination of non-edge vectors prevents the supplemental codebook from being dominated by vectors that correspond to certain high detail texture or random patterns that occur in many natural images. Distortion is not easily noticed in such areas and there is no reason to waste precious side-information in improving the quality here.

In order to keep the bit rate low the supplemental codebook must be kept small. It is also important to remember that, assuming that the entire codebook size is kept constant, the supplemental vectors replace vectors that could be in the main codebook. We have chosen a supplemental codebook containing 32 codevectors. These vectors are encoded as side-information by quantizing each pixel in each vector with a six-bit uniform quantizer. It is possible to encode this data more efficiently, but the potential gain in the overall compression factor is small. The main codebook is designed to have 224 vectors, giving a total codebook size of 256, allowing the index of each codevector to be transmitted using exactly eight bits. With scheme for generating side information we must transmit

$\beta = (32)(16)(6) / \{(512)(512)\} = 0.012 \text{ bpp}$
for each 512x512 image.

RESULTS

We now discuss the performance of the various algorithms described in the previous sections.

The MR algorithm efficiently removes a great deal of highly correlated information from an image. An indication of this removal is seen when we examine the intensity histograms of the residual images after mean-subtraction. It is found that the histograms are almost identical even though those of the originals are wildly different. As a final evaluation of the MR algorithm we present Fig. 3, where only the mean-tree information obtained by the MR algorithm has been used to generate a coded image at a bit rate of 0.05 *bpp*.



Fig. 3: "Womanhat" Reconstructed From the Mean Information (rate=0.05 *bpp*)

It is informative to observe what percentages of the quad-tree nodes terminate at what level. This information is given for two examples in Table 2.

seg.	dimension	percentage of image	
		"F16 jet"	"womanhat"
mean	32x32	48	32
	16x16	20	37
	8x8	20	22
	4x4	12	9
detail	32x32	34	17
	16x16	13	20
	8x8	13	20
	4x4	40	43

Table 2: Percentage of Images Segmented with the Different Block Sizes

Using the methods explained earlier, we can encode the residual data at rates around 0.25 to 0.35 *bpp*, with the actual rate depending on the image. The coded 512x512 images have good SNR's with final rates generally between 0.3 and 0.4 *bpp*; the SNR for "womanhat" was 29.8 dB, and that of "F16 jet" 30.3 dB. The definition of the SNR that was used is:

$$SNR = 10 \log \left[255^2 / \sum_{i=1}^{N \times N} (x_i - c_i)^2 \right] ,$$

with x_i and c_i are the i_{th} pixels from the original and coded images

respectively and N is the side-dimension of the image. The fully coded "womanhat" image is shown in Fig. 4, where the total rate is 0.35 *bpp*; the rate for "F16-jet" was 0.32 *bpp*. Edges are reproduced much better than in standard fixed-block size VQ at 0.5 *bpp* and only small distortions are noticed in the shade regions.



Fig. 4: The Fully Coded "Womanhat" (rate=0.35 *bpp*)

The supplemental codebook has been found to substantially improve the perceived quality in the high-detail regions when compared with the fixed codebook results; the SNR has also been improved by 0.5 dB. It is recommended that this technique, or some variation of it, be included in all VQ schemes.

CONCLUSIONS

A new approach to image coding has been introduced based on a quad-tree partitioning of image blocks. New algorithms involved in this approach have been presented and evaluated. These algorithms include: an efficient mean-subtraction procedure, a hybrid TC/VQ encoding procedure with a variable block-size and a method to generate perceptually important side-information about the details that are unique to each image. These algorithms use the idea of quad-trees in order to segment the image into regions with different properties. The new techniques were found to give good encoding results at rates around 0.33 and 0.4 *bpp*.

REFERENCES

- [1] B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images," *IEEE Trans. Commun.*, vol. COM-34, pp. 1105-1115, Nov., 1986.
- [2] R. Baker and R. Gray, "Differential Vector Quantization of Achromatic Imagery," *Proc. Picture Coding Symp.*, pp. 105-106, March 1983.
- [3] R. Aravind and A. Gersho, "Low-Rate Image Coding with Finite-State Vector Quantization," *Proc. ICASSP*, pp. 137-140, April 8-11, 1986.
- [4] H. Sauet, "The Quad Tree and Related Hierarchical Data Structures," *ACM Computing Surveys*, Vol.16, No.2, June 1984, pp. 188-260.
- [5] P. Farrellle and A. Jain, "Quad-Tree Based Two Source Image Coding," *Proc. MILCOM*, pp. 49.2.1-49.2.5, 1986.
- [6] M. Kunt et.al., "Second Generation Image Coding Techniques," *Proc. IEEE*, Vol.73, No.4, April 1985, pp. 549-574.
- [7] Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun. Technol.*, vol. COM-28, pp. 84-95, Jan. 1980.
- [8] A. Gersho and B. Ramamurthi, "Image Coding Using Vector Quantization," *Proc. ICASSP*, pp. 428-431, May 3-5, 1982.
- [9] M. Goldberg and H. Sun, "Image Sequence Coding Using Vector Quantization," *IEEE Trans. Commun.*, vol. COM-34, pp. 703-710, July 1986.

The Quadtree and Related Hierarchical Data Structures

HANAN SAMET

Computer Science Department, University of Maryland, College Park, Maryland 20742

A tutorial survey is presented of the quadtree and related hierarchical data structures. They are based on the principle of recursive decomposition. The emphasis is on the representation of data used in applications in image processing, computer graphics, geographic information systems, and robotics. There is a greater emphasis on region data (i.e., two-dimensional shapes) and to a lesser extent on point, curvilinear, and three-dimensional data. A number of operations in which such data structures find use are examined in greater detail.

Categories and Subject Descriptors: E.1 [Data]: Data Structures—trees; H.3.2 [Information Storage and Retrieval]: Information Storage—file organization; I.2.1 [Artificial Intelligence]: Applications and Expert Systems—cartography; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—representations, data structures, and transforms; I.3.3 [Computer Graphics]: Picture/Image Generation—display algorithms; viewing algorithms; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—curve, surface, solid, and object representations; geometric algorithms, languages, and systems; I.4.2 [Image Processing]: Compression (Coding)—approximate methods; exact coding; I.4.7 [Image Processing]: Feature Measurement—moments; projections; size and shape; J.6 [Computer-Aided Engineering]: Computer-Aided Design (CAD)

General Terms: Algorithms

Additional Key Words and Phrases: Geographic information systems, hierarchical data structures, image databases, multiattribute data, multidimensional data structures, octrees, pattern recognition, point data, quadtrees, robotics

INTRODUCTION

Hierarchical data structures are becoming increasingly important representation techniques in the domains of computer graphics, image processing, computational geometry, geographic information systems, and robotics. They are based on the principle of recursive decomposition (similar to *divide and conquer* methods [Aho et al. 1974]). One such data structure is the quadtree. As we shall see, the term *quadtree* has taken on a generic meaning. In this survey it is our goal to show how a number of data

structures used in different domains are related to each other and to quadtrees. This presentation concentrates on these different representations and illustrates how a number of basic operations that use them are performed.

Hierarchical data structures are useful because of their ability to focus on the interesting subsets of the data. This focusing results in an efficient representation and improved execution times and is thus particularly useful for performing set operations. Many of the operations that we describe can often be performed equally as

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1984 ACM 0360-0300/84/0000-0187 \$00.75.

Computing Surveys, Vol. 16, No. 2, June 1984

CONTENTS

INTRODUCTION

1. OVERVIEW OF QUADTREES

2. REGION DATA

- 2.1 Neighbor-Finding Techniques
- 2.2 Alternative Ways to Represent Quadrees
- 2.3 Conversion
- 2.4 Set Operations
- 2.5 Transformations
- 2.6 Areas and Moments
- 2.7 Connected Component Labeling
- 2.8 Perimeter
- 2.9 Component Counting
- 2.10 Space Requirements
- 2.11 Skeletons and Medial Axis Transforms
- 2.12 Pyramids
- 2.13 Quadtree Approximation Methods
- 2.14 Volume Data

3. POINT DATA

- 3.1 Point Quadrees and *k*-d Trees
- 3.2 Region-Based Qualities
- 3.3 Comparison of Point Quadrees and Region-Based Quadrees
- 3.4 CIP Quadrees
- 3.5 Bucket Methods

4. CURVILINEAR DATA

- 4.1 Strip Trees
- 4.2 Methods Based on a Regular Decomposition
- 4.3 Comparison

5. CONCLUSIONS

ACKNOWLEDGMENTS

REFERENCES

efficiently, or more so, with other data structures. However, hierarchical data structures are attractive because of their conceptual clarity and ease of implementation.

As an example of the type of problems to which the techniques described in this survey are applicable, consider a cartographic database consisting of a number of maps and some typical queries. The database contains a contour map, say at 50-foot elevation intervals, and a land use map classifying areas according to crop growth. Our wish is to determine all regions between 400- and 600-foot elevation levels where wheat is grown. This will require an intersection operation on the two maps. Such an analysis could be rather costly, depending on the way the data are represented. For example, areas where corn is grown are

of no interest, and we wish to spend a minimal amount of effort searching such regions. Yet, traditional region representations such as the boundary code [Freeman 1974] are very local in application, making it difficult to avoid examining a corn-growing area that meets the desired elevation criterion. In contrast, hierarchical methods such as the region quadtree are more global in nature and enable the elimination of larger areas from consideration. Another query might be to determine whether two roads intersect within a given area. We could check them point by point, but a more efficient method of analysis would be to represent them by a hierarchical sequence of enclosing rectangles and to discover whether in fact the rectangles do overlap. If they do not, then the search is terminated, but if an intersection is possible, then more work may have to be done, depending on which method of representation is used. A similar query can be constructed for point data—for example, to determine all cities within 50 miles of St. Louis that have a population in excess of 20,000 people. Again, we could check each city individually, but using a representation that decomposes the United States into square areas having sides of length 100 miles would mean that at most four squares need to be examined. Thus California and its adjacent states can be safely ignored. Finally, suppose that we wish to integrate our queries over a database containing many different types of data (e.g., points, lines, and areas). A typical query might be, "Find all cities with a population in excess of 5000 people in wheat-growing regions within 20 miles of the Mississippi River." In the remainder of this survey we shall present a number of different ways of representing data so that such queries and other operations can be efficiently processed.

The coverage and scope of the survey are focused on region data, and are concerned to a lesser extent with point, curvilinear, and three-dimensional data. Owing to space limitations, algorithms are presented only in a descriptive manner. Whenever possible, however, we have tried to motivate critical steps by a liberal use of examples. The concept of a pyramid is discussed only

briefly, and the reader is referred to the collection of papers edited by Rosenfeld [1983] for a more comprehensive exposition. Similarly, we discuss image compression and coding only in the context of hierarchical data structures. Results from computational geometry, although related to many of the topics covered in this survey, are only discussed briefly in the context of representations for curvilinear data. For more details on early results involving some of these and related topics, the interested reader may consult the surveys by Bentley and Friedman [1979], Edelsbrunner [1984], Nagy and Wagle [1979], Requicha [1980], Srihari [1981], Samet and Rosenfeld [1980], and Toussaint [1980]. Overmars [1983] has produced a particularly good treatment of point data. A broader view of the literature can be found in related bibliographies, for example, Edelsbrunner and van Leeuwen [1983] and Rosenfeld [1984]. Nevertheless, given the broad and rapidly expanding nature of the field, we are bound to have omitted significant concepts and references. In addition we at times devote a disproportionate amount of attention to some concepts at the expense of others. This is principally for expository purposes as we feel that it is better to understand some structures well rather than to give the reader a quick run-through of "buzz words." For these indiscretions, we beg your pardon.

1. OVERVIEW OF QUADTREES

The term *quadtree* is used to describe a class of hierarchical data structures whose common property is that they are based on the principle of recursive decomposition of space. They can be differentiated on the following bases: (1) the type of data that they are used to represent, (2) the principle guiding the decomposition process, and (3) the resolution (variable or not). Currently, they are used for point data, regions, curves, surfaces, and volumes. The decomposition may be into equal parts on each level (i.e., regular polygons and termed a *regular decomposition*), or it may be governed by the input. The resolution of the decomposition (i.e., the number of times that the decom-

position process is applied) may be fixed beforehand, or it may be governed by properties of the input data.

Our first example of quadtree representation of data is concerned with the representation of region data. The most studied quadtree approach to region representation, termed a *region quadtree*, is based on the successive subdivision of the image array into four equal-sized quadrants. If the array does not consist entirely of 1's or entirely of 0's (i.e., the region does not cover the entire array), it is then subdivided into quadrants, subquadrants, etc. until blocks are obtained (possibly single pixels) that consist entirely of 1's or entirely of 0's; that is, each block is entirely contained in the region or entirely disjoint from it. Thus the region quadtree can be characterized as a variable resolution data structure. For example, consider the region shown in Figure 1a, which is represented by the 2^3 by 2^3 binary array in Figure 1b. Observe that the 1's correspond to picture elements (termed *pixels*) that are in the region and the 0's correspond to picture elements that are outside the region. The resulting blocks for the array of Figure 1b are shown in Figure 1c. This process is represented by a tree of degree 4 (i.e., each nonleaf node has four sons). The root node corresponds to the entire array. Each son of a node represents a quadrant (labeled in order NW, NE, SW, SE) of the region represented by that node. The leaf nodes of the tree correspond to those blocks for which no further subdivision is necessary. A leaf node is said to be BLACK or WHITE, depending on whether its corresponding block is entirely inside or entirely outside of the represented region. All nonleaf nodes are said to be GRAY. The quadtree representation for Figure 1c is shown in Figure 1d.

At this point it is appropriate to define a few terms. We use the term *image* to refer to the original array of pixels. If its elements are either BLACK or WHITE then it is said to be *binary*. If shades of gray are possible (i.e., gray levels), then the image is said to be a *gray-scale* image. In our discussion we are primarily concerned with binary images. The *border* of the image is the outer boundary of the square corresponding

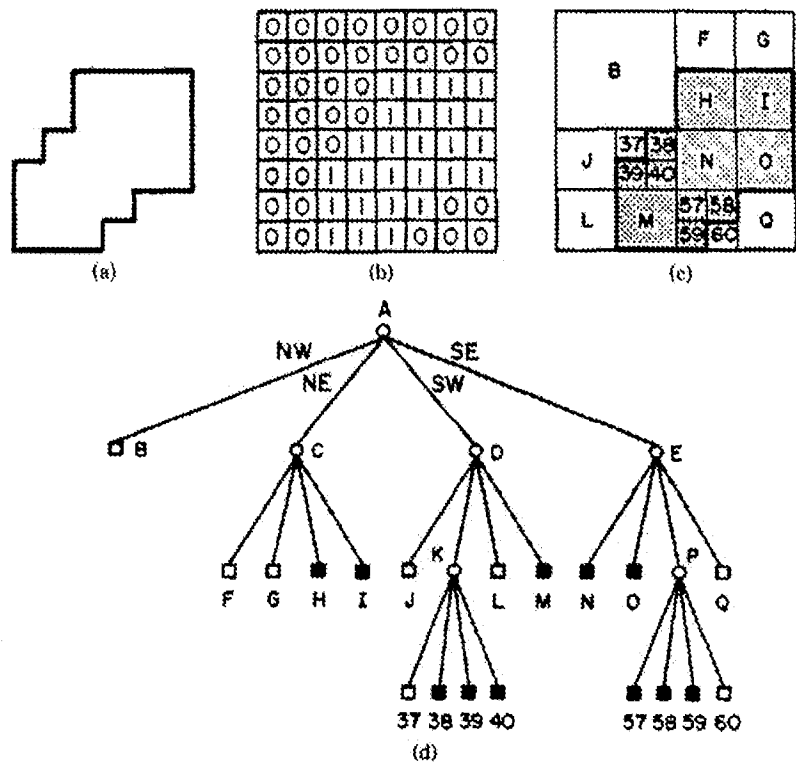


Figure 1. A region, its binary array, its maximal blocks, and the corresponding quadtree. (a) Region. (b) Binary array. (c) Block decomposition of the region in (a). Blocks in the region are shaded. (d) Quadtree representation of the blocks in (c).

to the array. Two pixels are said to be *4-adjacent* if they are adjacent to each other in the horizontal or vertical directions. If the concept of adjacency also includes adjacency at a corner (i.e., diagonal adjacencies), then the pixels are said to be *8-adjacent*. A **BLACK** region is a maximal *four-connected* set of **BLACK** pixels, that is, a set S such that for any pixels p, q , in S there exists a sequence of pixels $p = p_0, p_1, \dots, p_n = q$ in S such that p_{i+1} is 4-adjacent to p_i , $0 \leq i < n$. A **WHITE** region is a maximal *eight-connected* set of **WHITE** pixels, which is defined analogously. A pixel is said to have four edges, each of which is of unit length. The *boundary* of a **BLACK** region consists of the set of edges of its constituent pixels that also serve as edges of **WHITE** pixels. Similar definitions can be formulated in terms of blocks. For ex-

ample, two disjoint blocks, P and Q , are said to be *4-adjacent* if there exists a pixel p in P and a pixel q in Q such that p and q are 4-adjacent. Eight-adjacency for blocks is defined analogously.

Unfortunately, the term *quadtree* has taken on more than one meaning. The region quadtree, as shown above, is a partition of space into a set of squares whose sides are all a power of two long. This formulation is due to Klinger [1971; Klinger and Dyer 1976], who used the term *Q-tree*, whereas Hunter [1978] was the first to use the term *quadtree* in such a context. Actually, a more precise term would be *quadtrie*, as it is really a trie structure [Fredkin 1960] (i.e., a data item or key is treated as a sequence of characters, where each character has M possible values and a node at level i in the trie represents an M -

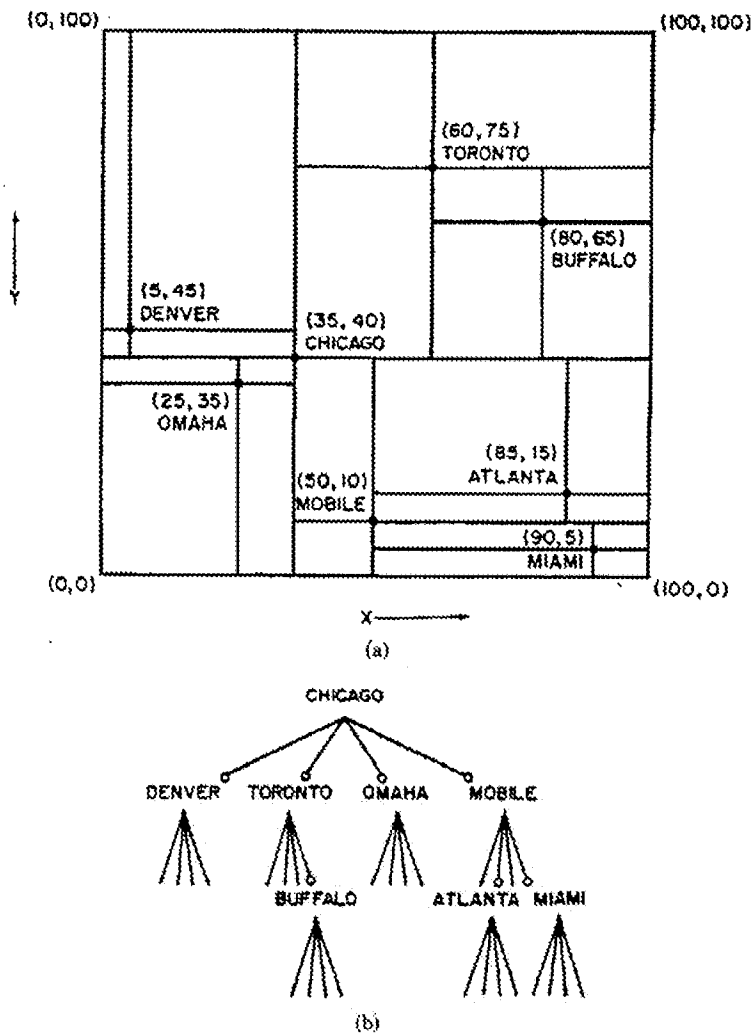


Figure 2. A point quadtree (b) and the records it represents (a).

way branch depending on the i th character). A similar partition of space into rectangular quadrants, also termed a quadtree, was used by Finkel and Bentley [1974]. It is an adaptation of the binary search tree [Knuth 1975] to two dimensions (which can be easily extended to an arbitrary number of dimensions). It is primarily used to represent multidimensional point data, and we shall refer to it as a *point quadtree* when confusion with a region quadtree is possible. As an example, consider the point

quadtree in Figure 2, which is built for the sequence Chicago, Mobile, Toronto, Buffalo, Denver, Omaha, Atlanta, and Miami.¹ Note that its shape is highly dependent on the order in which the points are added to it.

¹ We have taken liberty in the assignment of coordinates to city names so that the same example can be used throughout the text to illustrate a variety of concepts.

The origin of the principle of recursive decomposition upon which, as we have said, all quadtrees are based is difficult to ascertain. Below, in order to give some indication of the uses of the quadtree, we briefly, and incompletely, trace some of its applications to geometric data. Most likely it was first seen as a way of aggregating blocks of zeros in sparse matrices. Indeed, Hoare [1972] attributes a one-level decomposition of a matrix into square blocks to Dijkstra. Morton [1966] used it as a means of indexing into a geographic database. Warnock [1969; Sutherland et al. 1974] implemented a hidden surface elimination algorithm by using a recursive decomposition of the picture area. The picture area is repeatedly subdivided into successively smaller rectangles while a search is made for areas sufficiently simple to be displayed. The SRI robot project [Nilsson 1969] used a three-level decomposition of space to represent a map of the robot's world. Eastman [1970] observes that recursive decomposition might be used for space planning in an architectural context. He presents a simplified version of the SRI robot representation. A quadtree-like representation in the form of production rules called depth-first (DF)-expressions is discussed by Kawaguchi and Endo [1980] and Kawaguchi et al. [1980]. Tucker [1984a] uses quadtree refinement as a control strategy for an expert vision system.

Parallel to the above development of the quadtree data structure there has been related work by researchers in the field of image understanding. Kelly [1971] introduced the concept of a plan which is a small picture whose pixels represent gray-scale averages over 8 by 8 blocks of a larger picture. Needless effort in edge detection is avoided by first determining edges in the plan and then using these edges to search selectively for edges in the larger picture. Generalizations of this idea motivated the development of multiresolution image representations, for example, the recognition cone of Uhr [1972], the preprocessing cone of Riseman and Arbib [1977], and the pyramid of Tanimoto and Pavlidis [1975]. Of these representations, the pyramid is the closest relative of the region quadtree. A pyramid is an exponentially tapering stack

of arrays, each one-quarter the size of the previous array. It has been applied to the problems of feature detection and segmentation. In contrast, the region quadtree is a variable-resolution data structure.

In the remainder of this paper we discuss the use of the quadtree and other hierarchical data structures as they apply to region representation, and to a lesser extent, point data and curvilinear data. Section 2 deals with region representation. We are primarily concerned with two-dimensional binary regions and how basic operations common to computer graphics, image processing, and geographic information systems can be implemented when the underlying representation is a quadtree. Nevertheless, we do show how the quadtree can be extended to represent surfaces and volumes in three dimensions. A brief overview of pyramids and their applications is also presented. For more details, the reader is urged to consult Tanimoto and Klinger [1980] and Rosenfeld [1983]. In Section 3 we present various hierarchical representations of point data. Our attention is focused primarily on the point quadtree and its relative, the k -d tree. A more extensive discussion of point-space data structures can be found in the survey of Bentley and Friedman [1979]. In Section 4 we show how hierarchical data structures are used to handle curvilinear data. We demonstrate the way in which the region quadtree can be adapted to cope with such data and compare this adaptation with other hierarchical data structures.

2. REGION DATA

There are two major approaches to region representation: those that specify the boundaries of a region and those that organize the interior of a region. Owing to the inherent two-dimensionality of region information, our discussion focuses on the second approach.

The *region quadtree* (termed a quadtree in the rest of this section) is a member of a class of representations that are characterized as being a collection of maximal blocks that partition a given region. The simplest such representation is the run length code, where the blocks are restricted to 1 by m

rectangles [Rutovitz 1968]. A more general representation treats the region as a union of maximal square blocks (or blocks of any desired shape) that may possibly overlap. Usually, the blocks are specified by their centers and radii. This representation is called the *medial axis transformation* (MAT) [Blum 1967; Rosenfeld and Pfaltz 1966].

The quadtree is a variant on the maximal block representation. It requires that the blocks be disjoint and have standard sizes (i.e., sides of lengths that are powers of two) and standard locations. The motivation for its development was a desire to obtain a systematic way to represent homogeneous parts of an image. Thus, in order to transform the data into a quadtree, a criterion must be chosen for deciding that an image is homogeneous (i.e., uniform). One such criterion is that the standard deviation of its gray levels is below a given threshold t . By using this criterion the image array is successively subdivided into quadrants, subquadrants, etc. until homogeneous blocks are obtained. This process leads to a regular decomposition. If one associates with each leaf node the mean gray level of its block, the resulting quadtree then will completely specify a piecewise approximation to the image, where each homogeneous block is represented by its mean. The case where $t = 0$ (i.e., a block is not homogeneous unless its gray level is constant) is of particular interest, since it permits an exact reconstruction of the image from its quadtree.

Note that the blocks of the quadtree do not necessarily correspond to maximal homogeneous regions in the image. Most likely there exist unions of the blocks that are still homogeneous. To obtain a segmentation of the image into maximal homogeneous regions, we must allow merging of adjacent blocks (or unions of blocks) as long as the resulting region remains homogeneous. This is achieved by a "split and merge" algorithm [Horowitz and Pavlidis 1976]. However, the resulting partition will no longer be represented by a quadtree; instead, the final representation is in the form of an adjacency graph. Thus the quadtree is used as an initial step in the segmen-

tation process. For example, Figure 3b, c, and d demonstrate the results of the application, in sequence, of merging, splitting, and grouping to the initial image decomposition of Figure 3a. In this case, the image is initially decomposed into 16 equal-sized square blocks. Next, the "merge" step attempts to form larger blocks by recursively merging groups of four homogeneous "brothers" (e.g., the four blocks in the NW and SE quadrants of Figure 3b). The "split" step recursively decomposes blocks which are not homogeneous (e.g., the NE and SW quadrants of Figure 3c). Finally, the "grouping" step aggregates all homogeneous 4-adjacent BLACK blocks into one region apiece; the 8-adjacent WHITE blocks are likewise aggregated into WHITE regions.

An alternative to the quadtree representation is to use a decomposition method that is not regular (i.e., rectangles of arbitrary size rather than squares). This alternative has the potential of requiring less space. However, its drawback is that the determination of optimal partition points necessitates a search. The homogeneity criterion that is ultimately chosen to guide the subdivision process depends on the type of region data that is being represented. In the remainder of this section we shall assume that our domain is a 2^n by 2^n binary image with 1 or BLACK corresponding to foreground and 0 or WHITE corresponding to background (e.g., Figure 1). It is interesting to note that Kawaguchi et al. [1983] use a sequence of m binary-valued quadtrees to encode image data of 2^m gray levels, where the various gray levels are encoded by use of Gray codes [McCluskey 1965]. This should lead to compaction (i.e., larger sized blocks), since the Gray code guarantees that adjacent gray-level values differ by only one binary digit.

In general, any planar decomposition for image representation should possess the following two properties:

- (1) The partition should be an infinitely repetitive pattern so that it can be used for images of any size.
- (2) The partition should be infinitely decomposable into increasingly finer patterns (i.e., higher resolution).

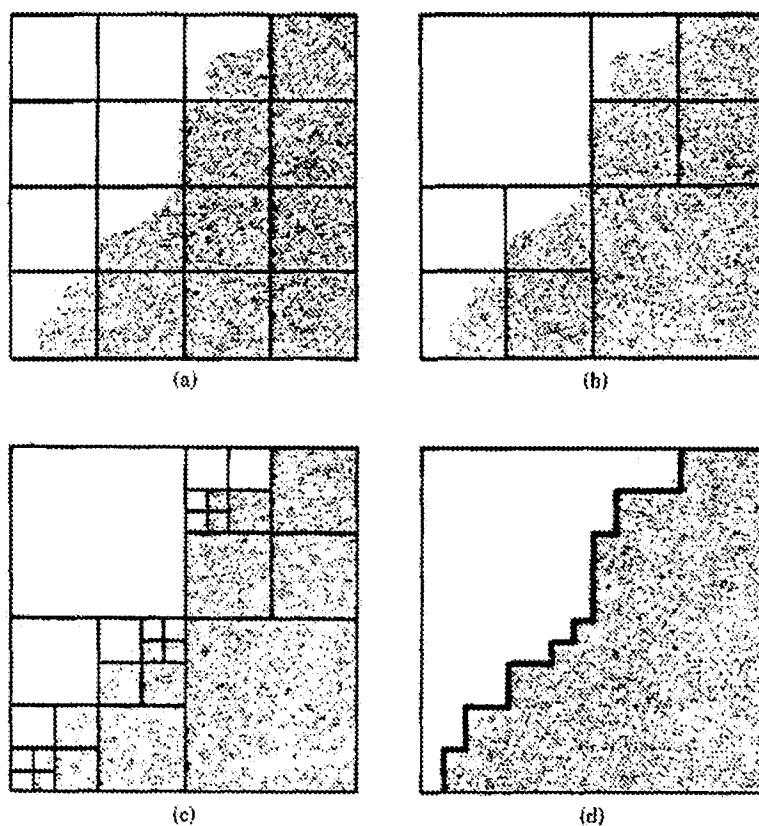


Figure 3. Example illustrating the "split and merge" segmentation procedure. (a) Start. (b) Merge. (c) Split. (d) Grouping.

Bell et al. [1983] discuss a number of tilings of the plane (i.e., tessellations) that satisfy Property (1). They also present a taxonomy of criteria to distinguish among the various tilings. Most relevant to our discussion is the distinction between limited and unlimited hierarchies of tilings. A tiling that satisfies Property (2) is said to be *unlimited*. An alternative characterization of such a tiling is that each edge of each tile lies on an infinite straight line composed entirely of edges. Four tilings satisfy this criterion; of these $[4^4]$,² consist-

²The notation is based on the degree of each vertex taken in order around the "atomic" tiling polygon. For example, for $[4.8^2]$ the first vertex of a constituent triangle has degree 4, while the remaining two vertices have degree 8 apiece.

ing of square atomic tiles (Fig. 4a), and $[6^3]$, consisting of equilateral triangle atomic tiles (Figure 4b), are well-known regular tessellations [Ahuja 1983]. For these two tilings we consider only the molecular tiles given in Figure 5a and b. The tilings $[4^4]$ and $[6^3]$ can generate an infinite number of different molecular tiles where each molecular tile consists of n^2 atomic tiles ($n \geq 1$). The remaining nonregular triangular tilings $[4.8^2]$ (Figure 4c) and $[4.6.12]$ (Figure 4d) are less well understood. One way of generating $[4.8^2]$ and $[4.6.12]$ is to join the centroids of the tiles of $[4^4]$ and $[6^3]$, respectively, to both their vertices and midpoints of their edges. Each of the resulting tilings has two types of hierarchy: in the case of $[4.8^2]$ an ordinary (Figure 5c) and a rotation hierarchy (Figure

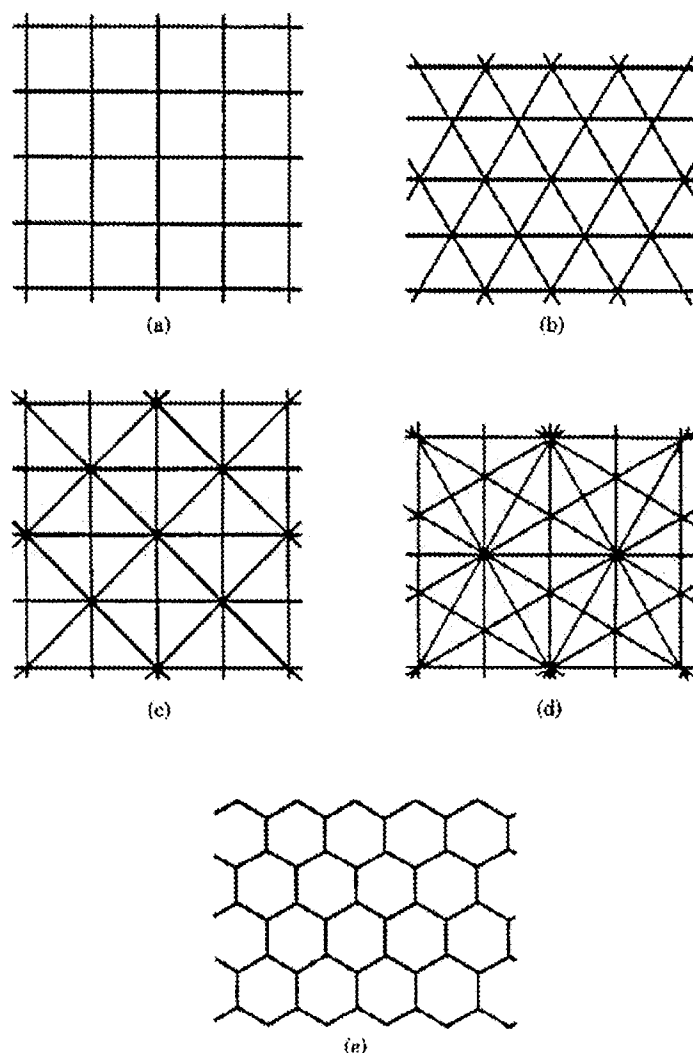


Figure 4. Sample tessellations. (a) $[4^4]$ square. (b) $[6^3]$ equilateral triangle. (c) $[4.8^2]$ isosceles triangle. (d) $[4.6.12]$ 30-60 right triangle. (e) $[3^6]$ hexagon.

5e) and in the case of $[4.6.12]$ an ordinary (Figure 5d) and a reflection hierarchy (Figure 5f). Of the *limited* tilings, many types of hierarchies may be generated [Bell et al. 1983]; however, they cannot, in general, be decomposed beyond the atomic tiling without changing the basic tile shape. This is a serious deficiency of the hexagonal tessellation $[3^6]$ (Figure 4e), which is, however,

regular, since the atomic hexagon can only be decomposed into triangles.

Thus we see that to represent data in the Euclidean plane any of the unlimited tilings could have been chosen. For a regular decomposition, the tilings $[4.8^2]$ and $[4.6.12]$ are ruled out. Upon comparing "square" $[4^4]$ and "triangular" $[6^3]$ quadtrees we find that they differ in terms of adjacency and

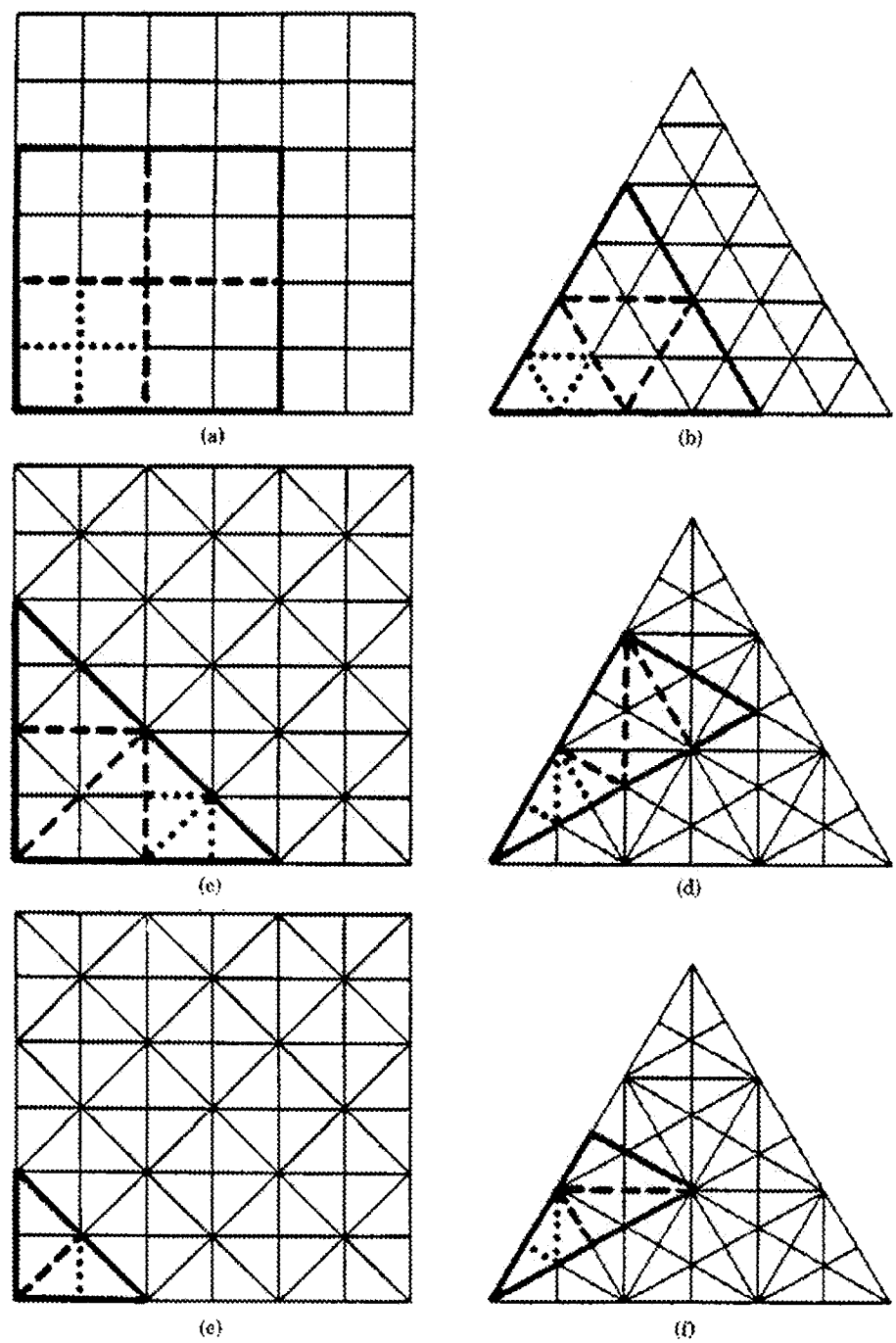


Figure 5. Examples illustrating unlimited tilings. (a) $[4^4]$ hierarchy. (b) $[8^4]$ hierarchy. (c) Ordinary $[4.8^2]$ hierarchy. (d) Ordinary $[4.6.12]$ hierarchy. (e) Rotation $[4.8^2]$ hierarchy. (f) Reflection $[4.6.12]$ hierarchy.

orientation. For example, let us say that two tiles are *neighbors* if they are adjacent either along an edge or at a vertex. A tiling is *uniformly adjacent* if the distances between the centroid of one tile and the centroids of all its neighbors are the same. The adjacency number of a tiling is the number of different intercentroid distances between any one tile and its neighbors. In the case of $[4^4]$, there are only two adjacency distances, whereas for $[6^3]$ there are three adjacency distances. A tiling is said to have *uniform orientation* if all tiles with the same orientation can be mapped into each other by translations of the plane that do not involve rotation or reflection. Tiling $[4^4]$ displays uniform orientation, whereas that of $[6^3]$ does not. Thus we see that $[4^4]$ is more useful than $[6^3]$. It is also very easy to implement. Nevertheless, $[6^3]$ has its uses. For example, Yamaguchi et al. [1984] use a triangular quadtree to generate an isometric view from an octree (a three-dimensional region quadtree discussed in greater detail in Section 2.14) representation of an object.

The type of quadtree used often depends on the grid formed by the image sampling process: Square quadtrees are appropriate for square grids and triangular quadtrees are appropriate for triangular grids. In the case of a hexagonal grid [Burt 1980], since a hexagon cannot be decomposed into hexagons, a *rosettelike* molecule of seven hexagons (i.e., septrees) must be built. Note that these rosettes have jagged edges as they are merged to form larger units (e.g., Figure 6). The hexagonal tiling is regular, has a uniform orientation, and most importantly displays a uniform adjacency. These properties are exploited by Gibson and Lucas [1982] in the development of algorithms for septrees (called *generalized balanced ternary* or *GBT* for short) analogous to those existing for quadtrees. Although the septree can be built up to yield large septrees, the smallest resolution in the septree must be decided upon in advance, since its primitive components (i.e., hexagons) cannot be decomposed into septrees later. Thus the septree yields only a partial hierarchical decomposition in the sense that the components can always be merged into larger

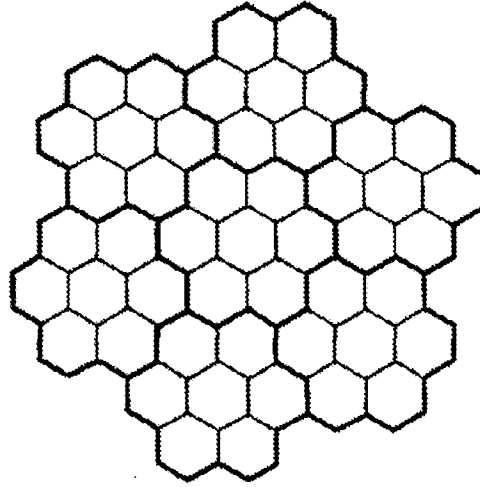


Figure 6. Example septree or "rosette" for a hexagonal grid.

units, but they cannot always be broken down.

2.1 Neighbor-Finding Techniques

A natural by-product of the treelike nature of the quadtree representation is that many basic operations can be implemented as tree traversals. The difference among them is in the nature of the computation that is performed at the node. Often these computations involve the examination of nodes whose corresponding blocks are "adjacent" to the block corresponding to the node being processed. We shall speak of these adjacent nodes as "neighbors." However, we must be careful to note that adjacency in space does not imply that any simple relationship exists among the nodes in the quadtree. This relationship is the subject of this section. In order to be more precise, we digress briefly and discuss the concepts of adjacency and neighbor in greater detail.

Each node of a quadtree corresponds to a block in the original image. We use the terms *block* and *node* interchangeably. The term that will be used depends on whether we are referring to decomposition into blocks (i.e., Figure 1c) or a tree (i.e., Figure 1d). Each block has four sides and four

corners. At times we speak of sides and corners collectively as directions. Let the four sides of a node's block be called its N, E, S, and W sides. The four corners of a block are labeled NW, NE, SW, and SE with the obvious meaning. Given two nodes P and Q whose corresponding blocks do not overlap, and a direction D , we define a predicate *adjacent* such that $\text{adjacent}(P, Q, D)$ is true if there exist two pixels p and q , contained in P and Q , respectively, such that either q is adjacent to side D of p , or corner D of p is adjacent to the opposite corner of q . In such a case, nodes P and Q are considered to be neighbors. For example, nodes J and 39 in Figure 1 are neighbors, since J is to the west of 39 , as are nodes 38 and H since H is to the NE of 38 . Two blocks may be adjacent both along a side and along a corner (e.g., B is both to the north and NE of J ; however, 39 is to the east of J but not to the SE of J). Note that the adjacent relation also holds for nonterminal (i.e., GRAY) as well as terminal (i.e., leaf) nodes.

Unfortunately, the neighbor relation is not a function in a mathematical sense. The problem is that given a node P , and a direction D , there is often more than one node, say Q , that is adjacent. For example, nodes 38 , 40 , K , and D are all western neighbors of node N . Similarly, nodes 40 , K , and D are all NW neighbors of node 57 . This means that in order to specify a neighbor more precise information is necessary about its nature (i.e., leaf or nonterminal) and location. In particular, it is necessary to be able to distinguish between neighbors that are adjacent to the entire side of a node (e.g., B is a northern neighbor of J) and those that are only adjacent to a segment of a node's side (e.g., 37 is one of the eastern neighbors of J). An alternative characterization of the difference is that in the former case we are interested in determining a node Q such that its corresponding block is the smallest block (possibly GRAY) of size greater than or equal to the block corresponding to P , whereas in the latter case we specify the neighbor in greater detail, in our case, by indicating the corner of P to which Q must be adjacent. The same distinction can also be made for corner directions. Below we define these

relations more formally. In the construction of names we use the following correspondence: G for "greater than or equal," C for "corner," S for "side," and N for "neighbor."

- (1) $\text{GSN}(P, D) = Q$. Node Q corresponds to the smallest block (it may be GRAY) adjacent to side D of node P of size greater than or equal to the block corresponding to P .
- (2) $\text{CSN}(P, D, C) = Q$. Node Q corresponds to the smallest block that is adjacent to side D of the C corner of node P .
- (3) $\text{GCN}(P, C) = Q$. Node Q corresponds to the smallest block (it may be GRAY) opposite the C corner of node P of size greater than or equal to the block corresponding to P .
- (4) $\text{CCN}(P, C) = Q$. Node Q corresponds to the smallest block that is opposite to the C corner of node P .

For example, $\text{GSN}(J, E) = K$, $\text{GSN}(J, S) = L$, $\text{CSN}(J, E, SE) = 39$, $\text{GCN}(H, NE) = G$, $\text{GCN}(H, SW) = K$, and $\text{CCN}(H, SW) = 38$. From the above we see that GCN is the corner counterpart of GSN and likewise CCN for CSN. It should be noted that the block corresponding to a node returned as the value of GCN or CCN must overlap some of the region bounded by the designated corner. Thus $\text{CCN}(J, NE) = B$ and not 37 . The following observations are also in order. First, none of GSN, CSN, GCN, or CCN define a 1-to-1 correspondence (i.e., a node may be a neighbor in a given direction of several nodes, e.g., $\text{GSN}(J, N) = B$, $\text{GSN}(37, N) = B$, and $\text{GSN}(38, N) = B$). Second, GSN, CSN, GCN, and CCN are not necessarily symmetric. For example, $\text{GSN}(H, W) = B$ but $\text{GSN}(B, E) = C$.

In the remaining discussions in this survey we focus strictly on GSN and GCN. When we use the term *neighbor*, that is, P is a neighbor of Q , we mean that P is a node of size greater than or equal to Q . For example, node 40 in Figure 1d (or equivalently block 40 in Figure 1c) has neighbors 38 , N , 57 , M , 39 , and 37 . A block that is not adjacent to a border of the image has a minimum of five neighbors. This can be seen by observing that a node cannot be adjacent to two nodes of greater size on

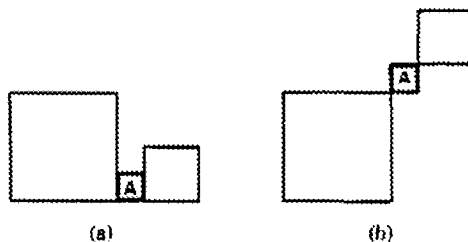


Figure 7. Impossible node configurations in a quadtree.

opposite sides (e.g., Figure 7a) or on opposite corners (e.g., Figure 7b). For further clarification, we observe that a split of a block creates four subblocks of equal size. Each subblock is 4-adjacent to two other subblocks (one horizontally adjacent neighbor and one vertically adjacent neighbor) at one of its vertices and 8-adjacent to the remaining subblock (corner adjacent neighbor) at the same vertex. As an example, given node P such that nodes Q and R are adjacent to its eastern and western sides, respectively, then at most one of nodes Q and R can be of greater size than P . Thus a node can have at most two larger sized neighbors adjacent to its nonopposite sides. One of these neighbors can overlap three neighboring directions, while the other can overlap two neighboring directions. The remaining three neighbors must be of equal size. For example, for node 37 in Figure 1, node B overlaps the NW, N, and NE neighboring directions, node J overlaps the W and SW directions, and the remaining neighbors are nodes 38, 40, and 39 in the E, SE, and S directions, respectively. A node has a maximum of eight neighbors, in which case all but one of the neighbors in the corner direction correspond to blocks of equal size. For example, for node N in Figure 1, the neighbors are nodes H, I, O, Q, P, M, K, and B. It is interesting to observe that for any BLACK node in the image, its neighbors cannot all be BLACK since otherwise merging would have taken place and the node would not be in the image. The same result holds for WHITE nodes.

As mentioned above, most operations on quadtrees can be implemented as tree traversals, with the operation being performed

by examining the neighbors of selected nodes in the quadtree. In order that the operation be performed in the most general manner, we must be able to locate neighbors in a way that is independent of both position (i.e., the coordinates) and size of the node. We also do not want to maintain any additional links to adjacent nodes. In other words, we only use the structure of the tree and no pointers in excess of the four links from a node to its four sons and one link to its father for a nonroot node. This is in contrast to the methods of Klinger and Rhodes [1979], which make use of size and position information, and those of Hunter [1978] and Hunter and Steiglitz [1979a, 1979b], which locate neighbors through the use of explicit links (termed ropes and nets). Yet another approach is to hypothesize a point across the boundary in the desired direction and then search for it. This is undesirable for two reasons. First, hypothesizing a point requires that we know the size of the block whose neighbor we are seeking. Second, the search requires that we make use of coordinate information.

Locating adjacent neighbors in the horizontal or vertical directions (i.e., GSN) is relatively straightforward [Samet 1982a]. The basic idea is to ascend the quadtree until a common ancestor with the neighbor is located, and then descend back down the quadtree in search of the neighboring node. It is obvious that we can always ascend as far as the root quadtree and then start our descent. However, our goal is to find the nearest common ancestor, as this minimizes the number of nodes that must be visited. Suppose, for example, that we wish to find the western neighbor of node N in Figure 1, that is, $GSN(N, W)$. The nearest common ancestor is the first ancestor node which is reached via its NE or SE son (i.e., the first ancestor node of which N is not a western descendant). Next, we retrace the path used to locate the nearest common ancestor, except that we make mirror image moves about an axis formed by the common boundary between the nodes. In the case of a western neighbor, the mirror images of NW and SW are NE and SE, respectively. Therefore the western neighbor of node N in Figure 1 is node K. It is located by

ascending the quadtree until the nearest common ancestor A has been located. This requires going through a NW link to reach node E, and a SE link to reach node A. Node K is subsequently located by backtracking along the previous path with the appropriate mirror image moves (i.e., by following a SW link to reach node D, and a NE link to reach node K).

Neighbors in the horizontal or vertical directions need not correspond to blocks of the same size. If the neighbor is larger, then only part of the path from the nearest common ancestor is retraced. Otherwise the neighbor corresponds to a block of equal size and a pointer to a BLACK, WHITE, or GRAY node, as is appropriate, of equal size is returned. If there is no neighbor (i.e., the node whose neighbor is being sought is adjacent to the border of the image in the specified direction), then NIL is returned.

Locating a neighbor in a corner direction (i.e., GCN) is considerably more complex [Samet 1982a]. Once again, we traverse ancestor links until a common ancestor of the two nodes is located. This is a process that requires two or three steps. First, we locate the given node's nearest ancestor, say *P*, which is also adjacent (horizontally or vertically) to an ancestor, say *Q*, of the sought neighbor (to see how this is determined, please read on!). If the node *P* does not exist, then we are at the true nearest common ancestor (e.g., when we are at node D when trying to find the SE neighbor of node J in Figure 1). Otherwise, the second step is one that finds *Q* by using the procedure for locating horizontally and vertically adjacent neighbors. The final step retraces the remainder of the path while it makes directly opposite moves (e.g., a SE move becomes a NW move). The nearest ancestor of the first step is the first ancestor node that is not reached by a link equal to the direction of the desired neighbor (e.g., to find a SE neighbor, the nearest such ancestor is the first ancestor node that is not reached via its son in the SE direction).³ As an example of the corner neigh-

bor-finding process, suppose that we wish to locate the SE neighbor of node 40 in Figure 1, which is 57, that is, GCN(40, SE). It is located by ascending the quadtree until we find the nearest ancestor D, which is also adjacent (horizontally in this case) to an ancestor of 57, that is, E. This requires that we go through a SE link to reach K and a NE link to reach D. Node E is now reached by applying the horizontal neighbor-finding techniques in the direction of the adjacency (i.e., east). This forces us to go through a SW link to reach node A. Backtracking results in descending a SE link to reach node E. Finally, we backtrack along the remainder of the path by making 180-degree moves; that is, we descend a SW link to reach node P and a NW link to reach node 57. Note that neighbors in the corner directions need not correspond to blocks of the same size. If the neighbor is larger, then it is handled in the same manner as outlined above for the horizontal and vertical directions (i.e., only part of the path from the nearest common ancestor is retraced). Webber [1984] discusses proofs of the correctness of the various neighbor-finding algorithms presented in this section.

Hunter [1978] and Hunter and Steiglitz [1979a, 1979b] describe a number of algorithms for operating on images represented by quadtrees by using explicit links from a node to its neighbors. These links connect adjacent nodes in the vertical and horizontal directions. A *rope* is defined as a link between two adjacent nodes of equal size where at least one of them is a leaf node. For example, there is a rope between nodes K and N in Figure 1. A *D-adjacency tree* in direction *D* exists whenever there is a rope between a leaf node, say *X*, and a GRAY node, say *Y*. In such a case, the *D-adjacency tree* of *X* is said to be the binary tree rooted at *Y* whose nodes consist of all the descendants of *Y* (BLACK, WHITE, or GRAY) that are adjacent to *X*. For example, Figure 8 contains the S-adjacency tree of node B

³ If the ancestor node is reached by a link directly opposite to the required direction, then we are already at the nearest common ancestor of the sought neigh-

bor. Otherwise, we obtain the neighbor in the direction that did not change (i.e., this determines whether we go in the N, E, S, or W direction for Step 2).

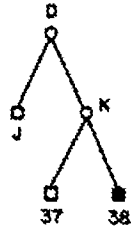


Figure 8. Adjacency tree corresponding to the rope between nodes D and B in Figure 1 (i.e., B's S-adjacency tree).

corresponding to the rope between nodes B and D that crosses the S side of node B.

The process of finding a neighbor by using a roped quadtree is quite simple. The rope is essentially a way to short-circuit the need to find a nearest common ancestor. Suppose that we want to find the neighbor of node X on side N using a rope. If a rope from X on side N exists, then it leads to the desired neighbor. Otherwise the desired neighbor is larger. Next, the tree is ascended until a node having a rope on side N, which will lead to the desired neighbor, is encountered. What we are doing is ascending the S-adjacency tree of the northern neighbor of node X. For example, to find the northern neighbor of node 38 in Figure 1, we ascend through node K to node D, which has a rope along its north side leading to node B (i.e., B's S-adjacency tree).

At times it is not even desirable to ascend nodes in the search for a rope. In such a case Hunter and Steiglitz make use of a *net*. This is a linked list whose elements are all the nodes, regardless of their relative size, that are adjacent along a given side of a node. For example, in Figure 1 there is a net for the southern side of node B consisting of nodes J, 37, and 38.

The advantage of ropes and nets is that the number of nodes that must be visited in the process of finding neighbors is reduced. However, the disadvantage is that the storage requirements are increased considerably. In contrast, our methods [Samet 1982a] only make use of the structure of the quadtree, that is, four links from a nonleaf node to its sons and a link from a nonroot node to its father. Using a suitably

defined model, Samet [1982a] and Samet and Shaffer [1984] have shown that in order to locate a neighbor of greater than or equal size in the horizontal or vertical direction, on the average, less than four nodes will be visited when using the nearest common ancestor techniques, whereas less than two nodes must be visited on the average when using ropes.⁴ Empirical results confirming this have been reported by Rosenfeld et al. [1982], Samet and Shaffer [1984], and Tucker [1984b]. Thus in practice it is not necessary to add the extra overhead of roping and netting of a quadtree, particularly upon considering that it requires extra storage. It should be noted that, at times, the algorithms that perform the basic operations on the image can be reformulated so that they do not require the computation of the neighbors. This is achieved by transmitting the neighbors of each node in the principal directions as actual parameters. Such techniques are termed *top down* in contrast with the *bottom-up* methods discussed earlier. One such technique is used by Jackins and Tanimoto [1983] in the computation of an n -dimensional perimeter. Their algorithm requires making n passes over the data and works only for neighbors that are adjacent along a side rather than at a corner. Independently, a similar algorithm was devised that does not require n passes but only uses one pass [Rosenfeld et al. 1982b; Samet and Webber 1982]. Another top-down algorithm that is able to compute all neighbors (i.e., adjacent along a side as well as a corner) with just one pass is reported by Samet [1985a].

2.2 Alternative Ways to Represent Quadtrees

As is shown in Section 1 the most natural way to represent a quadtree is to use a tree structure. In this case each node is represented as a record with four pointers to the records corresponding to its sons. If the node is a leaf node, it will have four pointers

⁴ A similar result is reported by DeMillo et al. [1978] in the context of embedding a two-dimensional array in a binary tree.

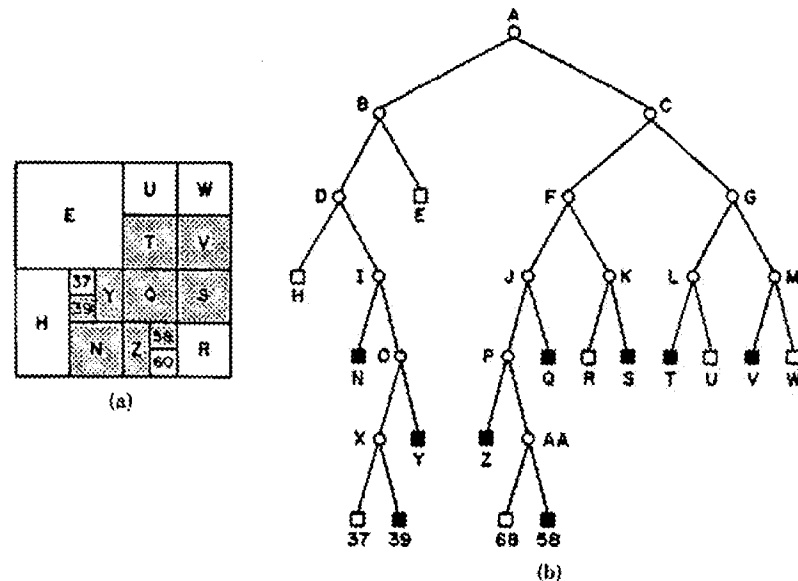


Figure 9. The bintree corresponding to Figure 1. (a) Block decomposition. (b) Bintree representation of the blocks in (a).

to the empty record. In order to facilitate certain operations an additional pointer is at times also included from a node to its father. This greatly eases the motion between arbitrary nodes in the quadtree and is exploited in a number of algorithms in order to perform basic image processing operations.

An alternative tree structure that uses an analogy to the *k*-d tree [Bentley 1975b] (see Section 3.1) is the *bintree* [Knowlton 1980; Samet and Tamminen 1984; Tamminen 1984a]. In essence, the space is always subdivided into two equal-sized parts alternating between the *x* and *y* axes. The advantage is that a node requires space only for pointers to its two sons instead of four sons. In addition, its use generally leads to fewer leaf nodes. Its attractiveness increases further when dealing with higher dimensional data (e.g., three dimensions) since less space is wasted on NIL pointers for terminal nodes and many algorithms are simpler to formulate. For example, Figure 9 is the bintree representation corresponding to the image of Figure 1.

The problem with the tree representation of a quadtree is that it has a considerable

amount of overhead associated with it. For example, given an image that can be aggregated to yield *B* and *W* BLACK and WHITE nodes, respectively, $(B + W - 1)/3$ additional nodes are necessary for the internal (i.e., GRAY) nodes. Moreover, each node requires additional space for the pointers to its sons. This is a problem when dealing with large images that cannot fit into core memory. Consequently, there has been a considerable amount of interest in pointerless quadtree representations. They can be grouped into two categories. The first treats the image as a collection of leaf nodes. The second represents the image in the form of a traversal of the nodes of its quadtree. The following discussion briefly summarizes the type of operations that can be achieved using such representations. Some of these operations are discussed in greater detail in subsequent sections in the context of pointer-based quadtree representations.

When an image is represented as a collection of the leaf nodes comprising it, each leaf is encoded by a base 4 number termed a *locational code*, corresponding to a sequence of directional codes that locate the

leaf along a path from the root of the quadtree. It is analogous to taking the binary representation of the x and y coordinates of a designated pixel in the block (e.g., the one at the lower left corner) and interleaving them (i.e., alternating the bits for each coordinate). It is difficult to determine the origin of this technique. It was used as an index to a geographic database by Morton [1966] and is termed a *Morton matrix*. Klinger and Rhodes [1979] presented it as a means of organizing quadtrees on external storage. It has also been widely discussed in the literature in the context of multidimensional point data (see Section 3.5). A base 5 variant of it (although all arithmetic operations on the locational code are performed by using base 4), which has an additional code as a *don't care*, is used by Gargantini [1982a] and Abel and Smith [1983] (see also Burton and Kollias [1983], Cook [1978], Klinger and Dyer [1976], Oliver and Wiseman [1983a], Weber [1978], and Woodward [1982]) to yield an encoding where each leaf in a 2^n by 2^n image is n digits long. A leaf corresponding to a 2^k by 2^k block ($k < n$) will have $n - k$ *don't care* digits. As an example, assuming that codes 0, 1, 2, and 3 correspond to quadrants NW, NE, SW, and SE, respectively, and 4 denotes a *don't care*, block H in Figure 1 is represented by the base 5 number 124. Such an encoding has the interesting property that when the codes of the leaf nodes are sorted in increasing order, the resulting sequence is the postorder (also preorder or inorder since the nonleaf nodes are excluded) traversal of the blocks of the quadtree.

Actually, in the representation described above there is no need to include the locational code of every leaf node. Gargantini [1982a] only retains the locational codes of the BLACK nodes and terms the resulting representation a *linear quadtree*. The codes for the WHITE blocks can be obtained by using the ordering imposed by the sort without having physically to construct the quadtree. Lauzon et al. [1984] propose that the collection of the leaf nodes be represented by using a variant of the run length code [Rutovitz 1968] termed a *two-dimensional run encoding*. They make use of a

Morton matrix. Once the codes of the leaf node have been sorted in increasing order, the resulting list is viewed as a set of subsequences of codes corresponding to blocks of the same color. The final step in its construction is to discard all but the first element of each subsequence of blocks of the same color. The codes of the intervening blocks can be reconstructed by knowing the codes of two successive blocks. In comparison to linear quadtrees, this representation is more compact and more efficient for superposition. However, translation and rotation by multiples of 90 degrees are easier with the linear quadtree [Gargantini 1983]. In addition, given a code for a particular BLACK node, its horizontal and vertical neighbors can be obtained by performing arithmetic operations on the locational code [Abel and Smith 1983; Gargantini 1982a]. However, this often involves search, and can be made more efficient by special-purpose hardware. Nevertheless, this result is significant in that many of the standard quadtree algorithms that rely on neighbor computation can be applied to images represented by linear quadtrees. Abel [1984] describes an organization of the postorder sequence in the form of a B⁺-tree [Comer 1979].

Jones and Iyengar [1984] (see also Raman and Iyengar [1983]) introduced the concept of a forest of quadtrees that is a decomposition of a quadtree into a collection of subquadtrees, each of which corresponds to a maximal square. The maximal squares are identified by refining the concept of a nonterminal node to indicate some information about its subtrees. An internal node is said to be of type GB if at least two of its sons are BLACK or of type GB. Otherwise the node is said to be of type GW. For example, in Figure 10, nodes C, E, and F are of type GB and nodes A, B, and D are of type GW. Each BLACK node or an internal node with a label of GB is said to be a maximal square. A forest is the set of maximal squares that are not contained in other maximal squares and that span the BLACK area of the image. Thus the forest corresponding to Figure 10 is {C, E, F}. The elements of the forest are identified by base 4 locational codes. Such a

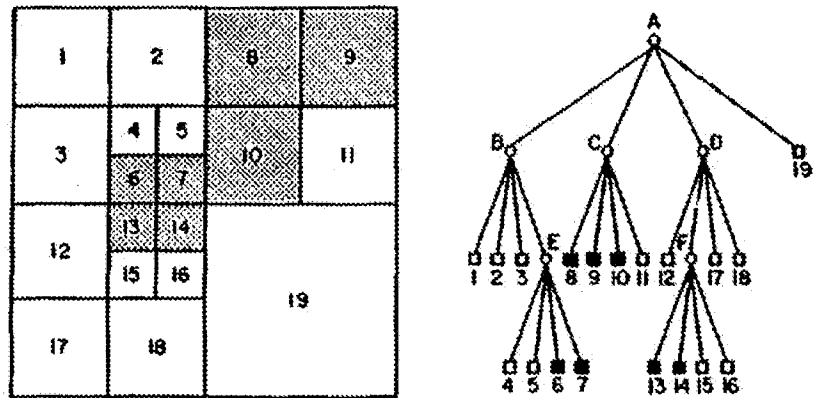


Figure 10. A sample image and its quadtree illustrating the concept of a forest.

representation can lead to a savings of space since large WHITE items are ignored by it.

The second pointerless representation is in the form of a preorder tree traversal (i.e., depth first) of the nodes of the quadtree. The result is a string consisting of the symbols "(", "B", "W" corresponding to GRAY, BLACK, and WHITE nodes, respectively. This representation is due to Kawaguchi and Endo [1980] and is called a DF-expression. For example, the image of Figure 1 has

(W(WWBB(W(WBBBWB(BB(BBBWW as its DF-expression (assuming that sons are traversed in the order NW, NE, SW, SE). The original image can be reconstructed from the DF-expression by observing that the degree of each nonterminal (i.e., GRAY) node is always 4. DeCoulon and Johnsen [1976] use a very similar scheme termed *autoadaptive block coding*. The difference is that the alphabet consists solely of two symbols, "0" and "1". The "0" corresponds to a block composed of WHITE pixels only. Otherwise, a "1" is used and the block is subdivided into four subblocks. Therefore the "0" is analogous to "W" and the "1" is analogous to "(" and "B". In other words, there is no merging of BLACK pixels into blocks, and thus the coding scheme is asymmetric, whereas the DF-expression method is symmetric with respect to both BLACK and WHITE. The two methods are shown to yield encodings

that require a comparable number of bits. A binary tree variant of the DF-expression based on the bintree is discussed by Tamminen [1984b].

Kawaguchi et al. [1983] show how a number of basic image processing operations can be performed on an image represented by a DF-expression. In particular, they demonstrate centroid computation, rotation, scaling, shifting, and set operations. Representation of an image using a preorder traversal is also reported by Oliver and Wiseman [1983a]. They show how to perform operations as mentioned above as well as merging, masking, construction of a quadtree from a polygon, and area filling. Neighbor finding is also possible when traversal-based representations are used, although it is rather cumbersome and time consuming.

In the remainder of this survey we shall be using the pointer-based quadtree representation unless specified otherwise. This should not pose a problem as we have already discussed some of the problems associated with the pointerless representations (i.e., that neighbor finding is more complicated, etc.).

2.3 Conversion

The quadtree is proposed as a representation for binary images because its hierarchical nature facilitates the performance of a large number of operations. However, most images are traditionally represented

by use of methods such as binary arrays, rasters (i.e., run lengths), chain codes (i.e., boundaries), or polygons (vectors), some of which are chosen for hardware reasons (e.g., run lengths are particularly useful for rasterlike devices such as television). Techniques are therefore needed that can efficiently switch between these various representations.

The most common image representation is probably the binary array. There are a number of ways to construct a quadtree from a binary array. The simplest approach is one that converts the array to a complete quadtree (i.e., for a 2^n by 2^n image, a tree of height n with one node per pixel). The resulting quadtree is subsequently reduced in size by repeated attempts at merging groups of four pixels or four blocks of a uniform color that are appropriately aligned. This approach is simple, but is extremely wasteful of storage, since many nodes may be needlessly created. In fact, it is not inconceivable that available memory may be exhausted when an algorithm employing this approach is used, whereas the resulting quadtree fits in the available memory.

We can avoid the needless creation of nodes by visiting the elements of the binary array in the order defined by the labels on the array in Figure 11 (which corresponds to the image of Figure 1). This order is also known as a Morton matrix [Morton 1966] (discussed in Section 2.2). By using such a method a leaf node is never created until it is known to be maximal. An equivalent statement is that the situation does not arise in which four leaves of the same color necessitate the changing of the color of their parent from GRAY to BLACK or WHITE as is appropriate. For example, we note that since pixels 25, 26, 27, and 28 are all BLACK, no quadtree nodes were created for them; that is, node H corresponds to the part of the image spanned by them. This algorithm is shown to have an execution time proportional to the number of pixels in the image [Samet 1980b].

At times the array must be scanned in a row-by-row manner as we build the quadtree (e.g., when a raster representation is used). For example, the pixels of the image

1	2	5	6	17	18	21	22
3	4	7	8	19	20	23	24
9	10	13	14	25	26	29	30
11	12	15	16	27	28	31	32
33	34	37	38	49	50	53	54
35	36	39	40	51	52	55	56
41	42	45	46	57	58	61	62
43	44	47	48	59	60	63	64

Figure 11. Binary array representation of the region in Figure 1a.

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	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 12. A labeling of the pixels of the region in Figure 1 that indicates the order of visiting them in the process of constructing a quadtree from the raster representation.

of Figure 1 would be visited in the order defined by the labels on the array of Figure 12. The amount of work that is required depends on whether an odd-numbered or even-numbered row is being processed. For an odd-numbered row, the quadtree is constructed by processing the row from left to right, adding a node to the quadtree for each pixel. As the quadtree is constructed, nonterminal nodes must also be added in such a way that at any given instant, a valid quadtree exists. Even-numbered rows require more work since merging may also take place. In particular, a check for a possible merger must be performed at every even-numbered vertical position (i.e., every even-numbered pixel in a row). Upon the creation of any merger, it must be checked to determine whether another merger is possible. In particular, for pixel position $(a \cdot 2^i, b \cdot 2^j)$ where $(a \bmod 2) = (b \bmod 2) = 1$, a maximum of $k = \min(i, j)$ mergers is possible. In this discussion, a pixel position is the coordinate of its lower right corner with respect to an origin in the upper

left corner of the image. For example, at pixel 80 of Figure 12, that is, position (4, 8), a maximum of two merges is possible. An algorithm using these techniques, which has an execution time proportional to the number of pixels in the image, is described by Samet [1981a]. Unnikrishnan and Venkatesh [1984] present an algorithm for converting rasters to linear quadtrees.

As output is usually produced on a raster device, we need a method for converting a quadtree representation into a suitable form. The most obvious method is to generate an array corresponding to the quadtree, but this method may require more memory than is available and thus is not considered here. Samet [1984] describes a number of quadtree-to-raster algorithms. All of the algorithms traverse the quadtree by rows and visit each quadtree node once for each row that intersects it. For example, a node that corresponds to a block of size 2^k by 2^k is visited 2^k times, and each visit results in the output of a sequence of 2^k 0's or 1's as is appropriate. Some of the algorithms are *top down* and others are *bottom up*. The bottom-up algorithms visit adjacent blocks by use of neighbor-finding techniques, whereas the top-down method starts at the root each time it visits a node. The bottom-up methods are superior as the image resolution gets larger (i.e., n for a 2^n by 2^n image) since the number of nodes that must be visited in locating neighbors is smaller than that necessary when the process is constantly restarted from the root. All of the algorithms have execution times that depend only on the number of blocks in the image (irrespective of their color) and not on their particular configuration. In addition, they do not require memory in excess of that necessary to store the quadtree being output. For example, the two images shown in Figure 13 require the same amount of time to be output since they both have 11 blocks of size 2 by 2 pixels and 20 blocks of 1 pixel. This is important when considerations such as refresh times, etc. must be taken into account.

The chain code representation [Freeman 1974] (also known as a boundary or border code) is very commonly used in carto-

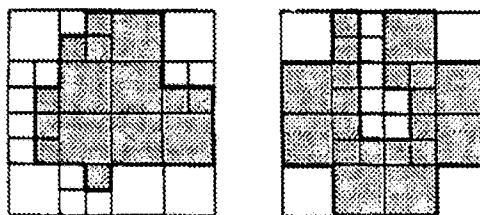


Figure 13. Two images that require the same amount of work to be converted from a quadtree to a raster representation.

graphic applications. It can be specified, relative to a given starting point, as a sequence of unit vectors (i.e., one pixel wide) in the principal directions. We can represent the directions by numbers; for example, let i , an integer quantity ranging from 0 to 3, represent a unit vector having a direction of $90 + i$ degrees. For example, the chain code for the boundary of the BLACK region in Figure 1, moving clockwise starting from the midpoint of the extreme right boundary, is

3²2³3¹2¹3¹2³1³0¹1¹0¹1²0⁴3².

The above is a four-direction chain code. Generalized chain codes involving more than four directions can also be used. Chain codes are not only compact, but they also simplify the detection of features of a region boundary, such as sharp turns (i.e., corners) or concavities. On the other hand, chain codes do not facilitate the determination of properties such as elongatedness, and it is difficult to perform set operations such as union and intersection as well. Thus it is useful to be able to construct a quadtree from a chain code representation of a binary image. Such an algorithm described by Samet [1980a] is briefly outlined below.

The algorithm has two phases. The first phase traces the boundary in the clockwise direction and constructs a quadtree with BLACK nodes of size unit code length. All terminal nodes are said to be at level 0 and correspond to blocks that are adjacent to the boundary and are within the region whose boundary is being traced. The process begins by choosing a link in the chain code at random and creating a node for it, say P. Next, the following link in the chain

code, say NEW, is examined, and its direction is compared with that of the immediately preceding link, say OLD. At this point, three courses of action are possible. If the directions of NEW and OLD are the same, then a node, say Q, which is a neighbor of P in direction OLD, may need to be added (see Figure 14a). If NEW's direction is to the right of OLD, a new node is unnecessary (see Figure 14b); but if NEW's direction is to the left of OLD, then we may have to add two nodes. First, a node, say Q, that is a neighbor of P in direction OLD is added (if not already present). Second, a node, say R, that is a neighbor of Q in direction NEW is added (see Figure 14c). These nodes are added to the quadtree by using the neighbor-finding techniques discussed previously. As the various links in the chain code are processed, some nodes may be encountered more than once, indicating that they are adjacent to the boundary on more than one side. This information is recorded for each node. Figure 15 shows the block decomposition and partial quadtree after the application of the first phase to the boundary code representation corresponding to Figure 1. The BLACK nodes have been labeled in the order in which they have been visited, starting at the midpoint of the extreme right boundary of the image and proceeding in a clockwise manner. All uncolored nodes in Figure 15 are depicted as short lines emanating from their father.

The first phase of the algorithm leaves many nodes uncolored since it only marks nodes adjacent to the boundary as BLACK. The second phase of the algorithm performs a postorder traversal of the partial quadtree resulting from the first phase and sets all the uncolored nodes to BLACK or WHITE as is appropriate. For an uncolored node to eventually correspond to a BLACK node, it must be totally surrounded by BLACK nodes since otherwise it would have been adjacent to the boundary and could not be uncolored. The algorithm therefore sets every uncolored node to BLACK, unless any of its neighbors is WHITE, or if one of its neighbors is BLACK with a boundary along the shared side. This information is easy to ascertain

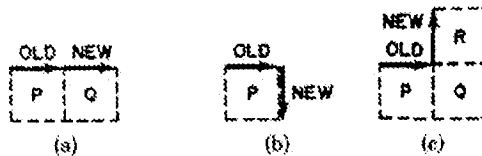


Figure 14. Examples of the actions to be taken when the chain code (a) maintains its direction, (b) turns clockwise, and (c) turns counterclockwise.

by virtue of the boundary adjacency information that is recorded for each BLACK terminal node during the first phase. Also, any GRAY node that has four BLACK sons is replaced by a BLACK node. The above algorithm has a worst-case execution time that is proportional to the product of the region's perimeter (i.e., the length of the chain code) and the log of the diameter of the image (i.e., n for a 2^n by 2^n image) [Samet 1980a]. Webber [1984] presents a variation of this algorithm that shifts the chain code to an optimal position before building the quadtree. The total cost of the shift and build operations is proportional to the region's perimeter.

It is also useful to be able to convert a quadtree representation of a region to its chain code [Dyer et al. 1980]. This is achieved by traversing the boundary in such a way that the region always lies to the right once an appropriate starting point has been determined. The boundary consists of a sequence of (BLACK, WHITE) node pairs. Assume for the sake of this discussion that P is a BLACK node, Q is a WHITE node, and that the block corresponding to node P is to the north of Q. For each BLACK-WHITE adjacency, a two-step procedure is executed. First, the chain link associated with that part of P's boundary that is adjacent to Q is output. The length of the chain is equal to the minimum of the sizes of the two blocks.

Second, the (BLACK, WHITE) node pair that defines the subsequent link in the chain as we traverse the boundary is determined. There are three possible relative positions of P and Q as outlined in Figure 16: (1) P extends past Q (Figure 16a), (2) Q extends past P (Figure 16b), or (3) P and Q meet at the same point (Figure 16c). In order to determine the next pair, the adja-

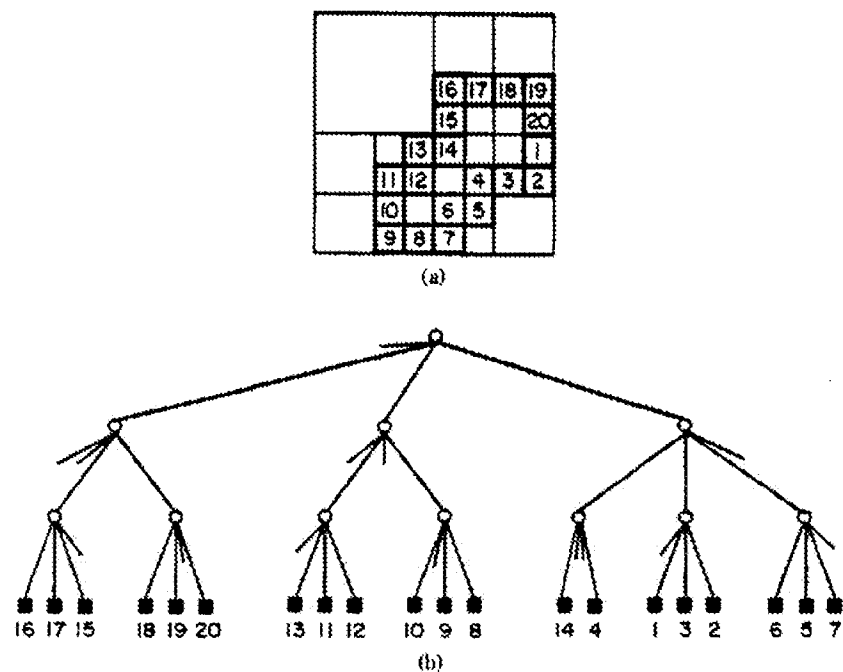


Figure 15. Block decomposition (a) and quadtree (b) of the region in Figure 1 after application of phase one of the chain code to quadtree algorithm.

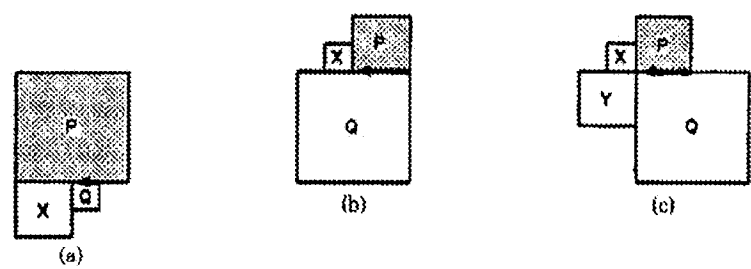


Figure 16. Possible overlap relationships between the (BLACK, WHITE) adjacent node pair (P, Q). The arrow indicates the boundary segment just output. (a) P extends past Q. (b) Q extends past P. (c) P and Q meet at the same point.

cent nodes X and Y are located by using the neighbor-finding techniques discussed previously. At this point the next pair can be determined by referring to Figure 17 and choosing the two blocks that are adjacent to the arrow in the appropriate case. Note that we assume that the region is four-connected so that blocks touching only at a corner are not adjacent. For example, the new pair in Figure 17g is (P, X); that is, the

boundary turns right regardless of the type of node Y. The algorithm has an average execution time that is proportional to the region's perimeter [Dyer et al. 1980]. In the case where a region contains holes, the algorithm can be extended by systematically traversing all BLACK nodes upon completion of the first boundary-following sequence. Whenever a BLACK node is encountered with a boundary edge unmarked

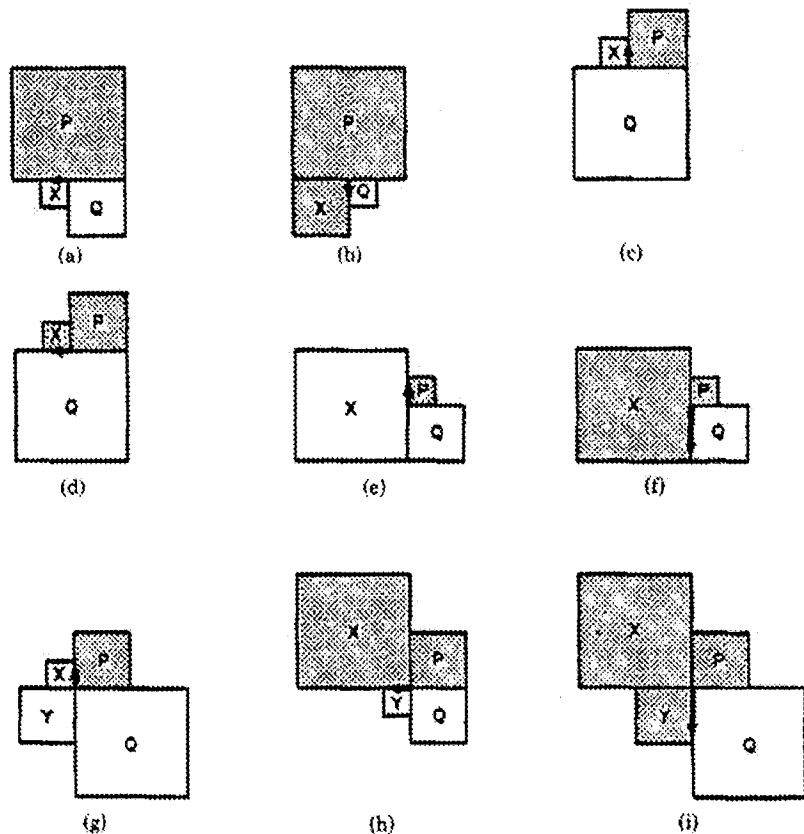


Figure 17. Possible configurations of P, Q, and their neighbor blocks in determining the next (BLACK, WHITE) pair. The arrow indicates the next boundary segment to be output.

by the boundary follower, its boundary is followed, after which the traversal of the quadtree continues.

The chain code can be used as an approximation of a polygon by unit vectors. It is also common to represent polygonal data by a set of vertices, or even a point and a sequence of vectors consisting of pairs (i.e., (magnitude, direction)). Hunter [1978] and Hunter and Steiglitz [1979a, 1979b] address the problem of representing simple polygons (i.e., polygons with non-intersecting edges and without holes) by using quadtrees. A polygon is represented by a three-color variant of the quadtree. In essence, there are three types of nodes—interior, boundary, and exterior. A node is said to be of type *boundary* if an edge of the polygon passes through it. Boundary nodes

are not subject to merging (they are analogous to BLACK nodes in the matrix (MX) quadtree described in Section 3.2). *Interior* and *exterior* nodes correspond to areas respectively within and outside of the polygon and can be merged to yield larger nodes. Figure 18 illustrates a sample polygon and its quadtree corresponding to the definition of Hunter and Steiglitz [1979a]. One disadvantage of such a representation for polygonal lines is that a width is associated with them, whereas in a purely technical sense these lines have a width of zero. Also a shift in operations may result in information loss. (For more appropriate representations of polygonal lines see Section 4.)

Hunter and Steiglitz present two algorithms for building a quadtree from a polygon. The first is a top-down algorithm that

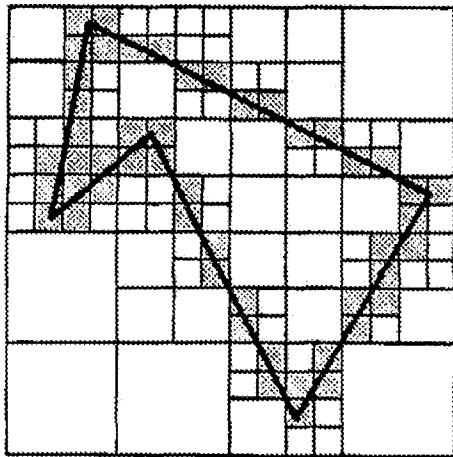


Figure 18. Hunter and Steiglitz's [1979a] quadtree representation of a polygon.

starts at the root and splits the space into four blocks, creating the necessary nodes. Each node whose block (which is not a pixel) intersects the polygonal boundary is recursively split. Given a polygon with v vertices and a perimeter p (in units of pixel width), construction of a quadtree within a 2^n by 2^n space from a polygon has an execution time of $O(v + p + n)$. Unfortunately, the quadtree from the polygon construction algorithm does not distinguish between an interior and an exterior node. A coloring algorithm is then performed that propagates the color of the boundary nodes inward by initially traversing the boundary and stacking all sides that are within the polygon for each boundary node. Coloring is propagated by examining stack entries and their adjacent leaves. For stack entry S , if the block corresponding to its adjacent leaf node, say T , is not smaller and is uncolored, then T is colored and all of its sides with the exception of S are placed on the stack. S is removed from the stack and colored. The key to the algorithm is that boundary nodes (i.e., pixels) are small and their neighbors get larger as the center of the polygon is approached. This algorithm makes use of a netted quadtree to compute neighboring nodes. It has been shown to have an execution time proportional to the number of nodes in the quadtree being colored.

The second algorithm for constructing a quadtree from a polygon is termed an *outline algorithm*. It combines a top-down decomposition of the space in which the polygon is embedded with a traversal of the boundary, resulting in a roped quadtree. During the construction process neighbors are computed as a by-product of the top-down decomposition process. The outline algorithm similarly has an execution time of $O(v + p + n)$. Combining the outline algorithm, a netting process, and the coloring algorithm leads to a quadtree for polygon algorithm with execution time of $O(v + p + n)$.

2.4 Set Operations

The quadtree is especially useful for performing set operations such as the union (i.e., overlay) and intersection of several images. This is described in greater detail by Hunter [1978], Hunter and Steiglitz [1979a], and Shneier [1981a]. For example, obtaining the quadtree corresponding to the union of S and T merely requires a traversal of the two quadtrees in parallel, an examination of corresponding nodes, and construction of the resulting quadtree, say in U . If either of the two nodes is BLACK, then the corresponding node in U is BLACK. If one node is WHITE, say in S , then the corresponding node in U is set to the other node, that is, in T . If both nodes are GRAY, then U is set to GRAY and the algorithm is applied recursively to the sons of S and T . However, once the sons have been processed, when both nodes are GRAY, a check must be made if a merger is to take place since all four sons could be BLACK. For example, consider the union of the quadtrees of Figures 19 and 20. Node B in Figure 19 and node E in Figure 20 are both GRAY. However, the union of their corresponding sons yields four BLACK nodes, which must be merged to yield a BLACK node in U , where the corresponding nodes in S and T were GRAY. Figure 21 shows the result of the union of Figures 19 and 20.

Computing the intersection of two quadtrees is just as simple. The algorithm described above for union is applied, except

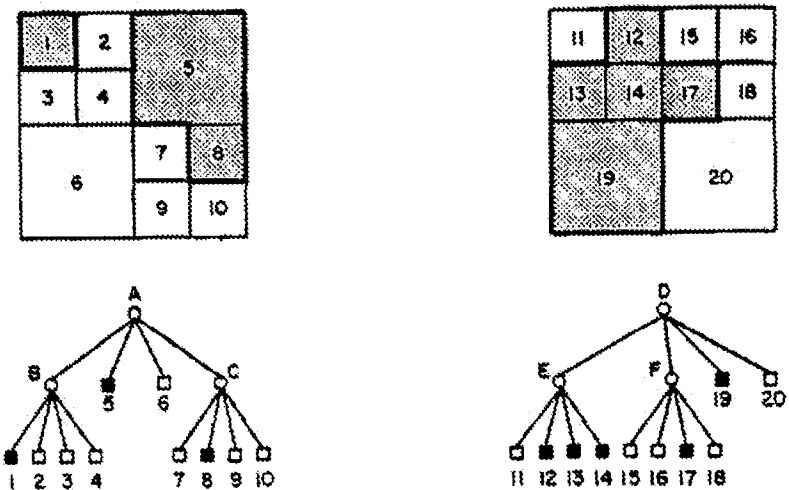


Figure 19. Sample image and its quadtree.

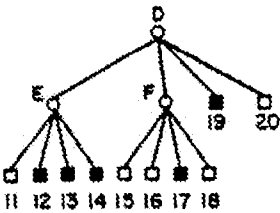


Figure 20. Sample image and its quadtree.

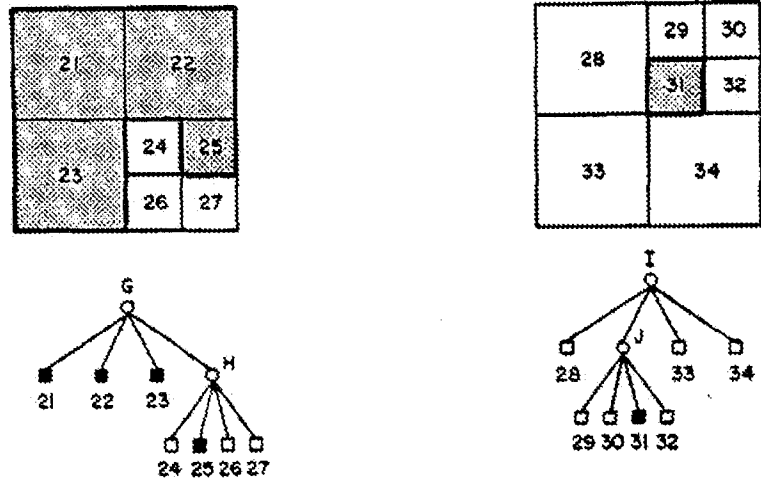


Figure 21. Union of the images in Figures 19 and 20.

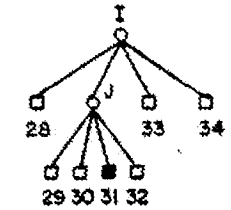


Figure 22. Intersection of the images in Figures 19 and 20.

that the roles of BLACK and WHITE are interchanged. When both nodes are GRAY, the check for a merger is performed to determine if all four sons are WHITE. Figure 22 shows the result of the intersection of Figures 19 and 20.

The time required for these algorithms is proportional to the minimum of the number of nodes at corresponding levels of the two quadtrees. In order to achieve this time bound, the resulting quadtree is composed

of subtrees from the quadtrees serving as operands of the set operation. If a new quadtree is constructed, then the operations have an execution time that is proportional to the number of nodes in the quadtrees. An upper bound on this time is the size of the smaller of the two quadtrees. The ability to perform set operations quickly is one of the primary reasons for the popularity of quadtrees over alternative representations such as the chain code. The

chain code can be characterized as a local data structure, since each segment of the chain code conveys information only about the part of the image to which it is adjacent; that is, the image is to its right. Performing an overlay operation on two images represented by chain codes thus requires a considerable amount of work. In contrast, the quadtree is a hierarchical data structure that yields successive refinements at lower levels in the tree. Of course, a hierarchical chain code can be defined, but this is primarily useful in handling extreme cases (null intersection, etc.).

Hunter [1978] suggested a novel approach to solving the problem of determining whether or not two polygons intersect when polygons are represented as quadtrees. One constructs the two quadtrees from the polygons, intersects them, and then checks the result to see whether it is the empty quadtree. This process has an execution time of $O(v + p + n)$ (see Section 2.3). Of course, this time bound is a function of the accuracy required and is subject to errors resulting from limitations imposed by the digitization process. In contrast, Shamos and Hoey [1975] show that the problem can be solved in $O(v \log v)$ time. The reader is cautioned that in actuality the different nature of the representations that are involved may make it difficult to compare the two algorithms (i.e., the constants and quantities are considerably different).

2.5 Transformations

One of the primary motivations for the development of the quadtree concept is a desire to provide an efficient data structure for computer graphics. Warnock [1969] has used recursive decomposition as the basis for the hidden surface elimination algorithm. Hunter's doctoral thesis [Hunter 1978], which addressed the problem of efficiently performing animation by computer, was a significant extension of the quadtree concept from both a theoretical and practical standpoint. In order to do this, the system must have the capability of performing a number of basic transformations. Scaling by a power of two is trivial

when using quadtrees since it is simply a reduction in resolution. Rotation by multiples of 90 degrees is equally simple, that is, a recursive rotation of sons at each level of the quadtree. For example, Figure 23b is the result of rotating Figure 23a by 90 degrees counterclockwise. Notice how the NW, NE, SW, and SE sons have become SW, NW, SE, and NE sons, respectively, at each level in the quadtree.

It is also useful to transform a quadtree encoding of a picture in the form of a collection of polygons and holes into another quadtree by applying a linear operator. One simple algorithm [Hunter and Steiglitz 1979b] traces all the polygons in the input quadtree to find vertices. The images of the vertices that result from the application of the linear operator determine the polygons in the output quadtree. The outline and color algorithm [Hunter and Steiglitz 1979a] (see Section 2.3) is used to construct the actual output quadtree for each polygon (as well as holes). The final step is the superposition of the polygons, which is performed by using techniques discussed in Section 2.4. The outline algorithm saves some work by ignoring the boundaries of the input polygons that will not be visible in the output. By assuming that the transformation does not change the resolution (or scale) of the input picture, it can be shown that the transformation algorithm requires time and space of $O(t + p)$ [Hunter and Steiglitz 1979b], where t is the total number of nodes in the input quadtree and p is the total perimeter of the nonbackground visible portions of the input picture.

The linear transformation algorithm and the scaling and rotation operations share a common failing. With the exception of scaling by a power of two, translations, or rotations in multiples of 90 degrees, they result in approximations. Straight lines are not necessarily transformed into straight lines. This failing is often mistakenly attributed to the quadtree representation, whereas in fact it is a direct result of the underlying digitization process. It manifests itself no matter what underlying representation is used when doing raster graphics. (For a quadtree-based representation that is free of such a problem see the

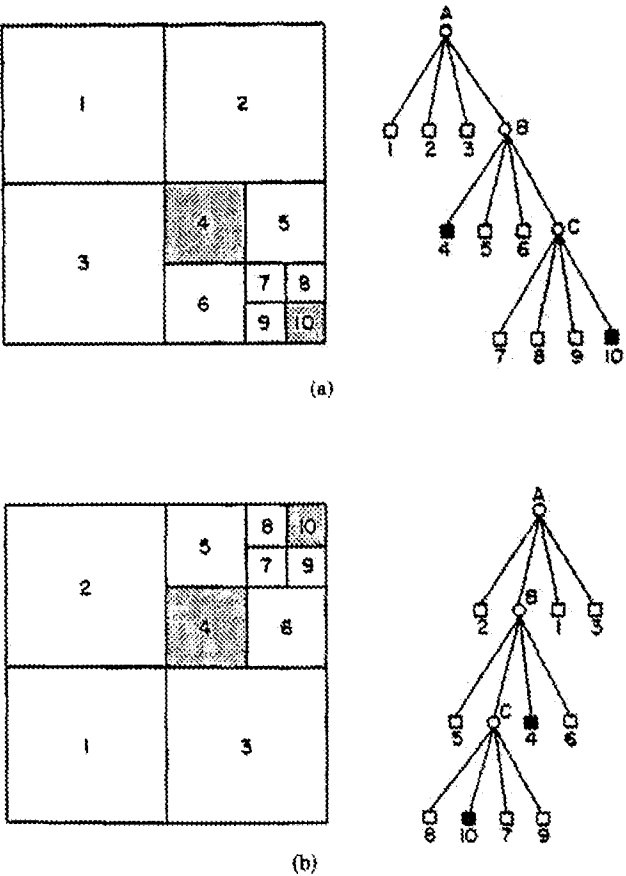


Figure 23. Rotating (a) by 90 degrees counterclockwise yields (b).

discussion of the PM quadtree [Samet and Wehber 1983b] in Section 4.2).

Another operation that is useful in graphics applications is termed *windowing*. It is the process of extracting a rectangular window from an image represented by a quadtree and building a quadtree for the window. An algorithm designed to achieve this effect for a square window of size 2^k by 2^k at an arbitrary position in a 2^n by 2^n image is described by Rosenfeld et al. [1982b]. In essence, the new quadtree is constructed as the input quadtree is decomposed and relevant blocks are copied into the new quadtree. The execution time of this process depends both on the relative position of the center of the window with respect to the center of the input quadtree,

and on the sizes of the blocks in the input quadtree that overlap the window. A generalization of this windowing algorithm for pointer-based quadtrees [Peters 1984] and linear quadtrees [van Lierop 1984] performs the calculation of a general linear transformation (including scaling and rotation) without extracting the polygon from the quadtree and then rebuilding the quadtree from the transformed polygon [Hunter and Steiglitz 1979b]. For rectangular windows, windowing is trivial to implement if the *squarecode* representation of Oliver and Wiseman [1983b] is used. The *squarecode* is a variant of the locational code (see Section 2.2), which is used to represent the image as a collection of disjoint squares of arbitrary side length and at arbitrary posi-

tions by recording the length and the address of one of the square's corners.

Quadtrees have also been used for image-processing operations that involve gray-scale images rather than binary images. Some examples include image segmentation [Ranade et al. 1980], edge enhancement [Ranade 1981], image smoothing [Ranade and Shneier 1981], and threshold selection [Wu et al. 1982].

2.6 Areas and Moments

Areas and moments for images represented by quadtrees are extremely simple to compute. To find the area it is necessary to traverse the quadtree in postorder and accumulate the sizes of the BLACK blocks. Assume that the root of a 2^n by 2^n image is at level n and the number of pixels in such an image is 2^{2n} . For a BLACK block at level k , the contribution to the area is 2^{2k} . Moments are obtained by summing the moments of the BLACK blocks. The position of each BLACK block is easy to ascertain because the path that was taken to reach the block is known when processing starts at the root of the quadtree. Knowledge of the area and the first moments permits the computation of the coordinates of the centroid, and thereupon central moments relative to the centroid can be obtained. It should be noted that all of these algorithms have an execution time proportional to the number of nodes in the quadtree [Shneier 1981a]. Chien and Aggarwal [1984] use a normalized representation of the quadtree with respect to the centroid to match noisy objects against models. This method also relies on the selection of a principal axis and scaling to a fixed resolution.

2.7 Connected Component Labeling

Connected component labeling is one of the basic operations of an image-processing system. It is analogous to finding the connected components of a graph. For example, the image of Figure 24 has two components. Given a binary array representation of an image, the traditional method of performing this operation [Rosenfeld and Pfaltz 1966] would be a

"breadth-first" approach, which scans the image row by row from left to right and assigns the same label to adjacent BLACK pixels that are found to the right and in the downward direction. During this process pairs of equivalences may be generated, thus necessitating two more steps: one to merge the equivalences and the second to update the labels associated with the various pixels to reflect the merger of the equivalences. (However, Lumia [1983] and Lumia et al. [1983] have made improvements on Rosenfeld and Pfaltz's [1966] method of keeping track of equivalences.)

Using a quadtree to perform the same operation involves an analogous three-step process [Samet 1981b]. The first step is a postorder tree traversal (in order NW, NE, SW, SE), where for each BLACK node that is encountered, say A , all adjacent BLACK nodes on the southern and eastern sides of A are found, and assigned the same label as A . The adjacency exploration is done by using the neighbor-finding techniques of Samet [1982a] (see Section 2.1). At times, the adjacent node may already have been assigned a label, in which case the equivalence is noted. The second step merges all the equivalence pairs that were generated during the first step. The third step performs another traversal of the quadtree and updates the labels on the nodes to reflect the equivalences generated by the first two steps of the algorithm.

As an example, consider the image of Figure 24a, whose quadtree block decomposition is given in Figure 24b and c. All blocks are labeled with a different identifying number in their upper left corner; their lower right corner contains the label assigned by the first step of the connected component labeling process. Two items are worthy of further note. First, when block 15 is processed, neither it nor block 31, its southern neighbor, have been labeled yet and thus label B is assigned to them. When block 20 is processed, it has no label, but its southern neighbor, 31, has already been assigned B as a label, and thus block 20 is assigned label B as well. Second, Figure 24b shows the status of the image at the conclusion of the second step of the algorithm. It has three different labels (i.e., A, B, and

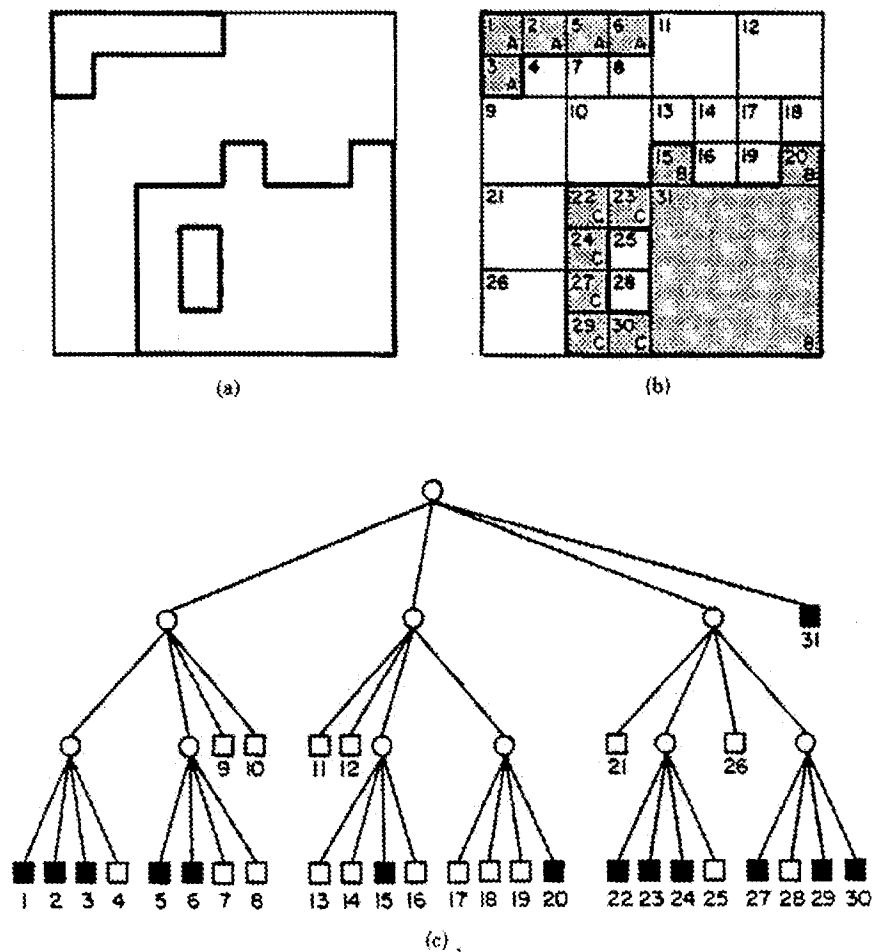


Figure 24. An image, its maximal blocks, and the corresponding quadtree. Blocks in the image are shaded; background blocks are blank. (a) Image. (b) Block decomposition of the image in (a). (c) Quadtree representation of the blocks in (b).

C) with B equivalent to C. This equivalence was generated when the eastern adjacency of block 23 was explored. In essence, block 23 was labeled with C when block 22's eastern adjacency was explored, whereas block 31 was labeled with B when block 15's southern adjacency was explored. Thus we see that the third step of the algorithm will have to be applied, thereby relabeling all C blocks with B.

The execution time of connected component labeling is obtained by examining the three steps of the algorithm. Let B be the number of BLACK nodes in the quad-

tree. Step 1 is a tree traversal in which neighbors are examined as well. Since the average cost of examining a neighbor is a constant [Samet 1982a; Samet and Shaffer 1984], Step 1 is $O(B)$. Step 3 is also a tree traversal and is $O(B)$ as well. Step 2, the merger of equivalence classes, can be done in $O(B \log B)$ time; the algorithm thus has an average execution time that is $O(B \log B)$. In fact, almost linear average execution time can be obtained by combining Steps 1 and 2 by using the UNION-FIND algorithm [Tarjan 1975]. This is a very important result because it means that the exe-

putation time of the connected component labeling process is dependent only on the number of blocks in the image and not on their size. In contrast, the analogous algorithm for the binary array [Rosenfeld and Pfaltz 1966] has an execution time that is proportional to the number of pixels and hence to the area of the blocks. Thus we see that the hierarchical structure of the quadtree data structure saves not only space but time. The cost of the neighbor computation process is avoided when the top-down algorithm of Samet [1984d] is used.

Note that the coloring algorithm of Hunter and Steiglitz [1979a] by itself does not achieve the same effect as the connected component labeling algorithm described above. The coloring algorithm starts with a polygon and traverses its boundary before propagating the boundary color inward and thereby colors the interior nodes. It does not need to merge equivalence classes since the polygon is itself one equivalence class. However, the combination of the coloring algorithm with a polygon identification step will, in effect, yield a connected component labeling algorithm. These results are similar to those achieved using a "depth-first" approach. Use of such an approach leads to algorithms having execution times that are a function of the perimeter of the individual polygons. We also observe that some of the speed of the coloring algorithm is derived from the use of nets that avoid neighbor computation; however, this speed is achieved at the expense of extra storage for the added links as well as the stack.

The algorithm that we have described makes use of a pointer-based quadtree representation. Connected component labeling can also be performed by using some of the pointerless quadtree representations described in Section 2.2. There are two ways to proceed. The first method simply mimics the breadth-first algorithm above and requires use of search to implement the neighbor-finding operation. An alternative method [Samet and Tamminen 1984, 1985] makes use of a staircaselike data structure to remember components on sides of blocks that have already been processed. This

method works for perimeter computation (Section 2.8), component counting (Section 2.9), and also three-dimensional data. Both the linear quadtree [Gargantini 1982a] and the DF-expression [Kawaguchi and Endo 1980] representations can be used.

2.8 Perimeter

Computing the perimeter of an image represented by a quadtree can be done in a manner analogous to Step 1 of the connected component labeling process described in Section 2.7. The difference is that in the previous case, when connected components were labeled, the algorithm searched for adjacent BLACK nodes, whereas adjacent WHITE nodes must be searched for when computing the perimeter. In other words, a postorder tree traversal is performed, and for each BLACK node that is encountered its four adjacent sides are explored in the search for adjacent WHITE nodes. For each adjacent WHITE node that is found the length of the corresponding shared side is included in the perimeter.

Use of such an algorithm will result in a certain amount of duplication of effort because each adjacency between two BLACK blocks is explored twice, and neither of these adjacency explorations contributes to the value of the perimeter (e.g., the eastern side of node N and the western side of node O in Figure 1). An alternative algorithm performs adjacency exploration only for southern and eastern neighbors. That is, for each BLACK node a search is made for adjacent WHITE southern and eastern neighbors, and for each WHITE node, a search is made for adjacent BLACK southern and eastern neighbors. The only problem with such a method is that the northern and western boundaries of the image are never explored. This can be alleviated by embedding the image in a white region, as shown in Figure 25.

Both formulations of the algorithm have expected execution times that are proportional to the number of nodes in the quadtree [Samet 1981c]. Jackins and Tanimoto [1983] have developed an asymptotically faster alternative perimeter computation

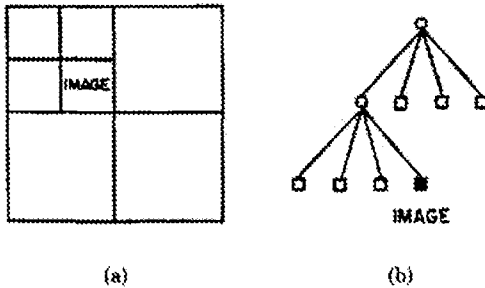


Figure 25. An image totally surrounded by background.

algorithm that works for an arbitrary number of dimensions. This method achieves its efficiency by transmitting the neighbors as parameters rather than by having to rely on neighbor exploration as happens in the approach described above. However, it requires that a separate pass be made over the data for each dimension (see also Samet [1985a]).

2.9 Component Counting

Once the connected components of the image have been labeled, it is easy to count them, since the result is the same as the number of different equivalence classes resulting from Step 2 of the connected component labeling algorithm. An alternative quantity is known as the Euler number or genus, say G , which is $V - E + F$, where V , E , and F correspond to the number of vertices, edges, and faces, respectively, in a planar graph [Harary 1969]. It is defined as the difference between the number of connected components and the number of holes. It is well known [Minsky and Papert 1969] that for a binary image represented by a binary array $G = V - E + F$, where V , E , and F are defined as follows. Let a BLACK pixel be represented by 1 and a WHITE pixel by 0. V is the number of 1's in the image, E is the number of horizontally adjacent pairs of 1's (i.e., 11) or vertically adjacent pairs of 1's, and F is the number of 2 by 2 blocks of 1's.

Dyer [1980] has obtained the same result for a quadtree representation of a binary

image by redefining V , E , and F in the context of a quadtree. This is accomplished by letting V be the number of BLACK blocks, E be the number of pairs of adjacent BLACK blocks in the horizontal and vertical directions (see Figure 26), and F be the number of triples or quadruples surrounding a point, that is, 2 by 2 blocks of pixels that are contained by three or four BLACK blocks (e.g., Figure 27). The algorithm for the computation of the genus is a postorder tree traversal that is analogous to Step 1 of the connected component labeling algorithm with the additional proviso that when F is determined, the leaf nodes surrounding the southeastern corner of a BLACK node must be examined. The algorithm's expected execution time is proportional to the number of blocks in the image.

The value of Dyer's result lies not in the mechanics of the algorithm, but in the theory generated thereby. He has shown that the quadtree representation is hierarchical in the sense that the most critical measure is the number of blocks and not their size. It also demonstrates another instance of the use of an algorithm originally formulated for the binary array representation being used in an analogous manner for a quadtree representation by treating blocks (of possibly different sizes) as if they were pixels. This technique was used previously in the labeling of connected components.

2.10 Space Requirements

The prime motivation for the development of the quadtree has been the desire to reduce the amount of space necessary to store data through the use of aggregation of homogeneous blocks. As the previous discussion has demonstrated, an important by-product of this aggregation has been to decrease the execution time of a number of operations (e.g., connected component labeling and component counting). Nevertheless, the quadtree is not always the ideal representation. The worst case for a quadtree of a given depth in terms of storage requirements occurs when the region corresponds to a checkerboard pattern, as in Figure 28. The amount of space required is

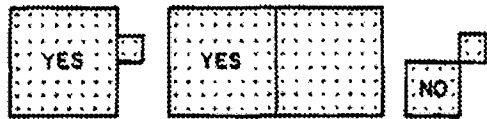


Figure 26. Examples of adjacencies for the computation of the genus of an image.

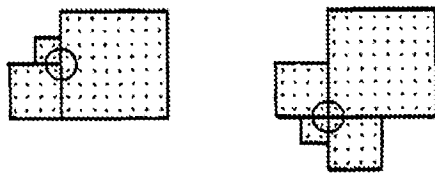


Figure 27. Example of three or four blocks meeting at a corner.

obviously a function of the resolution (i.e., the number of levels in the quadtree). Hunter [1978] presents a fundamental theorem on the space complexity of quadtrees. He has shown that for a simple polygon (i.e., nonintersecting edges) of perimeter p (measured in pixel widths) and a resolution n , the number of nodes in the quadtree is of $O(p + n)$. He goes on to show that $O(p + n)$ is an attainable upper bound. From this, Hunter obtains a corollary of great importance, showing that the quadtree grows linearly in number of nodes as the resolution is doubled, whereas a binary array representation leads to a quadrupling of the number of pixels through doubling the resolution. For experiments on cartographic data that verify the linear growth see Rosenfeld et al. [1982].

The amount of space occupied by a quadtree is extremely sensitive to its orientation. Dyer [1982] has shown that arbitrarily placing a square of size 2^m by 2^m at any position in a 2^n by 2^n image requires an average of $O(2^{m+2} + n - m)$ quadtree nodes. An alternative characterization of this result is that the amount of space necessary is $O(p + n)$, where p is the perimeter (in pixel widths) of the block. Shifting the image within the space in which it is embedded can reduce the total number of nodes. Grosky and Jain [1983] have shown that for a region such that d is the maximum of its horizontal and vertical extent

(measured in pixel widths) and $2^{n-1} < d < 2^n$, the optimal grid resolution is either n or $n + 1$. In other words, embedding the region in a larger area than 2^{n+1} by 2^{n+1} and shifting it around will not result in fewer nodes. This result is used by Li et al. [1982] to obtain an algorithm that finds the configuration of the quadtree requiring a minimum number of nodes. The algorithm proceeds by using a binary array representation of the image and attempting translations of magnitude power of 2 in the vertical, horizontal, and corner directions. When d is defined as above, the algorithm requires $O(2^{2n})$ space and has an execution time that is $O(n \cdot 2^{2n})$.

In Section 2.2 we showed that a tree implementation of a quadtree has overhead in terms of the internal nodes. For an image with B and W BLACK and WHITE blocks, respectively, $(4/3)(B + W)$ nodes are required. In contrast, a binary array representation of a 2^n by 2^n image requires only 2^{2n} bits; however, this quantity grows quite quickly. Furthermore, if the amount of aggregation is minimal (e.g., a checkerboard image), then the quadtree is not very efficient. The overhead for the internal nodes is avoided by using some of the pointerless representations discussed in Section 2.2 such as the linear quadtree and the DF-expression. In fact, the DF-expression requires at most two bits per node. The compression characteristics of DF-expressions compared to boundary and run length codes are discussed by Tamminen [1984b].

2.11 Skeletons and Medial Axis Transforms

The medial axis of a region [Blum 1967; Duda and Hart 1973; Pfaltz and Rosenfeld 1967; Rosenfeld and Kak 1982] is a subset of points, each of which has a distance from the component of the region (e.g., its boundary), using a suitably defined metric, which is a local maximum. The medial axis transform (MAT) consists of the set of medial axis or *skeleton* points and their associated distance values. Before proceeding any further we shall review the definition of a metric.

Let d be a function that maps pairs of points into nonnegative numbers. It is

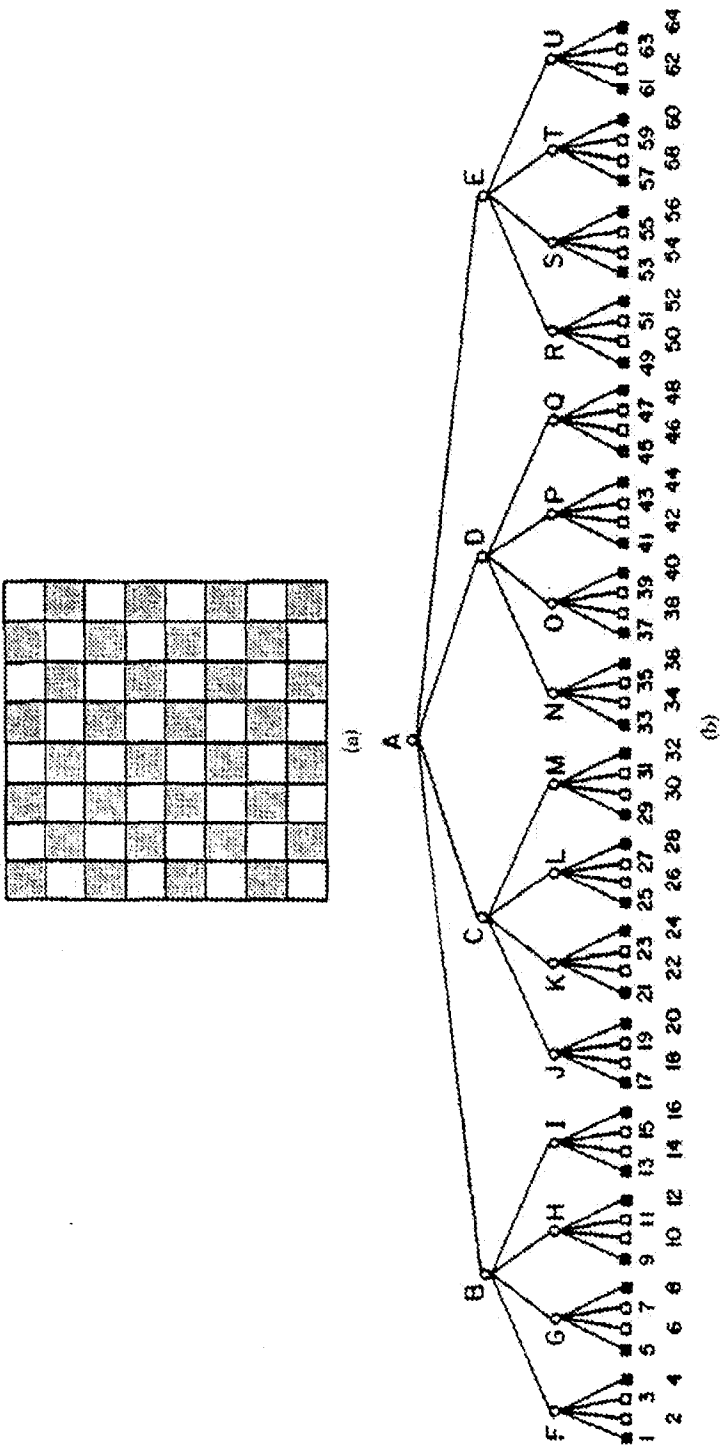


Figure 28. A checkerboard (a) and its quadtree (b).

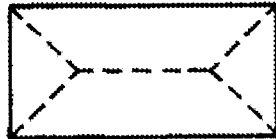


Figure 29. A rectangle and its skeleton using d_x .

called a *metric* or a *distance function* if for all points p , q , and r the following relations are satisfied:

- (1) $d(p, q) \geq 0$ and $d(p, q) = 0$
if and only if $p = q$
(positive definiteness);
- (2) $d(p, q) = d(q, p)$
(symmetry);
- (3) $d(p, r) \leq d(p, q) + d(q, r)$
(triangle inequality).

Some of the more common metrics are examined below in the context of the points $p = (p_x, p_y)$ and $q = (q_x, q_y)$. By far the most popular metric is the *Euclidean metric*

$$d_E(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

Two other metrics which are used in image processing are the absolute value metric, also known as the *city block metric* (or *Manhattan metric*),

$$d_A(p, q) = |p_x - q_x| + |p_y - q_y|,$$

and the maximum value metric, also known as the *Chessboard metric*,

$$d_M(p, q) = \max\{|p_x - q_x|, |p_y - q_y|\}.$$

The set of points having $d_E(p, q) \leq T$ are those points contained in a circle centered at p having radius T . Similarly, $d_A(p, q) \leq T$ yields a diamond, centered at p , with side length $T \cdot \sqrt{2}$, and $d_M(p, q) \leq T$ yields a square, centered at p , with side length $2 \cdot T$. For example, by using the Euclidean metric, the skeleton of a circle is its center. The skeleton of the rectangle in Figure 29 is the set of dashed lines within it. An alternative characterization of a skeleton is achieved by drawing an analogy to a "brush fire." That is, imagine that the boundary of the object is set on fire; the remains would be the skeleton.

Skeletons and medial axis transforms are traditionally used in image processing for the purpose of obtaining an approximation of the image. We wish here to obtain an exact representation of the image. The application of the concept of metric to an image represented by a quadtree is discussed by Shneier [1981b] and Samet [1982b]. In particular, Samet [1982b] has shown that the Chessboard metric is most appropriate for an image represented by a quadtree since it has the property that the set of points $\{q\}$ such that $d_M(p, q) \leq T$ is a square. This metric is used to define the Chessboard distance transform for a quadtree as a function *DIST*, which yields for each BLACK block in the quadtree the Chessboard distance from the center of the block to the nearest point which is on a BLACK-WHITE boundary. In addition, *DIST* of a WHITE or GRAY block is said to be zero, and the border of the image is assumed to be BLACK. For example, in Figure 30, node 1 has a *DIST* value of 6, whereas node 12 has a *DIST* value of 0.5, assuming a 2^4 by 2^4 image. The process of computing the Chessboard distance transform is relatively simple. It consists of a postorder tree traversal where for each BLACK block the eight adjacent (horizontal, vertical, and corner) neighbors are examined to determine the closest WHITE block. This process is analogous to that used for connected component labeling and perimeter computation.

The quadtree skeleton is defined as follows. Given a BLACK block b , it is convenient to use $S(b)$ to refer to the set of pixels in the image spanned by a square with side width $2 \cdot \text{DIST}(b)$ centered about block b . Let the set of BLACK blocks in the image be denoted by B . The *quadtree skeleton* is the set T of BLACK blocks, denoted by t_i , satisfying the following properties:

- (1) the set of pixels in $B = \bigcup_i S(t_i)$;
- (2) for any t_j in T there does not exist b_k in B ($b_k \neq t_j$) such that $S(t_j) \subseteq S(b_k)$;
- (3) for all b_i in B there exists t_j in T such that $S(b_i) \subseteq S(t_j)$.

For example, for the quadtree of Figure 30, the quadtree skeleton consists of nodes

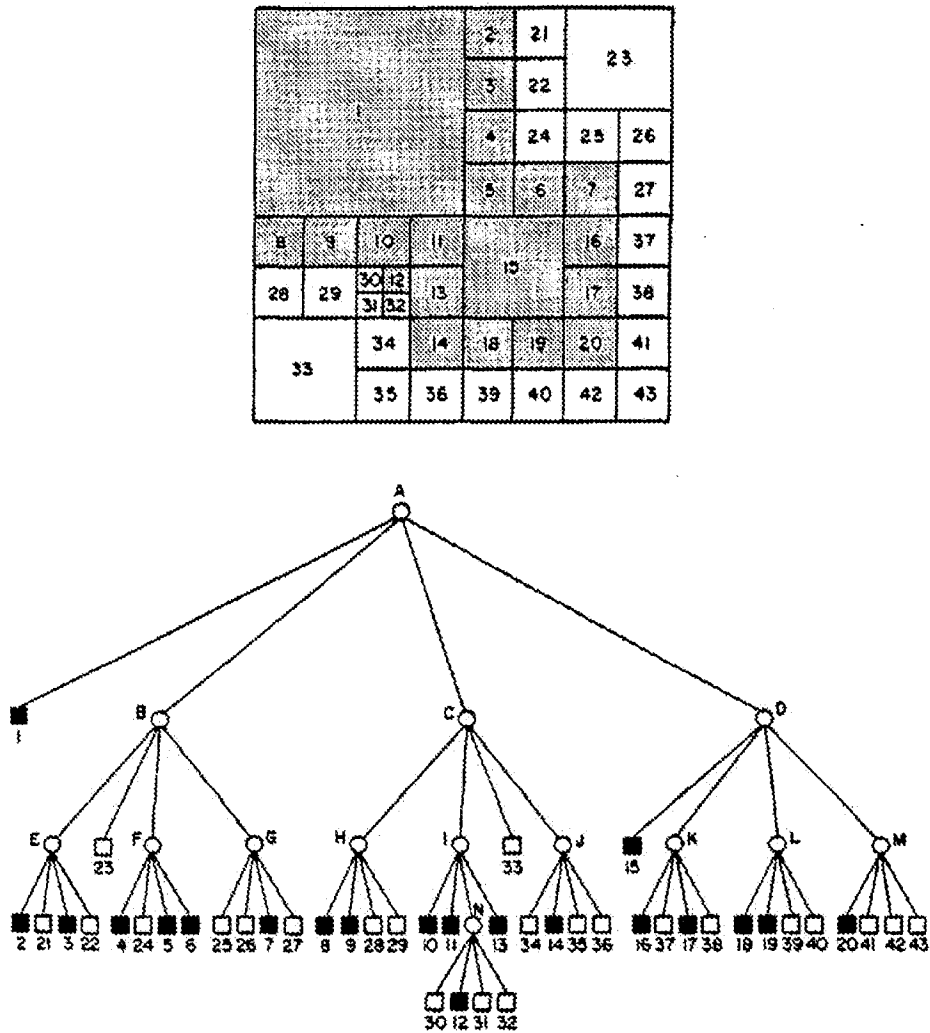


Figure 30. Sample quadtree.

1, 11, and 15 with Chessboard distance transform values of 6, 2, and 4, respectively. Property (1) ensures that the entire image is spanned by the quadtree skeleton. Property (2) is termed the *subsumption* property, wherein b_j is *subsumed* by b_k when $S(b_j) \subseteq S(b_k)$. Property (2) means that the elements of the quadtree skeleton are the blocks with the largest distance transform values. Property (3) ensures that no block in B and not in T requires more than one

element of T for its subsumption. Therefore the case where one-half of the block is subsumed by one element of T and the other half is subsumed by another element of T is not permitted. Samet [1983] has shown that the quadtree skeleton of an image is unique.

The *quadtree medial axis transform* (QMAT) of an image is the quadtree whose BLACK nodes correspond to the BLACK blocks comprising the quadtree skeleton

and their associated Chessboard distance transform values. All remaining nodes in the QMAT are WHITE and GRAY with distance value zero. For example, Figure 31 contains the block and tree representations of the QMAT of Figure 30. The algorithm for the construction of a QMAT from its quadtree is straightforward [Samet 1983]. In essence, it is a modified postorder tree traversal where GRAY nodes are processed first, and for each BLACK node a check is made to see if it is subsumed by one of its eight neighbors. If it is, then the node is changed from BLACK to WHITE. This algorithm is facilitated by Property (3), which ensures that there is no need to check whether a node is subsumed partially by one neighbor and partially by another neighbor. The reverse process of reconstructing a quadtree from its QMAT is also possible [Samet 1985b]. It is potentially useful for thinning an image.

The QMAT has a number of important properties. First, it results in a partition of the image into a set of possibly nondisjoint squares having sides whose lengths are sums of powers of two rather than, as in the case of quadtrees, a set of disjoint squares having sides of lengths that are powers of two. Second, the QMAT is more compact than the quadtree, as it never contains more nodes than the quadtree and often contains considerably fewer nodes (e.g., compare Figures 30 and 31). Third, the QMAT representation is less sensitive to shift operations in the sense that a small shift of the image will not, in general, cause the QMAT to get as large as the shifted quadtree. This should be apparent when we realize that the QMAT is most economical storagewise, vis à vis the quadtree when large blocks are surrounded by smaller blocks; this is normally the situation when a shift operation takes place. For example, compare Figure 32 and the result of shifting it by one pixel to the right, as shown in Figure 33.

2.12 Pyramids

Given a 2^n by 2^n image array, say $A(n)$, a *pyramid* is a sequence of arrays $\{A(i)\}$ such that $A(i-1)$ is a version of $A(i)$ at half the

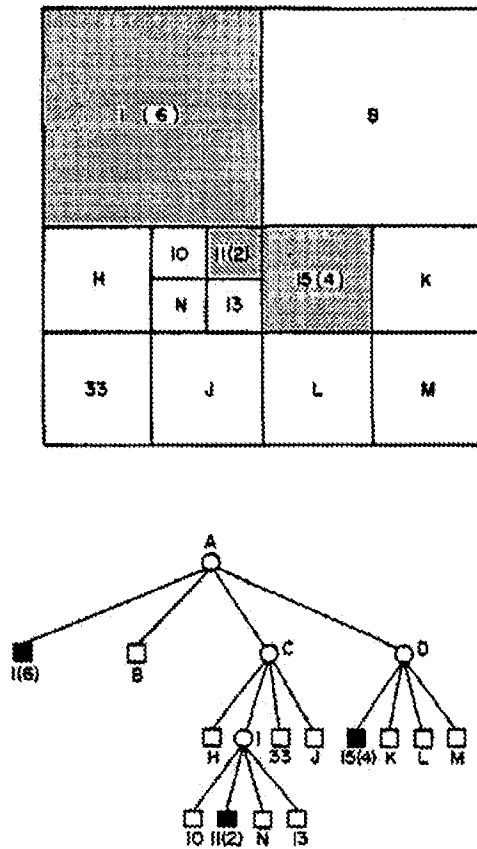


Figure 31. QMAT corresponding to the quadtree of Figure 30.

resolution of $A(i)$, etc. $A(0)$ is a single pixel. For example, Figure 34 shows the structure of a pyramid having three levels. It should be clear that a pyramid can also be defined in a more general way by permitting finer scales of resolution than the power of two scale.

At times it is more convenient to define a pyramid in the form of a tree. Again, assuming a 2^n by 2^n image, a recursive decomposition into quadrants is performed, just as in quadtree construction, except that we keep subdividing until we reach the individual pixels. The leaf nodes of the resulting tree represent the pixels, whereas the nodes immediately above the leaf nodes correspond to the array $A(n-1)$, which is

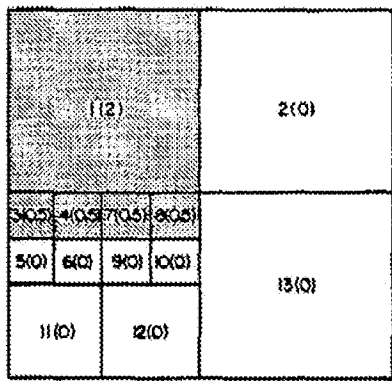
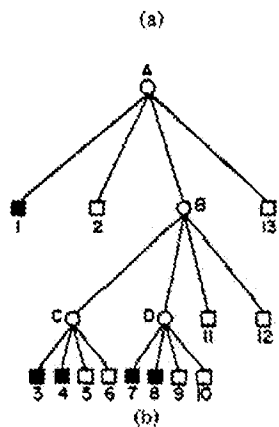
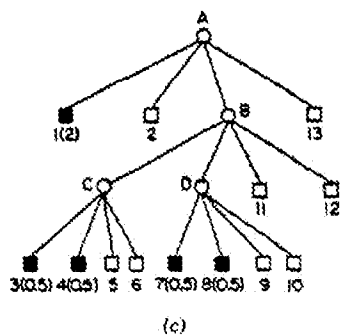


Figure 32. Sample quadtree. (a) Block decomposition. (b) Tree representation. (c) QMAT.

of size 2^{n-1} by 2^{n-1} . The nonterminal nodes are assigned a value that is a function of the nodes below them (i.e., their sons) such as the average gray level.

The above definition of a pyramid is based on a nonoverlapping 2 by 2 blocks of

pixels. An alternative definition uses overlapping blocks of pixels. One of the simplest schemes makes use of 4 by 4 blocks that overlap by 50 percent in both the horizontal and vertical directions [Burt et al. 1981]. For example, Figure 35 is a 2^3 by 2^3 array, say $A(3)$, whose pixels are labeled 1-64. Figure 36 is $A(2)$ corresponding to Figure 35, and its elements are labeled A-P. The 4 by 4 neighborhood corresponding to element F in Figure 36 consists of pixels 10-13, 18-21, 26-29, and 34-37. This method implies that each block at a given level participates in four blocks at the immediately higher level. Thus the containment relations between blocks no longer form a tree. For example, pixel 28 participates in blocks F, G, J, and K in the next higher level (see Figure 37 where the four neighborhoods corresponding to F, G, J, and K are drawn as squares). In order to avoid treating border cases differently, each level is assumed to be cyclically closed (i.e., the top row at each level is adjacent to the bottom row and likewise for the columns at the extreme left and right of each level). Once again, we say that the value of a node is the average of the values of the nodes in its block on the immediately lower level. The overlapped pyramid may be compared with the QMAT (see Section 2.11) in the sense that both may result in nondisjoint decompositions of space.

Pyramids are used for feature detection and extraction since they can be used to limit the scope of the search. Once a piece of information of interest is found at a coarse level, the finer resolution levels can be searched. This approach was used for approximate pattern matching by Davis and Roussopoulos [1980]. Pyramids can also be used for encoding information about edges, lines, and curves in an image [Shneier 1981c]. One note of caution: The reduction of resolution will affect the visual appearance of edges and small objects [Tanimoto 1976]. In particular, at a coarser level of resolution edges tend to get smeared and region separation may disappear. Pyramids have also been used as the starting point for a "split and merge" segmentation algorithm [Pietikainen et al. 1982].

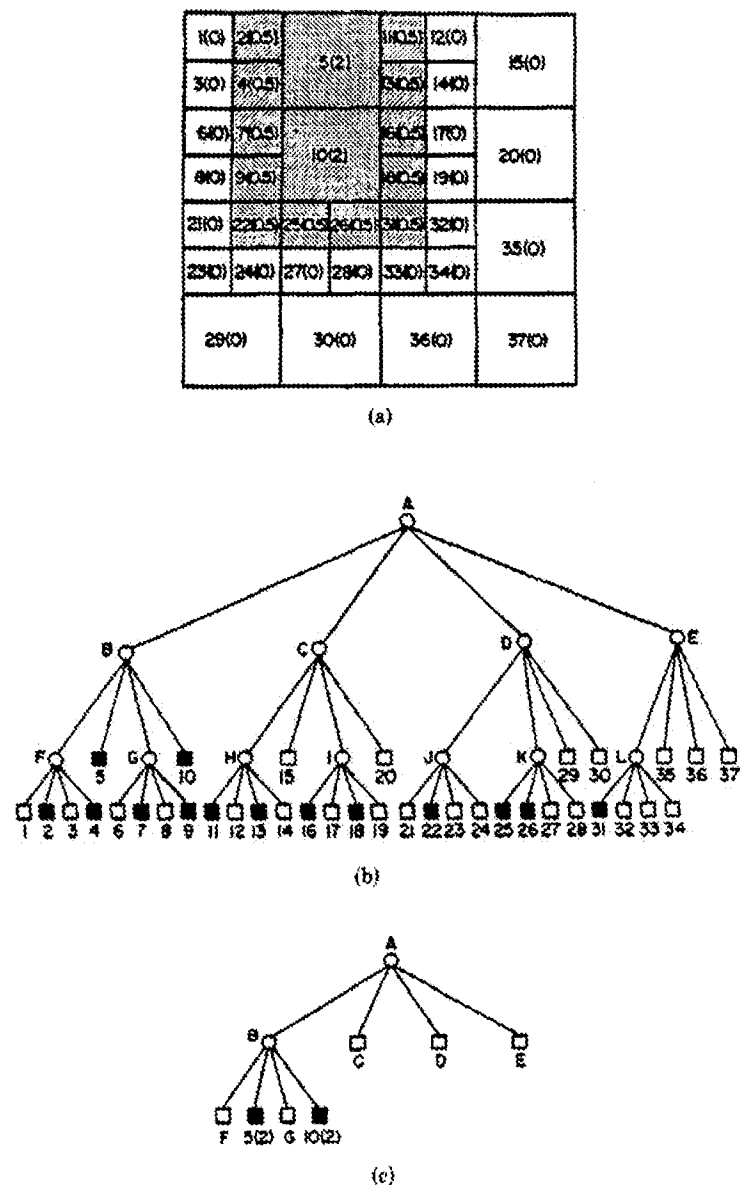


Figure 33. The result of shifting the image in Figure 32 by one pixel to the right. (a) Block decomposition. (b) Tree representation. (c) QMAT.

Before leaving this section, it is important to reiterate a comment made in Section 1 that pyramids and quadtrees, although related, are different entities. A pyramid is a multiresolution representation, whereas the quadtree is a variable resolution representation. Another analogy

is that the pyramid is a complete quadtree [Knuth 1975, p. 401].

2.13 Quadtree Approximation Methods

The quadtree can be used as an image approximation device. By truncating the

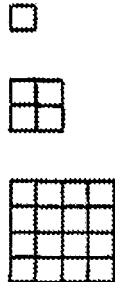


Figure 34. Structure of a pyramid having three levels.

quadtree (i.e., ignoring all nodes below a certain level), a crude approximation can be realized. Ranade et al. [1982] define a sequence of inner and outer approximations to an image and use it for shape approximation. The inner approximation consists of treating GRAY nodes as WHITE, whereas the outer approximation treats them as BLACK. Rosenfeld et al. [1982] discuss a quadtree truncation technique that treats a GRAY node as BLACK or WHITE, depending on the type of the majority of its constituent pixels. This is shown to lead to a very gentle degradation of the image in contrast to the abruptness of the inner and outer approximation methods.

Quadtree-based approximation methods have also been devised for use in transmission of binary and gray-scale images. In such a case, it is desirable for the chosen method to exhibit compression as well as be progressive. By progressive we mean that as more of the image is transmitted, the receiving device progressively constructs a better approximation. At the end of the transmission, the original image is to be reconstructed perfectly. Progressive approximation should be contrasted with facsimile techniques that transmit the image a line at a time. Thus the goal is to receive a crude picture first and the details later, thereby enabling browsing operations.

Sloan and Tanimoto [1979] (see also Tanimoto [1979]) propose a number of pyramid-based approaches to the problem of transmitting a gray-scale image. In Section 2.12 a pyramid was described as a sequence

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	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 35. Example pyramid $A(3)$.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Figure 36. $A(2)$ corresponding to Figure 35.

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	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 37. The overlapping blocks in which pixel 28 participates.

of arrays $\{A(i)\}$ such that $A(i - 1)$ is a version of $A(i)$ at half the resolution; $A(0)$ is said to be a single element. Given a 2^n by 2^n image, the pyramid can be considered to be a complete quadtree with $A(n)$ corresponding to the image. The simplest transmission technique that they propose is analogous to a breadth-first traversal of the complete quadtree. The shortcoming of this approach is that redundant information must be transmitted (i.e., one-third more information), and thus no compression exists. Sloan and Tanimoto propose a number of refinements to this method. First, a level number and coordinates for each pixel are included, but are transmitted only if they

differ from the value of the pixel's predecessor. The second refinement requires that the receiver deduce one pixel's value from those of its predecessor and its three siblings. By using such a method, there is no need for redundant pixel information to be transmitted; however, there is no compression since the amount of information transmitted is equal to the number of pixels. The predecessor's value can be a sum of the values of the four sons; an even better method, in the sense that less computational overhead is involved, is simply to use one of the values of the sons.

Knowlton [1980] discusses techniques for transmission of both gray-scale and binary images. He makes use of a binary tree version of a quadtree (i.e. bintree). In essence, an image is repeatedly split into two halves alternating between vertical and horizontal splits. For gray-scale images, he describes each two-pixel group (hence the binary subdivision) by two numbers such that the first is analogous to an average, termed a *composite value*, and the second is like a differentiator that enables the computation of their corresponding intensities. These two pixel groups are recursively aggregated in groups of two to form a binary tree. Knowlton shows that all that needs to be transmitted is the composite value for the root of the tree and the successive sets of differentiators. Thus the sequence of transmission is a breadth-first traversal of the binary tree of differentiator values. The result is that an image of p pixels of 2^b gray levels can be transmitted and reformatted by using $p \cdot b$ bits. Use of the composite values leads to successively better approximations to the image until an exact reconstruction is obtained at the end of the transmission. Compression can be achieved by using Huffman codes [Huffman 1952] to encode the differentiator values.

Knowlton also presents a technique for the progressive transmission of binary images. Again, the binary tree version of the quadtree is used to represent the image. Nodes are labeled BLACK, WHITE, or GRAY. The basic unit of decomposition is a pixel and these are aggregated into 2 by 3 rectangles. At this level all nodes are described by using a seven-valued entity

corresponding to the number of constituent BLACK pixels. The image is transmitted in order of a breadth-first traversal of the binary tree. Whenever a block of size greater than 2 by 3 is described as BLACK or WHITE, it ceases to participate in the remainder of the transmission process. In order to obtain an approximation, two values are transmitted for the 2 by 3 GRAY blocks. The first set of values is just a five-valued number indicating the shade of GRAY for each block. Next, the exact details of each pixel in each block are transmitted. Knowlton makes considerable use of coding methods to obtain compression factors as high as 8:1 (i.e., for a 2^n and 2^n image, instead of transmitting 2^{2n} bits, as few as 2^{2n-3} bits are necessary). It should be noted that these high compression factors do not necessarily result from the use of a bintree over a quadtree. Instead, they result from the uniformity of the image (i.e., blocks of WHITE and BLACK), and when this is not the case, then they result from coding groups of pixels. Somewhat similar compression results, although they do not exhibit progressiveness, have been obtained by using DF-expressions [Kawaguchi and Endo 1980].

The notion of a forest (see Section 2.2) has been extended by Samet [Samet 1985c] to develop a sequence of approximations to a quadtree-encoded binary image that also exhibits compression as well as progressiveness. The approximation sequence consists of using the roots of the elements of the forest of Jones and Iyengar [1984]. Each successive approximation constructs a new forest for each element of the previous approximation that is not a terminal node. The only difference is that the second approximation uses a forest of maximal squares that span the WHITE area of the components of the first approximation. This process is repeated, alternating BLACK and WHITE approximations until all elements of the approximation are terminal nodes. This method works by alternating an overestimation of the BLACK area with an underestimation, and spiraling in to the true image. Thus it avoids the one-sidedness of the inner and outer approximation methods of Ranade et al.

[1982]. The nodes comprising the elements of the approximation sequences are specified by use of locational codes.

It can be shown that use of forest-based approximation methods leads to a number of interesting properties. First, the total number of nodes in the approximation sequence does not exceed the minimum of the BLACK or WHITE nodes in the original quadtree. Using a 512 by 512 image, reductions as high as 22 percent (with respect to the minimum of the BLACK or WHITE nodes) in the number of nodes have been obtained [Samet 1985c]. As larger images are used, the compression factor becomes considerably greater. Second, the methods yield a saving of space whenever the situation arises that three out of four sons have the same type (i.e., BLACK or WHITE). The worst-case scenario from a node-counting standpoint is the checkerboard (all the BLACK nodes must be transmitted). Finally, we observe that the forest method is biased in favor of approximating objects having the shape of a "panhandle," whereas the inner and outer approximations [Ranade et al. 1982] are insensitive to them.

Ismail and Steele [1980] make use of an approximation method, termed *aplc*, that is similar in spirit to the forest method [Samet 1985c]. They treat each M by M block in the image as BLACK if at least $M^2 - 1$ of its constituent pixels are BLACK. Similarly, a M by M block is treated as WHITE if at least $M^2 - 1$ of its pixels are WHITE. Otherwise, the block is decomposed into four blocks, and the same coding process is recursively attempted. The principal difference between the two methods is that the forest method is hierarchical, whereas the *aplc* approximation is not in that if four brother blocks of size 2 by 2 each contain one WHITE pixel, they are treated as four 2 by 2 BLACK blocks and not as one 4 by 4 BLACK block. Also, the *aplc* method does not lead to an exact reconstruction of the image, whereas the forest method does.

2.14 Volume Data

Extension of the quadtree to represent three-dimensional objects by use of octrees

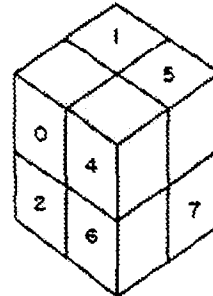


Figure 38. Labeling of octants in an octree (octant 3 is not visible).

has been proposed independently by many researchers [Hunter 1978; Jackins and Tanimoto 1980; Meagher 1982; Reddy and Rubin 1978]. The process begins with a 2^n by 2^n by 2^n object array of unit cubes or voxels [Jackins and Tanimoto 1980] (also termed obels [Meagher 1982]). The octree is an approach to object representation similar to the region quadtree, and is based on the successive subdivision of an object array into octants. If the array does not consist entirely of 1's or entirely of 0's, then it is subdivided into octants, suboctants, etc. until cubes (possibly single voxels) are obtained that consist of 1's or of 0's; that is, they are entirely contained in the region or entirely disjoint from it. This process is represented by a tree of out degree 8 in which the root node represents the entire object with octants labeled as in Figure 38, and the leaf nodes correspond to those cubes of the array for which no further subdivision is necessary. Leaf nodes are said to be BLACK or WHITE (alternatively, FULL or VOID), depending on whether their corresponding cubes are entirely within or outside of the object, respectively. All nonleaf nodes are said to be GRAY. Figure 39 contains an example object in the form of a staircase and its corresponding octree. The labels denote the octant numbers associated with each son by using the labeling convention of Figure 38.

Many of the algorithms obtained for the region quadtree, for example, the Boolean operations, are easily extended to the octree domain. Jackins and Tanimoto [1980] have

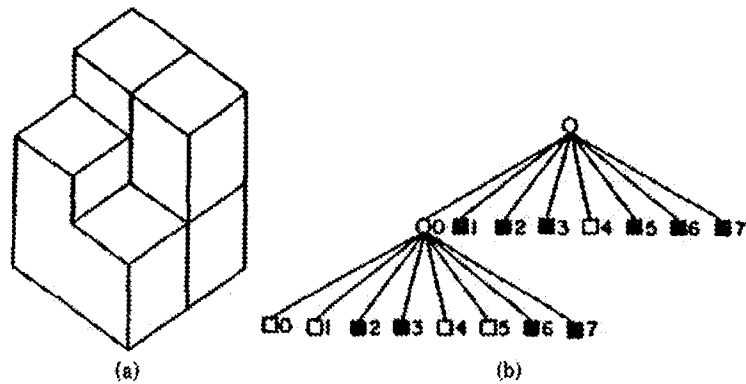


Figure 39. Example object (a) and its octree (b). ■ = BLACK = "Full"; □ = WHITE = "VOID" (empty); ○ = GRAY.

adapted Hunter and Steiglitz's [1979a] translation algorithm to the three-dimensional domain. They also discuss rotation by multiples of 90 degrees. Meagher [1982] and Ahuja and Nash [1984] discuss different approaches. Meagher [1982] also describes algorithms to perform scaling, rotation, perspective transformation, and hidden surface display. He observes that memory and processing time requirements for operations involving three-dimensional objects are proportional to the surface area. This is analogous to Hunter and Steiglitz' [1979a] observation that for quadrees these requirements are proportional to the perimeter of the object being represented. Tamminen and Samet [1984] show how to convert a boundary representation of a solid to its corresponding octree by use of connectivity labeling. Gillespie and Davis [1981] discuss the projection of an octree onto a plane formed by the axes resulting in a quadtree. This is useful for display purposes (see also Doctor and Torborg [1981]). Yau [1984] solves the projection problem for sections orthogonal to a coordinate axis as well as the more general problem of a projection onto a plane of arbitrary position and orientation. Yau and Srihari [1983] present an algorithm for constructing an N -dimensional octree-like representation from multiple $(N - 1)$ -dimensional cross-sectional images for use in processing medical images. Connolly [1984] treats a similar problem by using range data. Gargantini [1982b] makes use of a

pointerless representation termed a *linear octree* (analogous to the linear quadtree [Gargantini 1982a]) and shows how a number of primitive operations can be performed.

Reddy and Rubin [1978] discuss three representations for solid objects, one of which is the octree. The second is a three-dimensional generalization of the point quadtree [Finkel and Bentley 1974], that is, a decomposition into rectangular parallelepipeds (as opposed to cubes) with planes perpendicular to the x , y , and z axes. The third breaks the object into rectangular parallelepipeds that are not necessarily aligned with an axis. The parallelepipeds are of an arbitrary size and orientation. The top level of the tree has a branching factor of N . At this level of the tree, they store N transformation matrices, T_1 through T_N , where each matrix is a 4 by 4 transformation that converts the object space point into the coordinate system of its parallelepiped. Each parallelepiped is recursively subdivided into parallelepipeds in the coordinate space of the enclosing parallelepiped. Reddy and Rubin prefer the third approach for its ease of display. Situated somewhere in between the second and third approaches outlined above is the method of Brooks and Lozano-Perez [1983] (see also Lozano-Perez 1981), who use a recursive decomposition of space into an arbitrary number of rectangular parallelepipeds, with planes perpendicular to the x , y , and z axes, to model space in solving the *findpath* or

mover's problem in robotics. This problem arises when planning the motion of a robot in an environment containing known obstacles and the desired solution is a collision-free path that is obtained by use of a search. Faverjon [1984] discusses an approach to this problem that uses an octree.

Faugeras and Ponce [1983] describe a hierarchical data structure that they term a *prism tree*. It is a ternary tree structure that is a generalization of the strip tree of Ballard [1981] (see Section 4) to hierarchically approximate surfaces by using a prism as an enclosing box. The prism tree is built from an initial triangulation of an object by using a polyhedral approximation algorithm [Faugeras et al. 1984]. Algorithms are presented for intersecting surfaces and finding neighbors in the sense of Samet [1982a].

Representing surfaces (i.e., $2\frac{1}{2}$ -dimensional images) by hierarchical methods is an interesting area in which, unfortunately, only a limited amount of work has been done. DeFloriani et al. [1982] discuss a data structure for multilevel surface representation consisting of a nested triangulated irregular network [Lee and Schachter 1980] that is used for surface interpolation and also serves as a data compression mechanism. Gomez and Guzman [1979] use a data structure that is somewhat related to the point quadtree. It is a recursive subdivision of the surface into four triangles of unequal size, which uses a process that stops when a triangle matches the surface within a prespecified error. Carlson [1982] describes a quadtree-based data structure for representing surfaces to be used in the synthesis of three-dimensional objects in the domain of computer graphics. In general, the principle of recursive subdivision is of considerable importance in the processing of curved surfaces [Cohen et al. 1980]. (See also Mudur and Koparkar [1984].)

Multidimensional data in excess of three dimensions can also be represented by n -dimensional generalizations of the quadtree. Also interesting is the use of the fourth dimension to represent time [Gillespie and Davis 1981; Jackins and Tanimoto 1983; Yau and Srihari 1983]; however, this representation is somewhat difficult since the

dimensional units of the extra dimension are different (i.e., units of time instead of distance). Such techniques are potentially useful in dealing with time-varying imagery.

3. POINT DATA

Multidimensional point data can be represented in a variety of ways. The representation ultimately chosen for a specific task will be heavily influenced by the type of operations to be performed on the data. Our focus is on dynamic files (i.e., the number of data can grow and shrink at will) and applications involving search. Knuth [1973] lists three typical queries: (1) a *point query*, which determines whether a given data point is in the database, and if so, the address corresponding to it; (2) a *range query* (i.e., region search), which asks for a set of data points within a given range (this category includes the partially specified query); (3) a *Boolean query*, which consists of the previous type combined with the Boolean operations AND, OR, NOT, etc. A related operation is to find the n nearest neighbors of a given point [Bentley 1975a].

Nievergelt et al. [1984] group searching techniques into two categories: those that organize the data to be stored and those that organize the embedding space from which the data are drawn. In a more formal sense, the distinction is between *trees* and *tries*, respectively. The binary search tree [Knuth 1973] is an example of the former since the boundaries of different regions in the search space are determined by the data being stored. Address computation methods such as radix searching [Knuth 1973] (also known as digital searching) are examples of the latter, since region boundaries are drawn at locations that are fixed regardless of the content of the file. The distinction can also be seen by comparing the region quadtree [Klinger 1971] with the point quadtree [Finkel and Bentley 1974]; that is, the former is based on a regular decomposition, whereas the latter is not.

In the remainder of this section we further elaborate on the point quadtree and the k -d tree [Bentley 1975b]. Then, some representations that are based on the re-

gion quadtree (i.e., on a regular decomposition) are discussed and compared with the point quadtree. We also present an application of a region-based quadtree in representing small rectangles for very large-scale integration (VLSI) applications [Kedem 1981], concluding with a brief overview of methods that replace the hierarchical structure of quadtrees by address computation. These techniques are directed, in part, toward ensuring efficient access to disk data, and are termed *bucket methods*. In the same context some tree-based methods are also discussed. All of the examples are limited to two dimensions although they can be easily generalized to an arbitrary number of dimensions. It should be borne in mind that our presentation is very brief; that is, we do not analyze the performance of these methods. Actually, the field of multidimensional data structures is a rapidly developing one, and this discussion is necessarily limited to a detailed presentation of methods that can be viewed as direct applications of a quadtree-like recursive subdivision approach.

3.1 Point Quadtrees and k -d Trees

The point quadtree [Finkel and Bentley 1974] is a multidimensional generalization of a binary search tree. In two dimensions, each data point is a node in a tree having four sons, which are roots of subtrees corresponding to quadrants labeled in order NW, NE, SW, and SE. Each data point is assumed to be unique. The process of inserting into point quadtrees is analogous to that used for binary search trees. In essence, we search for the desired record on the basis of its x and y coordinates. At each node of the tree a four-way comparison operation is performed and the appropriate subtree is chosen for the next test. Reaching the bottom of the tree without finding the record means that it should be inserted at this position. The shape of the resulting tree depends on the order in which records are inserted into it. For example, the tree in Figure 2 is the point quadtree for the sequence Chicago, Mobile, Toronto, Buffalo, Denver, Omaha, Atlanta, and Miami. Deletion of a node is more complex [Samet

1980c] as is balancing [Overmars and van Leeuwen 1982].

Point quadtrees are especially attractive in applications that involve search. However, they have also been used to solve a measure problem for rectangular ranges in three-space [van Leeuwen and Wood 1981]. A typical query is one that requests the determination of all records within a specified distance of a given record, that is, all cities within 50 miles of Washington, D.C. The efficiency of the point quadtree lies in its role as a pruning device on the amount of search that is required. Thus many records will not need to be examined. For example, suppose that in the hypothetical database of Figure 2 we wish to find all cities within eight units of a data point with coordinates (83, 10). In such a case, there is no need to search the NW, NE, and SW quadrants of the root (i.e., Chicago with coordinates (35, 40)). Thus we can restrict our search to the SE quadrant of the tree rooted at Chicago. Similarly, there is no need to search the NW and SW quadrants of the tree rooted at Mobile (i.e., coordinates (50, 10)). Search operations using point quadtrees are analyzed by Bentley and Stanat [1975] and Lee and Wong [1977]. Note that the search ranges are usually orthogonally defined regions such as rectangles or boxes. Other shapes are also feasible as the above example demonstrated (i.e., a circle). In order to handle more complex search regions such as polygons, Willard [1982] defines a *polygon tree* where the x - y plane is subdivided by J lines that need not be orthogonal, although there are other restrictions on these lines. When $J = 2$, the result is a point quadtree with nonorthogonal axes.

Our examples of the use of the point quadtree have been limited to two dimensions. The problem with a large number of dimensions is that the branching factor becomes very large (i.e., 2^k for k dimensions), thereby requiring much storage for each node as well as many NIL pointers for terminal nodes. The k -d tree of Bentley [Bentley 1975b] is an improvement on the point quadtree that avoids the large branching factor. In principle, it is a binary search tree with the distinction that at each

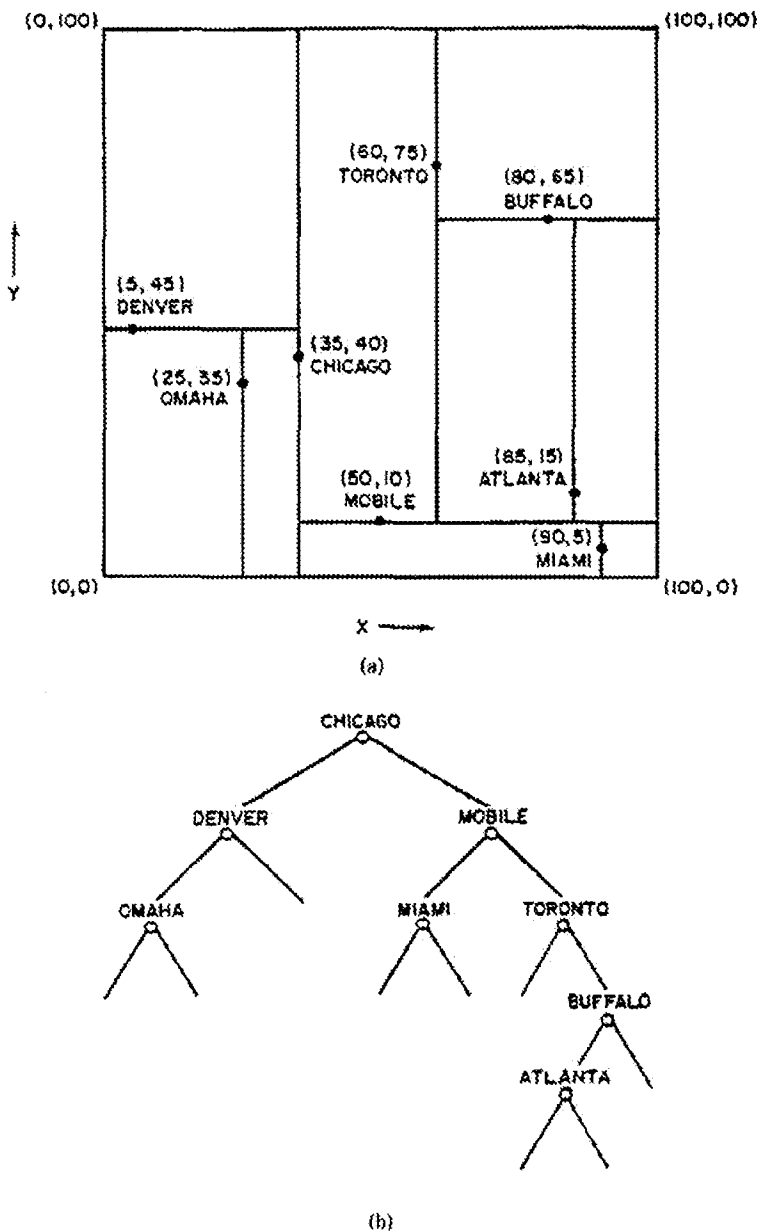


Figure 40. A k -d tree (b) and the records it represents (a).

level of the tree a different coordinate is tested when determining the direction in which a branch is to be made. Therefore in the two-dimensional case (i.e., a 2-d tree!), we compare x coordinates at the root and at even levels (assuming the root is at level 0) and y coordinates at odd levels. Each node has two sons. Figure 40 is the k -d tree corresponding to the point quadtree of Figure 2, where the records have been inserted

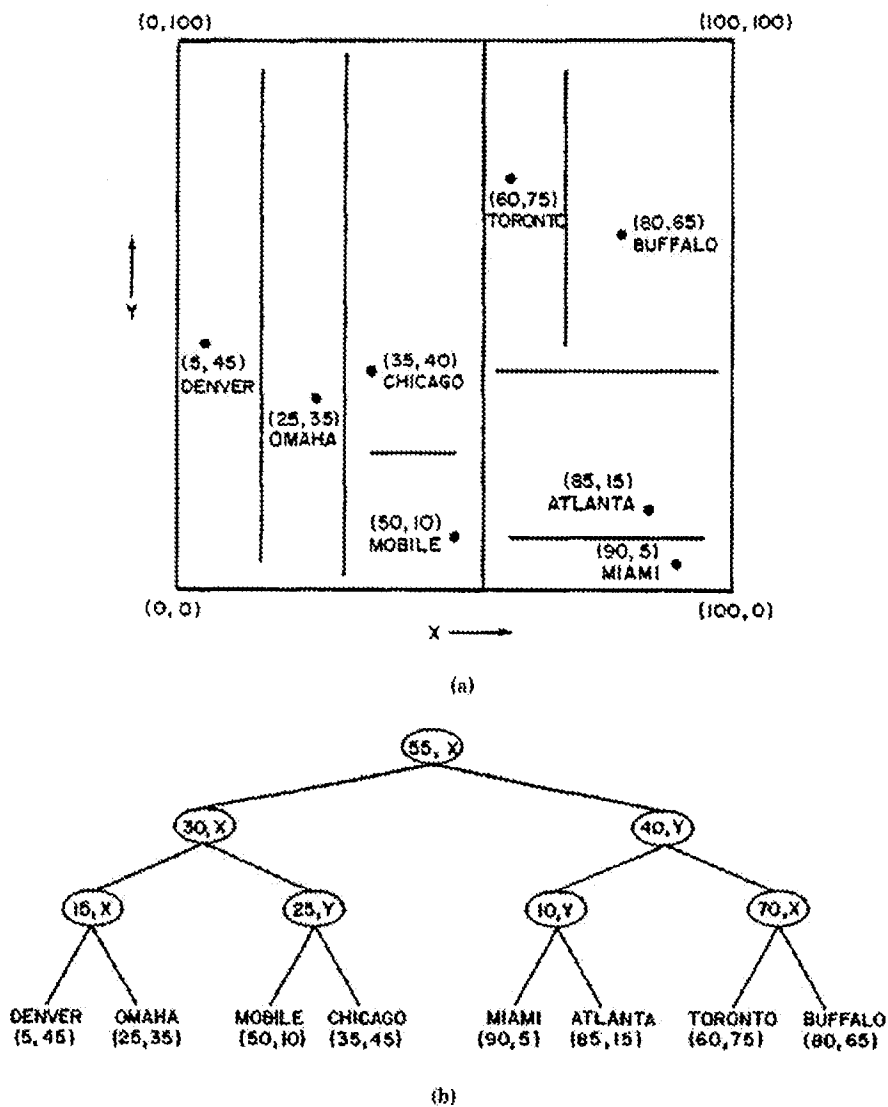


Figure 41. Adaptive k -d tree. (a) Set of points in 2-space. (b) 2-d tree.

in the same order. Friedman et al. [1977] report an improvement on the k -d tree that relaxes the requirement of alternating tests at the price of storing at each node an indication of which coordinate is being tested. Using this data structure, termed an *adaptive k -d tree*, we can construct a balanced k -d tree where records are stored only at the terminal nodes. Figure 41 is the adaptive k -d tree corresponding to the point

quadtree of Figure 2. Before constructing such a tree we must know all of the constituent records. Thus its shape is independent of the order in which the records were encountered. However, adding a new record requires rebuilding the tree. Thus it is not a dynamic data structure.

In general, k -d trees are superior to point quadtrees, with one exception: The point quadtree is an inherently parallel data

structure and thus the comparison operation can be performed in parallel for the k key values, whereas this cannot be done for the k -d tree. Thus we can characterize the k -d tree as a superior serial data structure and the point quadtree as a superior parallel data structure. Linn [1973] discusses the use of point quadtrees in a multiprocessor environment.

3.2 Region-Based Quadtrees

Although conceivably there are many ways of adapting the region quadtree to represent point data, our discussion is limited to two methods. The first method assumes that the domain of data points is discrete; they are treated as if they are BLACK pixels in a region quadtree. An alternative characterization is to think of the data points as nonzero elements in a square matrix. The resulting data structure is called an *MX quadtree* (MX for matrix), although the term *MX quadtree* would probably be more appropriate. The MX quadtree is organized in a similar way to the region quadtree. The difference is that leaf nodes are BLACK or empty (i.e., WHITE) corresponding to the presence or absence, respectively, of a data point in the appropriate position in the matrix. For example, Figure 42 is the 2^3 by 2^3 MX quadtree corresponding to the data of Figure 2. It is obtained by applying the mapping f such that $f(Z) = Z \text{ div } 12.5$ to both x and y coordinates. The result of the mapping is reflected in the coordinate values in the figure.

Each data point in an MX quadtree corresponds to a 1 by 1 square. For ease of notation and operation using modulo and integer division operations, the data point is associated with the lower left corner of the square. This adheres to the general convention followed throughout this presentation that the NE and SE quadrants are closed with respect to the x coordinate and the NW and NE quadrants are closed with respect to the y coordinate. Note that nodes corresponding to data points are not merged, whereas this is not the case for empty leaf nodes. For example, the NW and NE sons of node D in Figure 42 are NIL and likewise for the NW son of node

A. However, it is undesirable to merge nodes corresponding to data points as this results in a loss of the identifying information about the data points. Recall that each data point is different, whereas the empty leaf nodes have the absence of information as their common property and thus can be safely merged.

Data points are inserted into an MX quadtree by searching for them. This search is based on the location of the data point in the matrix (e.g., the discretized values of its x and y coordinate in the example of Figure 42). An unsuccessful search terminates at a leaf node. If this leaf node is NIL, the space spanned by it may have to be repeatedly subdivided until it is a 1 by 1 square. This process is termed *splitting* and for a 2^n by 2^n MX quadtree, it will have to be performed at most n times. The shape of the MX quadtree is independent of the order in which data points are inserted into it. Deletion of nodes is slightly more complex and may require collapsing of nodes—the direct counterpart of the node-splitting process outlined above.

The MX quadtree is useful in a number of applications. It serves as a basis of a quadtree matrix manipulation system [Samet and Krishnamurthy 1983]. It is used by Letelier [1983] to represent silhouettes of hand motions to aid in the telephonic transmission of sign language for the hearing impaired. DeCoulon and Johnsen [1976] describe its use in the coding of black and white facsimiles for efficient transmission.

The MX quadtree is adequate as long as the domain of the data points is discrete and finite. If this is not the case, then the data points cannot be represented since the minimum separation between the data points is unknown. This leads us to an alternative adaptation of the region quadtree to point data that associates data points (that need not be discrete) with quadrants. We call it a *PR quadtree* (P for point and R for region) although again the term *PR quadtree* would probably be more appropriate. The PR quadtree is organized in the same way as the region quadtree. The difference is that leaf nodes are either empty (i.e., WHITE) or contain a data

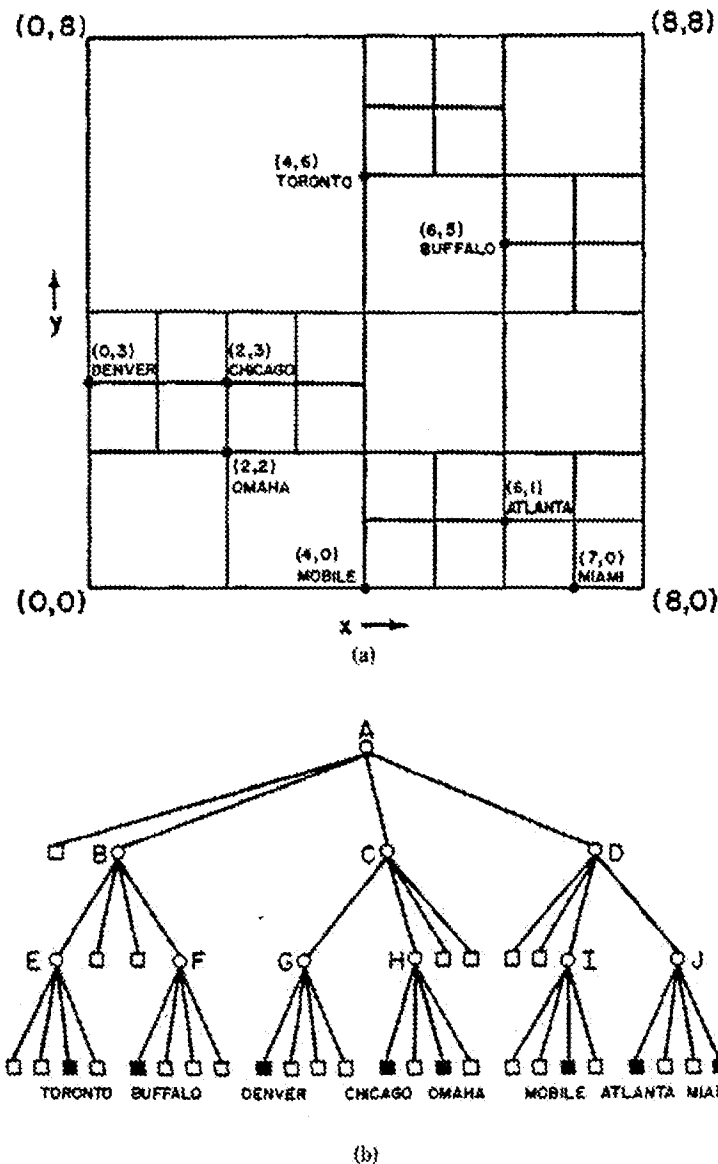


Figure 42. A MX quadtree (b) and the records it represents (a).

point (i.e., BLACK) and its coordinates. A quadrant contains at most one data point. For example, Figure 43 is the PR quadtree corresponding to the data of Figure 2. Orenstein [1982] describes an analogous data structure using binary trees rather than quadtrees. Such a data structure could be

called a *k-d PR quadtree* or even better simply a *k-d trie*. Data points are inserted into PR quad-trees in a manner analogous to that used to insert in a point quadtree; that is, a search is made for them. Actually, the search is for the quadrant in which the data point,

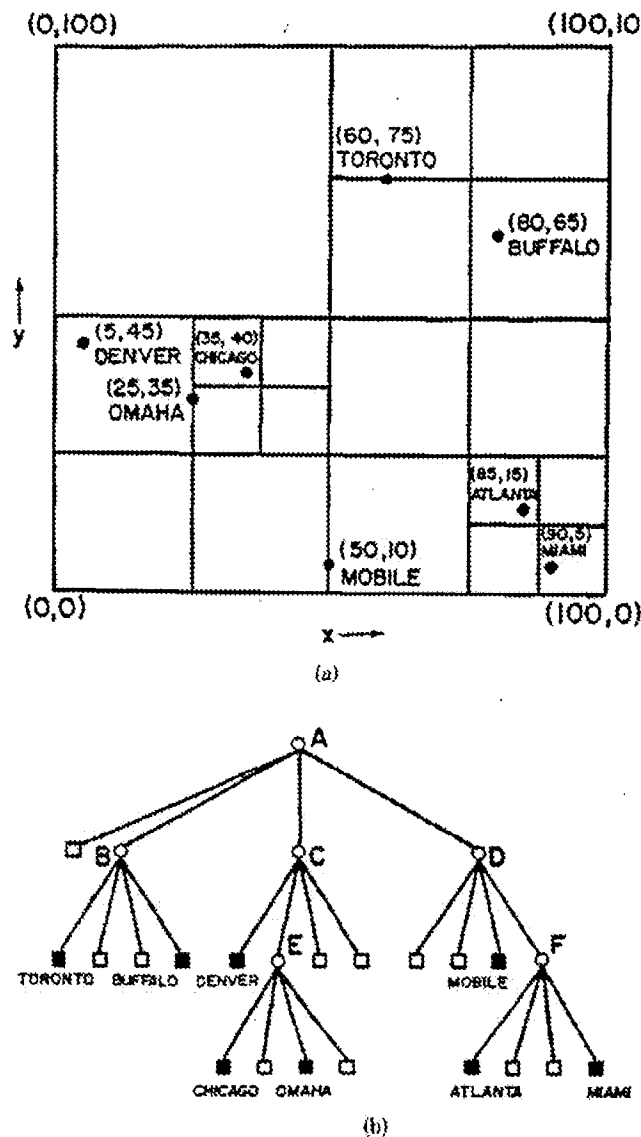


Figure 43. A PR quadtree (b) and the records it represents (a).

say *A*, belongs (i.e., a leaf node). If the quadrant is already occupied by another data point with different *x* and *y* coordinates, say *B*, then the quadrant must repeatedly be subdivided (termed *splitting*) until nodes *A* and *B* no longer occupy the same quadrant. This may result in many subdivisions, especially if the Euclidean distance between *A* and *B* is very small.

The shape of the resulting PR quadtree is independent of the order in which data points are inserted into it. Deletion of nodes is more complex and may require collapsing of nodes, that is, the direct counterpart of the node-splitting process outlined above.

Matsuyama et al. [1984] discuss the use of a PR quadtree in partitioning a point space into “buckets” of a finite capacity. As

a bucket overflows, a partition into four, equal-sized squares is made. Anderson [1983] makes use of a PR quadtree (termed a *uniform* quadtree) to store endpoints of line segments to be drawn by a plotter. The goal is to minimize pen plotting time by choosing the line segment to be output next whose end point is closest to the current pen position. Samet and Webber [1983] represent polygonal maps, for example, Voronoi diagrams, using a variant of the PR quadtree. It has the advantage that edges are represented exactly, thereby avoiding the edge width problem associated with the methods of Hunter and Steiglitz [1979a] for polygons.

3.3 Comparison of Point Quadrees and Region-Based Quadrees

The comparison of the MX, PR, and point quadrees reduces, in part, to a comparison of their respective decomposition methods. A major difference among the three data structures is in the size of the regions associated with each data point. For the point quadtree there is no a priori constraint on the size of the space spanned by the quadtree (i.e., the x and y coordinates of the data points). For both the MX and PR quadrees the space spanned by the quadtree is constrained to a maximum width and height. All three quadrees result in the association of one rectangular region with each data point. The point quadtree produces a rectangle that may, at times, be of infinite width and height. For the MX quadtree this region must be a square with a particular size associated with it. This size is fixed at the time the MX quadtree is defined and is the minimum permissible separation between two data points in the domain of the MX quadtree (equivalently, it is the maximum number of elements permitted in each row and column of the corresponding matrix). The PR quadtree also has a square region, and its size depends on what other data points are currently represented by nodes in the quadtree. In the case of the MX quadtree there is a fixed discrete coordinate system associated with the space spanned by the quadtree, whereas no such limitation exists for

the PR quadtree. The advantage of such a fixed coordinate system is that there is no need to store coordinate information with a data point's leaf node. The disadvantage is that the discretization of the domain of the data points limits the differentiation between data points.

The size and shape of a quadtree are important from the standpoint of efficiency of both storage and search operations. The size and shape of the point quadtree are extremely sensitive to the order in which data points are inserted into it during the process of building it. This means that for a point quadtree of M records, its maximum depth is $M - 1$ (i.e., one record is stored at each level in the tree), whereas its minimum depth is $\lceil \log_2(3 \cdot M) \rceil$ (i.e., each level in the tree is completely full), where we assume that the root of the tree has a depth of 0. In contrast, the shape and size of the MX and PR quadrees are independent of the insertion order. For the MX quadtree all nodes corresponding to data points appear at the same depth in the quadtree. The depth of the MX quadtree depends on the size of the space spanned by the quadtree and the maximum number of elements permitted in each row and column of the corresponding matrix. For example, for a 2^n by 2^n matrix, all data points will appear as leaf nodes at a depth of n . The size and shape of the PR quadtree depend on the data points currently in the quadtree. The minimum depth of a PR quadtree for $M > 1$ data points is $\lceil \log_2(M - 1) \rceil$ (i.e., all the data points are at the same level), whereas there is no upper bound on the depth in terms of the number of data points. In particular, for a square region of side length s , such that the minimum Euclidean distance separating two points is d , the maximum depth of the quadtree can be as high as $\lceil \log_2((s/d) \cdot \sqrt{2}) \rceil$.

The volume of data also affects the comparison among the three quadrees. When the volume is very high, the MX quadtree loses some of its advantage since an array representation may be more economical in terms of space, as there is no need for links. Whereas the size of the PR quadtree was seen to be affected by clustering of data points, especially when the number of data

points is relatively small, this is not a factor in the size of a point quadtree. However, when the volume of data is large and is uniformly distributed, the effect of clustering is lessened and there should not be much difference in storage efficiency between the point and PR quadrees.

3.4 CIF Quadrees

The *MX-CIF quadtree* is a quadtree-like data structure devised by Kedem [1981] (and called a *quad-CIF tree*, where CIF denotes Caltech Intermediate Form) for representing a large set of very small rectangles for application in VLSI design rule checking. The goal is to locate rapidly a collection of all objects that intersect a given rectangle. An equivalent problem is to insert a rectangle into the data structure under the restriction that it does not intersect existing rectangles. The MX-CIF quadtree is organized in a similar way to the region quadtree. A region is repeatedly subdivided into four equal-sized quadrants until blocks are obtained that do not contain rectangles. As the subdivision takes place, a set containing all of the rectangles that intersect the lines passing through it is associated with each subdivision point. For example, Figure 44 contains a set of rectangles and its corresponding MX-CIF quadtree. Once a rectangle is associated with a subdivision point, say P , it is not considered to be a member of any of the sons of the node corresponding to P . For example, in Figure 44, node D spans a space that contains both rectangles 11 and 12. However, only rectangle 11 is associated with node D, whereas rectangle 12 is associated with node F.

Our definition of an MX-CIF quadtree is very similar to that of an MX quadtree with the following differences. First, data are associated with both terminal and non-terminal nodes. Nevertheless, the analog of a WHITE node is present and is a NIL pointer in the direction of a quadrant that contains no rectangles. Second, we are representing rectangles rather than points. This is fortunate because it provides a termination condition for the subdivision process in forming an MX quadtree. Sec-

tion 3.2 demonstrates that MX quadrees are defined for a domain whose elements must be subdivision points of the space being represented. The nonzero width of the rectangles ensures that they overlap with the subdivision points.

The set of the rectangles that intersect the lines passing through a subdivision point is subdivided into two sets. For example, consider subdivision point P centered at (CX, CY) that partitions a $2 \cdot LX$ by $2 \cdot LY$ rectangular area. All input rectangles that intersect the line $x = CX$ form one set, and all input rectangles that intersect the line $y = CY$ form the other set. Equivalently, these sets correspond to the rectangles intersecting the y and x axes, respectively, passing through (CX, CY) . If a rectangle intersects both axes (i.e., it contains the subdivision point P), then we adopt the convention that it is stored with the set associated with the y axis. These subsets are implemented as binary trees, which in actuality are one-dimensional analogs of the MX quadtree. For example, Figure 45 illustrates the binary tree associated with the x and y axes passing through A , the root of the MX-CIF quadtree of Figure 44.

Rectangles are inserted into an MX-CIF quadtree by searching for the position that they are to occupy. We assume that the input rectangle does not overlap any of the existing rectangles. This position is determined by a two-step process. First, the first subdivision point must be located such that at least one of its axis lines (i.e., the quadrant lines emanating from the subdivision point) intersects the input rectangle. Second, having found such a point and an axis, say point P and axis V , the subdivision process is repeated for the V axis until the first subdivision point that is contained within the rectangle is located. During the process of locating the destination position for the input rectangle, the space spanned by the MX-CIF quadtree may have to be repeatedly subdivided (termed *splitting*), creating new nodes in the process. As was the case for the MX quadtree, the shape of the resulting MX-CIF quadtree is independent of the order in which the rectangles are inserted into it. Deletion of nodes is more complex and may require collapsing

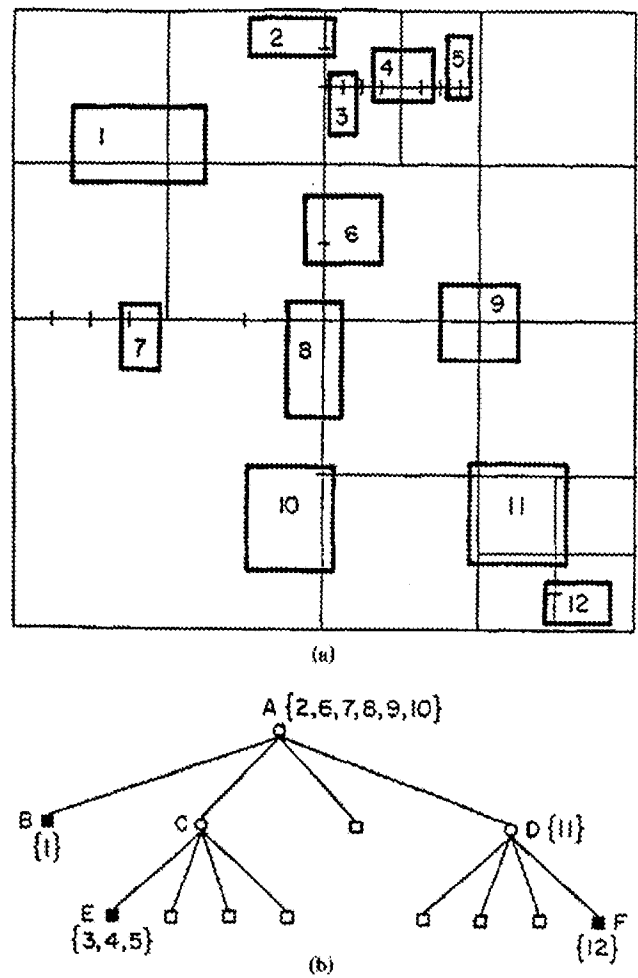


Figure 44. A MX-CIF quadtree (b) and the rectangles it represents (a).

of nodes, that is, the direct counterpart of the node-splitting process outlined above.

The most common search query is one that seeks to determine whether a given rectangle overlaps (i.e., intersects) any of the existing rectangles. It is a prerequisite to the successful insertion of a rectangle. Range queries can also be performed. However, they are more usefully cast in terms of finding all the rectangles within a given area. Another popular query seeks to determine whether one collection of rectangles can be overlaid on another collection with-

out any of the component rectangles intersecting one another. These two operations can be implemented by using variants of algorithms developed for handling set operations (i.e., union and intersection) in region-based quadtrees [Hunter and Steiglitz 1979a; Shneider 1981a]. In particular, the range query can be answered by intersecting the query rectangle with the MX-CIF quadtree. The overlay query can be answered by a two-step process. The two MX-CIF quadtrees are first intersected. If the result is empty, then they can be safely

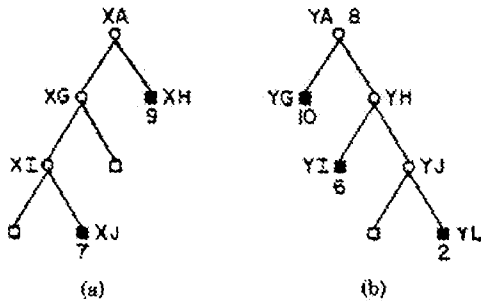


Figure 45. Binary trees for the (a) x axis and (b) y axis passing through node A in Figure 44.

overlaid and all that is needed is to perform a union of the two MX-CIF quadtrees. Boolean queries can also be handled easily.

The problem of determining whether a given rectangle overlaps any of an existing set of rectangles by use of quadtrees is also addressed by Abel and Smith [1983]. Each rectangle is associated with the node corresponding to the smallest block that totally encloses it (i.e., its minimum bounding "quadrant"). This is the same representation used by Kedem, except that Kedem uses binary trees to organize the rectangle associated with each quadtree node. Abel and Smith do not use a pointer-based quadtree representation. Instead, they represent each rectangle by the locational code (see Section 2.2) of its node. These codes are subsequently organized in a B^+ -tree [Comer 1979]. Note that many rectangles have identical locational codes, and thus the rectangle dimensions must also be stored along with the locational codes in the B^+ -tree. For example, for the set of rectangles in Figure 44, rectangles 2, 6, 7, 8, 9, and 10 have the same locational code as do rectangles 3, 4, and 5, albeit a different one. Hinrichs and Nievergelt [1983] describe another approach to this problem that is based on use of the Grid File, a hierarchical organization of point data, discussed in Section 3.5.

3.5 Bucket Methods

All of the data structures discussed above are primarily designed for in-core applications, although their uses can be extended

elsewhere. The problem is that when data are stored in external storage, the need to follow pointers may lead to page faults. To overcome this, methods have been designed that collect the points into sets (termed *buckets*) corresponding to the storage unit (i.e., page) of the disk. The remaining task is to organize the access to these buckets; this is often done by replacing the tree structure with an array, thereby facilitating address computation. We term such techniques *bucket methods*, and their aim is to ensure efficient access to disk data. The simplest bucket method is the fixed-grid (or cell) method [Knuth 1973, p. 554; Bentley and Friedman 1979], which is popular among cartographers. It divides the space into equal-sized cells (i.e., squares and cubes for two- and three-dimensional data, respectively) having the width equal to the search radius. If data are sought by using only a fixed-search radius, then the fixed grid is an efficient structure. It is also efficient when points are uniformly distributed (it corresponds to hashing [Knuth 1973]). For a nonuniform distribution it is less efficient, because buckets may be unevenly filled, leading to nearly empty pages as well as long overflow chains. The data structure is essentially a directory in the form of a k -dimensional array with one entry per cell. Each cell may be implemented as a linked list to represent the points within it. Figure 46 is an example in which a grid representation for the data of Figure 2 is shown for a search radius consisting of a square of size 20 by 20; that is, by assuming a 100 by 100 coordinate space, we have 25 squares of equal size. Its deficiency is a fixed size for the blocks, which results in both overflow and underflow. The methods presented below are examples of attempts to address this deficiency from both hierarchical and nonhierarchical viewpoints. We conclude with a discussion of some related work from the hashing area.

The *Grid File* of Nievergelt et al. [1984] is a variation of the grid method, which relaxes the requirement that cell division lines be equidistant. Its goal is to retrieve records with at most two disk accesses. This is done by using a grid directory consisting of grid blocks, which are analogous to the

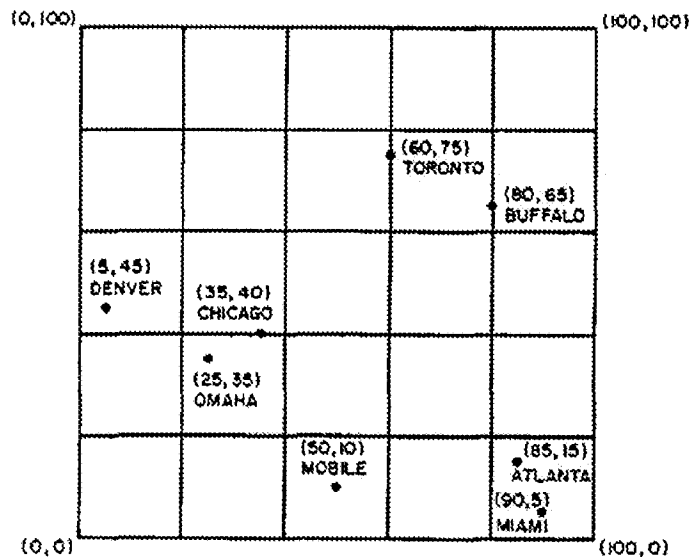


Figure 48. Grid representation corresponding to Figure 2 with a search radius of 20.

cells of the fixed-grid method. All records in one grid block are stored in the same bucket. However, several grid blocks can share a bucket as long as the union of these grid blocks forms a k -dimensional rectangle (i.e., a convex region) in the space of records. Although the regions of the buckets are piecewise disjoint, together they span the space of records.

The purpose of the grid directory is to maintain a dynamic correspondence between the grid blocks in the record space and the data buckets. The grid directory consists of two parts. The first is a dynamic k -dimensional array, which contains one entry for each grid block. The values of the elements are pointers to the relevant data buckets. Usually buckets will have a capacity of 10–1000 records. Thus the entry in the grid directory is small in comparison to a bucket. We are not concerned with how records are organized within a bucket (e.g., linked list and tree). The grid directory may be kept on disk. The second part of the grid directory is a set of k one-dimensional arrays called *linear scales*. These scales define a partition of the domain of each attribute and enable the accessing of the appropriate grid blocks by aiding in the computation of

their address on the basis of the value of the relevant attributes. The linear scales are kept in core. It should be noted that the linear scales are useful in guiding a range query by indicating the grid directory elements that overlap the query range.

As an example, consider Figure 47, which shows the Grid File representation for the data in Figure 2. The bucket capacity is two records. There are $k = 2$ different attributes. The grid directory consists of nine grid blocks and six buckets labeled A–F. We refer to grid blocks as if they are array elements; that is, grid block (i, j) is the element in row i (starting at the bottom) and column j (starting at the left) of the grid directory. Grid blocks $(2, 2)$, $(3, 1)$, and $(3, 3)$ are empty; however, they do share buckets with other grid blocks. In particular, grid block $(3, 1)$ shares bucket D with grid block $(2, 1)$, grid blocks $(3, 2)$ and $(3, 3)$ share bucket B, and grid blocks $(2, 2)$ and $(2, 3)$ share bucket E. The sharing is indicated by the broken lines. Figure 48 contains the linear scales for the two attributes (i.e., the x and y coordinates). For example, executing a FIND command with $x = 80$ and $y = 65$ causes the access of the bucket associated with the grid block in row 2 and

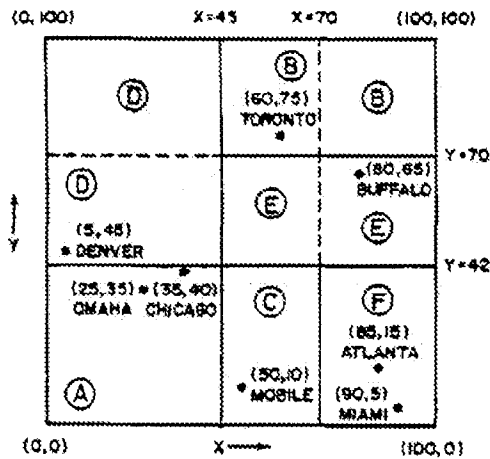


Figure 47. Grid directory for the data of Figure 2.

column 3 of the grid directory of Figure 47. The Grid File is attractive, in part, because of its graceful growth as more and more records are inserted. As the buckets overflow, a splitting process is applied that results in the creation of new buckets and a movement of records. Two types of bucket splits are possible. The first, and most common, occurs when several grid blocks share a bucket that has overflowed. For example, suppose Boise at (10, 80) and Fargo at (15, 75) are inserted in sequence in Figure 47. Boise is inserted in bucket D because it belongs in grid block (3, 1), which currently shares bucket D with grid block (2, 1). Fargo also belongs to grid block (3, 1); however, bucket D is now full. In this case, we merely need to allocate a new bucket and adjust the mapping between grid blocks and buckets. The second type of a split arises when we must refine a grid partition. It is triggered by an overflowing bucket, all of whose records lie in a single grid block (e.g., the overflow of bucket A upon insertion of Kansas City at (30, 30) in Figure 47). In this case there exists a choice with respect to the dimension (i.e., axis) and the location of the splitting point (i.e., we do not have to split at the midpoint of an interval). The counterpart of splitting is merging. There are two possible instances when

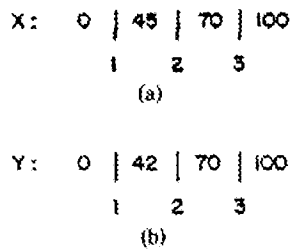


Figure 48. Linear scales for (a) x and (b) y corresponding to the grid directory of Figure 47.

merging is appropriate: (1) *Bucket merging*, the most common instance, arises when a pair of neighboring buckets are empty or nearly empty and their coalescing has resulted in a convex bucket region; (2) *directory merging* arises when two adjacent cross sections in the grid directory each have identical bucket values. For example, in the case of the two-dimensional grid directory of Figure 49, where all grid blocks in column 2 are in bucket C and all grid blocks in column 3 are in bucket D, if the merging threshold is satisfied, then buckets C and D can be merged and the linear scales modified to reflect this change. Generally, directory merging is of little practical interest since, even if merging is allowed to occur, it is probable that splitting will soon have to take place. Merrett and Otoo describe a technique termed *multipaging* [Merrett 1978; Merrett and Otoo 1982] which is very similar to the Grid File. It also uses a directory and maintains a set of linear scales called *axial arrays*. In fact, the Grid File uses multipaging as an index to a paged data structure. A database that is organized by using multipaging differs from the Grid File in that it requires bucket overflow areas. This means that it has a different bucket overflow criterion. Thus it does not guarantee that every record can be retrieved with two disk accesses. In particular, multipaging makes use of a load factor and a probe factor, which are related to the number of overflowing data items. This makes insertion and deletion (as well as bucket splitting and merging) somewhat more complicated than when the Grid File is used.

A	C	D	E
A	C	D	F
B	C	D	G

Figure 49. Example grid directory illustrating directory merging.

The EXCELL method of Tamminen [1981] is a bintree together with a directory array providing access by address computation. It can also be viewed as an adaptation of extendible hashing [Fagin et al. 1979] to multidimensional point data. It implements EXHASH, the extendible hashing hash function, by interleaving the most significant bits of the data (analogous to the locational codes discussed in Section 2.2). Similar in spirit to the Grid File, it is based on a regular decomposition and is useful in providing efficient access to, and an efficient representation of, geometric data. It also makes use of a grid directory; however, all grid blocks are of the same size. The principal difference is that grid refinement for the Grid File splits only one interval in two and results in the insertion of a $(k - 1)$ -dimensional cross section. In contrast, a grid refinement for the EXCELL method splits all intervals in two (thus the partition points are fixed) for the particular dimension and results in doubling the size of the grid directory. Therefore the grid directory grows more gradually when the Grid File is used, whereas use of EXCELL reduces the need for grid refinement operations at the expense of larger directories in general because of a sensitivity to the distribution of the data. However, a large bucket size reduces the effect of nonuniformity unless the data consist entirely of a few clusters. The fact that all grid blocks define equal-sized regions (and convex as well) means that EXCELL does not require a set of linear scales to access the grid directory as is needed for the Grid File.

An example of the EXCELL method is considered in Figure 50, which shows the representation for the data in Figure 2.

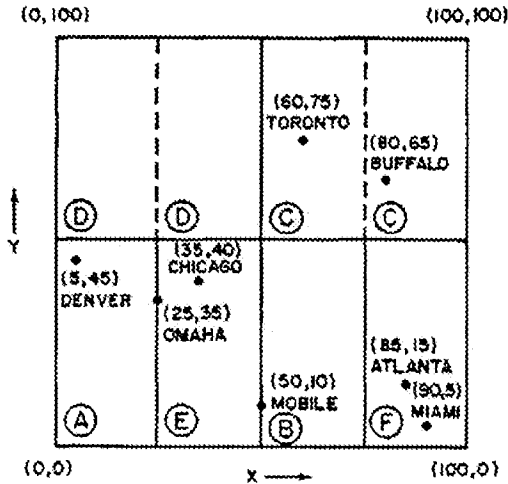


Figure 50. EXCELL representation corresponding to Figure 2.

Again, the convention is adopted that a rectangle is open with respect to its upper and right boundaries and closed with respect to its lower and left boundaries. The capacity of the bucket is two records. There are $k = 2$ different attributes. The grid directory is implemented as an array and in this case it consists of eight grid blocks (labeled in the same way as for the Grid File) and six buckets labeled A-F. Note that grid blocks (2, 3) and (2, 4) share bucket C, whereas grid blocks (2, 1) and (2, 2), despite being empty, share bucket D. The sharing is indicated by the broken lines. Furthermore, when a bucket size of 1 is used, the partition of space induced by EXCELL equals that of a PR k -d tree [Orenstein 1982].

As a database represented by the EXCELL method grows, buckets will overflow. This leads to the application of a splitting process that results in the creation of new buckets and a movement of records. As in the case of the Grid File, two types of bucket splits are possible. The first, and most common, occurs when several grid blocks share a bucket that has overflowed. In this case, a new bucket is allocated and the mapping between grid blocks and buckets is adjusted. The second type of a split arises when a grid partition must be re-

finer; this causes a doubling of the directory. It is triggered by an overflowing bucket that is not shared among several grid blocks (e.g., the overflow of bucket A upon insertion of Kansas City (30, 30) in Figure 50). The split occurs along the different attributes in a cyclic fashion (first split along attribute x , then y , then x , etc.). For both types of bucket splits, a situation may arise in which none of the elements in the overflowing buckets belongs to the newly created bucket, with the result that the directory will have to be doubled more than once. This results from the fact that the splitting points are fixed for EXCELL. For example, this will occur when we attempt to insert Kansas City at (30, 30) in Figure 50 since the first directory doubling at $y = 25$ and $y = 75$ will still have Chicago, Omaha, and Kansas City in the same grid block. Thus we see that the size of the EXCELL grid directory is sensitive to the distribution of the data. However, a large bucket size reduces the effect of nonuniformity unless the data consist entirely of a few clusters.

The counterpart of splitting is merging. However, it is considerably more limited in scope for EXCELL than for the Grid File. Also, it is less likely to arise because EXCELL has been designed primarily for use in geometrical applications in which deletion of records is not so prevalent. As with the Grid File, however, there are two cases where merging is appropriate, that is, bucket merging and directory merging.

Both the Grid File and EXCELL organize space into buckets and use directories in the form of arrays to access them. The similarity to the quadtree lies in the mappings induced by the directories (i.e., EXCELL with the region quadtree and the Grid File with the point quadtree). Trees can also be used to access the buckets [Knott 1971]. Matsuyama et al. [1984] compare a technique of accessing buckets by use of a PR quadtree with one that uses an adaptive k -d tree. Robinson [1981] introduces the k -d-B-tree, which is a generalization of the B-tree to allow multiattribute access. O'Rourke [1981; O'Rourke and Sloan 1984] makes use of an adaptive k -d tree, which he calls a *dynamically quantized*

space, to access buckets of data for use in multidimensional histogramming to aid in the focusing of the Hough transform. Sloan [1981; O'Rourke and Sloan 1984] addresses the same problem as O'Rourke, albeit with a different data structure, which he calls a *dynamically quantized pyramid*. It is based on the pyramid data structure (see Section 2.12). Here the number of buckets is fixed. It differs from the conventional pyramid in that the partition points at the various levels are allowed to vary rather than being fixed and are adjusted as data are entered. The result is somewhat related to a complete point quadtree [Knuth 1975, p. 401] with buckets.

There has also been considerable work on representing multidimensional point data by use of linear hashing. Linear hashing [Litwin 1980] methods are attractive because they provide for linear growth of the file (i.e., one bucket at a time) without requiring a directory. In contrast, extendible hashing [Fagin et al. 1979] (e.g., EXCELL) and the Grid File methods require extra storage in the form of directories. When a bucket overflows, the directory doubles in size in the case of extendible hashing, whereas in the case of the Grid File it results in the insertion of a $(k - 1)$ -dimensional cross section. Neither EXCELL nor the Grid File need overflow pages, whereas methods based on linear hashing generally do, although this may be unnecessary. Bit interleaving (e.g., attributed by Bentley [1975a] to McCreight, but see the discussion of the Morton matrix [Morton 1966] in Section 2.2) is used by a number of researchers [Burkhardt 1983; Orenstein 1983; Orenstein and Merrett 1984; Oukseil and Scheuermann 1983; Tropf and Herzog 1981] to create a linear order on the multidimensional domain of the data. Tropf and Herzog [1981] and Orenstein and Merrett [1984] discuss its use in range searching. Burkhardt [1983] terms it *shuffle order*, and adapts it to linear hashing in the same way that EXCELL adapts it to extendible hashing, and uses it to evaluate range queries. Oukseil and Scheuermann [1983] call it z order. Orenstein [1983] discusses the problems associated with such an approach. He points out

that the resulting file may contain a number of sparsely filled buckets, which will result in poor performance for sequential access. He goes on to propose a modification that unfortunately, unlike linear hashing, does not result in a bucket retrieval cost of one or two disk reads (for the hash operations). In contrast, directory-based methods such as the Grid File and EXCELL do not suffer from such a problem to the same extent because, since the directory consists of grid blocks and several grid blocks can share a bucket, the sparseness issue can be avoided.

4. CURVILINEAR DATA

Section 2 was devoted to approaches to region representation that are based on descriptions of their interiors. In this section we focus on representations that specify boundaries of regions. This is done in the more general context of data structures for curvilinear data. The simplest representation is the polygon in the form of vectors [Nagy and Wagle 1979], which are usually specified in the form of lists of pairs of x and y coordinate values corresponding to their start and end points. One of the most common representations is the chain code [Freeman 1974] (described in Section 2.3), which is an approximation of a polygon. Other popular representations include raster-oriented methods [Merrill 1973; Peuquet 1979], as well as a combination of vectors and rasters (e.g., vasters [Peuquet 1983]). There has also been considerable interest recently in hierarchical representations. These are primarily based on rectangular approximations to the data [Ballard 1981; Burton 1977; Peucker 1976]. In particular, Burton [1977] uses upright rectangles, Ballard [1981] uses rectangular strips of arbitrary orientation, and Peucker [1976] uses sets of bands. There also exist methods that are based on a regular decomposition in two dimensions, as reported by Hunter and Steiglitz [1979a], Shneier [1981c], and Martin [1982]. Note that our primary focus is on the facilitation of set operations and not ease of display, which is a characterization of B-splines and Bezier methods [Cohen et al. 1980].

In many applications polygons are not unrelated, but together form a partition of the study area (termed a *polygonal map*). It is possible to use the above representations for each curve that bounds two adjacent regions. However, it is often preferable to represent the complete network of boundaries with a single hierarchical data structure. Some examples include the line quadtree of Samet and Webber [1984], the PM quadtree of Samet and Webber [1983], and the edge variant of the EXCELL method of Tamminen [1981]. In order to avoid confusion with the point space formulation of the EXCELL method, discussed in Section 3.5, we shall use the term *edge-EXCELL*. In the remainder of this section, we elaborate further on the strip tree and also on the representations that are based on a regular decomposition, concluding with a brief comparison of these methods.

4.1 Strip Trees

The *strip tree* is a hierarchical representation of a single curve that is obtained by successively approximating segments of it by enclosing rectangles. The data structure consists of a binary tree whose root represents the bounding rectangle of the entire curve. For example, consider Figure 51, where the curve between points P and Q , at locations (x_P, y_P) and (x_Q, y_Q) , respectively, is modeled by a strip tree. The rectangle associated with the root, A in this example, corresponds to a rectangular strip of maximum width, enclosing the curve, whose sides are parallel to the line joining the endpoints of the curve (i.e., P and Q). Next, this rectangle is decomposed into two parts at one of the points (termed a *splitting point*) on the rectangle that is also part of the curve. There is at least one such splitting point. If there are more, then the decomposition is performed by using the point that is at a maximum distance from the line joining the endpoints of the curve. If the curve is both continuous and differentiable at the splitting point, then, of course, the boundaries of the rectangle that pass through these points are tangents to the curve. This splitting process is recur-

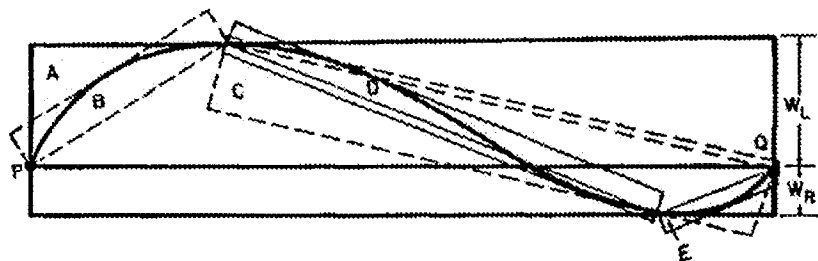


Figure 51. A curve and its decomposition into strips.

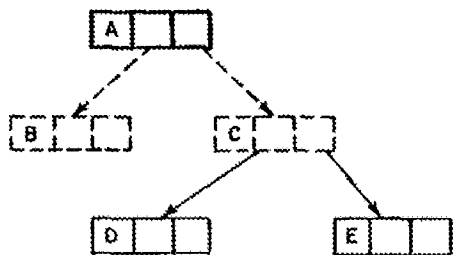


Figure 52. Strip tree corresponding to Figure 51.

sively applied to the two sons until every strip is of a width less than a predetermined value. For Figure 51, the first splitting operation results in the creation of strips B and C. Strip C is further split, creating strips D and E, at which point the splitting process ceases. Figure 52 shows the resulting binary tree. Note that each node in the strip tree is implemented as a record with eight fields. Four fields contain the x and y coordinates of the endpoints, two fields contain pointers to the two sons of the node, and two fields contain information about the width of the strip (i.e., w_L and w_R of Figure 51).

Figure 51 is a relatively simple example. In order to be able to cope with more complex curves, the notion of a strip tree must be extended. In particular, closed curves (e.g., Figure 53) and curves that extend past their endpoints (Figure 54) require some special treatment. The general idea is that these curves are enclosed by rectangles that are split into two rectangular strips, and from now on the strip tree is used as before. Note that the strip tree concept and the related algorithms are regarded by Ballard

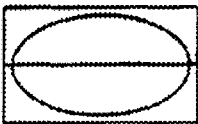


Figure 53. Modeling a closed curve by a strip tree.

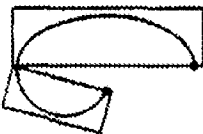


Figure 54. Modeling a curve that extends past its endpoints by a strip tree.

as completely expanded down to a primitive level of unit line segments on a discrete grid, even when the underlying curves are collinear. In order to be able to handle curves that consist of disconnected segments, strips are classified as either regular or not and a special bit is associated with each strip to indicate its status. Formally, a curve is said to be regular if it is connected and has its endpoints touching the ends of the strip.

Like point and region quadtrees, strip trees are useful in applications that involve search and set operations. For example, suppose that we wish to determine whether a road crosses a river. By using a strip tree representation for these features, answering this query means basically that we perform an intersection of the corresponding strip trees. Three cases are possible, as is shown in Figure 55. Figure 55a and b correspond to the answers NO and YES, re-

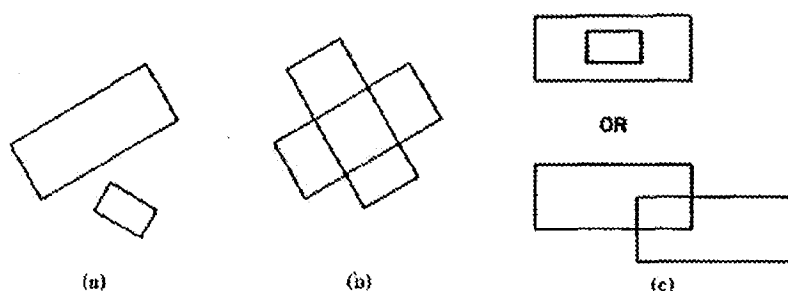


Figure 55. Three possible results of intersecting two strip trees. (a) Null. (b) Clear. (c) Possible.

spectively, whereas Figure 55c requires us to descend further down the strip tree. Other operations that can be performed efficiently by using the strip tree data structure include the computation of the union of two curves, length of a curve, areas of closed curves, intersection of curves with areas, point membership, etc. [Ballard 1981]. In particular, for closed curves that are well behaved, intersection and point membership have an expected execution time of $O(\log v)$, where v is the number of points describing the curve. Strip trees are also used by Gaston and Lozano-Perez [1984] in robotic tactile recognition and localization.

The strip tree can be characterized as a top-down approach to curve approximation. Burton [1977] defines a related structure termed a *BSPR* (binary searchable polygonal representation), which is a bottom-up approach to curve approximation. Once again, the primitive unit of approximation is a rectangle; however, in the case of the BSPR all rectangles are upright (i.e., they have a single orientation). The curve to be approximated is decomposed into a set of simple sections, where each simple section corresponds to a segment of the curve that is monotonic in both the x and y values of the points comprising it. The tree is built by combining pairs of adjacent simple sections to yield compound sections. This process is repeatedly applied until the entire curve is approximated by one compound section. Thus we see that terminal nodes correspond to simple sections and

nonterminal nodes correspond to compound sections. For a curve with 2^n simple sections, the corresponding BSPR has n levels.

As an example of a BSPR consider the regular octagon in Figure 56a having vertices A-H. It can be decomposed into four simple sections, that is, ABCD, DEF, FGH, and HA. Figure 56b shows a level 1 approximation to the four simple sections consisting of rectangles AIDN, DJFN, HMFK, and AMHL, respectively. Pairing up adjacent simple sections yields compound sections AIJF corresponding to AIDN and DJFN, and AFKL corresponding to HMFK and AMHL (see Figure 56c). More pairing yields the rectangle for compound section IJKL (see Figure 56d). The resulting BSPR tree is shown in Figure 56e. By using the BSPR, Burton shows how to perform point in polygon determination and polygon intersection. These operations are implemented by tree searches and splitting operations.

Both the strip tree and the BSPR share the property of being independent of the grid system in which they are embedded. An advantage of the BSPR representation over the strip tree is the absence of a need for special handling of closed curves. However, the BSPR is not as flexible as the strip tree. In particular, the resolution of the approximation is fixed (i.e., once the width of the rectangle is selected, it cannot be varied). In fact, this is similar to the advantage of quadtree-based decomposition methods over hexagon-based systems

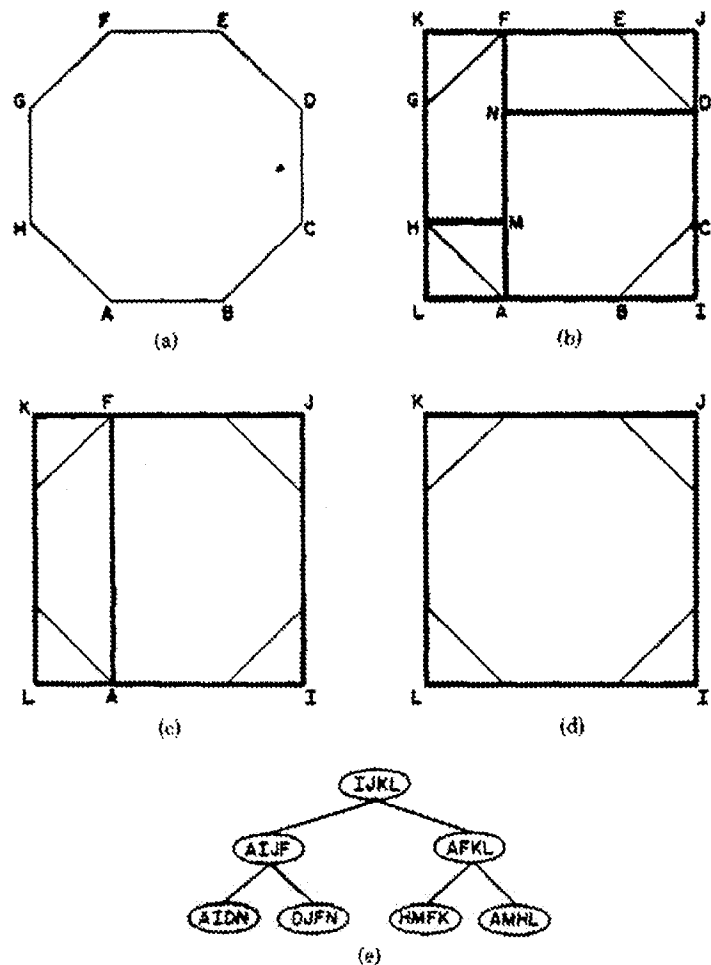


Figure 56. (a) A regular octagon and (b)–(d) the three successive approximations resulting from the use of BSPR. (e) The resulting BSPR.

pointed out in Section 2; that is, for the hexagon, we must decide a priori on the finest resolution.

4.2 Methods Based on a Regular Decomposition

Strip tree methods approximate curvilinear data with rectangles. Quadtree methods achieve similar results by use of a collection of disjoint squares having sides of length power of two. A number of variants of quadtrees are currently in use, and can be differentiated by the type of data that they

are designed to represent. All but the PM quadtree of Samet and Webber [1983] and the edge-EXCELL of Tamminen [1981] are pixel based and yield approximations whose accuracy is constrained in part by the resolution of the data that they represent. They can be used to represent both linear and nonlinear curves. The latter need not be continuous or differentiable. In contrast, the PM quadtree and the edge-EXCELL yield an exact representation of polygons or collections of polygons. The *edge quadtree* of Shneier [1981c] is an attempt to store linear feature infor-

mation (e.g., curves) for an image (binary and gray scale) in a manner analogous to that used for storing region information. A region containing a linear feature or part thereof is subdivided into four squares repeatedly until a square is obtained that contains a single curve that can be approximated by a single straight line. Each leaf node contains the following information about the edge passing through it: magnitude (i.e., 1 in the case of a binary image or the intensity in case it is a gray-scale image), direction, intercept, and a directional error term (i.e., the error induced by approximating the curve by a straight line using a measure such as least squares). If an edge terminates within a node, then a special flag is set and the intercept denotes the point at which the edge terminates. Application of this process leads to quad-trees in which long edges are represented by large leaves or a sequence of large leaves. However, small leaves are required in the vicinity of corners or intersecting edges. Of course, many leaves will contain no edge information at all. As an example of the decomposition that is imposed by the edge quadtree, consider Figure 57, which is the edge quadtree corresponding to the polygon of Figure 18 when represented on a 2^4 by 2^4 grid. Note that the edge quadtree in Figure 57 requires fewer blocks than Figure 18, which is the representation of the polygon when using the methods of Hunter and Steiglitz [1979a].

Closely related to the edge quadtree is the *least square quadtree* of Martin [1982]. In that representation, leaf nodes correspond to regions that contain a single curve that can be approximated by k (fixed a priori) straight lines within a least square tolerance. This enables the handling of curved lines and uses fewer nodes than the edge quadtree with greater precision. A cruder method is described by Omolayole and Klinger [1980], where all parts of the image that contain edge data are repeatedly decomposed until a 2 by 2 quadrant is obtained in which they store templatelike representations of the edges. This is quite similar to the MX quadtree, except that the data are edges rather than points. However, it is too low a level of representation in that

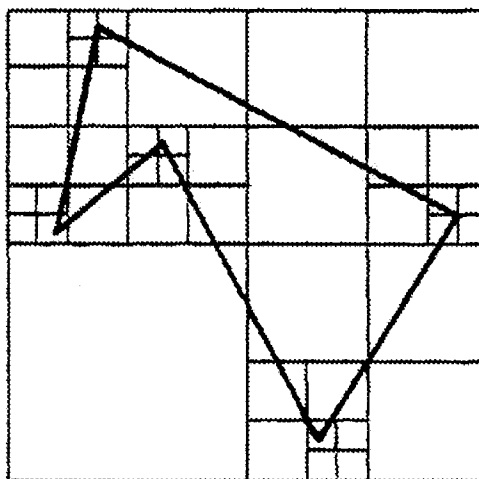


Figure 57. The edge quadtree corresponding to the polygon of Figure 18.

it does not take advantage of the hierarchical nature of the data structure.

The *line quadtree* of Samet and Webber [1984] addresses the issue of hierarchically representing images that are segmented into a number of different regions rather than mere foreground and background, as is the case for conventional quadtrees. In particular, it encodes both the area of the individual regions and their boundaries in a hierarchical manner. This is in contrast to the region quadtree, which encodes only areas hierarchically, and the strip tree, which encodes only curves hierarchically. The line quadtree partitions the set of regions (termed a *map*) via a recursive decomposition technique that successively subdivides the map until blocks (possibly single pixels) that have no line segments passing through their interior are obtained. With each leaf node, a code is associated that indicates which of its four sides form a boundary (not a partial boundary) of any single region. Thus, instead of distinguishing leaf nodes on the basis of being BLACK or WHITE, boundary adjacency information is used. This boundary information is hierarchical in that it is also associated with nonterminal nodes. In essence, wherever a nonterminal node does not form a T-junction with any of the boundaries of its descendants along a particular side, this side

is then marked as being adjacent to a boundary.

As an illustration of a line quadtree, consider the polygonal map of Figure 58 whose corresponding line quadtree (i.e., block decomposition) is shown in Figure 59. The bold lines indicate the presence of boundaries. Note that the south side of the block corresponding to the root is on a boundary that is also the border of the image. As another example, the western side of the SW son of the root in Figure 59 does not indicate the presence of a boundary (i.e., the side is represented by a light line) even though it is adjacent to the border of the image. The problem is that the SW son of the root has its NW and SW sons in different regions, as is signaled by the presence of a T-junction along its western side. Having the boundary information at the non-terminal nodes enables boundary following algorithms to be performed quickly, in addition to facilitating the process of superimposing one map on top of another. Observe also that the line quadtree has the same number of nodes as a conventional quadtree representation of the image. Boundaries of leaf nodes that are partially on the boundary between two regions can have their boundaries reconstructed by examining their neighbors along the shared boundary. For example, the southern side of the NW son of the SW son of the root in Figure 59, say A, represents a partial boundary. The exact nature of the boundary is obtained by examining the NE and NW sons of the southern brother of A.

The PM quadtree of Samet and Webber [1983] and the edge-EXCELL of Tamminen [1981] are attempts to overcome some of the problems associated with the following three structures: the line quadtree, the edge quadtree, and the quadtree formulation of Hunter and Steiglitz (termed an *MX quadtree*) for representing polygonal maps (i.e., collections of straight lines). In general, all three of these representations correspond to approximations of a map. The line quadtree is based on the approximation that results from digitizing the areas of the polygons comprising a polygonal map. For the edge and MX quadtrees, the result is still an approximation because the vertices

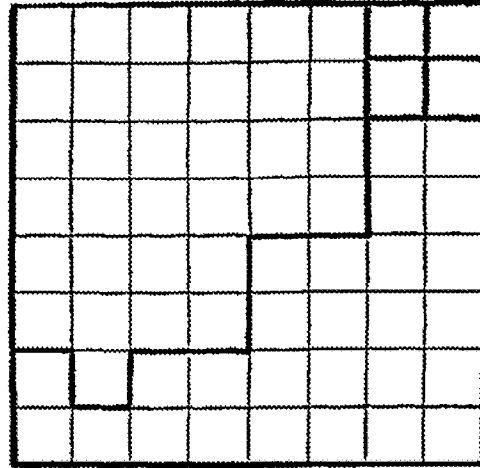


Figure 58. Example polygonal map to illustrate line quadtrees.

are represented by pixels in the edge quadtree and boundaries are represented by BLACK pixels in the MX quadtree. In other words, the MX quadtree is based on a digitization of the boundaries of the polygons, whereas the edge quadtree is based on a piecewise linear approximation of the boundaries of the polygons. Another disadvantage of these three representations is that certain properties of polygonal maps cannot be directly represented by them. For example, it is impossible for five line segments to meet at a vertex. In the case of the edge and MX quadtrees, we would have difficulty in detecting the vertex and for the line quadtree the situation cannot be handled because all regions comprising the map must be rectilinear. Note that it is impossible for five rectilinear regions to meet at a point. Other problems include a sensitivity to shift and rotation, which may result in a loss of accuracy in the original approximation. Finally, owing to their approximate nature, these data structures will most likely require a considerable amount of storage, since each line is frequently approximated at the pixel level, thereby resulting in quadtrees that are fairly deep.

The PM quadtree represents a polygonal map by using the PR quadtree discussed in Section 3.2. Each vertex in the map corresponds to a data point in the PR quadtree.

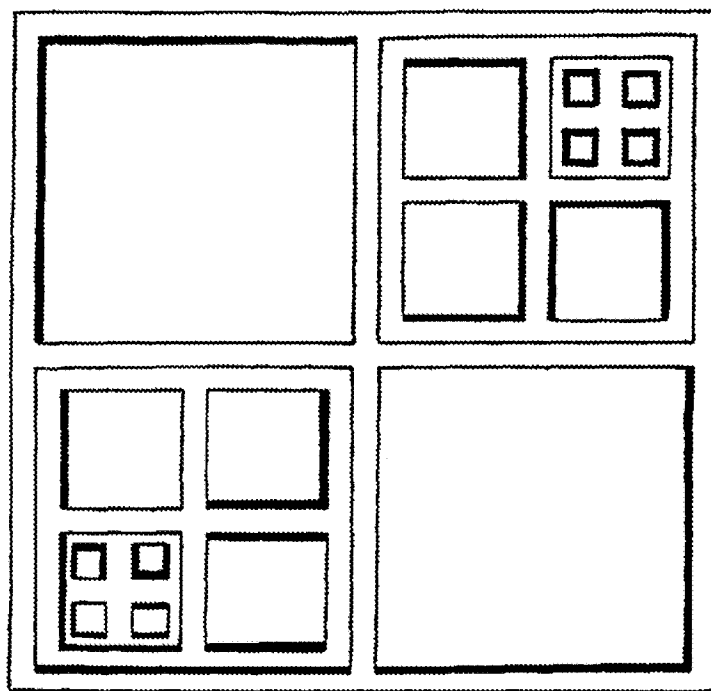


Figure 59. Line quadtree corresponding to Figure 58.

We define a *q-edge* to be a segment of an edge of the map that either spans an entire block in the PR quadtree (e.g., segment RS in Figure 61) or extends from a boundary of a block to a vertex within the block (i.e., when the block contains a vertex, e.g., segment CV in Figure 61).

For every leaf in the PR quadtree we partition all of its *q-edges* into seven classes. Each of these classes is stored in a balanced binary tree [Aho et al. 1974]. One class corresponds to the set of *q-edges* that meet at a vertex within the block's region. This class is ordered in an angular manner. The remaining *q-edges* that pass through the block's region must enter at one side and exit via another. This yields six classes: NW, NS, NE, EW, SW, and SE, where SW denotes *q-edges* that intersect both the southern and western boundaries of the block's region. Note that these classes are often empty. The *q-edges* of these classes are ordered by the position of their intercepts along the perimeter of the block's

region. A *q-edge* that coincides with the boundary of a leaf's region is placed in either class NS or EW as is appropriate. For example, consider the polygonal map of Figure 60 and its corresponding PM quadtree in Figure 61. The block containing vertex C has one balanced binary tree for the *q-edges* intersecting vertex C (three balanced binary tree nodes for *q-edges* CM, CN, and CV) and one balanced binary tree for the *q-edges* intersecting the NW boundary (one balanced binary tree node for *q-edge* ST). In total, the PM quadtree of Figure 61 contains seven quadtree leaf nodes, and nine nonempty balanced binary trees containing seventeen nodes.

The PM quadtree provides a convenient, reasonably efficient data structure for performing a variety of operations. Point-in-polygon determination is achieved by finding a bordering *q-edge* with respect to each of the seven classes and then selecting the closest of the seven as the true bordering *q-edge*. The execution time of this proce-

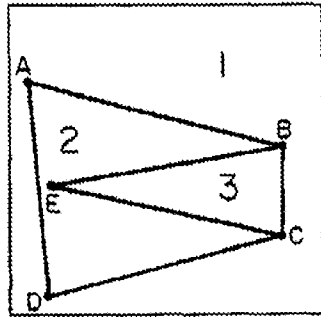


Figure 60. Example polygonal map to illustrate PM quadtrees.

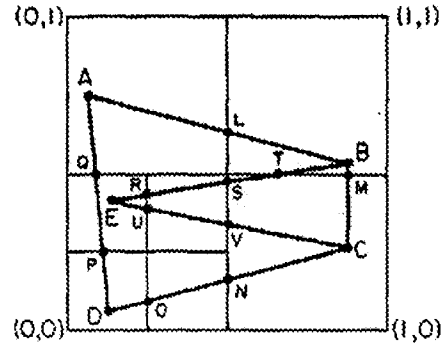


Figure 61. PM quadtree corresponding to Figure 60.

ture is proportional to the depth of the PM quadtree. It should be noted that the depth of the PM quadtree is inversely proportional to the log of the minimum separation between two vertices plus the log of the number of edges in the polygonal map [Samet and Webber 1983]. Besides point-in-polygon determination, there exist efficient algorithms for insertion of an edge into the map, overlaying two maps, clipping and windowing a map, and range searching (i.e., determining all polygons within a given distance of a point).

The *edge-EXCELL* method [Tamminen 1981] is an application of the EXCELL method for point data (described in Section 3.5) to polygonal maps. It is based on a regular decomposition. The principles guiding the decomposition process and the data structure are identical to those used for representing points. The only difference is that now the data consist of straight-line segments that intersect the cells (i.e., grid blocks). Once again, a grid directory is used that maps the cells into storage areas of a finite capacity (e.g., buckets), which often reside on disk. As the buckets overflow (i.e., the number of line segments intersecting them exceeds the capacity of the bucket), buckets are split into two equal-sized grid blocks, which may also lead to a doubling in the size of the directory. If the polygonal map contains a vertex at which m lines intersect, and m is greater than the bucket capacity, then no matter how many times the bucket is split, it will be impossible to store all the line segments in one bucket.

In such a case, *edge-EXCELL* makes use of overflow buckets. This is a disadvantage of *edge-EXCELL* when compared with the PM quadtree.

Using *edge-EXCELL*, point-in-polygon determination is achieved by a two-step process. First, the cell in which the point lies is located. Second, the corresponding polygon is determined by finding the closest polygon boundary in any given direction by use of a technique known as ray casting [Roth 1982] (similar to searching for the closest q -edge when using the PM quadtree). Tamminen [1983] has shown that, in practice, this requires on the average little more than one cell access. *Edge-EXCELL* has also been used to do hidden line elimination [Tamminen 1982].

4.3 Comparison

The chain code is the most compact of the representations. However, it is a very local data structure and is thus rather cumbersome when attempting to perform set operations such as intersection of two curves represented using it. In addition, like the strip tree, and to a lesser extent the BSPR, it is not tied to a particular coordinate system. This is a problem with methods based on a regular decomposition, although it is somewhat reduced for the PM quadtree and *edge-EXCELL*.

Representations based on a regular decomposition have a number of advantages over the strip tree. First, more than one curve can be represented with one instance

of the data structure—a very important feature for maps. Second, they are unique. In contrast, only one curve can be represented by a single strip tree. Also, the strip tree is not unique when the curve is closed, not regular, or contains more than one endpoint. However, the strip tree is invariant under shifts and rotations. The line quadtree is better than the MX quadtree of Hunter and Steiglitz [1979a], which represents boundary lines as narrow BLACK regions, for two reasons. First, narrow regions are costly in terms of the number of nodes in the quadtree. Second, it commits the use to a specific thickness of the boundary line, which may be unfortunate, for example, when regenerating the picture on an output device. Also such arbitrary decisions of representation accuracy are not very appropriate when data are to be stored in a database.

At this point it might be appropriate to speculate on some other data structures for curvilinear data. Representations based on regular decomposition are attractive because they enable efficient computation of set operations. In particular, at times it is convenient to perform the operations on different kinds of geometric entities (e.g., intersecting curves with areas). The strip tree is an elegant data structure from the standpoint of approximation. However, it has the disadvantage that decomposition points are independent of the coordinate system of the image (i.e., they are at arbitrary points dependent on the data rather than at predetermined positions as is the case with a data structure that is based on a regular decomposition). Thus answering a query such as, "Find all wheat growing regions within 20 miles of the Mississippi River," is not easy to do when the river is represented as a strip tree and the wheat-growing regions are represented by region quadtrees. The problem is that, whereas quadtree methods merely require pointer-chasing operations, strip tree methods may lead to complex geometric computations. What is desired is a regular decomposition strip tree or variant thereof.

The data structures discussed in this section are rooted in the image-processing area and were designed primarily to represent

curves and lines. Computational geometry is another area where similar problems arise [Edelsbrunner 1984; Toussaint 1980]. This is a rapidly changing field, which has its roots in the work of Shamos and Hoey [1975; Shamos 1978] and focuses on problems of asymptotical computational complexity of geometric algorithms. However, a full presentation of this field is beyond the scope of this survey. Nevertheless, in the following we do give a brief sample of the types of results attainable for similar problems. Many of the solutions (e.g., Lipton and Tarjan [1977]) are based on the representation of line segments as edges and vertices in a graph.

For example, an alternative to the PM quadtree and edge-EXCELL is the *K-structure* of Kirkpatrick [1983]. It is a hierarchical structure based on triangulation rather than a regular decomposition. The notion of hierarchy in the *K-structure* is radically different from that of a quadtree, in that instead of replacing a group of triangles by a single triangle at the next higher level, a group of triangles is replaced by a smaller group of triangles. Triangles are grouped for replacement because they share a common vertex. The smaller group results from eliminating the common vertex and then retriangulating. Kirkpatrick [1983] shows that at each level of the hierarchy, at least one twenty-fourth of the vertices can be eliminated in this manner. The vertices that have been eliminated are guaranteed to have degree of 11 or less, thus bounding the cost of retriangulating. Let v denote the number of vertices in a polygonal map. Then, the size of a *K-structure* is guaranteed to be $O(v)$ although the worst-case constant of proportionality is 24 times the amount of information stored at a node. It also leads to an $O(\log v)$ query time for point-in-polygon determination. The construction process has a worst-case execution time of $O(v)$ for a triangular subdivision and $O(v \log v)$ for a general one. The latter is dominated by the cost of triangulating the original polygonal map [Hertel and Mehlhorn 1983]. Since a triangulation constitutes a convex map, that is, a planar subdivision formed of convex regions, the work of Nievergelt and Preparata [1982] is

relevant. They show that the cost of performing a map overlay operation is $O(v \log v + s)$, where s is the number of intersections of all line segments in the two maps. Finally, it is worth noting that the hierarchical nature of the K-structure may lead to an efficient range-searching algorithm.

Another alternative to the K-structure is the *layered dag* of Edelsbrunner et al. [1985]. This structure is a modification of a binary tree, which is used to store a y -monotone subdivision of a polygonal map. A y -monotone subdivision of a polygonal map is created by partitioning the regions of a map until no vertical line intersects a region's boundary more than twice. Note that the resulting map need not be convex; however, the asymptotic worst-case analysis of the layered dag is identical to that of the K-structure. The number of line segments that must be inserted to check a y -monotone subdivision is usually considerably fewer than the number needed for triangulation.

When the K-structure and the layered dag are compared with the PM quadtree (and to some extent edge-EXCELL), the qualitative comparison is analogous to that of a point quadtree with a PR quadtree. All of these structures have their place; the one to use depends on the nature of the data and the importance of guaranteed worst-case performance. The K-structure and the layered dag organize the data to be stored, whereas the PM quadtree organizes the embedding space from which the data are drawn. The K-structure and the layered dag have better worst-case execution time bounds for similar operations compared with those considered for the PM quadtree. However, considerations such as ease of implementation and integration with representations of other data types must also be taken into account in making an evaluation. In the case of dynamic files, at present it would seem to be more convenient to use the PM quadtree since a general updating algorithm for the K-structure or the layered dag have not been reported.

5. CONCLUSIONS

In this paper we have attempted to give a survey of a large number of hierarchical

data structures and show how they are interrelated. Undoubtedly, some data structures may have been overlooked, but any such omissions were unintentional. The idea of recursive decomposition is the underlying basis of each structure discussed. In the past, there has been considerable confusion between these similar representations (e.g., the point quadtree and the region quadtree). Consequently, readers of the literature are cautioned about comparisons involving "quadtrees." Our focus has been on execution time efficiencies although storage requirements have also been taken into consideration.

For the future we foresee the following *quad* trends. Complexity measures for images represented by quadtrees are likely to be proposed [Creuzburg 1981]. Quadtrees and their variants will be used in computational domains [Rheinholdt and Mesztenyi 1980; Samet and Krishnamurthy 1983; Yerry and Shepard 1983]. With the current interest in VLSI, it is certain that increasingly these data structures will find themselves being implemented in hardware [Besslich 1982; Dyer 1981; Ibrahim 1984; Milford and Willis 1984; Woodwark 1982]. Another trend is the development of integrated geographical databases that permit the interaction between data of differing types [Matsuyama et al. 1984; McKeown and Denlinger 1984; Rosenfeld et al. 1982, 1983]. This would facilitate efficiently answering queries like, "Find all cities with a population in excess of 5000 people in wheat-growing regions within 20 miles of the Mississippi River." At this point we have traveled a full circle in our survey and if we still do not know how to answer this query, then, by now, maybe we at least know why—an appropriate point for us to get off and for you, the reader, to get on!

ACKNOWLEDGMENTS

I have benefited greatly from comments by Bernard Diaz, Gary Knott, Azriel Rosenfeld, Deepak Sherlekar, Markku Tamminen, and Robert E. Webber. The referees are also to be commended for the thoroughness of their reviews. This work was supported in part by the National Science Foundation under Grant MCS-83-02118.

REFERENCES

- ABEL, D. J. 1984. A B⁺-tree structure for large quad-trees. *Comput. Vision Gr. Image Process.* 27, 1 (July), 19-31.
- ABEL, D. J., AND SMITH, J. L. 1983. A data structure and algorithm based on a linear key for a rectangle retrieval problem. *Comput. Vision Gr. Image Process.* 24, 1 (Oct.), 1-13.
- AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. 1974. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass.
- AHUJA, N. 1983. On approaches to polygonal decomposition for hierarchical image representation. *Comput. Vision Gr. Image Process.* 24, 2 (Nov.) 200-214.
- AHUJA, N., AND NASH, C. 1984. Octree representations of moving objects. *Comput. Vision Gr. Image Process.* 26, 2 (May), 207-216.
- ANDERSON, D. P. 1983. Techniques for reducing pen plotting time. *ACM Trans. Gr.* 2, 3 (July), 197-212.
- BALLARD, D. H. 1981. Strip trees: A hierarchical representation for curves. *Commun. ACM* 24, 5 (May), 310-321. See also corrigendum, *Commun. ACM* 25, 3 (Mar. 1982), 213.
- BELL, S. B. M., DIAZ, B. M., HOLROYD, F., AND JACKSON, M. J. 1985. Spatially referenced methods of processing raster and vector data. *Image Vision Comput.* 1, 4 (Nov.), 211-220.
- BENTLEY, J. L. 1975a. A survey of techniques for fixed radius near neighbor searching. SLAC Rep. No. 186, Stanford Linear Accelerator Center, Stanford University, Stanford, Calif., Aug.
- BENTLEY, J. L. 1975b. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (Sept.), 509-517.
- BENTLEY, J. L., AND FRIEDMAN, J. H. 1979. Data structures for range searching. *ACM Comput. Surv.* 11, 4 (Dec.), 397-409.
- BENTLEY, J. L., AND STANAT, D. F. 1975. Analysis of range searches in quad trees. *Inf. Process. Lett.* 3, 6 (July), 170-173.
- BESSIER, P. W. 1982. Quadtree construction of binary images by dyadic array transformations. In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing* (Las Vegas, Nev., June). IEEE, New York, pp. 550-554.
- BLUM, H. 1967. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. M.I.T. Press, Cambridge, Mass., pp. 362-380.
- BROOKS, R. A., AND LOZANO-PEREZ, T. 1983. A subdivision algorithm in configuration space for findpath with rotation. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence* (Karlsruhe, West Germany, Aug.). Kaufmann, Los Altos, Calif., pp. 799-806.
- BURKHARDT, W. A. 1983. Interpolation-based index maintenance. *BIT* 23, 3, 274-294.
- BURT, P. J. 1980. Tree and pyramid structures for coding hexagonally sampled binary images. *Comput. Graphics Image Process.* 14, 3 (Nov.), 249-270.
- BURT, P. J., HONG, T. H., AND ROSENFELD, A. 1981. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Trans. Syst. Man Cybern.* 11, 12 (Dec.), 802-809.
- BURTON, W. 1977. Representation of many-sided polygons and polygonal lines for rapid processing. *Commun. ACM* 20, 3 (Mar.), 166-171.
- BURTON, F. W., AND KOLLIAS, J. G. 1983. Comment on the explicit quadtree as a structure for computer graphics. *Comput. J.* 26, 2 (May), 188.
- CARLSON, W. E. 1982. An algorithm and data structure for 3D object synthesis using surface patch intersections. *Comput. Gr. SIGGRAPH 82 Conf. Proc.* 16, 3, 255-264.
- CHEN, C. H., AND AGGARWAL, J. K. 1984. A normalized quadtree representation. *Comput. Vision Gr. Image Process.* 26, 3 (June), 331-346.
- COHEN, E., LYCHE, T., AND RIESENFELD, R. 1980. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Comput. Gr. Image Process.* 14, 3 (Oct.), 87-111.
- COMER, D. 1979. The ubiquitous B-tree. *ACM Comput. Surv.* 11, 2 (June), 121-137.
- CONNOLLY, C. I. 1984. Cumulative generation of octree models from range data. In *Proceedings of the International Conference on Robotics* (Atlanta, Ga., Mar.). IEEE, New York, pp. 25-32.
- COOK, B. G. 1978. The structural and algorithmic basis of a geographic data base. In *Proceedings of the First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems*, G. Dutton, Ed., Harvard Papers on Geographic Information Systems, Harvard University Press, Cambridge, Mass.
- CREUTZBURG, E. 1981. Complexities of quadtrees and the structure of pictures. Tech. Rep. N/81/74, Computer Science Dept., Friedrich-Schiller University, Jena, East Germany.
- DAVIS, L. S., AND ROUSSOPOULOS, N. 1980. Approximate pattern matching in a pattern database system. *Inf. Syst.* 5, 2, 107-119.
- DECOULON, F., AND JOHNSEN, U. 1976. Adaptive block schemes for source coding of black-and-white facsimile. *Electron. Lett.* 12, 3, 61-62. See also erratum, *Electron. Lett.* 12, 6 (1976), 152.
- DEFIORIANI, L., FALCIDIENO, B., NAGY, G., AND PIENOVI, C. 1982. Yet another method for triangulation and contouring for automated cartography. In *Proceedings of the American Congress on Surveying and Mapping* (Hollywood, Fla.), F. S. Cardwell, R. Black, and B. M. Cole, Eds. American Society of Photogrammetry, pp. 101-110.
- DEMILLO, R. A., EISENSTAT, S. C., AND LIPTON, R. J. 1978. Preserving average proximity in arrays. *Commun. ACM* 21, 3 (Mar.), 228-231.
- DOCTOR, L. J., AND TORRONG, J. G. 1981. Display techniques for octree-encoded objects. *IEEE Comput. Gr. Appl.* 1, 1 (July), 39-46.

- DUDA, R. O., AND HART, P. E. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.
- DYER, C. R. 1980. Computing the Euler number of an image from its quadtree. *Comput. Gr. Image Process.* 13, 3 (July), 270-276.
- DYER, C. R. 1981. A VLSI pyramid machine for parallel image processing. In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing* (Dallas, Tex.). IEEE, New York, pp. 381-386.
- DYER, C. R. 1982. The space efficiency of quadtrees. *Comput. Gr. Image Process.* 19, 4 (Aug.), 335-348.
- DYER, C. R., ROSENFELD, A., AND SAMET, H. 1980. Region representation: Boundary codes from quadtrees. *Commun. ACM* 23, 3 (Mar.), 171-179.
- EASTMAN, C. M. 1970. Representations for space planning. *Commun. ACM* 13, 4 (Apr.), 242-250.
- EDELSBRUNNER, H. 1984. Key-problems and key-methods in computational geometry. In *Proceedings of the Symposium of Theoretical Aspects of Computer Science* (Paris, France), Lecture Notes in Computer Science, vol. 166. Springer Verlag, New York, pp. 1-13.
- EDELSBRUNNER, H., AND VAN LEEUWEN, J. 1983. Multidimensional data structures and algorithms: A bibliography. Rep. F104, Institute for Information Processing, Technical University of Graz, Graz, Austria, Jan.
- EDELSBRUNNER, H., GUIBAS, L. J., AND STOLFI, J. 1985. Optimal point location in a monotone subdivision. *SIAM J. Comput.* in press.
- FAGIN, R., NIEVERGELT, J., PIPPENGER, N., AND STRONG, H. R. 1979. Extendible hashing--A fast access method for dynamic files. *ACM Trans. Database Syst.* 4, 3 (Sept.), 315-344.
- FAUGERAS, O. D., AND PONCE, J. 1983. Prism trees: A hierarchical representation for 3-d objects. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence* (Karlsruhe, West Germany, Aug.). Kaufmann, Los Altos, Calif., pp. 982-989.
- FAUGERAS, O. D., HERBERT, M., MUSSI, P., AND BOISSONNAT, J. D. 1984. Polyhedral approximation of 3-d objects without holes. *Comput. Vision Gr. Image Process.* 25, 2 (Feb.), 169-183.
- FAVERJON, B. 1984. Obstacle avoidance using an octree in the configuration space of a manipulator. In *Proceedings of the International Conference on Robotics* (Atlanta, Ga., Mar.). IEEE, New York, pp. 504-512.
- FINKEL, R. A., AND BENTLEY, J. L. 1974. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.* 4, 1, 1-9.
- FREDKIN, E. 1980. Trie memory. *Commun. ACM* 3, 9 (Sept.), 490-499.
- FREEMAN, H. 1974. Computer processing of line-drawing images. *ACM Comput. Surv.* 6, 1 (Mar.), 57-97.
- FRIEDMAN, J. H., BENTLEY, J. L., AND FINKEL, R. A. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.* 3, 3 (Sept.), 209-226.
- GARGANTINI, I. 1982a. An effective way to represent quadtrees. *Commun. ACM* 25, 12 (Dec.), 905-910.
- GARGANTINI, I. 1982b. Linear octrees for fast processing of three dimensional objects. *Comput. Gr. Image Process.* 20, 4 (Dec.), 365-374.
- GARGANTINI, I. 1983. Translation, rotation, and superposition of linear quadtrees. *Int. J. Man-Mach. Stud.* 18, 3 (Mar.), 253-263.
- GASTON, P. C., AND LOZANO-PEREZ, T. 1984. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 3 (May), 257-266.
- GIBSON, L., AND LUCAS, D. 1982. Vectorization of raster images using hierarchical methods. *Comput. Gr. Image Process.* 20, 1 (Sept.), 82-89.
- GILLESPIE, R., AND DAVIS, W. A. 1981. Tree data structures for graphics and image processing. In *Proceedings of the 7th Conference of the Canadian Man-Computer Communications Society* (Waterloo, Canada, June), pp. 155-161.
- GOMEZ, D., AND GUZMAN, A. 1979. Digital model for three-dimensional surface representation. *Geo-Process.* 1, 53-70.
- GROSS, W. I., AND JAIN, R. 1983. Optimal quadtrees for image segments. *IEEE Trans. Pattern Anal. Mach. Intell.* 5, 1 (Jan.), 77-83.
- HARARY, F. 1969. *Graph Theory*. Addison-Wesley, Reading, Mass.
- HERTEL, S., AND MEHLHORN, K. 1983. Fast triangulation of simple polygons. In *Proceedings of the 1983 International Foundations of Computation Theory Conference* (Borgholm, Sweden, Aug.), Lecture Notes in Computer Science, vol. 158. Springer Verlag, New York, pp. 207-216.
- HINRICHS, K., AND NIEVERGELT, J. 1983. The Grid File: A data structure to support proximity queries on spatial objects. Rep. 54, Institut für Informatik, ETH, Zurich, July.
- HOARE, C. A. R. 1972. Notes on data structuring. In *Structured Programming*, O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare, Eds. Academic Press, London, p. 184.
- HOROWITZ, S. L., AND PAVLIDIS, T. 1976. Picture segmentation by a tree traversal algorithm. *J. ACM* 23, 2 (Apr.), 368-388.
- HUFFMAN, D. A. 1952. A method for the construction of minimum-redundancy codes. *Proc. IRE* 40, 9 (Sept.), 1099-1101.
- HUNTER, G. M. 1978. Efficient computation and data structures for graphics. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, N. J.
- HUNTER, G. M., AND STEIGLITZ, K. 1979a. Operations on images using quad trees. *IEEE Trans. Pattern Anal. Mach. Intell.* 1, 2 (Apr.), 146-158.
- HUNTER, G. M., AND STEIGLITZ, K. 1979b. Linear transformation of pictures represented by quad

- trees. *Comput. Gr. Image Process.* 10, 3 (July), 289-296.
- IBRAHIM, H. A. H. 1984. The connected component labeling algorithm on the NON-VON supercomputer. In *Proceedings of the Workshop on Computer Vision: Representation and Control* (Annapolis, Md., Apr.). IEEE, New York, pp. 37-45.
- ISMAIL, M. G. B., AND STEELE, R. 1980. Adaptive pel location coding for bilevel facsimile signals. *Electron. Lett.* 16, (May), 361-363.
- JACKINS, C., AND TANIMOTO, S. L. 1980. Oct-trees and their use in representing three-dimensional objects. *Comput. Gr. Image Process.* 14, 3 (Nov.), 249-270.
- JACKINS, C., AND TANIMOTO, S. L. 1983. Quad-trees, oct-trees, and h -trees—A generalized approach to recursive decomposition of Euclidean space. *IEEE Trans. Pattern Anal. Mach. Intell.* 5, 5 (Sept.), 533-539.
- JONES, L., AND IVENGAR, S. S. 1984. Space and time efficient virtual quadtrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 2 (Mar.), 244-247.
- KAWAGUCHI, E., AND ENDO, T. 1980. On a method of binary picture representation and its application to data compression. *IEEE Trans. Pattern Anal. Mach. Intell.* 2, 1 (Jan.), 27-35.
- KAWAGUCHI, E., ENDO, T., AND YOKOTA, M. 1980. DF-expression of binary-valued picture and its relation to other pyramidal representations. In *Proceedings of the 5th International Conference on Pattern Recognition* (Miami Beach, Fla., Dec.). IEEE, New York, pp. 822-827.
- KAWAGUCHI, E., ENDO, T., AND MATSUNAGA, J. 1983. Depth-first expression viewed from digital picture processing. *IEEE Trans. Pattern Anal. Mach. Intell.* 5, 4 (July), 373-384.
- KEDEM, G. 1981. The Quad-CIF tree: A data structure for hierarchical on-line algorithms. Tech. Rep. 91, Computer Science Dept., The University of Rochester, Rochester, New York, Sept.
- KELLY, M. D. 1971. Edge detection in pictures by computer using planning. *Mach. Intell.* 6, 397-409.
- KIRKPATRICK, D. 1963. Optimal search in planar subdivisions. *SIAM J. Comput.* 12, 1 (Feb.), 28-35.
- KLINGER, A. 1971. Patterns and search statistics. In *Optimizing Methods in Statistics*, J. S. Rustagi, Ed. Academic Press, New York, pp. 303-337.
- KLINGER, A., AND DYER, C. R. 1976. Experiments in picture representation using regular decomposition. *Comput. Gr. Image Process.* 5, 1 (Mar.), 68-105.
- KLINGER, A., AND RHODES, M. L. 1979. Organization and access of image data by areas. *IEEE Trans. Pattern Anal. Mach. Intell.* 1, 1 (Jan.), 50-60.
- KNOTT, G. D. 1971. Expandable open addressing hash table storage and retrieval. In *Proceedings of SIGFIDET Workshop on Data Description, Access, and Control* (San Diego, Calif., Nov.). ACM, New York, pp. 187-206.
- KNOWLTON, K. 1980. Progressive transmission of grey-scale and binary pictures by simple, efficient, and lossless encoding schemes. *Proc. IEEE* 68, 7 (July), 885-896.
- KNUTH, D. E. 1973. *The Art of Computer Programming*, vol. 3, *Sorting and Searching*. Addison-Wesley, Reading, Mass.
- KNUTH, D. E. 1975. *The Art of Computer Programming*, vol. 1, *Fundamental Algorithms*, 2nd ed. Addison-Wesley, Reading, Mass.
- LAUZON, J. P., MARK, D. M., KIKUCHI, L., AND GUEVARA, J. A. 1984. Two-dimensional run-encoding for quadtree representation. Unpublished manuscript, Department of Geography, State University of New York at Buffalo, Buffalo, New York.
- LEE, D. T., AND SHACTER, B. J. 1980. Two algorithms for constructing a Delaunay triangulation. *Int. J. Comput. Inf. Sci.* 9, 3 (June), 219-242.
- LEE, D. T., AND WONG, C. K. 1977. Worst-case analysis for region and partial region searches in multidimensional binary search trees and quad trees. *Acta Inf.* 9, 1, 23-29.
- LETELIER, P. 1983. Transmission d'images à bas débit pour un système de communication téléphonique adapté aux sourds. Thèse de docteur-ingénieur, Université de Paris-Sud, Paris, Sept.
- LI, M., GROSSKY, W. I., AND JAIN, R. 1982. Normalized quadrees with respect to translations. *Comput. Gr. Image Process.* 20, 1 (Sept.), 72-81.
- LINN, J. 1973. General methods for parallel searching. Tech. Rep. 81, Digital Systems Laboratory, Stanford University, Stanford, Calif., May.
- LIPTON, R. J., AND TARJAN, R. E. 1977. Application of a planar separator theorem. In *Proceedings of the 18th Annual IEEE Symposium on the Foundations of Computer Science* (Providence, R. I., Oct.). IEEE, New York, pp. 162-170.
- LITWIN, W. 1980. Linear hashing: A new tool for file and table addressing. In *Proceedings of the 6th International Conference on Very Large Data Bases* (Montreal, Oct.). IEEE, New York, pp. 212-223.
- LOZANO-PEREZ, T. 1981. Automatic planning of manipulator transfer movements. *IEEE Trans. Syst. Man Cybern.* 11, 10 (Oct.), 681-698.
- LUMIA, R. 1983. A new three-dimensional connected components algorithm. *Comput. Vision Gr. Image Process.* 23, 2 (Aug.), 207-217.
- LUMIA, R., SHAPIRO, L., AND ZUNIGA, O. 1983. A new connected components algorithm for virtual memory computers. *Comput. Vision Gr. Image Process.* 22, 2 (May), 287-300.
- MARTIN, J. J. 1982. Organization of geographical data with quad trees and least square approximation. In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing* (Las Vegas, Nev., June). IEEE, New York, pp. 458-463.
- MATSUYAMA, T., HAO, L. V., AND NAGAO, M. 1984. A file organization for geographic information systems based on spatial proximity. *Com-*

- put. *Vision Gr. Image Process.* 26, 3 (June), 303-318.
- MCCLUSKEY, E. J. 1965. *Introduction to the Theory of Switching Circuits*. McGraw-Hill, New York, pp. 60-61.
- McKEOWN, D. M., JR., AND DENLINGER, J. L. 1984. Map-guided feature extraction from aerial imagery. In *Proceedings of the Workshop on Computer Vision: Representation and Control* (Annapolis, Md., Apr.). IEEE, New York, pp. 205-213.
- MEAGHER, D. 1982. Geometric modeling using octree encoding. *Comput. Gr. Image Process.* 19, 2 (June), 129-147.
- MERRETT, T. H. 1978. Multidimensional paging for efficient database querying. In *Proceedings of the International Conference on Management of Data* (Milan, Italy, June), pp. 277-289.
- MERRETT, T. H., AND OTOO, E. J. 1981. Dynamic multipaging: A storage structure for large shared data banks. In *Improving Database Usability and Responsiveness*, P. Scheuermann, Ed. Academic Press, New York, pp. 237-254.
- MERRILL, R. D. 1973. Representations of contours and regions for efficient computer search. *Commun. ACM* 16, 2 (Feb.), 69-82.
- MILFORD, D. J., AND WILLIS, P. C. 1964. Quad encoded display. *IEE Proc.* 131, E3 (May), 70-76.
- MINSKY, M., AND PAPERT, S. 1969. *Perceptrons: An Introduction to Computational Geometry*. M.I.T. Press, Cambridge, Mass.
- MORTON, G. M. 1966. A computer oriented geodetic data base and a new technique in file sequencing. Unpublished manuscript, IBM, Ltd., Ottawa, Canada.
- MUDUR, S. P., AND KOPARKAR, P. A. 1984. Interval methods for processing geometric objects. *IEEE Comput. Gr. Appl.* 4, 2 (Feb.), 7-17.
- NAGY, G., AND WAGLE, S. 1979. Geographic data processing. *ACM Comput. Surv.* 11, 2 (June), 139-181.
- NIEVERGELT, J., AND PREFARATA, F. P. 1982. Plane-sweep algorithms for intersecting geometric figures. *Commun. ACM* 25, 10 (Oct.), 739-746.
- NIEVERGELT, J., HINTERBERGER, H., AND SEVCIK, K. C. 1984. The Grid File: An adaptable, symmetric multikey file structure. *ACM Trans. Database Syst.* 9, 1 (Mar.), 38-71.
- NILSSON, N. J. 1969. A mobile automaton: An application of artificial intelligence techniques. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence* (Washington D.C.). Kaufmann, Los Altos, Calif., pp. 509-520.
- OLIVER, M. A., AND WISEMAN, N. E. 1983a. Operations on quadtree-encoded images. *Comput. J.* 26, 1 (Feb.), 83-91.
- OLIVER, M. A., AND WISEMAN, N. E. 1983b. Operations on quadtree leaves and related image areas. *Comput. J.* 26, 4 (Nov.), 375-380.
- OMOLAYOLE, J. O., AND KLINGER, A. 1980. A hierarchical data structure scheme for storing pictures. In *Pictorial Information Systems*, S. K. Chang and K. S. Fu, Eds. Springer-Verlag, Berlin and New York, 1980.
- ORENSTEIN, J. A. 1982. Multidimensional tries used for associative searching. *Inf. Process. Lett.* 14, 4 (June), 150-157.
- ORENSTEIN, J. A. 1983. A dynamic hash file for random and sequential accessing. In *Proceedings of the 6th International Conference on Very Large Data Bases* (Florence, Italy, Oct.). IEEE, New York, pp. 132-141.
- ORENSTEIN, J. A., AND MERRETT, T. H. 1984. A class of data structures for associative searching. In *Proceedings of the 3rd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (Waterloo, Ontario, Apr.). ACM, New York, pp. 181-190.
- O'ROURKE, J. 1981. Dynamically quantized spaces for focusing the Hough Transform. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence* (Vancouver, B.C., Aug.). Kaufmann, Los Altos, Calif., pp. 737-739.
- O'ROURKE, J., AND SLOAN, K. R., JR. 1984. Dynamic quantization: Two adaptive data structures for multidimensional squares. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 3 (May), 266-280.
- OUKSEL, M., AND SCHEUERMANN, P. 1983. Storage mappings for multidimensional linear dynamic hashing. In *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems* (Atlanta, Ga., Mar.). ACM, New York, pp. 90-105.
- OVERMARS, M. H. 1983. *The Design of Dynamic Data Structures*, Lecture Notes in Computer Science, vol. 156. Springer-Verlag, New York.
- OVERMARS, M. H., AND VAN LEEUWEN, J. 1982. Dynamic multi-dimensional data structures based on quad- and k-d trees. *Acta Inf.* 17, 3, 267-285.
- PETERS, F. 1984. An algorithm for transformations of pictures represented by quadtrees. Dept. of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.
- PEUCKER, T. 1976. A theory of the cartographic line. *Int. Yearb. Cartogr.* 16, 34-43.
- PEUQUET, D. J. 1979. Raster processing: An alternative approach to automated cartographic data handling. *Am. Cartogr.* 2 (Apr.), 129-239.
- PEUQUET, D. J. 1983. A hybrid data structure for the storage and manipulation of very large spatial data sets. *Comput. Vision Gr. Image Process.* 24, 1 (Oct.), 14-27.
- PFALTZ, J. L., AND ROSENFELD, A. 1987. Computer representation of planar regions by their skeletons. *Commun. ACM* 19, 2 (Feb.), 119-122.
- PIETIKAINEN, M., ROSENFELD, A., AND WALTER, I. 1982. Split-and-link algorithms for image segmentation. *Pattern Recognition* 15, 4, 287-293.
- RAMAN, V., AND IVENGAR, S. S. 1983. Properties and applications of forests of quadtrees for pictorial data representation. *BIT* 23, 4, 472-486.

- RANADE, S. 1981. Use of quadrees for edge enhancement. *IEEE Trans. Syst. Man, Cybern.* 11, 5 (May), 370-373.
- RANADE, S., AND SHNEIER, M. 1981. Using quadrees to smooth images. *IEEE Trans. Syst. Man Cybern.* 11, 5 (May), 373-376.
- RANADE, S., ROSENFELD, A., AND PREWITT, J. M. S. 1980. Use of quadrees for image segmentation. TR-878, Dept. of Computer Science, University of Maryland, College Park, Md., Feb.
- RANADE, S., ROSENFELD, A., AND SAMET, H. 1982. Shape approximation using quadrees. *Pattern Recognition* 15, 1, 31-40.
- REDDY, D. R., AND RUBIN, S. 1976. Representation of three-dimensional objects. CMU-CS-78-113, Computer Science Dept., Carnegie-Mellon University, Pittsburgh, Apr.
- REQUICHA, A. A. G. 1980. Representations of rigid solids: Theory, methods, and systems. *ACM Comput. Surv.* 12, 4 (Dec.), 437-464.
- RHEINBOLDT, W. C., AND MESZTENYI, C. K. 1980. On a data structure for adaptive finite element mesh refinements. *ACM Trans. Math. Softw.* 6, 2 (June), 166-187.
- RISEMAN, E. M., AND ARBIB, M. A. 1977. Computational techniques in the visual segmentation of static scenes. *Comput. Gr. Image Process.* 6, 3 (June), 221-276.
- ROBINSON, J. T. 1981. The k-d-B-tree: A search structure for large multidimensional dynamic indexes. In *Proceedings of the SIGMOD Conference* (Ann Arbor, Mich., Apr.). ACM, New York, pp. 10-18.
- ROSENFELD, A., ED. 1983. *Multiresolution Image Processing and Analysis*. Springer-Verlag, Berlin and New York.
- ROSENFELD, A. 1984. Picture processing 1984. *Comput. Vision Gr. Image Process.* 26, 3 (June), 347-384.
- ROSENFELD, A., AND KAK, A. C. 1982. *Digital Picture Processing*, 2nd ed. Academic Press, New York.
- ROSENFELD, A., AND PEALTY, J. L. 1966. Sequential operations in digital image processing. *J. ACM* 13, 4 (Oct.), 471-484.
- ROSENFELD, A., SAMET, H., SHAFFER, C., AND WEBER, R. E. 1982. Application of hierarchical data structures to geographical information systems. TR-1197, Computer Science Dept., University of Maryland, College Park, Md., June.
- ROSENFELD, A., SAMET, H., SHAFFER, C., AND WEBER, R. E. 1983. Application of hierarchical data structures to geographical information systems phase II. TR-1327, Computer Science Dept., University of Maryland, College Park, Md., Sept.
- ROTH, S. D. 1982. Ray casting for modeling solids. *Comput. Gr. Image Process.* 18, 2 (Feb.), 109-144.
- RETÖVITZ, D. 1968. Data structures for operations on digital images. In *Pictorial Pattern Recognition*, G. C. Cheng et al., Eds. Thompson Book Co., Washington D.C., pp. 105-133.
- SAMET, H. 1980a. Region representation: Quadrees from boundary codes. *Commun. ACM* 23, 3 (Mar.), 163-170.
- SAMET, H. 1980b. Region representation: Quadrees from binary arrays. *Comput. Gr. Image Process.* 13, 1 (May), 83-93.
- SAMET, H. 1980c. Deletion in two-dimensional quad trees. *Commun. ACM* 23, 12 (Dec.), 703-710.
- SAMET, H. 1981a. An algorithm for converting rasters to quadrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 3, 1 (Jan.), 93-95.
- SAMET, H. 1981b. Connected component labeling using quadrees. *J. ACM* 28, 3 (July), 487-501.
- SAMET, H. 1981c. Computing perimeters of images represented by quadrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 3, 6 (Nov.), 683-687.
- SAMET, H. 1982a. Neighbor finding techniques for images represented by quadrees. *Comput. Gr. Image Process.* 18, 1 (Jan.), 37-57.
- SAMET, H. 1982b. Distance transform for images represented by quadrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 4, 3 (May), 298-303.
- SAMET, H. 1983. A quadtree medial axis transform. *Commun. ACM* 26, 9 (Sept.), 680-693. See also corrigendum, *Commun. ACM* 27, 2 (Feb. 1984), 151.
- SAMET, H. 1984. Algorithms for the conversion of quadrees to rasters. *Comput. Vision Gr. Image Process.* 26, 1 (Apr.), 1-16.
- SAMET, H. 1985a. A top-down quadtree traversal algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* 7, 1 (Jan.) in press. Also TR-1237, Computer Science Dept., University of Maryland.
- SAMET, H. 1985b. Reconstruction of quadrees from quadtree medial axis transforms. *Comput. Vision Gr. Image Process.* 29, 2 (Feb.) in press. Also TR-1224, Computer Science Dept., University of Maryland.
- SAMET, H. 1985c. Data structures for quadtree approximation and compression. *Commun. ACM* 28 in press. Also TR-1209, Computer Science Dept., University of Maryland.
- SAMET, H., AND KRISHNAMURTHY, E. V. 1983. A quadtree-based matrix manipulation system. Unpublished manuscript of work in progress.
- SAMET, H., AND ROSENFELD, A. 1980. Quadtree structures for image processing. In *Proceedings of the 5th International Conference on Pattern Recognition* (Miami Beach, Fla., Dec.). IEEE, New York, pp. 815-818.
- SAMET, H., AND SHAFFER, C. A. 1984. A model for the analysis of neighbor finding in pointer-based quadrees. TR-1432, Computer Science Dept., University of Maryland, College Park, Md., Aug.
- SAMET, H., AND TAMMINEN, M. 1984. Efficient image component labeling. TR-1420, Computer Science Dept., University of Maryland, College Park, Md., July.
- SAMET, H., AND TAMMINEN, M. 1985. Computing geometric properties of images represented by

- linear quadtrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 7, 1 (Jan.) in press. Also TR-1359, Computer Science Dept., University of Maryland.
- SAMET, H., AND WEBBER, R. E. 1982. On encoding boundaries with quadtrees. TR-1162, Computer Science Dept., University of Maryland, College Park, Md., Feb.
- SAMET, H., AND WEBBER, R. E. 1983. Using quadtrees to represent polygonal maps. In *Proceedings of Computer Vision and Pattern Recognition 83* (Washington, DC, June). IEEE, New York, pp. 127-132. Also TR-1372, Computer Science Dept., University of Maryland.
- SAMET, H., AND WEBBER, R. E. 1984. On encoding boundaries with quadtrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 3 (May), 365-369.
- SHAMOS, M. I. 1978. Computational geometry. Ph.D. dissertation, Dept. of Computer Science, Yale University, New Haven, Conn.
- SHAMOS, M. I., AND HOEY, D. 1975. Closest-point problems. In *Proceedings of the 16th Annual IEEE Symposium on the Foundations of Computer Science* (Berkeley, Calif., Oct.). IEEE, New York, pp. 151-162.
- SHNEIER, M. 1981a. Calculations of geometric properties using quadtrees. *Comput. Gr. Image Process.* 16, 3 (July), 296-302.
- SHNEIER, M. 1981b. Path-length distances for quadtrees. *Inf. Sci.* 23, 1 (Feb.), 49-67.
- SHNEIER, M. 1981c. Two hierarchical linear feature representations: Edge pyramids and edge quadtrees. *Comput. Gr. Image Process.* 17, 3 (Nov.), 211-224.
- SLOAN, K. R., JR. 1981. Dynamically quantized pyramids. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence* (Vancouver, B.C., Aug.). Kaufmann, Los Altos, Calif., pp. 734-736.
- SLOAN, K. R., JR., AND TANIMOTO, S. L. 1979. Progressive refinement of raster images. *IEEE Trans. Comput.* 28, 11 (Nov.), 871-874.
- SRIHARI, S. N. 1981. Representation of three-dimensional digital images. *ACM Comput. Surv.* 13, 1 (Dec.), 399-424.
- SUTHERLAND, I. E., SPROULL, R. F., AND SCHUMACKER, R. A. 1974. A characterization of ten hidden-surface algorithms. *ACM Comput. Surv.* 6, 1 (Mar.), 1-55.
- TAMMINEN, M. 1981. The EXCELL method for efficient geometric access to data. *Acta Polytech. Scand. Mathematics and Computer Science Series No. 34*, Helsinki.
- TAMMINEN, M. 1982. Hidden lines using the EXCELL method. *Comput. Gr. Forum* 11, 3, 96-105.
- TAMMINEN, M. 1983. Performance analysis of cell based geometric file organizations. *Comput. Vision Gr. Image Process.* 24, 2 (Nov.), 163-181.
- TAMMINEN, M. 1984a. Comment on quad- and oct-trees. *Commun. ACM* 27, 3 (Mar.), 249-249.
- TAMMINEN, M. 1984b. Encoding pixel trees. *Comput. Vision Gr. Image Process.* 28, 1 (Oct.), 44-57.
- TAMMINEN, M., AND SAMET, H. 1984. Efficient oct-tree conversion by connectivity labeling. In *Proceedings of the SIGGRAPH 84 Conference* (Minneapolis, Minn., July). ACM, New York, pp. 43-51.
- TANIMOTO, S. 1976. Pictorial feature distortion in a pyramid. *Comput. Gr. Image Process.* 5, 3 (Sept.), 333-352.
- TANIMOTO, S. 1979. Image transmission with gross information first. *Comput. Graph. Image Process.* 9, 1 (Jan.), 72-76.
- TANIMOTO, S., AND KLINGER, A. EDS. 1980. *Structured Computer Vision*. Academic Press, New York.
- TANIMOTO, S., AND PAVLIDIS, T. 1975. A hierarchical data structure for picture processing. *Comput. Gr. Image Process.* 4, 2 (June), 104-119.
- TARJAN, R. E. 1975. Efficiency of a good but not linear set union algorithm. *J. ACM* 22, 2 (Apr.), 215-225.
- TOUSSAINT, G. T. 1980. Pattern recognition and geometrical complexity. In *Proceedings of the 5th International Conference on Pattern Recognition* (Miami Beach, Fla., Dec.). IEEE, New York, pp. 1324-1346.
- TROFF, H., AND HERZOG, H. 1981. Multidimensional range search in dynamically balanced trees. *Angew. Inf.* 2, 71-77.
- TUCKER, L. W. 1984a. Control strategy for an expert vision system using quadtree refinement. In *Proceedings of the Workshop on Computer Vision: Representation and Control* (Annapolis, Md., Apr.). IEEE, New York, pp. 214-218.
- TUCKER, L. W. 1984b. Computer vision using quadtree refinement. Ph.D. dissertation, Dept. Electrical Engineering and Computer Science, Polytechnic Institute of New York, Brooklyn, N.Y., May.
- UHR, L. 1972. Layered "recognition cone" networks that preprocess, classify, and describe. *IEEE Trans. Comput.* 21, 7 (July), 758-768.
- UNNIKRISHNAN, A., AND VENKATESH, Y. V. 1984. On the conversion of raster to linear quadtrees. Department of Electrical Engineering, Indian Institute of Science, Bangalore, India, May.
- VAN LEEUWEN, J., AND WOOD, D. 1981. The measure problem for rectangular ranges in d-space. *J. Algorithms* 2, 3 (Sept.), 282-300.
- VAN LIEROP, M. L. P. 1984. Transformations on pictures represented by leafcodes. Dept. of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.
- WARNOCK, J. L. 1969. A hidden surface algorithm for computer generated half tone pictures. TR 4-15, Computer Science Dept., University of Utah, Salt Lake City, June.
- WEBBER, R. E. 1984. Analysis of quadtree algorithms. Ph.D. dissertation, Computer Science Dept., University of Maryland, College Park.

- Md., Mar. Also TR-1976, Computer Science Dept., University of Maryland.
- WEBER, W. 1978. Three types of map data structures, their ANDs and NOTs, and a possible OR. In *Proceedings of the 1st International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems*, G. Dutton, Ed. Harvard Papers on Geographic Information Systems, Harvard Univ. Press, Cambridge, Mass.
- WILLARD, D. E. 1982. Polygon retrieval. *SIAM J. Comput.* 11, 1 (Feb.), 149-165.
- WOODWARK, J. R. 1982. The explicit quadtree as a structure for computer graphics. *Comput. J.* 25, 2 (May), 235-238.
- WU, A. Y., HONG, T. H., AND ROSENFELD, A. 1982. Threshold selection using quadtrees. *IEEE Trans. Pattern Anal. Mach. Intell.* 4, 1 (Jan.), 90-94.
- YAMAGUCHI, K., KUNII, T. L., FUJIMURA, K., AND TORIYA, H. 1984. Octree-related data structures and algorithms. *IEEE Comput. Gr. Appl.* 4, 1 (Jan.), 53-59.
- YAU, M. 1984. Generating quadtrees of cross-sections from octrees. *Comput. Vision Gr. Image Process.* 27, 2 (Aug.), 211-238.
- YAU, M., AND SRIHARI, S. N. 1983. A hierarchical data structure for multidimensional digital images. *Commun. ACM* 26, 7 (July), 504-515.
- YERRY, M. A., AND SHEPARD, M. S. 1983. A modified quadtree approach to finite element mesh generation. *IEEE Comput. Gr. Appl.* 3, 1 (Jan./Feb.), 39-46.

Received November 1983; final revision accepted June 1984.

Tel: +1-858-8455080
Fax: +1-848-6585673
Email: {peisongc,yye,martak}@qualcomm.com

Purpose: Proposal

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.

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:

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).

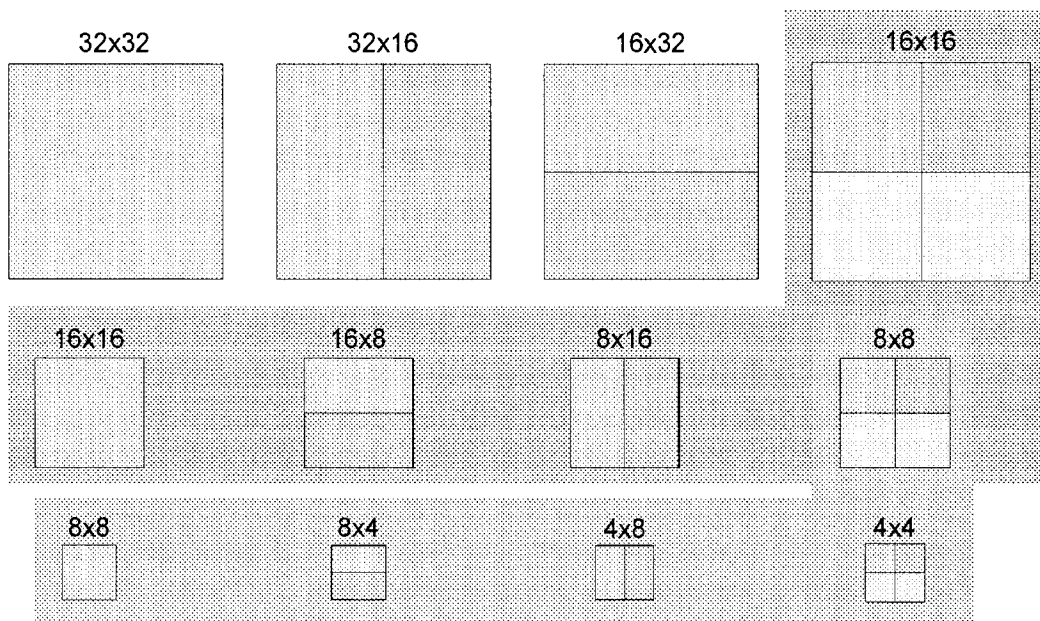


Figure 1: Motion partitions used in the proposed scheme

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).



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).

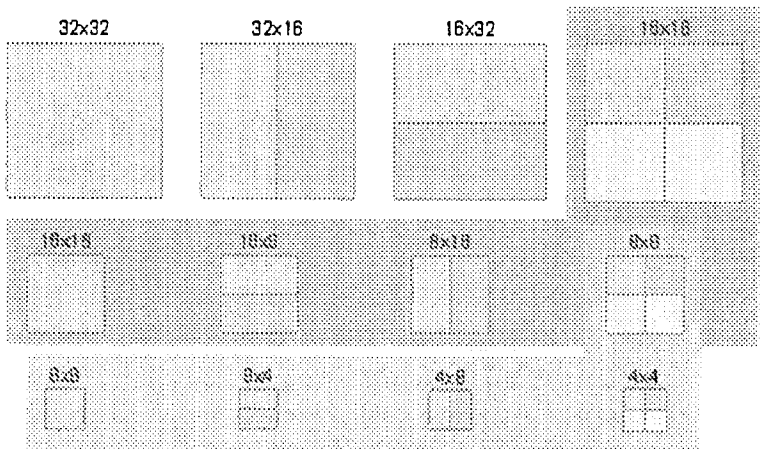


Figure 1: Motion partitions used in the proposed 32x32 blocks

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-A110. 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).



Publication number: KR20090065401 A
Publication country: SOUTH KOREA
Publication type: EXAMINED PATENT APPLICATION
Publication date: 20090622
Application number: KR20080012337
Application date: 20080211
Priority: KR2007013243520071217KR20070132435;
Assignee: ICU RESEARCH AND INDUSTRIAL COOPERATION GROUP ;
Assignee^{std}: ICU RES AND IND COOP GROUP ; KOREAN BROADCASTING SYSTEM ;
Inventor^{std}: PARK KEUN SOO ; KIM MUN CHURL ; PARK CHANG SEOB ; HAHM SANG JIN ;
International class⁸: H04N7/32; H04N7/32; H04N7/24; H04N7/24;
Family members: [KR100951301 B1](#) [KR20090065401 A](#)
Title: METHOD OF INTER-FRAME/INTRA BLOCK PREDICTIVE CODING IN VIDEO CODING
Title: 비디오 부호화에서의 화면간/화면내 예측 부호화 방법

Abstract:

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 blocks having various sizes(S102). inter-frame prediction for the divided sub blocks is performed(S106). One of the predictions is selected and encoded as a predictive encoding(S107).

Abstract:

본 발명은 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 관한 것으로, 부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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 왜곡(warping)을 이용하여 예측 정확도를 크게 향상시킬 수 있으므로 발생 잔차 신호의 양을 줄여 부호화 효율을 더욱 개선할 수 있는 효과가 있다.

Than 16x16 macro block units coded block of the present invention, the screen of the video encoding inter / intra prediction coding method relates to a larger size, consisting of at least one super macro expand into blocks, and each super macro blocks of various sizes sub- divided into blocks, and each super after performing a screen between the predicted macro block and sub-blocks partitioned for by selecting the screen in any one of them, as a result of inter-prediction coding and encoding, to improve the coding efficiency can be improved and are encoded using the conventional block matching method as well as the effect of pixel-based warping motion model (warping) to greatly improve the prediction accuracy by the occurrence of the residual signal by reducing the amount of coding efficiency can

be further improved in

Claims:

1.

비디오 부호화에서의 화면간 예측 부호화 방법에 있어서,

부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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.

2.

제1 항에 있어서,

각 슈퍼 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.

3.

제1 항에 있어서,

상기 슈퍼 매크로 블록의 크기는 16x16 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.

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.

5.

1. In the coding of the screen at the inter-prediction coding method

video,

block units

coding than 16x16 macro blocks consisting of a large size, at least one of the super macro is expanded into blocks, and each super macro blocks of various sizes is divided into sub-blocks, each including super macro block partitioned sub-block predictions about between screens, and then perform inter prediction in video coding, wherein the screen in any one of them, as a result of inter-prediction coding encoded by Screen method of encoding.

2. The method of claim 1

after performing transformation and quantization of the residual signal through the partitioned sub-block of each super macro block

including predictive coding, wherein the entropy coding of the screen in video coding methods.

3. The method of claim 1

square

Super macro block size of the horizontal and vertical length is larger than the size of the 16x16 macro block is defined as a power of 2 above, and it is split sub-defined as a power of 2, which includes the square and rectangular blocks consisting of the block being characterized in that screen between the predictive coding of video coding methods.

4. in the coding of the screen at the inter-prediction coding method

video,

input screen

and coding unit into blocks for motion estimation,

제4 항에 있어서,

상기 부호화 단위 블록은 매크로 블록 또는 16x16 매크로 블록보다 큰 크기의 슈퍼 매크로 블록인 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

6.

제4 항에 있어서,

상기 움직임 모델은 파라미터로 모형화 된 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

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.

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 변환계수에 대해 엔트로피 부호화를 수행하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

9.

비디오 부호화에서의 화면간 예측 부호화 방법에 있어서,

(a) 입력 화면을 미리 정의된 슈퍼 매크로 블록 단위로 확장하는 단계;

(b) 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하는 단계; 및

(c) 각 슈퍼 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

for each encoded unit block in the partitioned by selectively applying at least one of the motion model predictive coding, then any one of them between screens wherein the coding to predict results by selecting screen between the predictive coding of video coding methods

5. The method of claim 4,

coding unit block

above wherein the super macro block is larger than 16x16 macro block or macro block size of screen between the predictive coding of video coding methods.

6. The method of claim 4,

, the motion model wherein the modeling parameters to screen between the predictive coding of video coding methods.

7. The method of claim 4,

, the screen, wherein the translational motion model, doctor (? ?) (Affine) motion models, bilinear (Bilinear) fluoroscopic motion model or motion model (Perspective) that includes at least one of the motion model in video coding between the predictive coding method.

8. The method of claim 4,

, applying the motion model predictive coding process,

quantized transform coefficients and the motion model parameters, model parameter estimation and motion

see from the screen using the motion estimation, the estimated motion model parameters, and the residual signal by converting the quantized transform coefficients, and then of the screen to perform entropy coding for video coding that inter-predictive coding methods.

9. in the coding of the screen at the inter-prediction coding method

video,

predefined Super Macro

단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

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 부호화 하고자 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.

11.

제10 항에 있어서,

상기 단계(d')에서, 상기 움직임 모델 최종 결정은 움직임-왜곡 비용을 고려하여 결정하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

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.

13.

비디오 부호화에서의 화면내 예측 부호화 방법에 있어서,

부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록을

(a) input screen expansion unit block;

(b) steps each super macro blocks of various sizes is divided into sub-blocks, and

(c), and the residual signal obtained thereby determine the optimal block mode and motion vector encoding Super macro blocks as the unit each partitioned sub-block to perform motion estimation from a reference image for the steps to characterized in that it comprises a video of the screen between the predictive coding method coding.

10. in the coding of the screen at the inter-prediction coding method

video,

(a ') encoding for the target block motion model to obtain the motion model parameters, the selection step;

(b ') obtained above using the motion model parameters, see the screen from the steps to generate a motion compensation block of pixels;

motion compensation block pixel value and the current

(c ') the generated residual signal of the pixel values of the block to be encoded to obtain;

Selectable motion model for the above step (b ') and step (c') for

(d ') to perform the movement model for the current block to be encoded, and a final decision and

features that includes the steps to perform the encoding of video coding, motion estimation between screens

(e ') the final selected using a model movement between screens predictive coding methods.

11. according to claim 10,

,

rate

step (d '), the motion model, the final decision - determined by considering the cost of distorting wherein screen between the predictive coding of

로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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.

14.

제13 항에 있어서,

각 슈퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화하는 방향성 예측을 수행하고, 움직임 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화하는 것을 특징으로 하는 비디오 부호화에서의 화면내 예측 부호화 방법.

15.

비디오 부호화에서의 화면간 예측 부호화 방법에 있어서,

부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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.

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 것인지 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;

video coding methods.

12. according to claim 10 or claim 9,

video coding in order to initialize and block sequence coding for the screen, further comprising the step wherein prior to step (a) or step (a) above

screen between the predictive coding method.

13. video coding of intra prediction coding method in

block units

coding than 16x16 macro blocks consisting of a large size, at least one of the super macro is expanded into blocks, and each super macro blocks of various sizes is divided into sub-blocks, each including super macro block partitioned sub-block after intra prediction about video encoding that encoding the final coding results and compared the results with more favorable characteristics in intra prediction coding methods.

14. The method of claim 13

directional intra prediction coding for each Super

, including macro-block partitioned sub-block performs a prediction rate - that most distortion costs that occur less directional intra prediction as the final optimal prediction mode coding wherein the video coding of intra prediction coding methods.

15. in the coding of the screen at the inter-prediction coding method

video,

block units

coding than 16x16 macro blocks consisting of a large size, at least one of the super macro is expanded into blocks, and each super macro blocks of various sizes is divided into sub-blocks, each including super macro block partitioned sub-block my prediction about the screen, and then

(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 움직임 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 블록인지 판단하는 단계; 및

(l) 현재 단위 블록이 현재의 입력 화면에서 최종 블록이면, 현재의 입력 화면이 최종 화면인지 판단하여, 최종 화면이 될 때까지 상기 단계(a) 내지 단계(k)를 반복 수행하는 단계를 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

17.

제16 항에 있어서,

상기 단계(f)에서, 상기 움직임 모델을 적용하여 움직임 예측 수행 시 움직임 모델 파라미터를 추정하는 단계들 더 포함하는 것을 특징으로 하는 비디오 부호화에서의 화면간 예측 부호화 방법.

18.

제1 항 내지 제17 항 중 어느 한 항의 방법을 컴퓨터로 실행시킬 수 있는 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

perform the characteristics that predict more favorable comparisons between the predicted results to screen coding intra prediction results and the results to be encoded to the final coding in video coding between the predictions of the screen encoding methods.

16. in the coding of the screen at the inter-prediction coding method

video,

(a ") to be encoded to the screen to enter the stage;

(b) input screen coding unit into blocks that step:

screen between screens

(c ") the current input to perform predictive coding to determine whether the steps;

(d

") the current input screen rather than the screen between the predicted and the steps to perform the intra prediction coding;

screen between the predictions of the input current screen

(e ") if the initialization sequence encoded in the input screen of the unit block;

(f ") to be encoded in the unit block to perform motion estimation motion model to be applied to one selected to perform motion estimation step;

(g ") the selected motion estimation results using predictive coding;

(h ") (g") in the above steps results of predictive coding and rate-distortion cost to store a value

move above apply

(i ") model, the final model determines whether the final movement model and if the model determines the optimal motion, the final step to select the coding results

motion model

(j ") above apply step (f") to return to the final model, or steps;

block

(k ") in the current unit in the final screen of the current input block to determine whether, and

(l) the current unit the final block of the current block from the input screen, to determine the current input screen is the final screen, the above steps until the final screen to step (a ") (k") features that include repeat the steps to perform to screen between the predictive coding of video coding methods

17. The method of claim 16,

way to perform motion estimation motion model parameter estimation by applying the motion model, further comprising the step of

step (f "), wherein the screen between the predictive coding of video coding.

18. Section 1 to Section 17 of the

recorded to a computer program that can be run as one method of computer-readable recording medium.

Description:

Disclosure 명 세 서

[0001] 본 발명은 비디오 부호화에서 슈퍼 매크로 블록 또는 움직임 모델을 이용한 예측 부호화 방법에 관한 것으로, 기존의 16x16 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 왜곡을 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.

[0002] 일반적으로, ISO/IEC와 ITU-T에서 공동으로 표준화한 비디오 압축 부호화 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 효율

Disclosure Description:

0001] The present invention relates Super macro block in a video encoding or using a motion model predictive coding method, the existing 16 x 16 pixels macro block of more than after split to expand the size of the sub-blocks, and a variety of sizes, predictive coding and coding efficiency can be further improved by using it encoded using the conventional block matching method as well as the motion model of pixel-based warping, the accuracy of the predictions is greatly improved by reducing the amount of the residual signal coding efficiency can be improved so that in video coding between screens / screen predictive coding methods on.

The

[0002] Typically, the ISO / IEC and ITU-T in the co standardized video compression coding technology, ISO / IEC 14496-10 (MPEG-4 Advanced Video Coding) or H.264 and ISO / IEC

을에 있어 많은 진전을 이루었다.

[0003] 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(16x16) 단위로 분할한 후 이를 예측 부호화하는 것과는 달리, 매크로 블록을 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 및 4x4 단위로 더욱 세분화하여 분할하고 이들 하위 블록에 대해 예측 부호화를 수행하고 움직임-왜곡 비용의 관점에서 비용 최소화를 발생시키는 블록을 최적의 블록모드로 하여 부호화한다.

[0004] 이렇게 함으로써 미세한 움직임이나 복잡한 영상의 움직임 예측을 보다 효과적으로 수행하여 발생하는 잔차 신호를 크게 줄임으로써 압축 효율을 크게 향상할 수 있었다.

[0005] 도 1은 종래 기술의 H.264|AVC 부호화기에서 화면간 예측 부호화를 위한 16x16 매크로 블록 단위 블록의 분할 블록 타입을 나타낸 도면으로서, H.264에서 사용되는 7가지의 움직임 예측 블록 분할 타입을 나타낸다.

[0006] 도 1을 참조하면, 주로 작은 크기의 블록들(8x8, 8x4, 4x8 및 4x4)은 영상 내의 움직임 영역이 작거나 미세한 움직임의 경우 예측 성능의 큰 향상을 제공한다.

[0007] 반면에 크기가 큰 블록들(16x16, 16x8 및 8x16)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.

[0008] 결론적으로, 움직임 예측 부호화 시에 다양한 블록의 크기를 사용하여 움직임 예측 부호화를 수행하고, 이 중 하기의 수식식 1과 같이 움직임-왜곡 비용을 가장 적게 발생시키는 블록이 최적의 부호화 최종 블록모드로 결정된다.

[0009]

[KR20090065401_A_0001.TIF]

[0010] 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 움직임-왜곡 최적화를 수행하는 모든 블록모드(또는 크

14496 - 10 Amendment 3 (MPEG-4 Scalable Video Coding) standard SMPTE VC-1, only Audio Video coding Standard (AVS) specification, data and video compression efficiency has made a lot of progress.

[0003] improved video compression efficiency of these factors, but there may be several in particular, the existing video coding standards (MPEG-1 Video, MPEG-2 Video, MPEG-4 Part 2 Visual, H.261, Unlike H.263, etc.) to be encoded to the size of the screen divided into macro blocks (16 x 16) unit after it predictive coding, macro block is 16 x 16, 16 x 8, 8 x 16, 8 x 8, 8 x 4, 4 x 8 and 4 x 4 unit and further subdivided by splitting these sub-blocks to perform predictive coding and rate-coded by optimal block mode block causing distortion costs from the point of view of cost minimization should.

[0004] By doing so greatly improves the compression efficiency by reducing the residual signal occurs more effectively by performing complex or subtle movements of the image motion estimation could be.

1

[0005] also in the conventional technology of H.264 | AVC encoder as the drawing shows a block partitioning of block type predictive coding for inter-screen 16 x 16 macro block units, which is used in the H.264 different indicates the type of motion estimation block splitting.

mainly

[0006] Referring to FIG. 1, the small size of the block (8 x 8, 8 x 4, 4 x 8 and 4 x 4) movement within the image area, small or subtle movements of the prediction performance of the provide a major boost.

[0007], on the other hand, a large block (16 x 16, 16 x 8 and 8 x 16) image pixel values homogeneous area or moving a small area, or that the presence of the translational movement such as the movement area of motion estimation are often chosen when coding tendencies. To use the big block motion estimation such as that encoding motion vector generation at factors that increase the coding efficiency is reduced.

[0008] In conclusion, using a different block size motion estimation when the motion prediction coding, coding, and are less likely to Equation 1 as the rate-distortion cost of the most to the optimal block encoding is determined as the final

기)를 의미한다.

[0011] 예컨대, 영상의 균질 영역이 많은 부분을 차지할 경우나 움직임이 일어나지 않는 영역의 부분이 많은 경우 매크로 블록(16x16)이 최종 움직임 예측 블록으로 결정되는 빈도가 크게 증가한다.

[0012] 특히, 부호화하고자 하는 입력 영상의 해상도가 매우 큰 경우 이러한 영상의 균질한 영역이 많은 부분을 차지하나 16x16 크기의 매크로 블록이 최대 블록이므로, 즉 최대 처리블록 단위의 크기가 16x16으로 제한되어, 움직임 예측 부호화 효율을 증대시키는데 한계가 있다. 이로 인해 초고해상도 영상을 부호화하고자 하는 경우 기존의 최대 크기인 16x16 매크로 블록을 사용하면 부호화 효율 증대에 한계를 가지는 문제점이 있다.

[0013] 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.

[0014] 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 변화시켜 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.

[0015] 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.

[0016]

block mode.

[0009]

[KR20090065401_A_0001.TIF]

[0010], where, c and r means the current block and the reference block and D represents the distortion between the two blocks. Lagrangian multiplier λ_{MODE} the Lagrangian Multiplier. In addition, MODE (or size) to perform all of the blocks used in predictive coding, rate-distortion optimization mode means.

[0011] For example, if the homogeneous areas of the image occupy a large portion or movement that does not happen, part of the area in many cases macro block (16 x 16) final motion estimation block, which is determined by the frequency increased significantly.

[0012] In particular, one 16 x 16 macro block size of a large part of these images, if the resolution of the input image to be encoded to a very large homogeneous areas up to block, so the maximum throughput of the unit block size 16 x 16 is limited to improve the motion prediction coding efficiency, there is a limit. As a result, the ultra-high-resolution images to be encoded, if there is a problem to use the existing maximum size of 16 x 16 macro block to increase the coding efficiency has limits.

[0013] On the other hand, the existing video encoding methods when the motion prediction coding advantage of looking through a block matching the location of the most similar block of pixels in the area of the navigation screen within the block to be encoded to see motion vector transmission.

[0014] In this case, the block matching technique is used to perform motion estimation and assume translational motion basically is the case. However, the translational movement (Tracking), as well as the movement of the camera or the motion of objects in the image, zoom (Zoom-as in the position of the camera was fixed by changing the focal length of the zoom lens while on the subject by going to approach shots, Zoom-out of the camera position was fixed by changing the focal length of the zoom lens while moving away from the subject, as shot), rotation (Roll: rotate the camera by shooting a subject), panning (Panning: the vertical axis of the camera is fixed in the state of horizontal movement of the camera), Boom (Boom-up: move the vertical rise of the camera, Boom-down: the camera's vertical

[KR20090065401_A_0002.TIF]

[0017]

[KR20090065401_A_0003.TIF]

[0018] 여기서, u 와 v 는 움직임 벡터를 의미하고, f 는 카메라의 초점 거리, T_x , T_y , T_z 는 OXYZ 공간에서 각각 X축, Y축, Z축에서의 병진 속도 (Translational Velocity)를 나타낸다. 한편, R_x , R_y , R_z 는 각각 각속도 (Angular Velocity)를 나타낸다. x , y 는 투영 이미지 공간에서의 좌표 값을 의미한다.

[0019] 이와 같이 기존의 움직임 보상을 통한 예측 부호화는 병진 운동만을 고려한 움직임 예측이므로 실제 2차원 공간에서의 복잡한 움직임을 제대로 예측하지 못하여 이로 인한 예측 부호화 성능 향상의 한계점을 가지고 있다.

[0020] 본 발명은 전술한 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 목적은 비디오 부호화 시에 최대 16x16 매크로 블록을 통한 예측 부호화 방식을 확장하여 16x16 보다 큰 크기의 블록(이하, "수퍼 매크로 블록"이라 칭함)을 사용하여 움직임 예측 및 보상을 수행하여 보다 효과적인 예측 부호화 방법을 제공하는데 있다.

[0021] 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 복잡한 움직임을 예측하는데 한계가 있는 것을 극복하기 위해 움직임 모델을 이용하여 움직임을 보다 정확히 예측하고 이를 기반으로 예측 부호화를 수행함으로써 기대되는 잔차 신호(Residual Signals)의 발생량을 줄여 부호화 효율을 향상시키는 방법을 제공하는데 있다.

[0022] 전술한 목적을 달성하기 위하여 본 발명의 제1 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 수퍼 매크로 블록으로 확장하고, 각 수퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하여, 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.

[0023] 여기서, 각 수퍼 매크로 블록을 비롯한 분할된 하위 블록들을 통한 잔차 신호에 대하여 변환 및 양자화를 수행한 후, 그 결과를 엔트로피 부호화하는 것이 바람직하다.

descent to move), the camera tilting (Tilting: The horizontal axis of the camera is fixed in the state of the vertical movement), unlike (Dolly-in: approach to the subject, camera Dolly-out: the camera is from the subject retiring). To describe the type of motion generated by the movement of the camera video footage from the schematic drawings for the movement of the camera forms, as Figure 2 indicates.

[0015] 3 motion model to describe the projected two-dimensional image space as a concept for the movement, when the movement of the subject in a real three-dimensional space is projected onto a two-dimensional space in a two-dimensional space of the projected to Equation 2 and Equation 3 represents projected in the space of x-axis and y-axis of motion is expressed as a formula.

[0016]

[KR20090065401_A_0002.TIF]

[0017]

[KR20090065401_A_0003.TIF]

[0018]

here, the motion vectors u and v , f is the focal length of the camera. T_x , T_y , T_z in the space of a OXYZ each represent the X-axis, Y-axis, Z-axis translational velocity of the Translational Velocity. On the other hand, R_x , R_y , R_z represent the angular velocity (Angular Velocity). projection from the image space coordinates x , y value means.

[0019] Thus, through the conventional motion-compensated predictive coding motion estimation considering only translational motion, so can not correctly predict the actual two-dimensional space of complex movements, predictive coding performance improvement due to have limitations. / p>

[0020] The present invention is been made in order to solve the aforementioned problems, it is an object of the present invention through predictive coding encoded video at up to 16 x 16 macro block to expand the size is larger than the 16 x 16's performed by using motion estimation and compensation blocks (hereafter, referred to as "Super macro block") is to provide a method for more effective predictive coding.

[0024] 바람직하게, 상기 슈퍼 매크로 블록의 크기는 16x16 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.

[0025] 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.

[0026] 여기서, 상기 부호화 단위 블록은 매크로 블록 또는 16x16 매크로 블록보다 큰 크기의 슈퍼 매크로 블록인 것이 바람직하다.

[0027] 바람직하게, 상기 움직임 모델은 파라미터로 모델화 된 것이다.

[0028] 바람직하게, 상기 움직임 모델은 병진 움직임 모델, 의사(擬似)(Affine) 움직임 모델, 이중선형(Bilinear) 움직임 모델 또는 투시(Perspective) 움직임 모델 중 적어도 하나를 포함할 수 있다.

[0029] 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 변환 계수에 대해 엔트로피 부호화를 수행할 수 있다.

[0030] 본 발명의 제3 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, (a) 입력 화면을 미리 정의된 슈퍼 매크로 블록 단위로 확장하는 단계; (b) 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하는 단계; 및 (c) 각 슈퍼 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.

[0031] 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)

[0021] It is another object of the present invention in a block-based predictive coding by the translational motion of the existing reference in the navigation area of the screen up to find a similar block prediction coding by camera movement and object movement of a more complex composite using a motion model to predict the movement in order to overcome limitations in how to improve the coding efficiency reduces emissions more accurately predict movements and predictions based on the residual signal is expected by performing encoding Residual Signals is to provide.

[0022] The foregoing purpose to achieve a first aspect of the present invention, at least in the way the screen between the predictive coding of video coding, coding block of units consisting of 16 x 16 macro block size is larger than one Super macro block macro expansion, and each super block into sub-blocks of various sizes, and split the screen between the predicted macro block and sub-blocks partitioned for each Super, after one of them between screens wherein selected as a result of the predictive coding, a coding screen between the predictive coding of video coding is to provide a method.

perform the transformation and quantization of the residual signal through the macro-block, including the sub-blocks partitioned

[0023], where each super, a result that entropy coding is desirable.

[0024] Preferably, the super macro block size of the square, and the length of the horizontal and vertical size is larger than 16 x 16 macro block is defined as a power of 2, it split the sub-block square and rectangle includes a power of 2, which is defined as a block can be done

block for each coding unit, the split between the predictions of the screen in the video coding

[0025] The second aspect of the present invention, a method of encoding in the input screen, and a coding unit into blocks for motion estimation, at least screen in video coding encoded by selecting one of the screen between the predicted results that characterized inter-prediction coding is to provide a method after performing the movements of one model by selectively applying predictive coding.

Unit blocks

[0026], where the encoding Super macro block is

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 방법을 제공하는 것이다.

[0032] 상기 단계(d')에서, 상기 움직임 모델 최종 결정은 움직임-왜곡 비용을 고려하여 결정함이 바람직하다.

[0033] 바람직하게, 상기 단계(a) 또는 단계(a') 이전에 부호화를 위한 화면 순서 및 블록 순서를 초기화하는 단계를 더 포함할 수 있다.

[0034] 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.

[0035] 바람직하게, 상기 단계(c) 또는 단계(d')에서, 상기 화면간 예측을 통한 움직임-왜곡 비용을 계산하고 부호화하고자 하는 매크로 블록 또는 슈퍼 매크로 블록으로 할할 수 있다.

[0036] 본 발명의 제5 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화하고자 하는 화면을 입력하는 단계; 상기 입력된 화면을 미리 정의된 슈퍼 매크로 블록 단위로 분할하는 단계; 각 슈퍼 매크로 블록에 대해 하위 크기의 블록으로 분할하고, 각 슈퍼 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.

[0037] 본 발명의 제6 측면은 비디오 부호화에서의 화면내 예측 부호화 방법에 있어서, 부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하여, 각 슈퍼 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 부호화에서 화면내 예측 부호화 방법을 제공하는 것이다.

of size 16 x 16 macro block or macro block is larger than is desirable.

[0027] Preferably, the motion model parameters is modeled.

[0028] Preferably, the motion model of the translational motion model, doctor (??) (Affine) motion models, bilinear (Bilinear) fluoroscopic motion model or motion model (Perspective) of at least one can be

[0029] Preferably, the process of applying the motion model predictive coding reference frame motion estimation from the motion model parameters, and estimated the model parameters using the estimated motion and the residual signal by converting quantized transform coefficients, and then entropy coding for quantized transform coefficients and the motion model parameters, can be performed.

screen between the predictive coding of video coding in

[0030] The third aspect of the present invention, macro block units to extend the pre-defined (a) input screen Super (b) Each Super Macro to divide the block into sub-blocks of various sizes and (c) the partitioned sub-block for each super macro block units perform motion estimation from a reference image, and then determine the optimal block mode and thereby obtained which includes the step of encoding the motion vectors and the residual signal, wherein between the predictions of the screen in video coding is to provide a method of coding.

movement by selecting the model to obtain the motion model parameters for (a) coding the target block, the screen of the video encoding to the inter-prediction coding method

[0031] the fourth aspect of the present invention, step (b obtain the motion model parameters ") obtained above steps to generate the reference motion compensation block of pixels from the screen; (c) the generated pixel motion compensation block to be encoded and the current value of the block of pixel values using the residual signal and (e ') The final step; (d') the step (b '), and step (c') Selectable motion model for the movement model for the current block to be encoded to a final decision, and is to provide a screen between the predictive coding method wherein selected using a model movement between screens of motion estimation which includes the steps to perform the encoding of

[0038] 여기서, 각 슈퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 물-왜곡 비용을 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화함이 바람직하다.

[0039] 본 발명의 제7 측면은, 비디오 부호화에서의 화면간 예측 부호화 방법에 있어서, 부호화 블록 단위를 16x16 매크로 블록보다 큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 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.

[0040] 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 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 물-왜곡 비용을 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 판단하여, 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.

[0041] 상기 단계(f")에서, 상기 움직임 모델을 적용하여 움직임 예측 수행 시 움직임 모델 파라미터를 추정하는 단계를 더 포함함이 바람직하다.

[0042] 본 발명의 제9 측면은, 상술한 비디오 부호화에서의 화면간 예측 부호화 방법을 실행시키기 위한 프로그램을 기록한 기록매체를 제공한다.

video coding.

[0032] In step (d'), the final motion model to determine the rate-distortion cost determined by considering the desirability.

[0033] Preferably, the steps to initialize the sequence coding for the screen prior to the step (a) or step (a') and block order includes can be.

[0034] Preferably, the input screen, after step (a') after the step (c) in the motion prediction for coding in order to fit the screen, type, block units after splitting, and may further include the steps to initialize the block, the order of the inter-prediction coding is encoded in a single screen.

[0035] Preferably, the step (c) or step (d'), through the screen between the predicted rate-distortion and calculate the cost of coding a macro that you want to block or Super macro block can be summed. / p> between the predictions of the screen in the video coding

[0036] the fifth aspect of the present invention, a method of encoding in super macro block to be encoded to the screen to enter the predefined input screen divided into units steps for each macro block in the super, and split into sub-block size of the screen between the block and its sub-blocks in each super macro forecasts perform the steps of one of the blocks to select the type of the selected including steps to perform predictive coding between screens using the predicted results of block type wherein the screen between the predictive coding of video coding is to provide a method.

[0037] a sixth aspect of the present invention, the coding block in the coding of intra prediction in video coding methods, a unit consisting of a 16 x 16 macro block size is larger than at least one super macro block expansion, and each after performing Super macro blocks of various sizes is divided into sub-blocks, each super macro block and sub-blocks partitioned for intra prediction coding results more favorable, by comparing the results to the final coding intra prediction in video coding wherein is to provide a method of coding.

directional intra prediction coding for

[0038], where each super macro block and sub-blocks partitioned to predict the rate that causes the least distortion cost directional prediction modes, and perform in the final screen of the optimal encoded by the desirability of the prediction.

[0043] 이상에서 설명한 바와 같은 본 발명의 비디오 부호화에서의 화면간/화면내 예측 부호화 방법에 따르면, 부호화하고자 하는 블록의 크기를 기존의 16x16 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.

[0044] 또한, 본 발명에 따르면, 다양한 움직임 모델을 사용하여 보다 정확한 움직임 예측을 수행함으로써 발생하는 잔차 신호의 양을 줄여 부호화 효율을 증대시킬 수 있을 뿐만 아니라 움직임 모델에 의한 창조 영상의 왜곡(Warping)을 통해 정화소로부터 1/2-화소, 1/4-화소, 1/8-화소의 서브-픽셀(Sub-Pel) 단계로 움직임 보상을 수행함으로써 소모되는 움직임 예측의 많은 연산량의 감소를 얻을 수 있는 이점이 있다.

[0045] 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.

[0046] 먼저, 통상적인 화면간 예측은 부호화하고자 블록을 최대 16x16 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 물-왜곡 비용을 최소화하는 블록을 최종 선택하여 예측 부호화를 수행한다.

[0047] 전술한 바와 같이 부호화하고자 하는 블록 단위를 16x16 매크로 블록 단위로 제한하는 것은 영상 내의 균질 영역(Homogeneous Regions) 또는 움직임이 없는 영역에서 화면간 예측을 수행할 때 매크로 블록의 수가 많이 증가한다. 특히, 초고해상도 영상의 부호화에서는 매크로 블록의 크기가 상대적으로 너무 작아 부호화 효율을 떨어뜨리는 결과를 초래한다.

[0048] 도 4는 본 발명의 일 실시예에 적용된 16x16 매크로 블록 크기 이상의 슈퍼 매크로 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.

[0039] the seventh aspect of the present invention, made to units of 16 x 16 macro block is larger than the size of coded block in the way the screen between the predictive coding of video coding, at least to my Super macro block expansion, and after each super macro blocks of various sizes is divided into sub-blocks, each super macro block and sub-blocks partitioned for intra prediction, intra prediction results and the results to be encoded to be encoded between screens between the predictions of the screen characteristics that predict more favorable by comparing the predicted results to the final coding in video coding is to provide a method of encoding.

Article 8 of

[0040] The present invention side of the screen to enter the screen between the predictive coding of video coding in the way, (a ") to be encoded, (b) input screen coding unit block If you are not a prediction; (d ") of the input current screen steps to determine whether to perform predictive coding step (c) current input to split screen between the screen between the screen to perform intra prediction coding predictions; (e ") of the current screen, the screen for entering the liver; (f ") to be encoded in the unit block of steps to initialize the sequence encoded in the input screen of the unit block to perform motion estimation motion to apply the model after the one selected to perform motion estimation step (g) above, the selected motion estimation results using predictive coding; (h ") step (g) from the results of predictive coding and rate-distortion cost steps to save the value; (i ") the motion model applies judge whether the final model, the final movement model if the model determines the optimal motion, and that the final selection of the coding results; (j ") above apply motion model, or block the final step (i ") to return to step (k) units currently in the input screen of the current block to determine whether the final model, and (l) the current unit blocks the current input screen to determine the current input screen is the final screen in the final block, which includes the steps of (a ") to repeat step (k) to perform the above steps until the final screen, which features in video coding between screens is to provide a method for predictive coding.

[0041] in step (f ") on the steps to perform motion estimation motion model parameter estimation by applying the motion model, including the desirability of more.

[0042] the ninth aspect of the present invention provides a recording medium having recorded

[0049] 즉, 32x32 슈퍼 매크로 블록 및 그 하위 분할 블록, 64x64 슈퍼 매크로 블록 및 그 하위 분할 블록, 128x128 슈퍼 매크로 블록 및 그 하위 분할 블록 등의 형태로 블록 크기를 확장할 수 있다.

[0050] 또한, 부호화기의 복잡도를 고려하여 블록 크기의 확장을 제한하거나 그 하위 블록의 블록 분할 타입의 수를 제한할 수 있다. 실제 부호화 시에 전송한 수학적 1에서와 같이 물-왜곡 비용을 최소화하는 블록 모드(Mode)를 선택하여 부호화할 수 있다.

[0051] 뿐만 아니라 크기가 증가된 슈퍼 매크로 블록의 잔차 신호에 대해 변환 부호화(Transform Coding)를 적용할 경우 기존의 4x4, 8x8 또는 16x16 보다 큰 변환 커널(Kernel)을 사용함으로써 부호화 효율을 증가시킬 수 있다.

[0052] 또한, 화면간 예측 부호화에 부호화하고자 하는 블록에 대해 움직임 예측 모델을 선택적으로 적용하여 부호화 효율을 더욱 증가시킬 수 있다.

[0053] 이하에는 움직임 예측 모델에 대하여 구체적으로 설명하기로 한다.

[0054] 움직임 예측 모델을 보다 상세히 설명하기 위해 먼저, f_0 와 f_1 을 각각 부호화하고자 하는 현재 영상과 참조 영상으로 한다. (x, y) 를 f_0 의 부호화하고자 하는 현재 블록의 한 픽셀 좌표값을 나타낸다. 또한, 부호화하고자 하는 현재 프레임의 예측을

[KR20090065401_A_0004.TIF]

라고 하면 참조 영상으로부터 하기의 수학적 4와 같이 정의할 수 있다.[0055]

[KR20090065401_A_0005.TIF]

[0056] 여기서, (u, v) 는 참조 영상의 좌표값을 나타낸다. 상기의 수학적 4는 움직임 보상이 참조 영상의 좌표값을 현재 영상의 좌표값에 대응시키는 기하학적 변환(Geometric Transformation)을 의미하며 하기의 수학적 5와 같이 일반화하여 표현할 수 있다.

[0057]

[KR20090065401_A_0006.TIF]

[0058] 여기서, g_x 와 g_y 는 공간 변환 함수를 나타낸다. 이러한 공간 변환을 텍스처 매핑(Texture Mapping) 또는 이미지 매핑(Image Mapping)으로 불린다.

thereon a program for executing the aforementioned screen between the predictive coding of video coding methods, as

[0043] As described above, the screen of the present invention, such as in video encoding liver / intra prediction coding method, according to the size of the block to be encoded to expand more than the existing 16 x 16 macro block size, and again sub-divided into blocks and the optional use of the homogeneous areas of the image, and used effectively for many images, the movement to be encoded effectively reduces the motion estimation of the load vector and the reduction in the number of predicted coding can be performed in benefits is.

[0044] In addition, according to the present invention, to be more accurate motion prediction model using various movements caused by reducing the amount of the residual signal coding efficiency can be increased as well as the motion model by reference images of from warping (Warping) jeonghwaso takes 1/2- pixels, 1/8- 1/4- pixels, of the pixel sub-Pel (Sub-Pel) and step motion compensation performed by many of the computational complexity of motion estimation reduced There are benefits that can be obtained for.

less than

[0045] with reference to the accompanying drawings, an embodiment of the present invention will be described in detail for Embodiment of the present invention is illustrated in the following can be transformed into several different forms, however, is not the scope of the present invention is not limited to the embodiments detailed below. To explain more fully to those of ordinary skill in the art, the present invention is to provide an embodiment of the present invention will.

[0046] First of all, between the predictions of the conventional screen block to be encoded divided into units by up to 16 x 16 size, macro block coding. In particular, ISO / IEC 14496-10 (MPEG-4 Advanced Video Coding) or H.264 standard coding the macro block is divided into seven sub-blocks, as shown in FIG. 1 at the time of the final block to minimize the rate-distortion cost select and perform predictive coding.

[0047] As described above, limited to 16 x 16 macro block units to the units of the block to be encoded to Homogeneous Regions) images within a homogeneous area (or the area on the screen without any movement between the

[0059] 움직임 예측에 일반적으로 사용되는 블록 정합 방법인 2차원 병진 움직임 모형은 하기의 수학적 식 6과 같이 표현된다.

[0060]

[KR20090065401_A_0007.TIF]

[0061] 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.

[0062] 이러한 복잡한 움직임을 제대로 예측하기 위해서는 보다 정교한 움직임 모델이 필요하다. 하기의 수학적 식 7은 의사(擬似)(Affine) 움직임 모델, 수학적 식 8은 이중선형(Bilinear) 움직임 모델, 그리고 수학적 식 9는 투시(Perspective) 움직임 모델을 각각 나타낸다.

[0063]

[KR20090065401_A_0008.TIF]

[0064]

[KR20090065401_A_0009.TIF]

[0065]

[KR20090065401_A_0010.TIF]

[0066] 여기서, a_i 는 움직임 모델 파라미터를 나타낸다. 움직임 모델 기반의 예측 부호화를 수행할 경우, 상기 모델을 좀 더 정확하고 비용을 최소화하는 모델을 적용하여 부호화함으로써 부호화 효율을 증가시킬 수 있다.

[0067] 본 발명에서는 움직임 모델을 적용하는 단위 블록의 크기는 슈퍼 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.

[0068] 한편, 본 발명의 일 실시예에서는 비디오 부호화에서의 화면간 예측 부호화 방법 위주로 설명하였지만, 이에 국한하지 않으며, 비디오 부호화에서의 화면내 예측 부호화에 적용시킬 수도 있다.

[0069] 즉, 화면내 예측 부호화를 수행할 경우, 먼저, 부호화 블록 단위를 16x16 매크로 블록보다

predictions of the macro block when increase in the number of. In particular, the coding efficiency, ultra-high-resolution imaging of the coding of the macro block size is too small relative to the drop will result.

[0048] 4 drawings as block and block-split type, for example, shows the embodiment of the present invention applied to 16 x 16 macro block size than Super Mat macro expand the size of the block and sub-block split into various sizes by using the above-mentioned homogeneity region coding on the coding of images without movement very effectively can be applied.

[0049] In other words, the 64 x 32 x 32 Super macro block and block, subdivision 64 Super macro block and block subdivision, 128 x 128 Super Macro block and subdivision block in the form of block size can be extended.

[0050] In addition, considering the complexity of the encoder, limit or expand the size of the block, limiting the number of its sub-blocks of a block split type can. Can be encoded by the foregoing Equation (1) as shown in the actual coding block mode (Mode) to minimize the cost of the rate-distortion.

Transform Coding (Transform Coding) Super macro block of the residual signal for an increase in the size you want to apply

[0051], as well as the existing 4 x 4, 8 x 8 or 16 x 16 is larger than converting kernel (Kernel) By using the coding efficiency can be increased.

[0052] In addition, the motion prediction model for predictive coding between screens on the block to be encoded by selectively applying coding efficiency can be increased.

[0053] Hereinafter, the motion prediction model will be described in detail.

[0054] f_c and f_r , respectively, to be encoded to the current image and the reference image for motion prediction model to be described in more detail. Represents the value of the pixel coordinates (x, y) f_c of the current block to be coded. In addition, the prediction of the current frame to be encoded

[KR20090065401_A_0004.TIF]

Referred to Equation 4 and the reference image from [0055] can be defined.

큰 크기로 이루어진 적어도 하나의 슈퍼 매크로 블록으로 확장하고, 각 슈퍼 매크로 블록을 다양한 크기의 하위 블록들로 분할하며, 각 슈퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측을 수행한 후, 그 결과를 비교하여 더 유리한 부호화 결과를 선택하여 최종 부호화한다.

[0070] 이때, 각 슈퍼 매크로 블록을 비롯한 분할된 하위 블록들에 대해 화면내 예측 부호화는 방향성 예측을 수행하고, 움직임 벡터를 가장 적게 발생시키는 방향성 예측 모드를 최적의 최종 화면내 예측으로 하여 부호화함이 바람직하다.

[0071] 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.

[0072] 도 5를 참조하면, 먼저, 부호화를 위한 화면(i) 순서를 초기화($i=0$)한 후(S100), 부호화를 위한 순서에 맞게 화면(i)을 입력한다(S101).

[0073] 그런 다음, 상기 단계S101에서 입력된 화면(i)을 부호화 단위 블록(매크로 블록 또는 슈퍼 매크로 블록)으로 분할한 후(S102), 현재의 화면(i)이 화면간 예측부호화를 수행할 것인지 판단한다(S103).

[0074] 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).

[0075] 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 움직임 벡터를 저장한다(S108).

[0076] 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.

[0077] 그렇지 않고, 상기 단계S109에서의 판단 결과, 사용 움직임 모델이 최종 움직임 모델이면 최적의 움직임 모델을 결정하고 해당 부호화 결과를 최종 선택한다(S110).

[0078] 이후에, 현재 단위 블록(j)이 화면(i)에서 최

KR20090065401_A_0005.TIF]

[0056] where

, (u, v) coordinates of the reference image represents the value. Above Equation 4 to Equation 5 and the geometric transformation of the coordinate values of the reference image to the coordinates of the image corresponds to a value of the current (Geometric Transformation) refers to motion compensation can be expressed by the generalization.

[0057]

[KR20090065401_A_0006.TIF]

[0058] here, g_x and g_y represent space conversion function. These space conversion is referred to as texture or image mapping (Texture Mapping) mapping (Image Mapping).

block matching methods

[0059] motion estimation commonly used in the two-dimensional translational motion model to Equation 6 can be expressed as:

[0060]

[KR20090065401_A_0007.TIF]

As mentioned above, the actual two-dimensional projection

[0061] However, the translational motion, as well as the video zoom (Zoom), rotation (Roll), panning (Panning), boom (Boom), tilting (Tilting), unlike (Dolly a combination of movement, etc.) are mixed.

[0062] In order to correctly predict these complex movements than sophisticated motion model is required. Equation 8 to Equation 7 the doctor (??) (Affine) motion model, dual-linear motion model (Bilinear), and Equation 9 fluoroscopy (Perspective) motion model respectively.

[0063]

[KR20090065401_A_0008.TIF]

[0064]

[KR20090065401_A_0009.TIF]

종 블록인지를 판단하여(S111), 최종 블록이 아니면 다음 부호화 단위 블록(j)을 입력하고 상기 단계S106으로 리턴한다. [0065]

[KR20090065401_A_0010.TIF]

[0079] 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.

[0066] here, a_j represents the motion model parameters. Movements to perform model-based predictive coding, most of the models the rate-distortion to minimize the cost by applying the model by encoding the coding efficiency can be increased.

[0080] 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.

[0067] In the present invention, the size of the block motion model Super macro blocks and sub-blocks can be encoded, depending on the efficiency of the optimal motion models to choose from. On the other hand, an embodiment of the present invention is applied to the motion model parameters into modeling is desirable.

[0081] 예컨대, 컴퓨터가 읽을 수 있는 기록매체는 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.

[0068] On the other hand, the described embodiment of the present invention, a way to focus on the screen between the predictive coding in video coding, but are not limited to, intra prediction in video coding, coding can be applied to.

[0082] 또한, 컴퓨터로 읽을 수 있는 기록매체는 컴퓨터 통신망으로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 읽을 수 있는 코드로서 저장되고 실행될 수 있다.

[0069] In other words, if you want to perform the intra prediction coding, the first coded block units 16 x 16 macro block than at least one super macro expand into blocks, and each super macro blocks consisting of a large size, a variety of size partitioned into sub-blocks of each super macro block and sub-blocks partitioned for intra prediction coding, by comparing the results after the final coding results more favorable is.

[0083] 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.

intra prediction coding for directional predictions, and each super macro block and partitioned sub-block the

Description Drawings 도면의 간단한 설명

[0070] At this time, the final optimal rate that causes the least distortion cost directional prediction mode screen encoded by the desirability of the prediction.

[0084] 도 1은 종래 기술의 H.264/AVC 부호화기에서 화면간 예측 부호화를 위한 16×16 매크로 블록 단위 블록의 분할 블록 타입을 나타낸 도면.

the

[0085] 도 2는 비디오 영상에서의 카메라의 움직임에 의한 움직임 생성 타입을 설명하기 위한 도면.

[0071] 5 as a flow diagram for the overall inter prediction in video coding screen in accordance with an embodiment of the present invention a method of encoding to describe in the video coding, selective motion model-based predictive coding of the screen shows how

[0086] 도 3은 2차원 영상 공간에 투영된 움직임 모델을 설명하기 위한 개념도.

First, the screen coding for (i) the order of initialization (i = 0) after

[0087] 도 4는 본 발명의 일 실시예에 적용된 16×16 매크로 블록 크기 이상의 슈퍼 매트릭스 블록

및 블록 분할 타입의 일 예를 나타낸 도면.

[0088] 도 5는 본 발명의 일 실시예에 따른 비디오 부호화에서의 화면간 예측 부호화 방법을 설명하기 위한 전체적인 흐름도.

[0072] Referring to FIG. 5 (S100), coding for in order to fit screen (i) the input (S101).

[0073] Then, split the screen that you entered in step S101 (i) Super macro block or macro block coding units (blocks) after (S102), the current screen (i) between the screen Predictive Coding whether you want to perform is determined (S103).

[0074] If the step S103, in the judgment of the results between the predictions of the current screen (i) the screen and perform the intra prediction coding (S104), otherwise the current screen (i) of the screen order to initialize the screen (i) within the coding block (j) of the unit between the predicted if (j = 0) is (S105).

[0075], after the encoding units that you want to block (j) to perform motion estimation to a motion model to be applied, and then perform motion estimation (S106), and the results using predictive coding After you have performed (S107), and the resulting predictive coding and rate-distortion cost value (S108),

[0076] Then, using the motion model judging by the final movement model is (S109), if the model is not the final movement, returned to the step S106, then apply to the movement model and motion prediction and process is repeated.

[0077] Otherwise, the result determined in step S109, the final results of the model determines the optimal motion and the corresponding coding the motion model using the final motion model selection (S110) / p> unit block (j) screen, (i) or (S111), the judge in the final block is the final block coding input and

[0078], after the current block (j) units above step S106 returns to the should.

[0079] Otherwise, the result determined in step S111, the current unit block (j) of the screen (i) if the final block in the current screen (i) the final screen, decide whether (S112), the final screen otherwise exit and is returned to the step S101, the next screen will walk you through the process of input and tactics.

[0080] On the other hand, according to an embodiment of the present invention between screens in video coding / intra prediction coding method also can be implemented as computer readable code on a computer readable recording medium is. A computer-readable recording medium includes any data storage device that

can store data which can be read by a computer system.

[0081] For example, computer-readable recording medium as the Romans (ROM), RAM (RAM), CD-Rom (CD-ROM), a magnetic tape, hard disk, floppy disk, removable storage devices, and non-volatile memory (Flash Memory), and optical data storage devices, and carrier waves (eg, be implemented as a form of transmission through the Internet) are included

[0082] In addition, the distributed computer systems connected to computer networks, computer-readable recording medium as a code that you can read in a distributed manner can be stored and executed.

in video coding

[0083] The foregoing in accordance with the present invention between screens / intra predictive coding method for the preferred embodiment described, but the present invention is not limited to, details of the patent claims and invention describes is also possible to conduct several variations within the scope and the accompanying drawings of the present invention belongs to.

BRIEF DESCRIPTION OF THE DRAWINGS Strong

Description Drawings [0084] 1 from the conventional technology of H.264 | AVC coder predictive coding for inter-screen 16 x 16 macro block of the block split block type diagram illustrating.

2

[0085] also to describe the type of motion generated by the movement of the camera video footage from the drawing.

3

[0086] also for the concept of motion model to describe the projected two-dimensional image space.

Super Mat embodiment of the present invention applied to 16 x 16 macro block size greater than 4 shows an example of block and block-split type,

[0087] also drawings.

between screens in video coding according to an embodiment of the present invention describes how predictive coding for

[0088] 5 overall flowchart.

Cited by: KR101379188 B1 ;

Notice of Allowability	Application No. 17/351,649	Applicant(s) Zheng et al.	
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1.☒ This communication is responsive to RCE filed 6/22/2023.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2.☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3.☒ The allowed claim(s) is/are See Continuation Sheet . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4.☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:
a) ☐ All b) ☐ Some* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5.☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6.☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)
1.☒ Notice of References Cited (PTO-892)
2.☐ Information Disclosure Statements (PTO/SB/08),
 Paper No./Mail Date _____.
3.☐ Examiner's Comment Regarding Requirement for Deposit
 of Biological Material _____.
4.☒ Interview Summary (PTO-413),
 Paper No./Mail Date. See Attached.

5. ☒ Examiner's Amendment/Comment
6. ☒ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

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

Continuation of 3. The allowed claim(s) is/are: 20-22,24-27 and 29-39

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Response to Arguments

In response to the amendment filed June 22nd, 2023 before the agreed to Examiner's Amendment:

Applicant amended claims 20, 22, 25, 27, 32, 34, 36, and 38 beyond formalities, Objections, and 112 Rejections.

Applicant added new claims 40 – 43.

Applicant canceled claims 23 and 28.

The pending claims are 20 – 22, 24 – 27, and 29 – 43 [Page 7 lines 1 – 4].

Applicant comments on Examiner's Objections, 112 Rejections, and ODP Rejections [Page 7 lines 5 – 10]. However, no specific arguments were made but it seems the ODP Rejection was to be held in abeyance. The Examiner observes the new dependent claims would address several of the Claim Objections made.

Applicant's arguments filed June 22nd, 2023 [Page 7 lines 11 – 15] have been fully considered but they are not persuasive.

Applicant appears to argue Examiner's interpretation of the PTAB decision in a previous case as limited to the use of the previous "in the case where" claims, but the footnote is to indicate the analysis is not limited to such language as the partitioning claimed could be considered conditional (e.g. "further split").

Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Applicant's arguments do not comply with 37 CFR 1.111(c) because they do not clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. Further, they do not show how the amendments avoid such references or objections.

While the Applicant's points may be understood, the Examiner respectfully disagrees. While the Rejection may be maintained, an Examiner's Amendment is entered to address the outstanding Rejections.

Priority

Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, 365(c), or 386(c) is acknowledged (US Provisional Application 61/222,177 filed July 1st, 2009).

Information Disclosure Statement

The information disclosure statements (IDS) submitted on June 18th, 2021 and October 26th, 2021 were filed before the mailing date of the First Action on the Merits (mailed August 18th, 2022). The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the Examiner.

Due to the excessively lengthy Information Disclosure Statement submitted by applicant, the examiner has given only a cursory review of the listed references. In accordance with MPEP 609.04(a), applicant is encouraged to provide a concise explanation of why the information is being submitted and how it is understood to be relevant. Concise explanations (especially those which point out the relevant pages and lines) are helpful to the Office, particularly where documents are lengthy and complex and applicant is aware of a section that is highly relevant to patentability or where a large number of documents are submitted and applicant is aware that one or more are highly relevant to patentability. Applicant is required to comply with this statement for any non-English language documents. See 37 CFR § 1.56 Duty to Disclose Information Material to Patentability.

Terminal Disclaimer

The terminal disclaimer filed on August 25th, 2023 disclaiming the terminal portion of any patent granted on this application which would extend beyond the expiration date of **US Patent #11,082,697** has been reviewed and is accepted. The terminal disclaimer has been recorded.

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in an interview with the Attorney of Record on August 25th, 2023 (see attached Interview Summary).

The application has been amended as follows:

Note: The Examiner in the interest of brevity only lists the claims amended and not previously presented or original claims.

Regarding claim 20)

Claim 20) (Currently Amended) 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 32x32, 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.

Regarding claim 25)

Claim 25) (Currently Amended) 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 32x32, 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.

Regarding claim 32)

Claim 32) (Currently Amended) 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 32x32, 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.

Regarding claim 36)

Claim 36) (Currently Amended) 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 32x32, 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.

Allowable Subject Matter

Claims 20 – 22, 24 – 27, and 29 – 39 are allowed.

The following is an examiner's statement of reasons for allowance: Claim 20 is taken as the representative claim. Claim 20 in view of the previous PTAB decision recites a negative limitation (the large block is not further partitioned) which while the prior art renders obvious, the prior art does not render obvious signaling the intra prediction mode used for the large block. Even in the case of a 32x32 or 32x16 block, Chen and Moriya do not disclose an intra prediction mode for the whole 32x32 block (e.g. the 32 x 32 Intra DC mode proposed by the Applicant or unifying the 4 16x16 partition modes together) and rely on partitioning the larger block into smaller prediction units with basic / base block sizes for intra prediction. When the limitation is viewed as a whole, the prior art fails to contemplate intra modes as claimed by the Applicant in view of the video codecs being improved.

Claims 25, 32, and 36 cite similar allowable Subject Matter as independent claim 20 and thus are similarly Allowed.

Dependent claims 21 – 22, 24, 26 – 27, 29 – 35, and 37 – 39 depend on allowed independent claims and thus are similarly Allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Po, et al. (US PG PUB 2009/0219991 A1 referred to as "Po" throughout) teaches in Figures 3 – 4 and Paragraphs 8 at least the decisions to merge / split blocks with hierarchy of intra prediction modes / splits.

References previously cited against the claims: Chen, et al. (US PG PUB 2010/0086029 A1 referred to as "Chen" throughout in which citations will come from US Provisional Applications 61/166,631 (referred to as "Chen '631" throughout) and 61/144,357 (referred to as "Chen '631" throughout)) [The US PG PUB is cited in Applicant's June 18th, 2021 IDS as US PG PUB Item #24 – The Examiner will cite from the US Provisional References where "Chen" refers to the collective of "Chen '631" and "Chen '357"]; Moriya, et al. (US PG PUB 2009/0046781 A1 referred to as "Moriya" throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #20]; Moriya, et al. (US PG PUB 2008/0123947 A1 referred to as "Moriya 47" throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #16 and 17].

References cited in updated search including Interference Search: Chen, et al. (US PG PUB 2010/0086030 A1 referred to as "Chen" throughout) teaches in Paragraphs 120 – 125 considerations for intra predicting a large block not further split, but without description of unifying the intra modes for the 16 x 16 blocks into a 32 x 32 mode for the large block (each partitioned intra block is treated separately).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tyler W Sullivan whose telephone number is (571)270-5684. The examiner can normally be reached IFP.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Czekaj can be reached on (571)-272-7327. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To

file and manage patent submissions in Patent Center, visit: <https://patentcenter.uspto.gov>. Visit <https://www.uspto.gov/patents/apply/patent-center> for more information about Patent Center and <https://www.uspto.gov/patents/docx> for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

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

<i>Notice of References Cited</i>	Application/Control No. 17/351,649		Applicant(s)/Patent Under Reexamination Zheng et al.	
	Examiner Tyler W Sullivan		Art Unit 2487	Page 1 of 1

U.S. PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20090219991-A1	09-2009	Po; Lai-Man	H04N19/147	375/E7.126
*	B	US-20100086030-A1	04-2010	Chen; Peisong	H04N19/147	375/E7.243
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS		
*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	



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

NOTICE OF ALLOWANCE AND FEE(S) DUE

15800 7590 09/01/2023
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

EXAMINER

SULLIVAN, TYLER

ART UNIT PAPER NUMBER

2487

DATE MAILED: 09/01/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654

TITLE OF INVENTION: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	12/01/2023

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

15800 7590 09/01/2023
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

	(Typed or printed name)
	(Signature)
	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

17/351,649 06/18/2021 Yunfei Zheng ICVW_2009P00073 US02 2654

TITLE OF INVENTION: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	12/01/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
SULLIVAN, TYLER	2487	375-240020

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____
- 3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required)

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

- ☐ Electronic Payment via Patent Center or EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)
- ☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	09/01/2023	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	
Suite 1700			PAPER NUMBER	
Philadelphia, PA 19103			2487	
DATE MAILED: 09/01/2023				

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. 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 30 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, Virginia 22313-1450. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.** 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.

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 disclosure of these records is required by the Freedom of Information Act.
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 Cooperation Treaty.
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 inspection 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.



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	09/11/2023	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	PAPER NUMBER
Suite 1700			2487	
Philadelphia, PA 19103			MAIL DATE	DELIVERY MODE
			09/11/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

CORRECTED Notice of Allowability	Application No. 17/351,649	Applicant(s) Zheng et al.	
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1.☒ This communication is responsive to IDS filed 8/28/2023.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2.☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3.☒ The allowed claim(s) is/are See Continuation Sheet . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4.☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5.☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6.☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)	5. ☐ Examiner's Amendment/Comment
2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date <u>9/1/2023</u> .	6. ☐ Examiner's Statement of Reasons for Allowance
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____.	7. ☐ Other _____.
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date. _____.	

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

Continuation of 3. The allowed claim(s) is/are: 20-22,24-27 and 29-39

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: Tyler Sullivan

Title: 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 75-84 listed on the attached Form PTO-1449 are enclosed herewith.

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.	English Language Counterpart	Cite No.
KR 200960065401 A	76	PatDocs English Translation of KR20090065401 A	84
WO 2008-083633 A1	77	EP 2081388 A1	75

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: August 28, 2023

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
215-558-5740

/Brian J. Hubbard/

Brian J. Hubbard
Registration No. 45,873

Substitute for 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	17/351,649
				Filing Date	June 18, 2021
				First Named Inventor	Yunfei Zheng
				Art Unit	2487
				Examiner Name	Tyler Sullivan
Sheet	1	of	1	Attorney Docket Number	ICVW_2009P00073 US02

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number	Country Code	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)	T
/TWS/	75	2081388	EP	A1	2009-07-22	HUAWEI TECH CO LTD		
↓	76	20090065401	KR	A	2009-06-22	ICU RES & IND COOP GROUP		x
	77	2008-083633	WO	A1	2008-07-17	HUAWEI TECH CO LTD		

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
/TWS/	78	Vaisey, et al., "Image Compression with Variable Block Size Segmentation", IEEE, Transactions on Signal Processing, Vol 40, No. 8, August 1992, 3 pages	
↓	79	Vaisey, et al., "Variable Block-Size Image Coding", Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, IEEE, 1987, 4 pages	
	80	Samet, et al., "The Quadtree and Related Hierarchical Data Structures", Computer Surveys, Vol 16, No. 2, June 1984, 74 pages	
	81	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, 36 th Meeting: San Diego, USA, October 8-10, 2008, 3 pages	
	82	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	
↓	83	Wen, et al., "Intra Coding using Variable Block Sizes", ITU-Telecommunications Standardization Sector, STUDY GROUP 16 Question 6, Video Coding Experts Group (VCEG), VCEG-O31, 15 th Meeting, Pattaya, Thailand, December 4-6, 2001, 12 pages	
	84	PatDocs English Translation of KR 20090065401 A	

EXAMINER SIGNATURE			
Examiner Signature	/TYLER W. SULLIVAN/	Date Considered	09/07/2023
*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.			

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<i>Applicant-Initiated Interview Summary</i>	Application No. 17/351,649	Applicant(s) Zheng et al.		
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (First Inventor to File) Status No	Page 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	Title	Type
Tyler W Sullivan	Primary Examiner	Telephonic
Cassandra Katz	Attorney Representative	
Brian Hubbard (Reg. No. 45,873)	Attorney	

Date of Interview: 27 September 2023

Issues Discussed:

Other

Attorney's Representative contacted the Examiner to request reconsideration of the June 18th, 2021 IDS filed regarding references that were lined out. Attorney's Representative contended the references were cited in US Application 15/806,893 in the IDS filed 11/8/2017. Examiner verified the references were filed as contended and reconsidered the 6/18/2021 ids in the pending Application (attached). Examiner left a message for Attorney's representative regarding the consideration of the 6/18/2021 IDS in the current Application.

The IDS consideration does not change the Allowable Subject Matter previously agreed to.

/TYLER W. SULLIVAN/ Primary Examiner, Art Unit 2487	
Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04 Please further see: MPEP 713.04 Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b) 37 CFR § 1.2 Business to be transacted in writing	

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	10/02/2023	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	PAPER NUMBER
Suite 1700			2487	
Philadelphia, PA 19103			MAIL DATE	DELIVERY MODE
			10/02/2023	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

CORRECTED Notice of Allowability	Application No. 17/351,649	Applicant(s) Zheng et al.	
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1.☒ This communication is responsive to IDS Filed 6/18/2021.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2.☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3.☒ The allowed claim(s) is/are See Continuation Sheet . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4.☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5.☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6.☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)	5. ☐ Examiner's Amendment/Comment
2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date <u>6/18/2021</u> .	6. ☐ Examiner's Statement of Reasons for Allowance
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____.	7. ☐ Other _____.
4. ☒ Interview Summary (PTO-413), Paper No./Mail Date. <u>See Attached</u> .	

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

Continuation of 3. The allowed claim(s) is/are: 20-22,24-27 and 29-39

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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†	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/TWS/	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.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/	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		

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

<div>Substitute for 1449/PTO</div> <div>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</div> <div>(use as many sheets as necessary)</div>				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
/TWS/ 	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	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

Substitute for 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				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
/TWS/ 	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	/TYLER W. SULLIVAN/	Date Considered	07/18/2022
*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.			

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/



UNITED STATES
PATENT AND TRADEMARK OFFICE

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #
17/351,649

RECEIPT DATE / TIME
12/01/2023 03:38:34 PM Z ET

ATTORNEY DOCKET #
ICVW_2009P00073 US02

Title of Invention

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

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2654	FILED BY	Melissa Inzaris
PATENT CENTER #	63460253	FILING DATE	06/18/2021
CUSTOMER #	15800	FIRST NAMED INVENTOR	Yunfei Zheng
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Brian Hubbard

Documents

TOTAL DOCUMENTS: 8

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
ICVW_2009P00073US02_R CE.pdf	3	Request for Continued Examination (RCE)	467 KB
Warning: This is not a USPTO supplied RCE fillable form. Data in the form cannot be automatically loaded to other USPTO systems.			
ICVW_2009P00073US02_SI DS-1449.pdf	1	Information Disclosure Statement (IDS) Form (SB08)	137 KB
Warning: This is not a USPTO supplied IDS fillable form. Data in the form cannot be automatically loaded to other USPTO systems.			
ICVW_2009P00073US02_SI DS_Transmittal.pdf	2	Transmittal Letter	83 KB

89-US__PROV__61144357.pdf	10	Non Patent Literature	311 KB
87- Faundez_Zanuy_Contribution stoInterframe__October1994__ 6pp.pdf	6	Non Patent Literature	427 KB
86-JP__2000__308060.pdf	34	Foreign Reference	4227 KB
88- McCann_SamsungsRespons e__April2010__42pp.pdf	42	Non Patent Literature	4245 KB
90-US__PROV__61166631.pdf	97	Non Patent Literature	3853 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
ICVW_2009P00073US02_RCE. pdf	711B36B7DA7640DDAA441A6D9BF3742CADFF0A169A3A4160 C1D3D43F39C9DD854DDEF4875A3E71C0FB05DE575F6ECCF BA239BE9B80F018F7E82E6903F1EFD648
ICVW_2009P00073US02_SIDS -1449.pdf	3B44724C179A12129DF753E53741A7E8ECDBDCDE86399455F 2B461F338AE99E2DC008A16989AC6CF3249BA93B495594DBA DDB4D2CC3C21CB74A0013623D515D1
ICVW_2009P00073US02_SIDS __Transmittal.pdf	BE4987E31577DE8744DCB13D060A0C36D1DB08A538E6A4EE 5177F2514908CCBA38473C6D6673006BEEB2B432105A5DE67 BA0B14D4E5531253F7C0F41CD7499A4
89-US__PROV__61144357.pdf	6D6D225977AC3DBCCE0005862C77C602CEF6F93DA852E7948 2A0469F993CD17888C4106D84551AAF2831E93D2A3D418C9F9 7AF24C1FC6A19B3717BD63BECD747
87- Faundez_Zanuy_Contributionst oInterframe__October1994__6pp.	E5F54C3D95064CD99D51D27DA3E5711389AB69D4CB5B896A1 C1C7CD53D54C4FFC8F1FA9D99CC00D5CB84043DF178B8CD 797E4D90890B57DDCE0F591A86BD6561

pdf

86-JP__2000__308060.pdf	234229EFB2A9EB04392D76E3B49E8FA766DF3509F2018288C2989A24E285C50734AAE323D09F10F952C1A7CFACD19E838D6A4FC6C7372E43DBA984D500E1F777
88-McCann__SamsungsResponse__April2010__42pp.pdf	3AAD01E9C44927BA06CDFFBB1AC228C4D3B964138846D8769B99261BC63EAA45CAE10E3D4326D8B12EFA37F01EBA2CF48A4918A9C4D54F86FEFF49DC3CB4AE69
90-US__PROV__61166631.pdf	5AC8ABCF4748D09FFF636C4B32ED1BD863E64279409787A3275B5034AAC1B74D56D1E8984E838BC633741BD204AE1110707CC6DB74C4BB4E1B1FAD0D2E80B143

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 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.

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL (Submitted Only via EFS-Web)							
Application Number	17351649	Filing Date	2021-06-18	Docket Number (if applicable)	ICVW_2009P00073 US02	Art Unit	2487
First Named Inventor	Yunfei Zheng			Examiner Name	TYLER SULLIVAN		
This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application. Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV							
SUBMISSION REQUIRED UNDER 37 CFR 1.114							
Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).							
<div>☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.<div>☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on <div></div>☐ Other <div></div></div></div> <div>☒ Enclosed<div>☐ Amendment/Reply☒ Information Disclosure Statement (IDS)☐ Affidavit(s)/ Declaration(s)☐ Other <div></div></div></div>							
MISCELLANEOUS							
<div>☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months <div></div><div>(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)</div>☐ Other <div></div></div>							
FEES							
<div>☒ The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.<div>The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 505519</div></div>							
SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED							
<div>☒ Patent Practitioner Signature☐ Applicant Signature</div>							

Signature of Registered U.S. Patent Practitioner			
Signature	/Brian J. Hubbard/	Date (YYYY-MM-DD)	2023-11-30
Name	Brian J. Hubbard	Registration Number	45873

This collection of information is required by 37 CFR 1.114. 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 12 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

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 Cooperation Treaty.
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.

Substitute for 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	17/351,649
				Filing Date	June 18, 2021
				First Named Inventor	Yunfei Zheng
				Art Unit	2487
				Examiner Name	TYLER SULLIVAN
Sheet	1	of	1	Attorney Docket Number	ICVW_2009P00073 US02

U.S. PATENTS						
Examiner Initial*	Cite No	Publication Number	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)
	85	6978048	B1	2005-12-20	Higginbottom et al.	

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number	Country Code	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)	T
	86	2000-308060	JP	A	2000-11-02	Canon KK		

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	
	87	FAUNDEZ-ZANUY et al., "Contributions to Interframe Coding", Workshop on image analysis & synthesis in image coding, Berlin, October 1994, 6 pages		
	88	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, April 15-23, 2010, pp. 1-42		
	89	CHEN, et al., "US Provisional Patent Application No. 61/144,357", January 13, 2009, 10 pages		
	90	CHEN, et al., "US Provisional Patent Application No. 61/166,631", April 3, 2009, 97 pages		

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.			



Espacenet

Bibliographic data: JP2000308060 (A) — 2000-11-02

METHOD AND DEVICE FOR ENCODING

Inventor(s): PAUL RAYMOND HIGGINBOTTOM; MAJANI ERIC; ANDREW JAMES PHILIP ± (PAUL RAYMOND HIGGINBOTTOM, ; MAJANI ERIC, ; ANDREW JAMES PHILIP)

Applicant(s): CANON KK ± (CANON INC)

Classification: - international: G06T9/00; H03M7/30; H04N1/41; H04N19/63; H04N19/635; H04N19/645; H04N7/26; H04N7/30;
(IPC1-7): H03M7/30; H04N1/41; H04N7/30
- cooperative: H04N19/63 (EP, US); H04N19/635 (EP, US); H04N19/645 (EP, US); H04N19/42 (EP, US)

Application number: JP20000069353 20000313

Priority number(s): AU1999PP09186 19990312

Also published as: EP1035511 (A2) EP1035511 (A3) EP1035511 (B1) JP4532653 (B2) US6978048 (B1)

Abstract of JP2000308060 (A)

PROBLEM TO BE SOLVED: To execute DWT(discrete wavelet transformation) at a plurality of levels with a small buffer capacity. SOLUTION: In a method and device for encoding digital images into a prescribed resolution level one block by one block by DWT, images are divided into tiles and subjected to the DWT at every tile. Then HH, HL, and LH subband factors are encoded to a bit stream 402 and LL subband factors as subjected to the DWT after a prescribed number of factors is stored. Until the corresponding LL subband factor is encoded to the bit stream 402, processing is recursively executed at every DWT level.

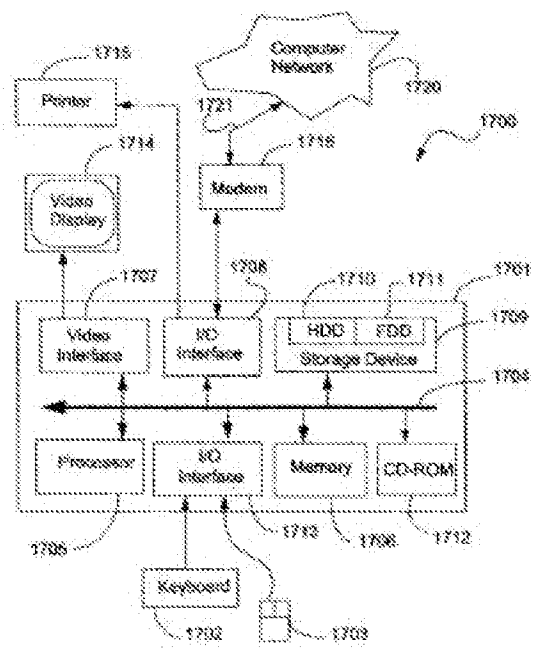


FIG. 17

(51)Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
H 0 4 N 7/30		H 0 4 N 7/133	Z
H 0 3 M 7/30		H 0 3 M 7/30	A
H 0 4 N 1/41		H 0 4 N 1/41	Z

審査請求 未請求 請求項の数48 O L (全 32 頁)

(21)出願番号	特願2000-69353(P2000-69353)	(71)出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	平成12年3月13日(2000.3.13)	(72)発明者	ポール レイモンド ヒギンボトム オーストラリア国 2113 ニュー サウス ウェールズ州、ノース ライド、トーマ ス ホルト ドライブ 1 キヤノン イ ンフォメーション システムズ リサーチ オーストラリア プロプライエタリー リミテツド内
(31)優先権主張番号	P P 9 1 8 6	(74)代理人	100076428 弁理士 大塚 康徳 (外2名)
(32)優先日	平成11年3月12日(1999.3.12)		
(33)優先権主張国	オーストラリア (AU)		

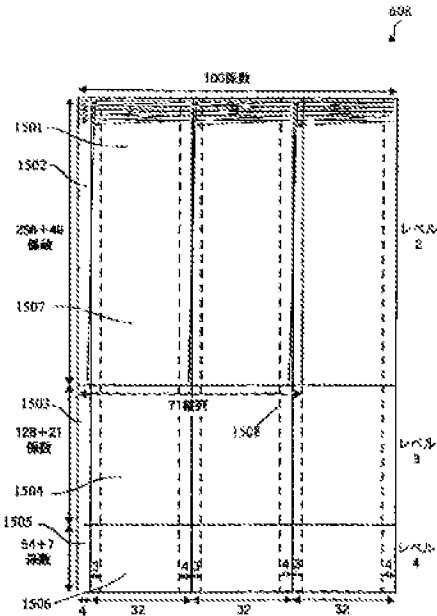
最終頁に続く

(54)【発明の名称】 符号化方法及び装置

(57)【要約】

【課題】 少ないバッファ容量で、複数レベルのDWTを実行する。

【解決手段】 DWTによってデジタル画像(502)を、ブロック単位で所定の分解レベルに符号化する符号化方法及び装置であって、画像はタイルに分割され、タイル毎にDWTされる。次に、HH、HLおよびLHサブバンド係数はビットストリーム(402)に符号化され、LLサブバンド係数は、所定数蓄積されDWTされる。対応するLLサブバンド係数がビットストリーム(402)に符号化されるまで、処理がDWTレベル毎に回帰的に実行される。



【特許請求の範囲】

【請求項1】 離散ウェーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する方法であって、

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように、画像を複数のタイルに分割する工程と、
b) 現タイルを選択する工程と、
c) DWTを使用して前記現タイルを分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と、
d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する工程と、
e) LLサブバンドの係数を蓄積するとともに、LLサブバンドの所定数の係数が蓄積されるまで、前記工程b)からe)を繰り返す工程と、
f) 現タイルに、蓄積されたLLサブバンドの係数の前記所定数を割り当てる工程と、
g) 分解の所定レベルに達するまで、前記工程c)からg)を繰り返す工程と、
h) LLサブバンドをビットストリームに符号化する工程と、
を含むことを特徴とするデジタル画像を符号化する符号化方法。

【請求項2】 各サブバンドの係数の蓄積は、異なるタイルの対応するサブバンド内の係数の蓄積を含むことを特徴とする請求項1に記載の符号化方法。

【請求項3】 前記符号化は、各ブロックが蓄積されるように各ブロックを実質的に符号化することを特徴とする請求項1に記載の符号化方法。

【請求項4】 前記特定のブロックサイズは、前記一次元内のH係数に前記二次元内のW係数を掛けたものであることを特徴とする請求項1に記載の符号化方法。

【請求項5】 前記一次元内の係数の数は、前記二次元の係数の数に等しい($W=H$)ことを特徴とする請求項4に記載の符号化方法。

【請求項6】 前記所定レベルがJであり、各タイルの一次元内の画素の最小数は、OをDWTフィルタによって必要とされるオーバーラップとしたとき、 $2^J \cdot H + O$ ($2^J - 1$)によって規定されることを特徴とする請求項4に記載の符号化方法。

【請求項7】 各タイルの二次元内の画素数が、一次元内の画素数よりも少なく、 $2 \cdot W + O$ で規定されることを

特徴とする請求項6に記載の符号化方法。

【請求項8】 前記DWTフィルタが、Daubechies 9/7フィルタであって、必要とするオーバーラップが7画素或は係数($O=7$)であることを特徴とする請求項1に記載の符号化方法。

【請求項9】 前記DWTフィルタが、Haarフィルタであって、必要とするオーバーラップは画素がない、或は係数($O=0$)であることを特徴とする請求項1に記載の符号化方法。

【請求項10】 離散ウェーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する方法であって、

a) 各タイルがDWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素のほぼ最小数を有し、また、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素のほぼ最小数より少なくなるように画像を複数のタイルに分割する工程と、
b) 現タイルとして画像のタイルを選択する工程と、
c) DWTを使用して前記現タイルをLL、LH、HLおよびHHサブバンドの複数の係数を提供するように分解する工程と、
d) 現レベルのLH、HL、HHサブバンドの係数をビットストリームに符号化する工程と、
e) 前記現レベルが所定レベルであるかどうかを判定し、
e a) 前記現レベルが所定レベルであれば、LLサブバンドの係数をビットストリームに符号化し、前記工程b)からe)を繰り返す、
e b) 前記現レベルが所定レベルでなければ、LLサブバンドの前回符号化されていない係数を記憶する工程と、
f) 現レベルにおけるLLサブバンドの前記記憶された係数の数が少なくとも所定数であるかどうかを判定する工程と、
f a) 前記記憶されたLL係数の数が少なくとも前記所定数であれば、現タイルとしてLL係数の前記所定数を現タイルに割り当て、前記工程c)からf)を繰り返す工程と、
f b) 記憶されたLL係数の数が前記所定数より少なければ、前記工程b)からf)を繰り返す工程と、
を含むことを特徴とするデジタル画像を符号化する符号化方法。

【請求項11】 前記特定のブロックサイズは、前記一次元内のH係数に前記二次元内のW係数を掛けたものであることを特徴とする請求項10に記載の符号化方法。

【請求項12】 前記一次元内の係数の数が、前記二次元の係数の数に等しい($W=H$)ことを特徴とする請求

項11に記載の符号化方法。

【請求項13】 前記所定レベルがJであり、各タイルの二次元内の画素の最小数が、OをDWTフィルタによって必要とされるオーバーラップとしたとき、 $2^J \cdot H + O$ ($2^J - 1$) によって規定されることを特徴とする請求項11に記載の符号化方法。

【請求項14】 各タイルの二次元内の画素数が、二次元内の画素数よりも少なく、 $2W + O$ で規定されることを特徴とする請求項13に記載の符号化方法。

【請求項15】 前記DWTフィルタは、Daubechies 9/7フィルタであって、必要とするオーバーラップが7画素或は係数 ($O=7$) であることを特徴とする請求項10に記載の符号化方法。

【請求項16】 前記DWTフィルタはhaarフィルタであって、必要とするオーバーラップは画素がない、或は係数 ($O=0$) であることを特徴とする請求項10に記載の符号化方法。

【請求項17】 前記符号化は、ビットプレーンエントロピー符号化器によって実行されることを特徴とする請求項10に記載の符号化方法。

【請求項18】 前記符号化は、演算符号化器によって実行されることを特徴とする請求項10に記載の符号化方法。

【請求項19】 前記符号化は、演算およびビットプレーンエントロピー符号化器を含むハイブリッド・符号化器によって実行されることを特徴とする請求項10に記載の符号化方法。

【請求項20】 離散ウェーブレット変換(DWT) により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように、画像を複数のタイルに分割する分割手段と、

b) 現タイルを選択する選択手段と、

c) DWTを使用して前記現タイルを分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解手段と、

d) LLサブバンド係数の所定数を蓄積する手段と、

e) 蓄積されたLLサブバンド係数の前記所定数を現タイルとして割り当てる手段と、

f) 前記現タイルをさらなるレベルの分解度に分解するために現タイルを前記分解手段に帰還させる手段と、

g) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定のブロックサイズを形成する

とともに、前記所定レベルでLLサブバンド内の前記特定のブロックサイズの少なくとも一つを蓄積する手段と、

h) 各前記ブロックをビットストリームに符号化する符号化手段と、

を有することを特徴とする符号化装置。

【請求項21】 離散ウェーブレット変換(DWT) により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化装置であって、

各画像の部分がDWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素のほぼ最小数を有し、また、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少なくなるように前記画像の少なくとも一部を記憶する手段と、

リニア変換を前記画像の部分の一次元および二次元それぞれに適用して、各サブパスバンドが少なくとも一つの係数を含むLL、LH、HLおよびHHサブバンドを備えている第1および第2フィルター手段と、

LLサブバンドの係数を所定数蓄積し、第1および第2フィルター手段によってリフィルタして次のレベル分解を達成するための画像部分として蓄積されたLL係数を使用する部分バンド記憶手段と、

LL、HLおよびHHサブバンド内の各レベルに対する特定サイズの前記ブロックを蓄積し、また前記所定レベルの分解レベルにおけるLLサブバンド内の特定サイズの前記ブロックを蓄積するためのサブバンド記憶手段と、

各蓄積したブロックをビットストリームに符号化する符号化手段と、

を具備することを特徴とする符号化装置。

【請求項22】 前記符号化手段は、ビットプレーンエントロピー符号化器を含むことを特徴とする請求項20に記載の符号化装置。

【請求項23】 前記符号化手段は、ビットプレーンエントロピー符号化器を含むことを特徴とする請求項21に記載の符号化装置。

【請求項24】 前記符号化手段は、演算符号化器を含むことを特徴とする請求項20に記載の符号化装置。

【請求項25】 前記符号化手段は、演算符号化器を含むことを特徴とする請求項21に記載の符号化装置。

【請求項26】 前記符号化手段は、演算およびビットプレーンエントロピー符号化器を含むハイブリッド・符号化器であることを特徴とする請求項20に記載の符号化装置。

【請求項27】 前記符号化手段は、演算およびビットプレーンエントロピー符号化器を含むハイブリッド・符号化器であることを特徴とする請求項21に記載の符号

化装置。

【請求項28】 離散ウェーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読取可能な記憶媒体において、前記プログラムが、

- a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように、画像を複数のタイルに分割する工程のためのコードと、
 - b) 現タイルを選択する工程のためのコードと、
 - c) DWTを使用して前記現タイルを分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解工程のためのコードと、
 - d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する工程のためのコードと、
 - e) LLサブバンドの係数を蓄積するとともに、サブバンドの所定係数が蓄積されるまで、工程b)からe)を繰り返す工程のためのコードと、
 - f) 現タイルに、蓄積されたLLサブバンドの係数の前記所定数を割り当てる割り当て工程のためのコードと、
 - g) 前記分解の所定レベルに達するまで、工程c)からg)を繰り返す繰り返す工程のためのコードと、
 - h) LLサブバンドをビットストリームに符号化する符号化工程のためのコードと、
- を含むことを特徴とする記憶媒体。

【請求項29】 各サブバンド内の係数を蓄積するためのコードが、異なるタイルの対応するサブバンド内の係数を蓄積するための工程を含むことを特徴とする請求項28に記載の記憶媒体。

【請求項30】 符号化のための前記コードが、各ブロックが略蓄積されるように各ブロックを符号化するための工程を含むように適用されていることを特徴とする請求項28に記載の記憶媒体。

【請求項31】 ブロックのサイズを特定するための前記コードが、前記一次元内のH係数に前記二次元内のW係数を掛けることにより特定するための工程を含むように適用されていることを特徴とする請求項28に記載の記憶媒体。

【請求項32】 前記コードが、一次元内の係数が二次元の係数に等しく($W=H$)なるように適用されていることを特徴とする請求項31に記載の記憶媒体。

【請求項33】 前記所定レベルがJであり、各タイルの一次元内の画素の最小数が、OをDWTフィルタによって必要とされるオーバーラップとしたとき、 $2^J \cdot H + O$ ($2^J - 1$)によって規定されるように前記コードが適用されていることを特徴とする請求項31に記載の記憶媒体。

【請求項34】 各タイルの二次元内の画素数が、一次元内の画素数よりも少なく、 $2W + O$ で規定されるように前記コードが適用されていることを特徴とする請求項33に記載の記憶媒体。

【請求項35】 前記DWTフィルタがDaubechies 9/7フィルタであって、必要とするオーバーラップが7画素或いは7係数($O=7$)であるように前記コードが適用されていることを特徴とする請求項28に記載の記憶媒体。

【請求項36】 前記DWTフィルタが、Haarフィルタであって、必要とするオーバーラップがない、或いは係数($O=0$)がないように前記コードが適用されていることを特徴とする請求項28に記載の記憶媒体。

【請求項37】 離散ウェーブレット変換(DWT)により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータにより読取可能な記憶媒体において、前記プログラムが、

- a) 各タイルが、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、またDWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように、画像を複数のタイルに分割する工程のためのコードと、
- b) 現タイルとして画像のタイルを選択する選択工程のためのコードと、
- c) DWTを使用して前記現タイルをLL、LH、HLおよびHHサブバンドの複数の係数を提供するように分解する工程のためのコードと、
- d) 現在レベルのLH、HL、HHサブバンドの係数をビットストリームに符号化する符号化工程のためのコードと、
- e) 前記現在レベルが所定レベルであるかどうかを判定する判定工程のためのコードと、
- e a) 前記現在レベルが所定レベルであれば、LLサブバンドの係数をビットストリームに符号化する工程と；工程b)からe)を繰り返すためのコードと；
- e b) 前記現在レベルが所定レベルでなければ、LLサブバンドの前回符号化されていない係数を記憶する記憶工程のためのコードと；
- f) 現在レベルにおけるLLサブバンドの前記記憶され

た係数が、少なくとも所定数であるかどうかを判定する工程のためのコードと；

f a) 記憶されたLL係数の数が少なくとも前記所定数であれば、現タイルとしてLL係数の前記所定数を割り当てる割当工程と；工程c) からf) を繰り返す工程のためのコードと；

f b) 記憶されたLL係数の数が前記所定数より小さければ、工程b) からf) を繰り返す繰返工程のためのコードと；を含むことを特徴とする記憶媒体。

【請求項38】 特定するブロックのサイズが、前記一次元内のH係数に前記二次元内のW係数を掛けたものであることを特徴とする請求項37に記載の記憶媒体。

【請求項39】 一次元内の係数が、二次元の係数に等しく ($W=H$) なるように前記コードが適用されていることを特徴とする請求項38に記載の記憶媒体。

【請求項40】 前記所定レベルがJであり、各タイルの一次元内の画素の最小数が、OをDWTフィルタによって必要とされるオーバーラップとしたとき、 $2^J H + O$ ($2^J - 1$) によって規定されるように、前記コードが適用されていることを特徴とする請求項38に記載の記憶媒体。

【請求項41】 各タイルの二次元内の画素数が、一次元内の画素数よりも少なく、 $2W + O$ で規定されるように前記コードが適用されていることを特徴とする請求項40に記載の記憶媒体。

【請求項42】 前記DWTフィルタが、Daubechies 9/7 フィルタであって、必要とするオーバーラップが7画素或いは7係数 ($O=7$) であるように前記コードが適用されていることを特徴とする請求項37に記載の記憶媒体。

【請求項43】 前記DWTフィルタが、Haarフィルタであって、必要とするオーバーラップがない、或いは係数がない ($O=0$) ように、前記コードが適用されていることを特徴とする請求項37に記載の記憶媒体。

【請求項44】 符号化のための前記工程が、ビットプレーンエントロピー符号化・プロセスを含むことを特徴とする請求項37に記載の記憶媒体。

【請求項45】 符号化のための前記工程が、演算符号化器によって実行されることを特徴とする請求項37に記載の記憶媒体。

【請求項46】 符号化のための前記工程が、演算およびビットプレーンエントロピー符号化・プロセスを含むハイブリッド符号化プロセスによって実行されることを特徴とする請求項37に記載の記憶媒体。

【請求項47】 離散ウェーブレット変換(DWT) により変換して所定レベルに分解でき、かつ各ブロックが一次元および二次元の複数の係数を含む特定のブロックサイズを有しブロック単位で符号化される複数の画素を含むデジタル画像を符号化する符号化方法であって、

a) 各タイルが、DWT分解の前記所定レベルで前記ブ

ロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、またDWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように画像を複数のタイルに分割する工程と、

b) 現タイルを選択する工程と、

c) DWTを使用して前記現タイルを分解の少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と、

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する工程と、

e) LLサブバンドの係数を蓄積するとともに、サブバンドの所定数の係数が蓄積されるまで、前記工程b) からe) を繰り返す工程と；

f) 現タイルに、蓄積されたLLサブバンドの前記所定数を割り当てる工程と；

g) 分解の所定レベルに達するまで、前記工程c) からg) を繰り返す工程と；

h) LLサブバンドをビットストリームに符号化する工程と；を含むことを特徴とする符号化方法。

【請求項48】

a) 画像をタイルに分割する分割ステップと、

b) 各タイルを離散ウェーブレット変換し、複数のサブバンドLL、LH、HL、HHを生成する工程と、

c) 注目タイルを選択する工程と、

d) DWTを使用して前記注目タイルを分割し、前記複数のサブバンドLL、LH、HL、HHを含む複数のサブバンドを形成する工程と、

e) サブバンドLL、LH、HL、HHを符号化する工程と、

f) LLサブバンドの係数を蓄積するとともに、LLサブバンドの所定数の係数が蓄積されるまで、前記工程c) から工程e) を繰返す工程と、

g) 前記蓄積された所定数のLLサブバンドを注目タイルに割り当てる工程と、

h) 所定分解レベルに達するまで、前記工程d) から工程g) を繰返す工程と、

i) LLサブバンドを符号化する工程と、を備えることを特徴とする符号化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタル画像圧縮分野、特に画像のウェーブレット分解を実行する符号化方法および装置に関する。

【0002】

【従来の技術】デジタル・データ圧縮、および特にデジタル画像圧縮の分野は、一時非常に注目されていたが、特に最近、離散ウェーブレット変換(DWT) に基づく圧縮

技術がますます一般的になってきている。これはDWTが画像の非冗長階層的分解を提供し、また、画像の圧縮結果が良好な定歪み統計値を提供するからである。

【0003】一般的に、画像の離散ウェーブレット変換(DWT)は、一連の一次元DWTを用いて実行される。信号(すなわち、画像の行)の一次元DWTが、ロウパスおよびハイパスフィルタリングされ、各フィルタされた信号を2つおきを選んで実行される。この2つおきに取り出すのは、フィルタリング処理されたサンプルを2つおきに計算して保持することを意味する。たまたみこみ(フィルタリング)が実行されると、フィルタは、普通は一つのサンプルであるのとは代わって、一度に二つのサンプル毎に進行する。この方法において、N個のサンプルの信号に対して、N個のDWTのサンプル、すなわち、実質的に $N/2$ のロウパス・サンプルと実質的に $N/2$ のハイパス・サンプルがある。

【0004】

【発明が解決しようとする課題】これはランダム・アクセス・メモリ(RAM)のようなコンピュータ・メモリ内で全画像をバッファリングすることによって実行されるのが一般的で、バッファされた画像にDWTが実行され、その変換された画像が符号化される。あいにく、このアプローチには大量のメモリ、特に2000画素/2000画素或いはそれ以上を必要とし、十分な処理速度を達成しなければならない場合には、相当なバンド幅を必要とする。

【0005】本発明は上記従来例に鑑みてなされたもので、本発明の目的は、既存の構成の一つのまたはそれ以上の欠点を実質上克服するか、ないしは少なくともこれを改善することにある。例えば、少ないバッファの容量で、複数レベルのDWT実行できる構成を提供する。

【0006】

【課題を解決するための手段】本発明の第1の態様によれば、離散ウェーブレット変換(DWT)により変換して所定レベルに分解され、各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化できる複数の画素を含むデジタル画像を符号化する方法を提供することにあり、この方法は、
a) 画像を複数のタイルに分割する工程であって、各タイルがまず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように分割する工程と；
b) 現在のタイルを選択する工程と；
c) DWTを使用して前記現在のタイル分割し、少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と；

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する工程と；

e) LLサブバンドの係数を蓄積するとともに、サブバンドの係数が所定数蓄積されるまで、工程b)からe)を繰り返す工程と；

f) 現在のタイルに、前記蓄積された所定数のLLサブバンドの係数を割り当てる工程と；

g) 分解の所定レベルに達するまで、工程c)からg)を繰り返す工程と；

h) LLサブバンドをビットストリームに符号化する工程と；を含んでいる。

【0007】本発明の第2の態様によれば、複数の画素を含むデジタル画像を符号化する方法であって、前記画像は離散ウェーブレット変換(DWT)によって所定レベルに分解でき、かつ、各ブロックが一次元および二次元において複数の係数を有し、特定ブロック・サイズを有するブロック毎のベースに符号化される複数の画素を含むデジタル画像を符号化する方法を提供することにあり、前記方法は、

a) 画像を複数のタイルに分割する工程であって、各タイルがまず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における複数の係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの一次元における複数の係数を発生するのに必要とされる画素の略最小数より少ない数を有するように分割する工程と；
b) 現在のタイルを選択する工程と；

c) DWTを使用して前記現在のタイルを分解し、LL、LH、HLおよびHHサブバンドの複数の係数を得る工程と；

d) 現在レベルのLH、HL、HHサブバンドの係数をビットストリームに符号化する工程と；

e) 前記現在レベルが所定レベルであるかどうかを判定し；

e a) 前記現在レベルが所定レベルであれば、LLサブバンドの係数をビットストリームに符号化し；工程b)からe)を繰り返す；

e b) 前記現在レベルが所定レベルでなければ、LLサブバンドの前記符号化されていない係数を記憶する工程と；

f) 現在レベルにおけるLLサブバンドの前記記憶された係数が、少なくとも所定数であるかどうかを判定する工程と；

f a) 記憶されたLL係数が少なくとも前記所定数であれば、前記所定数のLL係数を現在タイルとして割り当てる工程と；工程c)からf)を繰り返す工程と；

f b) 記憶されたLL係数が前記所定数より少ないければ、工程b)からf)を繰り返す工程と；を含んでいる。

る。

【0008】本発明の第3の態様によれば、画像が離散ウェーブレット変換(DWT)により変換されて所定レベルに分解でき、かつ各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化できる画像を符号化する装置を提供することであって、この装置が、

a) 画像を複数のタイルに分割する手段であって、各タイルが、まず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように分割する手段と；

b) 現在のタイルを選択する選択手段と；

c) DWTを使用して、前記現在のタイルを少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解手段と；

d) LLサブバンド係数の係数を所定数蓄積する手段と；

e) 蓄積されたLLサブバンド係数の前記所定数を、現在タイルとして割り当てる手段と；

f) 前記現在のタイルをさらなるレベルの分解度に分解するために現在タイルを分解手段に帰還させる手段と；

g) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記所定レベルでLLサブバンド内の前記特定サイズのブロックの少なくとも一つを蓄積する手段と；

h) 各前記ブロックをビットストリームに符号化するエンコーディング手段と；を含む画像を符号化する符号化手段と；を具備する。

【0009】本発明の第4の態様によれば、離散ウェーブレット変換(DWT)により変換して所定レベルに分解でき、かつ、各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化できる画像を符号化する装置を提供することであって、この装置が、前記画像の少なくとも一部を記憶する手段であって、各画像の一部が、まず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように記憶する手段と；前記画像部分の一次元および二次元それぞれにリニア変換を適用して、各サブバンドが少なくとも一つの係数を含むLL、LH、HLおよびHHサブバンドを提供する第1および第2フィルタ手段と；LLサブバンドの所定数の係数を蓄積し、蓄積されたLL係数を、前

記第1および第2フィルタ手段によってリフィルタリングするための画像部分として使用し、次の分解レベルを達成する部分バンド記憶手段と；LL、HLおよびHHサブバンド内の各レベルに対する特定サイズの前記ブロックを蓄積し、また前記所定レベルの分解におけるLLサブバンド内の特定サイズの前記ブロックを蓄積するサブバンド記憶手段と；各蓄積ブロックをビットストリームに符号化する符号化手段と；を具備する。

【0010】本発明の第5の態様によれば、離散ウェーブレット変換(DWT)により所定レベルに分解でき、かつ、各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化できる画像を符号化する装置のためのプログラムを記憶するコンピュータ読取可能記憶媒体を提供することであって、前記プログラムが、

a) 画像を複数のタイルに分割する工程のためのコードであって、各タイルが、まず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの二次元における係数を発生するのに必要とされる画素の略最小数より少ない数を有するように分割する工程のためのコードと；

b) 現在のタイルを選択する工程のためのコードと；

c) DWTを使用して前記現在のタイルを少なくとも一つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する分解工程のためのコードと；

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する蓄積工程のためのコードと；

e) LLサブバンドの係数を蓄積するとともに、サブバンドの所定数の係数が蓄積されるまで、工程b)からe)を繰り返す蓄積工程のためのコードと；

f) 現在のタイルに、蓄積されたLLサブバンドの係数の前記所定数として割り当てる割り当て工程のためのコードと；

g) 分解の所定レベルに達するまで、工程c)からg)を繰り返す繰り返し工程のためのコードと；

h) LLサブバンドをビットストリームに符号化するエンコーディング工程のためのコードと；を含んでいる。

【0011】本発明の第6の態様によれば、離散ウェーブレット変換(DWT)によって所定レベルに分解でき、かつ、各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化される複数の画素を含むデジタル画像を符号化する装置のためのプログラムを記憶するコンピュータ読取可能記憶媒体を提供することであって、前記プログラムが、

a) 画像を複数のタイルに分割する分割工程であって、各タイルが、まず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数より少ない数となるように分割する分割工程のためのコードと；

b) 現在のタイルとして画像のタイルを選択する選択工程のためのコードと；

c) DWTを使用して前記現在のタイルをLL、LH、HLおよびHHサブバンドの複数の係数を提供するように分解する分解工程のためのコードと；

d) 現在レベルのLH、HL、HHサブバンドの係数をビットストリームに符号化する符号化工程のためのコードと；

e) 前記現在レベルが所定レベルであるかどうかを判定する判定工程のためのコードと；

e a) 現在レベルが所定レベルであれば、LLサブバンドの係数をビットストリームに符号化する符号化工程と；工程b) からe) を繰り返すためのコードと；

e b) 前記現在レベルが所定レベルでなければ、LLサブバンドの前記符号化されていない係数を記憶する記憶工程のためのコードと；

f) 現在レベルにおけるLLサブバンドの記憶された係数が、少なくとも所定数であるかどうかを判定する判定工程のためのコードと；

f a) 記憶されたLL係数が少なくとも所定数であれば、現在タイルとしてLL係数の所定数を割り当てる割り当て工程と；工程c) からf) を繰り返す繰り返しの工程のためのコードと；

f b) 記憶されたLL係数が所定数より少なければ、工程b) からf) を繰り返す繰り返しの工程のためのコードと；を含んでいる。

【0012】本発明の第7の態様によれば、離散ウェーブレット変換(DWT)によって少なくとも二つのレベルである所定レベルに分解でき、かつ、さらに各ブロックが一次元および二次元における係数による特定ブロック・サイズを有するブロック単位で符号化される複数の画素を含むデジタル画像を符号化する方法を提供することであって、この方法が、

a) 画像を複数のタイルに分割する工程であって、各タイルが、まず第1に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数を有し、また第2に、DWT分解の前記所定レベルで前記ブロックの一次元における係数を発生するのに必要とされる画素の略最小数より少ない数となるように分割する工程と；

b) 現在のタイルを選択する工程と；

c) DWTを使用して前記現在のタイルを少なくとも一

つのレベルに分解してLL、LH、HLおよびHHサブバンドを含む複数のサブバンドを形成する工程と；

d) LH、HLおよびHHサブバンドの各サブバンド内の係数を蓄積して前記特定サイズのブロックを形成するとともに、前記各ブロックをビットストリームに符号化する工程と；

e) LLサブバンドの係数を蓄積するとともに、所定数のサブバンドの係数が蓄積されるまで、工程b) から

e) を繰り返す工程と；

f) 現在のタイルに、蓄積された前記所定数のLLサブバンドの係数を割り当てる工程と；

g) 分解の所定レベルに達するまで、工程c) からg) を繰り返す工程と；

h) LLサブバンドをビットストリームに符号化する工程と；を含んでいる。また本発明の符号化方法は、

a) 画像をタイルに分割する分割ステップと、

b) 各タイルを離散ウェーブレット変換し、複数のサブバンドLL、LH、HL、HHを生成する工程と、

c) 注目タイルを選択する工程と、

d) DWTを使用して前記注目タイルを分割し、前記複数のサブバンドLL、LH、HL、HHを含む複数のサブバンドを形成する工程と、

e) サブバンドLL、LH、HL、HHを符号化する工程と、

f) LLサブバンドの係数を蓄積するとともに、LLサブバンドの所定数の係数が蓄積されるまで、前記工程c) から工程e) を繰り返す工程と、

g) 前記蓄積された所定数のLLサブバンドを注目タイルに割り当てる工程と、

h) 所定分解レベルに達するまで、前記工程d) から工程g) を繰り返す工程と、

i) LLサブバンドを符号化する工程と、を備えることを特徴とする符号化方法。

【0013】

【発明の実施の形態】以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。なお、添付図面のいずれか一つまたはそれ以上において、同じ工程および(または)特徴に同じ参照符号を付して参照するものとし、強調しない限り、これらの工程および(または)特徴は、この説明の目的としては、同じ機能または動作を有するものである。

【0014】図1を参照して、4レベルまでの画像の離散ウェーブレット分散を示す。最高レベルにおいて、すなわち、レベル4において、ロー・ロウ(LL)周波数サブバンド(またはDCサブバンド)およびハイ・ロウ(HL4)周波数サブバンド、ロー・ハイ(LH4)周波数サブバンドおよびハイ・ハイ(HH4)周波数サブバンドがある。次のレベルに下がると、そのレベルに対してハイ・ロウ(HL3)周波数サブバンド、ロー・ハイ(LH3)周波数サブバンド、ロー・ハイ(LH3)

周波数サブバンドおよびハイーハイ（HH3）周波数サブバンドがある。また最低レベルにおいて、ハイーロウ（HL1）周波数サブバンド、ロウーハイ（LH1）周波数サブバンドおよびハイーハイ（HH1）周波数サブバンドがある。

【0015】各レベルは、（単一レベル）離散ウェーブレット変換(DWT)の適用によって発生される。従って、DWTを画像へ適用することは、第1レベルのDCサブバンドと、それよりも3つ高い周波数サブバンドHL1、LH1およびHH1が発生することになる。こうして、DWTが第1レベルのDC（またはLL）サブバンドに適用されて、第2レベルのDCサブバンドと、それよりも3つ高い周波数サブバンドHL2、LH2およびHH2が発生される。次に、第2レベルのDCサブバンドに更にDWTを適用することにより、第三レベルのDCサブバンドと、それよりも3つ高いサブバンドHL3、LH3およびHH3などが発生され、所望のレベルに分解される。その結果、上述の四つのレベルの分解が、（単一レベル）DWTの4つの適用、すなわち、画像への第1の適用と、上述した再帰（回帰）方法により得られたロウーロウ（LL）周波数サブバンドへの後続の適用とを必要とする。

【0016】本発明の好適な実施の形態は、メモリ効率（エフィシエント）と低メモリ・バンド幅（画素当りのバイト）を実現といった点で、所定のレベルの分解を達成する。好ましい実施の形態の更なる実施は、分解されるべき画像のサイズ（1K画素×1K画素より一般的に大きい画像に対して）とは実質上無関係である。すなわち、所定レベルにおける画像の離散ウェーブレット分解は、全画像を記憶するのに必要とする量よりも小さいメモリ容量で、また、画像の幅または高さとはほぼ無関係なメモリ・バッファによって提供される。

【0017】本発明の好適な実施の形態は、DWT分解の最大レベル数を固定にし、分解されるべき画像を複数のタイルに分割し、また、各タイルの一次元が、DWT分解の最大レベルにおけるDCブロックの対応する（第1）次元の所定サイズDCブロックを生成するのに必要な略最小画素数となるように分割することによって、十分なメモリ・バッファと低メモリ・バンド幅を達成する。さらに、各タイルの二次元は、所定サイズのDCブロックの対応する（第2）次元を生成するのに必要とする略最小画素数より少ない。好ましくは、各タイルの二次元は、画素で測定した最小長さを有していることであり、これは最大レベルにおける所定サイズDCブロックの対応する次元の（フラクション）サブ・マルチプルを生成することを必要とされるからである。

【0018】例えば、好ましい実施の形態によれば、Daubechies 9/7 DWTフィルタ（すなわち、9タップ・ロウパスと7タップ・ハイパス）、および各レベルにおいて32係数×32係数（高さ/幅）のエントロピー

符号化DWTブロック（アレイ）を使用して、各レベルにおいて、DWT分解の4レベルの最大まで、画像を複数のオーバーラップするタイルにサブ分割する必要がある。各タイルは一次元（例えば、タイルの高さ）内で617画素となり、二次元（例えば、タイルの幅）内で71画素となる。各タイルは、分解の各反復において、ロウーロウ周波数部分の中間バッファリングにより4レベルにDWT分解され、所定数のDC係数が生成される。この種のタイルで約9回処理すると、DC係数の32×32ブロックがリカバーされ、エントロピー符号化される。各反復を通して、より高い周波数係数（例えば、各レベルにおけるHL、LHおよびHH）の32×32ブロックもエントロピー符号化される。

【0019】次に、好ましい実施の形態につき、まず最初に、Daubechies 9/7 DWTフィルタを使用する画像の4レベルDWT分解を参照し、続いてDWT係数の32×32ブロックのエントロピー符号化を説明するが、本発明はこれに限定されるものではない。ブロックのエントロピー符号化の前に、分解レベルの異なる所定最大数、異なるフィルタおよび/または蓄積された係数の異なるブロックサイズが、本発明の範囲と概念から逸脱することなしに使用することができる。例えば、5レベルDWT分解が実行でき、および/またはDWT係数の64/64ブロックを、1つのブロックのエントロピー符号化の前に蓄積することができる。さらに、多数のウェーブレット・フィルタが「HAAR」フィルタ、「LeGall 5/3」、「LeGall 10/18」を含めて使用することができる。この種の変形例により、次に説明するハードウェア・アーキテクチャを対応して変更してもよい。

【0020】ここで、図2および図3を参照して、本発明の好適な実施の形態による画像のタイリングを示す。すなわち、好ましくは画像を部分的にオーバーラップさせて、複数のタイルにサブ分割する。各タイルの大きさは、分解のレベル、エントロピー符号化の前に蓄積されたブロック・サイズ、および使用されるDWTフィルタの種類等の所定パラメータを固定することによって決定される。各タイルは、タイルの一つの次元の71画素に、他の次元の617画素を掛けたものからなる。図2において、タイルは水平方向に配置され、すなわち、タイルの長手側が画像の走査線と略平行にレイアウトされている。図3は、タイルが垂直方向に配置された別の構成を示しており、短手側が画像の走査線と略平行である。図2および図3の各タイルは、「タイルn」の符号が付けられるが、ここでnは正の整数で、符号付け（ラベリング）は複数のタイルのうちの好ましい処理順序を示す。例えば、「タイル1」は「タイル2」の前に処理され、「タイル2」は「タイル3」の前に処理されるという具合である。さらに、各タイルは次に説明するように決定された近接のタイルと所定のオーバーラップ域200と201を有している。

【0021】一般的に、DWTの第2レベルにおいてサブバンドのHラインを生成するために、Daubechies 9/7 DWTフィルタを使用して、LL1サブバンドの $(2H+7)$ ラインが必要で、次に、これには画像の $(2(2H+7)+7=4H+14+7)$ ラインが必要である。2Hラインに7ラインを付加することは、Daubechies 9/7 DWTフィルタによって必要とされ、こ

$$2^J H + (2^{J-1} + 2^J + 2^6) \cdot 7 = 2^J H + 7(2^J - 1) \quad \text{式1}$$

の画像ラインが必要である。上式の第2項が必要とされるオーバーラップである。もちろん、これは他の次元に対しても同様に適用される。例えば、Daubechies 9/7フィルタを使用する4レベル分解($J=4$)と、32画素幅(水平方向) $W=32$ のブロックを蓄積するためには、式1によれば $(16 \times 32 + 7 \times 15 = 512 + 105)$ 幅のタイルを必要とし、ここでオーバーラップ200は、105画素である。すなわち、図2中の「タイル1」と「タイル11」間のオーバーラップ200は、水平方向の105画素に垂直方向の71画素 $(2H+7)$ を挿けたものである。

【0023】本実施の形態では、一度に $2H+7$ ラインを変換し、大きくなるラベル番号順に次の(オーバーラップした)タイルを処理して、HL1、LH1およびHH1サブバンドの各々からHラインを生成する。これらのサブバンド部分は、このような画素(Hライン倍した) W (この例では $W=32$ 画素)、すなわち、2H画素列が入力されると、上述した場合のように、直ちにコード化できる。タイルの第1パス(すなわち、一つの分解レベルにおける一つのタイル)の処理の終了時点において、LL1サブバンドの1ライン当たり $(2^{J-1}W+7(2^{J-1}-1))$ 倍したHラインをバッファする必要がある。LL1サブバンドの1ライン当たりのこの係数値は、トップ・レベル(レベルJ)における W 幅ブロックを形成するために必要である。図2において、タイル「タイル1」が処理された後、「タイル2」がほぼ同様にして処理される。すなわち、画像の他の $(2H+7) \times (2^J W + 7(2^J - 1))$ 画素のタイルが処理される。ここで、「タイル2」が処理されると、LL1サブバンドの他のHラインが発生され、二つのタイルが処理された後、合計2Hラインとなる。LL1サブバンドをさらに分解してLL2サブバンドのHラインを生成するためには、LL1サブバンドの $(2H+7)$ ラインが必要である。「タイル2」の処理後には、2Hラインが生成され、「タイル3」の処理後は、 $(2H+7)$ ラインを提供するための合計3Hを提供できる。これにより次のレベルの分解を実行してLL2サブバンドのHラインを生成する。即ち、各タイル毎にDWTを行い、LL1を生成するが、次のLL2を生成するために3つのタイルのLL1を用いて更にDWTを実行する。この方法によれば、小さいタイル単位(小さいバッファ容量)で2レベルのDWTが実行できる。同様にして、こ

のことが「オーバーラップ」200として表されている。好適には、このオーバーラップは、処理されるべき2Hラインの前3ラインと、処理されるべき2Hラインの後4ラインとして分布されることである。

【0022】数学的帰納法によれば、一つの次元において、いずれかのサブバンドのHラインを形成してレベルJにするには、

の方法を続行して少なくとも $(2^J H + 7(2^J - 1))$ ラインを処理すると、分解のJ番目のレベルにおける係数のHラインを生成することになる。即ち、レベル3(LL3)以降のLLnの作成のためにも、そのLLnの作成に必要な複数タイル分のLL $(n-1)$ を利用することにより、HラインのLL $(n-1)$ を最終的に生成する。ラインが $(2^{J-1}W+7(2^{J-1}-1))$ 画素(またはDWT係数の長さ)であり、画像の幅または長さを必要としないことに注意されたい。次のレベルの分解を得るために各々の繰り返しにおいて、LL周波数のサブバンドの合成係数がバッファされ、他方で現在レベルにおいて発生されたより高い周波数(すなわち、LH、HL、HH)の各 $H \times W$ ブロックが符号化のためにエン트로ピー符号化器に送られる。

【0024】普通、出力デバイスおよび特に表示デバイスは、ラスタ走査順に、左から右へ、また、上から下へ画像を表示する。従って、本実施の形態の好ましい実施においては、図3に示したタイル配列となる。オーバーラップするタイルは、タイルの最短次元が画像の走査ラインと平行になるように配列されるのが好ましい。好ましくは、タイルはタイルの次の列が処理される前に、画像の幅を横切って処理される。この配列は、タイルが左から右へ、また上から下へ(すなわち、ラスタ走査タイプ順)に処理できるので好ましいが、他の順序も本発明の範囲と概念から外れることなしに使用することができる。例えば、図2のタイル配列は図3のものとは別の配列である。

【0025】図3の配列において、各タイルはサイズ $(2^J H + 7(2^J - 1)) \times (2W + 7)$ 画素であり、ラスタ走査順で処理される。J=4レベルDWT分解に対し、またエン트로ピー符号化のために $H=32/W=32$ 係数ブロックサイズを得るために、画像の各タイルは、1タイルのライン当たり次元 $(512+105=617$ ライン) $\times (2 \times 32 + 7=71$ 係数(または画素))を有している。

【0026】図4を参照すると、図4はエンコーダ(符号化器)400のブロック図を示し、非圧縮画像401を取り入れて、画像を表す圧縮ビットストリーム402を生成する。このエンコーダ400は、所定量の外部メモリ403、好ましくはダイナミック・ランダム・アクセス・メモリDRAMと、それ自体所定量の内部メモリ405、一般的にはスタティック・アクセス・メモリS

RAMを含む圧縮エンジン404とを備えている。このエンコーダ400と、特に圧縮エンジン404は必要とされる工程を実行し、非圧縮画像401から圧縮画像ビットストリーム402を生成し、リニア変換(DWT)の実行と、その変換係数をエントロピー符号化するエントロピー符号化器を含んでいる。さらに、圧縮エンジン404は単一チップとして設けられ、そのチップを通してマスター画素クロック信号clkを使用するのが好ましい。

【0027】非圧縮画像401は、エンコーダ400の外部DRAMによりバッファリングされ、かつ、圧縮エンジン404によって制御される。外部DRAMは、これまでに説明したタイル(617×71画素又は係数)に略対応する非圧縮画像の一部をバッファするのに使用されるのが好ましい。圧縮エンジン404は、各ラインに沿った71画素または係数に対応するデータの短いバーストを読み取り、タイルをジグザグ順に上から下に走査する。

【0028】図5はジグザグ順500を示し、この順に各タイルが走査され、図3の配列よりも詳しいイラストを画像502の部分で示す。これまでに説明したように、タイルの幅は(2W+7)であり、W=32を所望のブロック幅とすると、タイル幅(またはタイル内のラインの長さ)は71画素(すなわち、64+7、7は水平方向のオーバーラップ201)となる。従って、圧縮エンジン404は、617ライン、すなわち、H=32としたとき、(2³H+7(2³-1))ラインが読み取られるまで、同時にジグザグ順で走査されてタイルの1本のラインを一つ読み取るのが好ましい。すなわち、前回の説明部分は、近接するタイル間のオーバーラップ域200の垂直方向の寸法を構成している。例えば、四つのレベルDWTに対し、オーバーラップ域200は、タイルの長さ方向における105画素に幅(71画素)を掛けたものである。より高い周波数要素(LH、HL、HH)の各32×32(H×W)変換ブロックが発生されると、このブロックは中間バッファリングなしにエントロピー符号化される。したがって、好ましい実施の形態において、水平方向における二つの近接するタイル(例えば、「タイル1」と「タイル2」)間のオーバーラップ域201は、略7画素幅にタイルの長さを掛けたもので、垂直方向の二つの近接するタイル(例えば、「タイル1」と「タイル1」)間のオーバーラップ域200は、略105画素長さにタイルの幅を掛けたものである。次に図6は、図4の圧縮エンジン404をより詳細に示す。圧縮エンジン404は、アドレス・バス603を介してDRAMに伝送されるメモリ・アドレスに応答して、外部DRAM403からの602データを受取るリオーダ(再配列)・バッファ・モジュール601と；制御状態マシン・モジュール604と；垂直フィルタ605と；水平フィルタ・モジュール606と；サブバンド

・バッファ607と；部分バンド・バッファ608と；エントロピー符号化ユニット(符号化器)609を含んでいる。

【0029】リオーダ・バッファ・モジュール601は、外部DRAM403に対する読み取りと書き込みを制御するとともに、DRAM403から受信したデータを組立て、所定順序で垂直フィルタ605で読み取りできるようにする。このリオーダ・バッファ・モジュール601は9バイト(72ビット)のデータを一度に、垂直フィルタ・モジュール605に送る。垂直フィルタ・モジュール605は、第1次元(図3のタイルの垂直方向)で、ロウバスおよびハイバス・フィルタリングを実行する。垂直フィルタ・モジュール605からの二つの出力611と610があり、それぞれデータのロウバスおよびハイバス・フィルタリングに対応している。各出力610および611は、水平フィルタ・モジュール606内の分離水平フィルタにパスされ、ここで同一のフィルタ操作が第2次元(図3のタイルの水平方向)において実行される。水平フィルタ・モジュール606は、LL、LH、HLおよびHHサブバンド信号に対応する四つの出力信号を提供する。高い周波数成分(すなわち、LH、HLおよびHH)を含むサブバンド信号が、エントロピー符号化器609によって符号化される前に、サブバンド・バッファ・モジュール607内で一時的にバッファされ、613に符号化ビットストリームを出力する。LLサブバンド信号が部分バンドバッファ608にパスされ、ここで、より大きいブロックに合成され、後続の変換レベルのために垂直フィルタ・モジュール605に帰還される。最終レベル(例えば、レベル4)においてのみ、LLサブバンド信号が出力ビットストリーム(出力613を介して)に符号化される。制御状態マシン・モジュール604が、圧縮エンジン404の動作を管理し、適切なデータが正しい場所に記憶されるか、また、その動作が正しい時刻に正しいデータで実行されるかどうかを確認する。

【0030】次に図7を参照して、図6のリオーダ・バッファ・モジュール601をより詳しく説明する。このリオーダ・バッファ・モジュール601は、外部メモリ403から受信した中間生データ602を記憶するランダム・アクセス・メモリ(以下、リオーダRAMという)700と、外部DRAMサブモジュール403を制御するためのDRAMコントローラを含むリオーダ・アドレス・ジェネレータ・サブモジュール701とを含んでいる。data_valid(データ・ヴァリッド)信号702は、(生)の画素データ401をDRAM403に供給する外部デバイスから供給される信号であって、DRAMデータ・ピン上の現在データが有効であることを示している。data_ack信号703は、データがDRAM403に読み取られたことを確認する。外部DRAM403が、その入力を外部画素供給源(図示省略)から直接取

り入れる。リオーダ・アドレス・ジェネレータ701は、アドレス704を決定してリオーダRAM700に供給し、そのRAM700への書き込みを制御する。

【0031】リオーダRAM700からのデータ・アウト(Data Out) (DO) ピンが、同じリオーダRAM700のデータ・イン(Data In) (DI) ピンに、8ビットシフトして接続されている。すなわち、DO (n+7:n) ピンがDI (n+15:n+8) ピンに接続され、ここで、nは低い方のビット数を表している。例えば、72ビット・ワード・データ・アウトの少なくとも8個の重要なビットを表すDO (7:0) ピンが、72ビット・ワード・データ・インの第2の少なくとも8個の重要なビットに対応するDI (15:8) に接続されている。データが、読取—修正—書込(リード・モディファイ)技術を使用してリオーダRAM700に書き込まれる。72ビット・ワード・データ・インの第2の少なくとも8個の重要なビット、すなわち、DI (7:0) ピンへのバイト入力が外部DRAM403から入り(DRAM Data)、一方残りの入力データ・ワードが上述したシフトされた態様でリオーダRAM700の出力から入力されている。従って、リオーダRAM700は、大きいバイト幅をアドレス可能なシフト・レジスタのように、9ワード深さと71ワード幅で作動する。データが、画像の走査線(すなわち、タイルのライン)に沿って71ワードのバースト内で外部DRAMから読み取られる。画像データの二つのライン(71バイト×2)が、垂直フィルタ・モジュール605によって使用されるサンプルの各列(ライン)に対してリオーダRAM700に書き込まれる。これはDWTによって必要とされる垂直方向内の画像データのダウン・サンプリング(1/2に減らすこと)を構成ないし提供する。

【0032】上述したように、リオーダRAM700からのリオーダ・データが、ro_dataバス705を介して、一度に9バイト(72ビット)が垂直フィルタ・モジュール605に供給される。

【0033】図8は図7のリオーダ・バッファ601の構成を表したものである。好適な実施の形態のリオーダ・バッファ601は、71画素幅と9ラインで構成されている。このバッファ601は、図7において矢印801で示したように左から右に書かれる。各ラインが左から右に書かれると、1ライン分の画素が一度にライン分プッシュダウンされる。すなわち、バッファの9ラインが書込まれると、(9×71=)639画素がバッファを占有し、639画素の最初(左上)の画素が、バッファの9番目のライン804の最初の位置にあるバッファのセル802の位置のバッファに入り、一方639画素の最後(右下)の画素が最初のライン805上のバッファのセル803の位置のバッファに入る。メモリ700(リオーダRAM)が72ビットの71ワードとして編成される。ここで72ビットは垂直方向のディメンジ

ョン(寸法)である9画素を表している。9ラインが同じ71アドレスに繰り返して(一度に8ビット)書き込まれると、全バッファ601が満たされる。639画素の最終71バイト(画素)がトップ・ラインに書き込まれると、読取—修正—書込サイクル(全72ビット)が使用されている間、データが読み取られて垂直フィルタ・モジュール605にロードされる。一度、リオーダ・バッファ601が満たされると、データがバッファ601にロードされる画像データのどの第2ラインからでも垂直フィルタ・モジュール605によって使用される。これは垂直方向におけるサブ・サンプリングとなる。

【0034】垂直方向におけるサブ・サンプリングは次のようにして達成される。最初に、垂直フィルタ・モジュール605がリオーダ・バッファの9ラインを処理する。このとき、9タップ・フィルタの内の1番目のタップが、リオーダ・バッファの最終ライン805からのデータを使用するようにアレンジされ、一方、フィルタの最終タップがリオーダ・バッファ601の第1ライン804からのデータを使用して、例えば、部分バンド・バッファ608内に記憶されることになるロウ・パス・フィルタされたデータの1ラインを生成する。当然、1番目のタップと最終タップの間のタップは、リオーダ・バッファ601の対応するラインからのデータを使用する。次に、リオーダ・バッファ601の各ラインのデータはプッシュダウンされ、リオーダ・バッファ601の最終ライン804は除去され(バッファから押し出され)、各先行ラインからのデータがリオーダ・バッファ601の次のラインに書き込まれる。すなわち、バッファの8番目のライン806からのデータが、9番目のライン804に、また、バッファの7番目のラインからのデータが8番目のライン806に書き込まれ、6番目のラインからのデータが7番目のラインにというようにして書き込まれる。リオーダ・バッファ601の初頭ライン805において、新しいラインデータが外部DRAM403から入力される。このプッシュダウンが、リオーダ・バッファ601に入力された全てのラインに対して繰り返される。こうして、第2ラインのデータがリオーダ・バッファ601に入力された後、垂直方向内のサブ・サンプリングが、リオーダ・バッファの9ラインの読み取りと処理によって実行される。このリオーダ・バッファは、タイルの開始時点で完全に再充填され、全画像が処理されるまで、上述の処理が繰り返される。

【0035】画像の略71ラインが外部メモリ403から検索され、かつ、リオーダ・バッファ601に伝送された後、サブバンド・バッファ607がエンコーディングを開始するための十分な変換係数を含んでいる。好適な実施の形態において、サブバンド・バッファ607がいっぱいになり、リオーダ・バッファ601がさらに更新される前に、係数が符号化される必要がある。より高いレベルの係数がフィルタリングされている間、次のレ

ベルのフィルタ操作の準備のために、新しいデータがリオーダ・バッファにロードされる。

【0036】リオーダ・アドレス・ジェネレータ701は、リオーダ・バッファ601のために、単純に0から70までの周期でアドレスを発生するが、読み取り中、図5を参照してこれまでに説明したように、ジグザグに外部DRAM403をアドレスする。このリオーダ・アドレス・ジェネレータ701は、画像を外部DRAM403に書き込み中、ラスタ走査順の制御もする。外部DRAM403は、画像の少なくともフルに617幅のラインを記憶するのに十分な大きさであるのが好ましい。実際には、十分なバンド幅はデュアル・ポートDRAMで実現でき、あるいはオプションとして、ビデオ・ランダム・アクセス・メモリ（VRAM）が外部メモリ403として使用できる。

【0037】図9は図6の垂直フィルタ・モジュール605をより詳細に示す図である。この垂直フィルタ・モジュール605はマルチプレクサ901を備えており、レベル1の分解を実行するときのリオーダ・バッファ601からのデータか、又は後続の分解レベル中、部分バンドバッファ608からのデータのいずれかを選択する。リオーダ・バッファ601から受信されたデータの9バイト×8ビットが16ビットの9ワードに変換される。これは未使用ビットをゼロにすることによって、部分バンド・バッファ608からのデータと一致する。マルチプレクサ901は、制御状態マシン・モジュール604によって制御される。一般的に、1画素データ当たり8ビットのカラー・チャネルが16ビットの精密DWT係数に変換され、これによって情報の最小量が係数値に対する丸め効果のために失われる。

【0038】さらに、垂直フィルタ・モジュール605は垂直ロウ・バス・フィルタ902と、垂直ハイ・バス・フィルタ903を含み、これらはマルチプレクサ901の選択に基づいてデータを受信する。垂直ロウ・バス・フィルタ902は、9×16ビットのサンプルを得て、データに対してマトリックス操作（乗算－蓄積）を実行し、入力全9ワードから16ビットの結果(vfl_data)を生成する。垂直ハイ・バス・フィルタ903は、7×16ビット・サンプルのみを必要とし、そのマトリックス操作を実行し、再度16ビットの結果(vfh_data)を生成する。これまでに説明したように、本実施の形態のフィルタはDaubechies 9/7 DWTに基づいているが、他のウェーブレット・フィルタも使用できる。

【0039】図10は、図6の水平フィルタ・モジュール606のより詳しい説明図である。この水平フィルタ・モジュール606は、垂直フィルタ・モジュール605からのデータを処理する二つのほぼ同様のセクション1004と1005を含んでいる。各セクションは9ステージ・シフト・レジスタ1001、水平ロウ・バス・フィルタ1002および水平ハイバス・フィルタ100

3を含んでいる。データはまず最初に、垂直フィルタ・モジュール605から受信された最大9個の連続するサンプルをバッファする、一対の9ステージ・シフト・レジスタを通過して同時に処理される。一方、二つのセクション1004と1005はほぼ同じ方法でデータを処理するが、各セクションのためのデータ源は異なる。一つのセクション1004が、垂直ロウ・バス・フィルタ信号vfl_dataを受取り、一方、他方のセクション1005が垂直ハイ・バス・フィルタ信号vfh_dataを受取る。

【0040】各セクションの水平ロウ・バス・フィルタ1002が9×16ビット・サンプルを取り込み、そのデータにマトリックス操作（乗算－蓄積）を実行し、入力全9ワードの各々の結果から16ビットを生成し、セクション1004と1005それぞれからhl_data 信号1006とhl_data 信号1007を生成する。各セクションの水平ハイ・バス・フィルタ1003は7×16ビット・サンプルのみを必要とし、マトリックス操作を実行し、再度各ハイ・バス・フィルタ1003が16ビット結果を生成する。セクション1004の水平ハイ・バス・フィルタが信号lh_data1008を提供し、セクション1005の水平ハイ・バス・フィルタが信号hh_data 1009を提供する。その結果得られたサブバンド・データ、すなわち、LL、LH、HLおよびHHサブバンドの係数に対応する信号ll_data lh_data hl_data、およびhh_data 信号がサブバンド・バッファ・モジュール607に送られる。

【0041】9ワード・サンプルの処理後、両9ステージ・シフト・レジスタがデータを2だけシフトし、水平方向にサブ・サンプリング（1/2に減少）のためにアカウンタする。

【0042】DWT分解の現在レベルにおけるLLサブバンドの係数に対応するll_data 信号1006も、分解の続くレベルのために部分バンド・バッファ・モジュール608に送られる。

【0043】図6のサブバンド・バッファ・モジュール607を図11により詳しく示す。このサブバンド・バッファ607は、LL、LH、HLおよびHHサブバンドの各々から所定数の係数を記憶する所定数のランダム・アクセス・メモリ（RAM）1101と；サブバンド・バッファ・アドレス・ジェネレータ・サブモジュール1102とマルチプレクサ1103を含んでいる。サブバンド・バッファ・アドレス・ジェネレータ・サブモジュール1102は、サブバンドRAM1101にアドレスを供給するとともに、制御状態マシン・モジュール604の制御下でRAMへの書き込みの制御もする。

【0044】マルチプレクサ1103は、エントロピー符号化ユニット（エントロピー符号化器）609によって符号化するために四つのサブバンドの一つから、データと係数を選択する。

【0045】図12を参照して、図11のサブバンド・

バッファのデータ構成ないし編成の態様を説明する。サブバンド・バッファは、 32×32 個の係数からなる四つのバンク1201を保有しており、各係数は16ビット値で表わされる。従って、サブバンドRAMは合計64キロビットの容量で、1ワード64ビットで1024ワードで構成されており、各メモリ位置で並行して、各サブバンドからの1つの係数をストアする。即ち、現レベルで、各サブバンドからの1つの係数、計四つの係数が、それぞれ64ビット・ワードの16ビットとして表わされる。バッファ1101、特に各バンク1201が、図12に示したジグザグ・パターン1202によって示された上から下へのラスタ操作順に満たされる。エンコーディング中、エントロピー符号化ユニットにおいて、サブバンドRAM1101が4分木(quadtree)方式で読み取りされる。

【0046】図13を参照して、図6のエントロピー符号化ユニット609をより詳しく説明する。このエントロピー符号化ユニット609は、ビットプレーン・コンバータ・モジュール1301と16個のビットプレーン・メモリを含むビットプレーン・バッファ・サブモジュール1302と；複数のビットプレーン・ツリー・ビルダー・ユニット1303と対応するビットプレーン・ツリー・バッファ・ユニット1304と；ビットプレーン・エンコーダ1305と；ビットプレーン・エンコーダ1305によって使用されるデータの中間記憶のための複数の中間バッファ1306とを備えている。サブバンドRAM1101に記憶されたデータがサブバンド・バッファ・モジュール607からクワドトリー順に読み取られる。ビットプレーン・コンバータ・モジュール1301は、各ビットプレーンのデータを分離するとともに、ビットプレーン・バッファ・サブモジュール1302内の16個のビットプレーン・メモリの各々に書き込む。データはまた、処理されたビットプレーン・ツリー構造情報を、対応するビットプレーン・ツリー・バッファ1304に書き込む複数のビットプレーン・ツリー・ビルダーによって走査される。ビットプレーン・ツリー1304とビットプレーン・バッファ1302の各々からのデータが、ビットプレーン・エンコーダ1305によって使用され、ビットストリームの発生を実行して、これが(バス613を介して)出力される。ビットプレーン・エンコーダ1305は制御状態マシン・モジュール604によって同期化(制御)される。中間バッファ1306は、重要な係数のリスト(LSC)を一時的に記憶するバッファと；重要でない係数(LIC)のリストを一時的に記憶するバッファと；域バッファの第1リストと域バッファの第2リストとしての二つのバッファとして実現できる古い重要でない域(OIR)リストを一時的に記憶するバッファ(図13では不図示)を含んでいる。さらに、エントロピー符号化ユニット609は、サイン・ビット・バッファ(図13では不図示)を

含み、これはビットプレーン・コンバータ・モジュール1301の出力と、サイン・ビットをビットストリームに符号化するビットプレーン・エンコーダ1305とに接続されている。エントロピー符号化ユニットの一例を本明細書中の、見出し「エントロピーエンコーダ」で後ほど詳細に説明する。

【0047】本発明の好適な実施の形態を、上述したクワドトリー・ビットプレーン・エンコーディングを実行するエントロピー符号化ユニットを参照して説明するが、他のエントロピー符号化器も本発明の範囲と概念から外れることなく使用することができる。例えば、ビットプレーン演算コードが変換係数を符号化するのに使用でき、または随意に、一部クワドトリー・エンコーディングとDWT係数の一部演算コーディングを実行するハイブリッド・エンコーダを使用できる。

【0048】図14を参照して、図6の部分バンド・バッファ・モジュールをより詳しく説明する。部分バンド・バッファ・モジュール608は：分解の各レベルに対するLLサブバンド・データ(すなわち、図6の水平フィルタ・モジュール606によって発生されたll_data1006)を記憶するRAM1401と；部分バンド・バッファ・アドレス・ジェネレータ・サブモジュール1402と、バレル・シフト・レジスタ(バレル・シフター)1403とを含んでいる。部分バンド・バッファ・アドレス・ジェネレータ1402がアドレスを部分バンドRAM1401に供給するとともに、制御状態マシン・モジュール604の制御下で、そのRAMへの書き込みを制御する。

【0049】部分バンド・バッファは、十分なサンプルが分解の次のレベルをフィルタするのに利用可能になるまで、圧縮の各レベルからLLサブバンド・データを保持する。一旦、十分なサンプルがいずれか一つのレベルにおいて利用可能になると、データがバレル・シフター1403を介して垂直フィルタ・モジュール605に読み取られ、分解の次のレベルがフィルタされる。概念的に、RAM1401は三つの別々のバッファを含み、各々100個のセル幅を有し、各セルは16ビット値を記憶することができる。これら三つの別個のバッファの内、第1バッファは305行(row)(ライン)深さで、第2バッファは149行の深さで、第3バッファは71行の深さであり、総計525行(ライン)になる。しかし、垂直フィルタ・モジュール605が、Daubechies 9/7フィルタによって必要とされる9個の値を取るもので、行(ライン)の合計は9の倍数であるのが望ましい。従って、余分の6行のセルがバッファに付加され、結果として合計531行(すなわち、9の59倍)になる。本実施の形態において、この結果は少なくとも $59 \times 100 \times 144 = 531 \times 100 \times 16$ ビット、すなわち、約0.81メガバイトのメモリ・サイズになる。

【0050】バレル・シフト・レジスタ1403が、デ

ータの単一クロック・アライメントを垂直フィルタ・モジュール605によって必要とされる所望の行にすることを許容する。部分バンドRAM1401から読み取って垂直フィルタ605のための単一ワードを生成するために所定のメモリ位置が必要となる。必要とされる単一ワードが9ラインの境界線に沿って包括される場合、二つの近接するメモリ位置が読み取られる必要がある。これは一般的に、9ライン・セット（各セットは9ラインからなる）の内8本に発生する。この場合、データは、両ワードが読み取りされ、全ての必要とされるビットが利用可能、かつ、適切に配置されるまで、バレル・シフター1403内にラッチされる。

【0051】図15を参照して、部分バンド・バッファ608のデータ編成の様態を説明する。部分バンド・バッファ608は、100係数幅と525係数高さと、上述したように、これに100係数幅の行（図15では不図示）につき、余分の6を加えたものである。部分バンド・バッファ608は、タイルの少なくとも一つのレベルDWT分解が実行された後、データ（係数）に書き込まれる。従って、分解の1レベル後で、画像のタイル $(2^j H + 7(2^j - 1)) \times (2W + 7)$ が、係数のタイルを少なくする。すなわち、分解の第1レベルが 305×32 係数のタイルまで少なくなった後、 617×71 画素タイルになる。好適な実施の形態の部分バンド・バッファ608は、100係数幅において、少なくとも幅方向に三つの32係数幅タイル1501と4係数幅ブロック1502を収容するように設計される。各32係数幅のタイルが、図15に示すように一度に1レベルで、左から右に、また、上から下の走査ライン順に書き込みされる。すなわち、第2レベル分解（各々、長さ $256 + 49 = 305$ 係数）のための三つの32係数幅のタイル1501が、各々ジグザグ順に満たされる。4係数幅のブロック1502が、少なくとも最初のタイル処理のために、最初の32係数幅タイル1501（以後、「パディング(padding)」ないし「パadded(padde)」という）の係数の最初の4カラムの鏡像で満たされる。このパディングは普通、画素境界線外の画素データが存在しない画像の境界線のためにアカウントするように実行される。他のパディング技術が、画像境界線のためにアカウントするのに利用できる。例えば、最も簡単な、しかしベストである必要はない、オプションが、画像の境界線をゼロ(0)でパッドするものである。

【0052】第2レベルの分解のための係数が、垂直フィルタ605と水平フィルタ606モジュールを介して処理されると、第三レベルの分解のための係数が、次のレベルの32係数幅のタイル1504と長さ方向 $(128 + 21 = 149)$ 係数に蓄積される。再度、最初に4係数幅ブロック1503が対応する（近接）タイル1504の係数でパッドされる。第三レベル分解の係数が垂

直フィルタ605と水平フィルタ606モジュールを介して処理されると、4レベル分解のための係数が次のレベル32係数幅のタイル1506に蓄積される。再度、最初に4係数幅のブロック1505が、対応する（近接）タイル1506の係数でパッドされる。画像の境界線において、4係数幅・ブロック1502、1503および1505がパッドされるべきとき以外の場合において、4係数幅ブロックが先行タイルの最終4カラムを含み、このレベルで処理される。

【0053】各レベルにおいて4係数幅ブロックが、水平線をフィルタするためのオーバーラップとして使用される。1レベルにおいて、一つの完全な32係数幅タイルを発生するために、先行レベルにおいて71係数幅が必要である。第2レベルの分解のための 305×71 係数からなる図15の隠れたエリア1507には、第三レベルの分解のための 149×32 のタイルを発生することが要求される。この隠れたエリア1507内の係数は、走査ライン順に、1ラインの左から右へ71カラム分、また上から下へ本読み取られる。一度、部分バンド・バッファ608が次の（高い）レベルで所望の量（例えば71カラム）まで満たされると、現在レベルにおけるカラム・アドレスが64カラムだけ右に増大し、アドレスが100番目のカラムを越えると第1カラムを包み込む。これは、例えば、現在レベルにおいて、係数の次の71カラムの処理が、バッファ608の64番目のカラム1508で開始され、バッファ608の第1カラムの近くまで包み込まれる。カラム64からカラム68までが、ここで4係数幅ブロック1509のオーバーラップの必要性を提供し、データのもう一つの71カラムを処理する必要がある。

【0054】部分バンド・バッファ608の各レベルがフィルタされると、フィルタ（LLサブバンド・係数）の結果が、図15に示した部分バンド・バッファ608の次の高いレベルに書き込まれる。最も高いレベルがフィルタされてしまうと（例えば、図15の「レベル4」）、得られたLLサブバンド・係数が、これらの係数を部分バンド・バッファ608にさらに書き込みせず、サブバンド・バッファ607に書き込み、さらにフィルタせずにストレートに符号化される。

【0055】部分バンド・バッファの読み取りの際に、 $9 \times 16 = 144$ ビットが一度に読み取られる。これらの9ワードはバッファの単一カラムからの9個の垂直係数を表している。このバッファから完全に読み取られるライン間で、ライン・アドレスが2だけ増大するので、144ビットストリームがバレル・シフタを通り、このデータと垂直フィルタへのデータ・バスに整列される必要がある。さらに、続いて2ワードを読み取り、一つの144ビット・ワードを得て、レジスタ内に第1ワードの必要とするビットを保持することによってワードの二つの部分を構築し、他方で第2アクセスから必要とされ

るビットがフェッチされる。ライン・アドレスの一致が9の倍数であるときのみ、一つのフェッチだけが必要である。一度に144ビットをフェッチできるようにするために、メモリは物理的に144ビット幅にしなければならない。余分なデコーディングが、メモリ・アレイに書き込み可能な形態にする必要がある（従って、ちょうど16ビットが同時に書き込みできるようにする）。

【0056】図16は、図6の制御状態マシン・モジュール604のための流れ図を示す。この制御状態マシン・モジュール604は、各モジュール601、604～609の機能を制御すること、およびデータを各モジュールにモジュール機能と同期させて（すなわち、正確な時間で）正しくルーティング（経路指示）する。

【0057】画像圧縮の開始1601において、「レベル」変数が1602で1に設定される。これは一般的に実行されるべき分解の第1レベルを表している。決定ブロック1603は、DWT分解の現在レベルが第1レベルにあるかどうかを判定する工程に入る。この決定ブロック1603が真（YES）の返答であれば、DRAM（図4）に記憶された画像の最初の9ラインがフィルタされ（1604）、サブバンド・データの第1ラインが生成される。入力データがレベル1のために外部DRAM403から取り出される。そうでなければ、決定ブロック1603が、偽（NO）の返答をし、ブロック1605にこれを送って、ここでデータが内部部分バンド・バッファ608から取り出され、フィルタ（次のレベルにDWT分解）される。十分なデータが部分バンド・バッファ608から取り出され、次のレベルのDWT係数のタイル内のラインを生成する。これらの結果はこれまでに説明したように対応するレベルのためのロケーションの部分バンド・バッファ608に記憶される。

【0058】データの71ラインがフィルタされた後、サブバンド・バッファ607の32ラインがいっぱいになる。この時点において、流れ図1600はサブバンド・バッファ607がいっぱいであるかどうかをチェックする。このチェック1606が偽（NO）の返答をすれば、処理流れは決定ブロック1603に戻り、決定ブロック1603を参照して上述した処理を続行する。チェック・ブロック1606が真（YES）を返答すれば、さらなる決定（ないしチェック）ブロック1607に進み、データがフィルタされるべきレベルのためのオーバーラップ域内にあるかどうかをチェックされる。この域になれば、ブロック1607が偽（NO）を返答し、サブバンド・バッファ607内に記憶された三つのサブバンドLH、HLおよびHHが符号化される（1608）。これら三つのサブバンドのエンコーディングに続いて、決定ブロック1609に進み、分解の現在レベルが最大レベル（すなわち、レベル4）であるかどうかをチェックされる。現在レベルが4であれば、決定ブロック1609が真（YES）を返答し、サブバンド・バ

ッファ607内のLLサブバンドも符号化され（1610）、流れ処理が決定ブロック1611において続行される。さらに、決定ブロック1609において、偽（NO）が返答されれば、サブバンド・バッファ607内のLLサブバンドが符号化されずに、流れ処理が決定ブロック1611で続行される。

【0059】決定ブロック1607を介して、ブロック1611への他のエントリがあり、ここで決定ブロック1607が真（YES）を返答してくる。即ち、処理されるべきデータが、近接するタイルのオーバーラップ域201または200内にある場合にブロック1611に進む。この決定ブロック1607において実行されたチェックは、オーバーラップ域の符号化を二度することを回避するためで、決定ブロック1607が真（YES）を返答したときに、決定ブロック1611に進行する。

【0060】決定ブロック1611において、部分バンド・バッファ608（図6）のレベル2において、（256÷105）行で71カラムの完全な係数のセットを有しているかどうかをチェックされる。もしそうであれば、「レベル」変数が1612で2にセットされ、処理が決定ブロック1603において再開され、上述したように工程1605から1611までが続行される。

【0061】レベル2で部分バンド・バッファがまだ一杯にされていないければ、即ち、完全なセット（すなわち、71×361係数からなるレベル2）が部分バンド・バッファ608になれば、偽が決定ブロック1611によって戻され、チェック（決定）ブロック1613に進む。チェック・ブロック1613が、部分バンド・バッファのレベル3において、（128÷21）行の71カラムの完全な係数セットが存在するかどうかをチェックする。もしそうであれば、「レベル」変数が1614で3にセットされ、処理が決定ブロック1603で再開され、部分バンド・バッファ608のレベル3から回収されたデータで、上述したように工程1605から1613が続行される。

【0062】レベル3で部分バンド・バッファがまだ一杯にされていないければ、即ち、完全なセット（即ち、71×149係数からなるレベル3）が部分バンド・バッファ608になれば、偽が決定ブロック1613によって戻され、チェック（決定）ブロック1615に進む。チェック・ブロック1615が、部分バンド・バッファのレベル4において、（64÷7）行の71カラムの完全な係数のセットが存在するかどうかをチェックする。もしそうであれば、「レベル」変数が1616で4にセットされ、処理が決定ブロック1603で再開され、部分バンド・バッファ608のレベル4から回収されたデータで、上述したように工程1605から1615が続行される。

【0063】レベル4で部分バンド・バッファがまだ一杯にされておらず、従って、偽（NO）が決定ブロック

1615によって返答されると、チェック（決定）ブロック1617に進む。ブロック1617において、全画像が完全に処理されたかどうかチェックされる。この場合に、決定ブロック1617が真（YES）を返答し、流れ処理が終了する（1618）。そうでなければ、決定ブロック1617が偽（NO）を返答し、処理がブロック1602で再開され、「レベル」変数に1を割り当てて、上述したように工程1602から1617が再度、外部DRAM403から取り出されたデータに対して処理が続行される。

【0064】説明した実施の形態において、9本のソース・ラインが係数の第1ラインを発生するのに必要とし、一方、2本のソース・ラインが次のおよび係数の各続くラインを発生するのに必要となる。71画素（または係数）の最小71ラインが32係数の32ラインを形成する必要がある。部分バンド・バッファ608が、次のレベルを処理するために、係数の三つのタイルを最初にロードする。次のレベルにおいて発生されるべき全ての連続するタイルに対して、入力二つの付加的なタイルのみが必要である。これは部分バンド・バッファ608が既に前回のロード・データから少なくとも一つのタイルを含んでいるからである。本実施の形態において、71カラムの係数が必要であり、部分バンド・バッファにおいて、現在レベルのために次のレベルフィルタする必要がある。すなわち、フィルタ操作は前回処理されたタイルから4カラムの係数で開始され、二つの付加的（完全な）タイル（各々32カラム）と、71カラムのタイルを形成するのに処理されるべき次のタイルからの3カラムとを使用する。随意に、部分バンド・バッファ608は、図15に示したような単一のバッファ・メモリとしてバッファ608を設けるのではなく、分解の各レベルの一つに対する三つの分離メモリ・ユニットとして設けることができる。このオプションは分離した読取・書込アドレス・ジェネレータを必要とする。

【0065】デジタル画像を符号化する方法は、図17に示したような従来の汎用コンピュータ・システム1700を使用して実行でき、例えば、図2、3および5に関する処理が、コンピュータ・システム1700内に存在するアプリケーション・プログラムのようなソフトウェアとして備えることができる。特に、デジタル画像を符号化する方法中の工程は、コンピュータによって実行されるソフトウェア内の命令によって実行される。このソフトウェアは二つのパーツに分割することができ、その一つはエンコーディング方法を実行し、他の一つは後者とユーザ間のユーザ・インターフェイスを管理する。ソフトウェアはコンピュータ読取可能媒体内に記憶することができ、例えば次に説明する記憶デバイスを含んでいる。このソフトウェアは、コンピュータ読取可能媒体からコンピュータにロードされ、コンピュータによって実行される。この種のソフトウェア、またはこれに記憶

されたコンピュータ・プログラムを有するコンピュータ読取可能媒体は、コンピュータ・プログラム製品である。コンピュータ内のコンピュータ・プログラム製品を使用すると、本発明の実施の形態に基づくデジタル画像を符号化するための装置が好適に実行される。

【0066】コンピュータ・システム1700は、コンピュータ・モジュール1701、キーボード1702およびマウス1703のような入力デバイス、プリンタ1715および表示デバイス1714を含む出力デバイスを備えている。モデム1716が、例えば、電話回線1721または他の機能媒体を介して接続可能な通信ネットワーク1720間を相互連絡するコンピュータ・モジュール1701によって使用される。モデム1716はインターネット、および、ローカル・エリア・ネットワーク（LAN）またはワイド・エリア・ネットワーク（WAN）のような他のネットワーク・システムへのアクセスを得るのに使用することも可能である。

【0067】コンピュータ・モジュール1701は、一般的に、少なくとも一つのプロセッサ・ユニット1705と、例えば、半導体ランダム・アクセス・メモリ（RAM）とリード・オンリ・メモリ（ROM）から形成されたメモリ・ユニット1706と、ビデオ・インターフェイス1707を含む入力／出力（I/O）インターフェイスと、キーボード1702とマウス1703およびオプションとしてジョイスティック（図示省略）のためのI/Oインターフェイス1713と、モデム1716のためのインターフェイス1708とを備えている。記憶デバイス1709を備えており、一般的にハードディスク・ドライブ1710とフロッピー・ディスク1711を含んでいる。磁気テープ・ドライブ（図示省略）も使用できる。CD-ROMドライブ1712が、一般的にデータの不揮発性ソースとして備えられている。コンピュータ・モジュール1701の要素1705から1713は、一般的に相互連結バス1704を介して、当該技術において知られているコンピュータ・システム1700の従来操作モードとなる方法で連絡されている。実施の形態を実行できるコンピュータの例としては、IBM-PCおよび互換性のあるサン・スパークステーションまたはこれから発展されたこれと同様のコンピュータ・システムが含まれる。

【0068】一般的に、好適な実施の形態のアプリケーション・プログラムは、ハードディスク・ドライブ1710に存在し、プロセッサ1705によるその実行が読み取られるとともに制御される。プログラムの中間記憶およびネットワーク1720からくどのデータも、多分ハードディスク・ドライブ1710と提携して半導体メモリ1706を使用して達成される。ある例において、アプリケーション・プログラムは、ユーザに提供され、CD-ROMまたはフロッピー・ディスクに符号化

され、対応するドライブ1712または1711を介して読み取られるか、或いは別の方法として、モデム・デバイス1716を介してネットワーク1720からユーザによって読み取ることできる。さらに、ソフトウェアも他のコンピュータ読取可能媒体からコンピュータ・システム1700にロードできる。この媒体には磁気テープ、ROMまたは集積回路、磁気・光学ディスク、コンピュータ・モジュール1701と他のデバイス間の無線または赤外線伝送チャネルと、PCMCIAカードのようなコンピュータ読取可能カード、およびインターネットと、Eメール伝送とウェブサイトに記憶された情報を含むイントラネット等々を含む。上述のものはコンピュータ読取可能媒体の単なる例である。他のコンピュータ読取可能媒体も、本発明の範囲と概念から外れることなく実行できる。

【0069】デジタル画像を符号化する方法は、別の方法としてデジタル画像を符号化する機能、又はサブ・ファンクションを実行する一つまたはそれ以上の集積回路のような専用ハードウェア内で実行できるのが好ましい。この種の専用ハードウェアにはグラフィック・プロセッサ、デジタル・プロセッサ、あるいは一つのまたはそれ以上のマイクロプロセッサおよび関連するメモリを含む。

〔エントロピー符号化器(エンコーダ)〕デジタル画像のコード化表現を発生するためのエンコーダを次に説明する。

【0070】画像の変換係数のコード化表現を生成するための方法を説明する。特に、この方法はクワドツリー(4分木)を使用する埋め込みブロック・コーディングに関する。

【0071】この方法は画像データのウェーブレット変換により最初に処理される。ウェーブレット処理の概観を添付図面を参照して次に説明する。

【0072】図18を最初に参照して、元の画像1801が離散ウェーブレット変換(DWT)を使用して四つのサブ画像1803乃至1806に変換される。このサブ画像ないしサブバンドは、通常LL1、HL1、HL1およびHH1で示される。サブバンド名に付いている1の添え字はレベル1を示している。LL1サブバンドは、元の画像のロウパス・デシメート・変形である。

【0073】ここで使用されるウェーブレット変換は、例えばハール(Haar)基本ファンクション、Daubechies基本ファンクション等で変形でき、またこれを含めることができる。LL1サブバンドは次に使用され、第2離散ウェーブレット変換が図19に示したように適用され、サブバンドLL2(1908)、HL2(1909)、LH2(1910)、HH2(1911)で示される。この処理は例えば図20に示したように続行され、ここにはLL4サブバンドが示されている。明らかに、分解の第四レベルは入力画像のサイズに依存して提供でき

る。最も低い周波数のサブバンドはDCサブバンドに該当される。図20の場合には、DCサブバンドはLL4サブバンドである。

【0074】次に、各単一レベルDWTは、元の画像を得るように順次変換される。従って、J-レベルDWTは一連のJ単一レベルの逆DWTに逆変換できる。

【0075】階層的に画像をコード化するために、DCサブバンドが最初にコード化される。こうして、残りのサブバンドがレベルを小さくする順にコード化される。これは4レベルDWTに対してであり、レベル4のサブバンドはDCサブバンド(LL4)の後にコード化される。すなわち、HL4、LH4およびHH4サブバンドである。レベル3(HL3、LH3およびHH3)のサブバンドが次にコード化され、これに続いてレベル2(HL2、LH2およびHH2)が、次にレベル1(HL1、LH1およびHH1)がコード化される。

【0076】標準画像では、符号化化サブバンドは通常、画像中に「詳細」情報を含んでいる。量子化後のサブバンドは、しばしば値が希薄なアレイになり、実質的な圧縮がその希薄なマトリックス形態の有効なエンコーディングによって達成される。

【0077】ここで図21を参照して、HH1のようなサブバンドのタイリングを示す。このサブバンドは、左上のコーナーから始まる32×32ブロックの係数を有するタイル2110、2120、2130、2140および2150であるのが好ましい。32×32なる用語は、それぞれ32の行(row)、×32列(column)を意味している。

【0078】以下の実施の形態の説明を進める前に、以後に使用する技術用語につき簡単に説明する。2進整数を表す数に対して、「ビットn」または「ビット数n」は、2進数nに該当し、例えば、8ビットの2進表現を假定すると、10進数の9は、0001001として表される。この数において、ビット3は1に等しく、ビット2、1および0はそれぞれ0、0、1に等しい。さらに、画像の変換は、ビット・シーケンスによって表された各係数による横列と縦列に配列された係数を有するマトリックスとして表すことができる。結果的に、マトリックスについて話すと、三つのディメンジョン(次元)を有していると考えることができ、一番目のディメンジョンは横方向、二番目は縦方向、三番目はビットのシーケンス方向である。同じビット番号で各ビット・シーケンスを通っているこの三つのディメンジョン(3次元)空間内の平面は、「ビットプレーン」または「ビットプレーン」と呼ばれる。用語「ビットプレーン数n」は、ビット番号nを通るビットプレーンを意味する。

【0079】説明を簡潔にするために、変換係数は、以後、固定小数点で符号のない2進整数フォーマットに、付加的な単一サイン・ビットを付して表す。すなわち、10進数の-9および9は、同じビット・シーケンス、

すなわち、1001で表し、前者は負数を示す1に等しいサイン・ビットを有し、後者は正数を表す0に等しいサイン・ビットを有する。整数表示を使用することにおいて、係数は最も近い整数値に既に暗黙のうちに量子化されているが、これは必要ではない。さらに、圧縮の目的で、端数ビットに含まれるどの情報も通常無視される。

【0080】画像フレームの域は、一連の隣接する画像係数を含んでいる。用語「係数」は、以後「画素」と交換可能に使用されるが、当該技術に習熟した人にとってよく理解される点では、前者が変換ドメイン（例えば、DWTドメイン）中で画素という意味で使用されるのが

$$|c_{i,j}| < 2^{\pi}, \text{ 全ての } c_{i,j} \in T \text{ に対して} \quad (1)$$

座標の集合類別Tにより、次の部分集合Tのコレクション(Tm)を定める。

【0082】

【数1】

$$T = \bigcup_{m=1}^{\infty} T_m, \quad T_n \cap T_m = \emptyset \quad \forall n \neq m$$

【0083】換言すれば、一つの $c_{i,j} \in T$ 、従って $c_{i,j} \in T_m$ は、ただ一つの部分集合 T_m である。本件のTは正方块であり、集合 $\{T_m\}$ は、Tの4象限からなる集合（セット）である。

【0084】提案された方法は、4分木を使用して埋め込み方法で一連の係数を符号化する。用語「埋め込み」の使用は、より低いビットプレーンでいずれかのビットをコード化する前に、より高いビットプレーン中の全てのビットをコード化することを意味している。次に、ビットプレーン6のいずれかのビットがコード化される前に、ビットプレーン7の各ビットがコード化され、順番に、ビットプレーン5のいずれかのビットがコード化される前に、ビットプレーン6の全ビットがコード化されるという順にコード化される。

【0085】提案された方法の好適な実施の形態は、次の疑似コードを使用して実行される。この提案された方法は、係数の正方ブロックを符号化するのが好ましく、ブロックのサイズは2の累乗（一般的に 32×32 の係数）したものである。更に、この提案方法は4分木類別を使用している。すなわち、各セット（集合）或いは域は、その四つの象限に類別され、これによって、常時2の累乗に等しいディメンジョンで正方域が維持される。開始時に、提案方法は三つのリストを初期化する。すなわち、重要でない域（有効数字でない部分）（LIR）のリストと；重要でない係数のリスト（LIC）と；重要な（有効数字部分）係数のリスト（LSC）である。単一の係数が、係数の重要性に依存して、重要でないセットのリスト（LIR）から除去されると、これらのセットは、重要でない係数のリスト（LIC）か、重要な係数のリスト（LSC）のいずれかに付加される。

【0086】この提案方法は次のように初期化される。

一般的である。これらのセットないし域Tは、 (i, j) を係数座標としたとき、変換画像係数 $\{c_{i,j}\}$ を有するものとして規定され、固定小数点でサインのない2進整数フォーマットに付加的な単一サイン・ビットを付して表わされる。

【0081】現在ビットプレーンにおける画素のセット或いは域Tは、域中の各係数のMSBの数が現在のビットプレーンの値より小さい場合、重要ではないと言える。域の意味の概念を正確にするために、数学的規定が式(1)で与えられる。画素のセット或いは域Tは、次式であれば、ビットプレーンnに関して（またはこれにおいて）重要でないと言える。

LICおよびLSCは空になるように初期化される。LIRは入力ブロックの4象限を包含するセット（集合）である。この方法は n_{max} を見つけて、コード化することによって開始され、ここで n_{max} はセット内の全ての係数に対して1ビットを含む最大のビットプレーンである。従って、この提案方法は次のようにして進行される：

1. $n = n_{max}$ に設定
2. 重要でない係数のリスト（LIC）中の各係数に対し、
 - ・ 係数のコード・ビットn（すなわち、その重要度（有効数字））
 - ・ ビットが1（即ち、重要である場合）、サイン・ビットにコード化される。係数をLSCの最後に付加され、LICから係数が除去される。

【0087】3. 重要でない域のリスト（LIR）中の各域Tに対し、

- ・ 重要なT（シグニフィカンス）をコード化。

【0088】・ Tが重要（有効数）であり、一つより多い係数から構成され、従って、Tがその4象限に仕切られ、それらがLIRの最後に付加される。Tがリストから除去される。

【0089】・ Tが単一係数であれば、

- ・ TがLIRから除去される。

【0090】・ Tが有効数であれば、サイン・ビットがコード化され、TがLSCの最後に付加される。

【0091】・ 他に、TをLICのリストの最後に付加する。

【0092】4. 重要な係数LSC（ステップ3中のリストに付加されたものを除く）のリスト中の各係数 $c_{i,j}$ に対し、

- ・ $c_{i,j}$ のビットnをコード化、
- 5. nを小さくして、ステップ2に進む。

【0093】上述のことから、出力ビットストリームは普通、次の形態をとる、

... LIC' LIR' LSC'

ここで、LIR'はステップ3で実行されたコード化表

現であり、LIC' はステップ2で実行されたコード化表現であり、LSC' はステップ4で実行されたコード化表現である。しかし、エンコーディング工程の最初の繰り返し中、LICとLSCは空であり、従って、最初の繰り返しの出力ビットストリームはLIR' の形態をとることに注意しなければならない。

【0094】この提案した方法に加えて、LICおよびLSCをコード化するとき、簡単なHuffman コード（またはより良いのはGolombコード）をビットのグループ（例えば4ビットのグループ）がコード化するのに使用される。さらに、域の各象限の有効数をコード化するとき、15レベルのHuffman コードが4象限（1象限は重要でなければならず、従って、重要なパターンは15（16ではない）の内の一つとなる）のそれぞれの異なるパターンの重要なパターンを示すのに使用できる。エントロピー符号化の他の形態として、例えば2進演算コーディングなどを使用して、残りの冗長性を利用する。

【0095】別の実施の形態として、Tが2×2ブロックの係数からなる場合、ステップ3における提案された方法は、次のサブステップを形成する。2×2ブロックの各係数の重要度（意味）をコード化して出力すると同時に、これらが重要であれば、対応するサインが出力されるとともに、その係数が適切にLSCまたはLICの最後に付加される。後者のサブステップにおいて、重要でない係数がLICリストに付加される。

【0096】好ましくは、この提案した方法は32×32ブロックのデータ・係数を符号化する、説明の目的のみで、4×4ブロックの係数の次なる例が、この提案した方法に基づいて符号化される。

【0097】

【数2】

31	16	6	0
15	17	0	0
9	7	1	6
5	3	1	0

【0098】上述のブロックは四つの象限A、B、CおよびDからなる。シンボルAは、ブロックの左上（2×2）象限、Bは右上、Cは左下、また、Dは右下の象限を表す。さらに、シンボルA1はそれぞれAの左上の画素、A2は右上、A3は左下、A4は右下の画素を示している。同様に、B1はBの左上、というように画素の残りの部分にも適用される。

【0099】この提案方法によれば、 n_{max} が最初に決定されるが、この場合4である。すなわち、各係数の最も重要なビットはビットプレーン4またはそれ以下にある。注意しなければならないのは、ビットプレーンの番号付けは0から始まることである。変数 n_{max} は4ビットでコード化される（係数が圧迫されているので、 n_{max} は0から15の間となる）。最初に、LIC=φ、LIR={A、B、C、D} およびLSC=φであり、

ここにシンボルφは、空リストを示すのに使用される。

【0100】従って、この提案方法によれば、ビットプレーンは繰り返してコード化される。この処理はビットプレーン $n=n_{\text{max}}=4$ で開始され、減少数nは各繰り返しにつき1ずつである。

1. $n=n_{\text{max}}=4$ につき、

・ 最初に、リストLIC中の各係数がコード化される。何もないので、コード化は実行されない。

・ 次に、リストLIR中の各域の重要度がコード化される。

【0101】・域Aに対して、ビットプレーン $n=4$ において重要であるから、1ビットが出力される。次に、Aの4象限が、すなわち、A1、A2、A3およびA4がリストLIRの最後に付加され、Aが除去される。ここでLIR={B、C、D、A1、A2、A3、A4}である。

【0102】・域Bに対して、ビットプレーン $n=4$ において重要でないから、0ビットが出力される。

【0103】・域Cに対して、0ビットが出力される。

【0104】・域Dに対して、0ビットが出力される。

【0105】・域A1に対して、1ビットが出力される。A1が単一係数31からなるので、A1がLIRから除去される。31（またはA1）が重要であるから、これがLSCに（またはそのロケーションがブロックに）付加される。A1のサイン・ビットは（0）も出力される。

【0106】・域A2に対して、1ビットが出力される。A2が単一の重要な係数16からなるので、LIRから除去され、LSCの最後に付加される。A2のサイン・ビット（0）も出力される。ここでLSC={31、16}である。

【0107】・域A3に対して、0ビットが出力される。単一の重要でない係数であるから、LIRから除去され、係数15がLICに付加される。ここに、LIC={15}である。

【0108】・域A4に対して、1ビットが出力される。A4が単一の重要な係数17からなるので、LIRから除去され、LSCの最後に付加される。A4のサイン・ビット（0）も出力される。ここで、LSC={31、16、17}である。

・ 最終ステップにおいて付加されなかったLSC中の各係数がここでコード化される。何もないので、コード化は実行されない。

【0109】従って、第1繰り返しにおいて、この提案した方法が次のビットストリームを出力する。

10001010010

この段階において、ビットプレーン4（及びこれより高い）中の全てのビットがコード化される。符号化されたビットストリームからビットを読み出すことにより、復

号器がビットプレーン4（及びより高い）を再構築できる。このデコーディング方法は、重要度の判定がビットストリーム（これが、なぜ重要度の判定がビットストリームに書き込まれるかの理由である）からの読取りによって決定されることを除いて、同じである。他の係数・ビットがそのまま簡単に読み取られる。デコード実行パスは、エンコーダのものと同様であり、これによってデコードは、読み取られた新しい各ビットの意味がわかっていることに注意しなければならない。

2. $n=3$ につき、最初に、 $LIC=\{15\}$ 、 $LIR=\{B, C, D\}$ および $LSC=\{31, 16, 17\}$ である。

・ 第1に、 LIC 中の各係数のビット $n=3$ をコード化する。すなわち、1ビットが係数15と、サイン・ビット（0）に対して出力される。これは重要であるので（1ビットが出力される）、サイン・ビットが出力され、係数15が LIC から除去され、 LSC の最後に付加される。これによって、 $LSC=\{31, 16, 17, 15\}$ となる。

・ LIR 中の域の各々の重要度がここでコード化される。

【0110】・ 域Bに対して、0ビットが出力される。

【0111】・ 域Cに対して、ビットプレーン $n=3$ において重要であるから、1ビットが出力される。域Cが4象限C00、C01、C10およびC11に仕切られ、 LIR の最後に付加される。Cが LIR から除去される。従って、ここに $LIR=\{B, D, C00, C01, C10, C11\}$ である。

【0112】・ 域Dに対して、0ビットが出力される。

【0113】・ 位置C00に対して、1ビットが出力される。C00が単一の重要な係数9からなるので、 LIR から除去され、 LSC の最後に付加される。C00のサイン・ビット（0）も出力される。ここに $LSC=\{31, 16, 17, 15, 9\}$ を得る。

【0114】・ 域C01に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数7を LIC に付加する。ここで $LIC=\{7\}$ である。

【0115】・ 域C10に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数5を LIC に付加する。ここに $LIC=\{7, 5\}$ である。

【0116】・ 域C11に対して、0ビットが出力される。単一の重要でない係数であるから、 LIR からこれを除去し、係数3を LIC に付加する。ここに $LIC=\{7, 5, 3\}$ である。

・ ここで、 LSC の各係数のビット $n=3$ をコード化する（上記に単に付加するだけではない）。

【0117】・ 31、16および17のビット $n=3$ として、それぞれ1、0および0を出力する。

【0118】従って、二回目の繰返しにおいて、この提案した方法は次のビットストリームを出力する。

1001010000100

3. $n=2$ につき、最初に、 $LIC=\{7, 5, 3\}$ 、 $LIR=\{C, D\}$ および $LSC=\{31, 16, 17, 15, 9\}$ を得る。

・ 第1に、 LIC 中の各係数のビット $n=2$ （またはビットプレーン $n=2$ における重要度と等価である）をコード化する。すなわち、7、5および3それぞれに対して1、1および0を出力する。加うるに、7（0）および5（0）のためのサイン・ビットが出力され、これらの係数が LSC に移動される。 LIC 中に3が残る。

・ 次に、 LIR 中の各域の有効数が符号化される。

【0119】・ 域Bに対して、0ビットが出力され、また、域Dに対して0ビットが出力される。

・ 最後に、 LSC 中の係数の各々に対するビット $n=2$ を更新する（上記に付加せず）。

【0120】・ 31、16、17、15および9に対して、それぞれ1、0、0、1および0を出力する。

【0121】従って、三回目の繰返しにおいて、この提案した方法は次のビットストリームを出力する。

101000010010

ビットプレーン0、またはいくつかの他の有線点まで、この態様を続行する。特定の有線コードを使用した場合、（三つの）サブパスのどれか一つの後、終了することができることに注意しなければならない。（基本的には、FFは有線コードとしてリザーブされ、有線コードを慎重に挿入しなければ、コード化ビットストリームを決して強制的にFFに含ませることはない。

【0122】これまでに説明したように、この方法は係数の 32×32 ブロックをエンコーディングするのに好適に利用できる。これらの状況において、元の象限A、B、C、Dは各々 16×16 の係数からなり、また域A1、A2、...、D4は各々 8×8 の係数からなる。従って、 32×32 ブロックの符号化において、ブロックは4分木方法に基づいて5回仕切られるが、一方、与えられた例においては 4×4 ブロックが2回のみで仕切られる。

【0123】デコーディング工程は、エンコーディング工程を単に模倣してコード化された表現から画素を再構築する。

【0124】図22を参照して、この提案方法を実行するための第1の好ましい実施の形態に基づくエンコーダを説明する。係数・エンコード2200は、対応するデータ2202を得て、出力符号化データ2204の連続した流れを提供するように設計されている。このエンコード2200は、次のロジック部を含んでいる。すなわち、ビットプレーン・コンバート2206；ビットプレ

ーン・ツリー・ビルダー2208-0から2208-15；ビットプレーン・エンコード2210である。

【0125】エンコード2200は、ビットプレーンで32×32の係数を記憶するためのメモリ2214と；係数のサイン・ビットを記憶するためのメモリ2216と；ビットプレーン0から15を記憶するためのメモリ2218-15から2218-0も含んでいる。エンコード2200は、域の第1リストを記憶するためのメモリ2226と；域の第2リストを記憶するためのFIFOメモリ2228と；LIC値のリストを記憶するためのメモリ2222と；LIC値のリストを記憶するためのメモリ2224をさらに含んでいる。

【0126】エンコード2220は、次のようにして動作する。最初に、32×32入力係数・データがラスタ順にメモリ2212に記憶される。ビットプレーン・コンバータ2206は、入力係数・データを読み取り、このデータを、ビットプレーン0からビットプレーン15までの16のビットプレーンに変換し、このデータは続いてメモリ2214に記憶される。ビットプレーン・コンバータ2206はまた32×32ブロックの係数のサイン・ビットも判定し、このサイン・ビットがメモリ2216に記憶される。

【0127】ビットプレーン・ツリー・ビルダー2208-0から2208-15が各々32×32係数・データを読み取り、域に対応するノードと1×1の画素を有する4分木構造を構築する。各ツリーにおいて、ノードは、域またはこのノードに対応する画素がビットプレーンのために重要であれば、1にセットされる。そのノードに対応する域又は画素のための重要なビットが、ビットプレーンより大きいか、又は小さければ、ノードはゼロにセットされる。

【0128】次に図23を参照して、32×32ブロックのためのビットプレーンnにおけるビットプレーン・ツリービルダー2208-nによって構成された構築ツリー2300を説明する。簡潔にする目的で、ツリーの一部のみを示す。このツリー2300は、1×1画素レベルまで下げたブロックの各4分木の仕切りを表すノードを含んでいる。この場合において、全32×32ブロックがシンボル0によって表される。16×16の左上象限がノードAで表され、16×16右上象限がBで表され、16×16の左下象限がノードCで表され、16×16の右下象限がノードDで表される。ノードA1、A2、A3およびA4が、4分木のAの左上、右上、左下および右下の8×8の4分木の仕切りをそれぞれ表している。同様に、ノードA11、A12、A13およびA14は、象限A1の4分木の左上、右上、左下および右下の4×4分木の仕切りをそれぞれ表している。同様に、ノードA111、A112、A113およびA114は、象限A11の4分木の左上、右上、左下および右下の2×2の4分木の仕切りをそれぞれ表

している。最後に、ノードA1111、A1112、A1113およびA1114は、象限A111の4分木の左上、右上、左下および右下の1×1分木の仕切りをそれぞれ表している。ツリーの残りの部分（図示省略）は32×32ブロックの各1×1象限まで下がって同様の方法で表される。

【0129】ビットプレーン・ツリー・ビルダー2208-nが、象限順（例えば、A1111、A1112、A1113およびA1114）で係数を読み取ることによってビットプレーンnを逆からこの種のツリーを構築する。そのノードに対応する域又は画素のための重要なビットが、ビットプレーンより大きいか、又は小さければ、ノードはツリービルダーによってゼロにセットされる。こうして、ビットプレーン・ツリー・ビルダーは、次のフォーマットで各ノードのための重要な情報を出力する。

【0130】ABCD A1 A2 A3 A4 A11 A12 A13 A14 A21,..., D44 A111,..., D444,..., A1111, D4444
ビットプレーン・ツリー・ビルダー2208-0から2208-15の各々からの出力は、それぞれのビットプレーン・ツリー・メモリ2218-0から2218-15に記憶される。

【0131】ビットプレーン・エンコード2210が、ビットプレーン・ツリー・メモリの各々を読み取り、次にビットプレーン・ツリー・メモリ2218-15で開始する。ビットプレーン・エンコード2210が、ノードA、B、CおよびDに対応するビットプレーン・メモリ2218-15内に記憶された四つの重要なビットA、B、CおよびDを次に読み取ることによって処理が開始される。ビットプレーン・エンコードは、メモリ2226内の域の第1リスト中のこれらのノード{A、B、C、D}のリストを記憶する。ビットプレーン・エンコード2210が次の操作に進行する：

1、ビットプレーン・エンコード2210が、第1域リスト上の第1ノード（域）に対応するビットプレーン・ツリー内のビットを読み取る。

【0132】a、ビットが重要であれば、エンコードが2進の1を出力する。エンコードがFIFO2228上の第2域リスト内のノードのチルドレン（子供）を記憶し、第1域リスト2226からのノードを除去する。

【0133】b、ビットが重要でない場合、エンコードが2進の0を出力し、第1域リスト2226上のノードを維持する。

【0134】2、ビットプレーン・エンコード2210が、第1域リスト内の全てのノードが読み取られるまで、ステップ1を繰り返す。

【0135】3、ビットプレーン・エンコードが、FIFO2228上の第2域リスト内の第1ノードに対応するビットプレーン・ツリー中のビットを読み取る。

【0136】a、ビットが重要であり、そのノードに関

する子供がある（すなわち、ツリー内のそのノードの直下にノードがある）場合、エンコーダが2進のゼロを出力し、このノードを第1域リスト2226上のノードに入れる。

【0137】b. ビットが重要であり、そのノードに関する子供がなければ、エンコーダが2進のゼロを出力し、このノードが対応する画素に対するインデックスとしてLICリスト2224に記憶される。

【0138】c. ビットが重要であり、そのノードに関する子供がなければ、エンコーダが2進の1と対応するサイン・ビットを出力し、このノードが対応する画素に対するインデックスとしてLICリスト2222に記憶される。

【0139】d. ビットが重要であり、そのノードに関する子供があれば、エンコーダが2進の1を出力し、第2域リストからノードを除去する。さらに、ノードの子供を第2域リストに付加する。

【0140】e. 第2域が空きであれば、エンコーディング工程がそのビットプレーン・ツリーを完成させる。

【0141】ビットプレーン・エンコーダが残りのビットプレーン・ツリーに対して次にこの操作を繰り返す。現在のビットプレーン・ツリーへの操作の開始時における域の第1リストには、前回のビットプレーン・ツリーへの前回操作から残っている域が含まれている。これらの出力は出力ストリームのLIC部分に対応する。現在のビットプレーン・ツリーに対する現在の操作の完了後、ビットプレーン・エンコーダがLICとLSCビットを符号化し、出力する。

【0142】ビットプレーン・エンコーダが、リスト上の第1画素に対するインデックスのためのLICリストを読み取り、ビットプレーン・メモリ2214から必要とされるビットを選択する現在ビットプレーン数を使用することによってLICビットを符号化する。選択されたビットが2進の0であれば、エンコーダがゼロを出力する。選択されたビットが2進の1であれば、エンコーダが2進の1を、メモリ2216からの画素のサイン・ビットとともに出力する。こうしてエンコーダはLICからインデックスを除去し、これをLSCリストに付加する。一度、インデックスがリストから除去されると、残りのインデックスは再編成されるのが好ましい。エンコーディング（符号化）は、一度、LICが通過すると、完了する。

【0143】ビットプレーン・エンコーダが、リスト上の第1画素に対するインデックスのためのLICリストを読み取り、ビットプレーン・メモリ2214から必要とされるビットを選択する現在ビットプレーン数を使用することによってLICビットを符号化する。こうして、選択されたビットが出力される。ビット・プレーン・エンコーダは、読み取られるべきLSCリスト中の画素数を示す長さ値を記憶するためのカウンタも含んでい

る。LSC符号化の終了後、この値が更新され、LICとLICからの新しい要素を付加することができる。

【0144】この提案方法のエンコーディング（符号化）とデコーディング（復号）工程は、図24に示したような従来の汎用コンピュータを使用して実行されるのが好ましく、その工程がこのコンピュータを実行するソフトウェアとして備えることができる。特に、コーディング及び（又は）デコーディング方法のステップは、コンピュータによって実行されるソフトウェア内の命令によって実行される。ソフトウェアは二つの別々のパーツに分かれており、その一つがデコーディング及び（又は）コーディング方法を実行し、また他方の一つが後者とユーザ間のユーザ・インターフェイスを制御する。ソフトウェアは、例えば次に説明する記憶デバイスを含むコンピュータにより読取可能な記憶媒体内に記憶できる。このソフトウェアはコンピュータ読取可能な記憶媒体からコンピュータ内にロードされ、コンピュータによって実行される。この種のソフトウェア、又はこれに記録されたコンピュータ・プログラムを有するコンピュータにより読取可能な記憶媒体は、コンピュータ・プログラム製品である。コンピュータ内でこのコンピュータ・プログラム製品の使用は、本発明によるデジタル画像をエンコーディング（符号化）し、また、デジタル画像のコード化された表現をデコーディングするための有利な装置を好適に実行する。

【0145】コンピュータ・システム2400は、コンピュータ2401と、ビデオ表示装置2416と、入力デバイス2402、2403を備えている。加うるに、コンピュータ・システム2400は、通信インターフェイス2421を介して、一つまたはそれ以上の他のコンピュータに、モデム通信バス、コンピュータ・ネットワーク等々のような適切な通信チャネル2422を使用して接続することができる。コンピュータ・ネットワークはローカル・エリア・ネットワーク（LAN）、ワイド・エリア・ネットワーク（WAN）、イントラネットおよび（または）インターネットを含めることもできる。

【0146】コンピュータ2401自体は、中央処理装置（以後単にプロセッサという）2405と、ランダム・アクセス・メモリ（RAM）とリード・オンリー・メモリ（ROM）とを含むメモリ2406と、入力／出力（I/O）インターフェイス2408と、ビデオ・インターフェイス2407と、図24のブロック2409で概略的に示された一つまたはそれ以上の記憶デバイスとを備えている。記憶デバイス2409は、次の内の一つまたはそれ以上を含んでいる。すなわち、フロッピー（登録商標）・ディスク、ハードディスク・ドライブ、磁気光学ディスク・ドライブ、CD-ROM、磁気テープまたは当該技術において周知の他の多数の不揮発性記憶デバイスである。要素2405から2409、2421はそれぞれ一般的に、データ、アドレスおよび制御

バスからなるバス2414を介して一つ、またはそれ以上の他のデバイスに接続される。

【0147】ビデオ・インターフェイス2407がビデオ表示装置2416に接続され、コンピュータ2401からのビデオ信号を提供してこのビデオ表示装置2416に表示する。コンピュータ2401を作動させるユーザ入力、一つ又はそれ以上の入力デバイス2402、2403によって提供できる。例えば、オペレータはキーボード2402および(または)マウス2403のような指示デバイスを使用することができる。

【0148】システム2400は図示の目的で簡潔に示されているが、他の形態も使用できる。実施の形態を実行できる例としてのコンピュータには、IBM-PC/ATまたは互換機を含み、その一つにPCのマッキントッシュ(商標名)ファミリ、サン・スパークステーション(商標名)等々がある。上述したものは実施の形態を実行できるコンピュータのタイプの単なる一例である。一般的には、以後に説明する実施の形態の処理は、プロセッサ2405を使用する読取りおよび制御されたコンピュータにより読取可能な記憶媒体としてハードディスク・ドライブ(図24において通常、ブロック2409で示される)に記憶されたソフトウェアまたはプログラムとして内在している。プログラムと画素データおよびネットワークから取り入れられたデータの間記憶は、半導体メモリ2406を使用して、ハード・ディスク・ドライブ2409と提携して達成することができる。

【0149】ある例において、プログラムは、CD-ROMまたはフロッピー・ディスク(両者ともブロック2423で概略的に示す)上に符号化されユーザに提供されるか、または別の方法として例えば、コンピュータに接続されたモデム・デバイスを介してネットワークからユーザが読み取ることができる。さらに、ソフトウェアは、他のコンピュータ読取可能媒体からコンピュータ・システム2400にロードすることもできる。この媒体には磁気テープ、ROMまたは集積回路、磁気-光学ディスク、コンピュータと他のデバイス間の無線または赤外線伝送チャネル、PCMCIAカードのようなコンピュータ読取可能カード、およびEメール通信を含むインターネットおよびイントラネットと、ウェブサイト等に記憶された情報を含む。上述したものは適切なコンピュータ読取可能媒体の単なる例である。他のコンピュータ読取可能媒体も実行可能である。

【0150】エントロピー符号化器から転じて符号化方法及び装置に説明を戻したが、逆に、符号化された画像を復号により再生しても良い。この場合、画像はまず最初に完全にエントロピー復号され、その復号された画像に、前述の符号化方法の略逆の処理が適用される。

【0151】上述の説明から明白なように、本発明の実施の形態はコンピュータおよび画像/データ処理産業に適用可能である。

【0152】これまでに説明したものは、本発明のある実施の形態のみで、修正および(または)変更も本発明の範囲と概念から外れることなく可能であり、実施の形態は例であって、限定ではない。

【0153】

【発明の効果】以上説明したように本発明によれば、例えば少ないバッファの容量で、複数レベルのDWTを効率的に行うことができるという効果がある。

【図面の簡単な説明】

【図1】画像の4レベルの離散ウェーブレット変換の種々のサブバンドを示す図である。

【図2】本発明の好適な実施の形態の変形例としての画像の水平タイリングを示す図である。

【図3】本発明の好適な実施の形態に係る画像の垂直タイリングを示す図である。

【図4】本発明の好適な実施の形態の符号化器を示す図である。

【図5】図3のタイリングと好適なタイル走査順を示す図である。

【図6】図4の圧縮エンジンをより詳しく示す図である。

【図7】本発明の好適な実施の形態に係るリオーダ・バッファを示す図である。

【図8】図7のリオーダ・バッファの充填順を示す図である。

【図9】本発明の好適な実施の形態に係る垂直フィルタのハードウェア構成を示す図である。

【図10】本発明の好適な実施の形態に係る水平フィルタのハードウェア構成を示す図である。

【図11】本発明の好適な実施の形態に係るサブバンド・バッファのハードウェア構成を示す図である。

【図12】図11のサブバンド・バッファの充填順および走査順を示す図である。

【図13】本発明の好適な実施の形態に係る制御状態マシンのハードウェア構成を示す図である。

【図14】本発明の好適な実施の形態に係る部分バンド・バッファのハードウェア構成を示す図である。

【図15】図14の部分バンド・バッファの充填順および走査順を示す図である。

【図16】図6の制御状態マシン・モジュールによる処理の流れを示すフローチャートである。

【図17】本発明の好適な実施の形態を実行する汎用コンピュータの概略ブロック図である。

【図18】元の画像とそのDWT変換を示す図である。

【図19】図18の元の画像の第2レベルDWT変換を示す図である。

【図20】図18の元になる画像の4レベルDWT変換を示す図である。

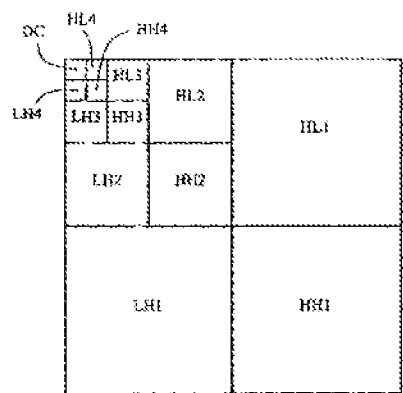
【図21】32×32ブロックにサブバンドをタイリングする図である。

【図22】符号化器の実施の形態を示す図である。

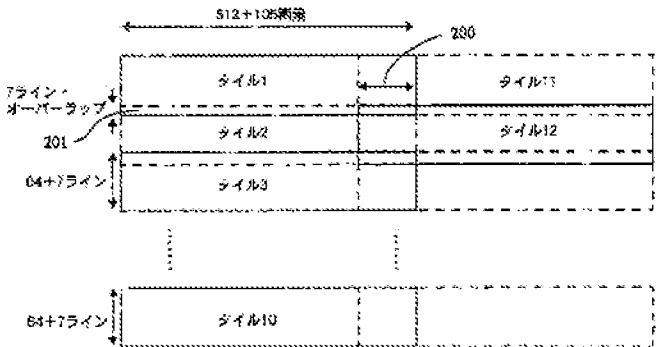
【図23】図22のビットプレーン・ツリー・ビルダーによって構築されたツリーの一部を示す図である。

【図24】説明した符号化方法を実行するための汎用コンピュータの構成を示すブロック図である。

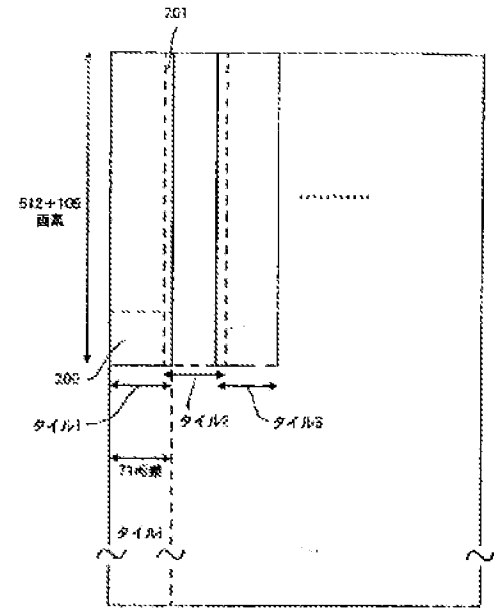
【図1】



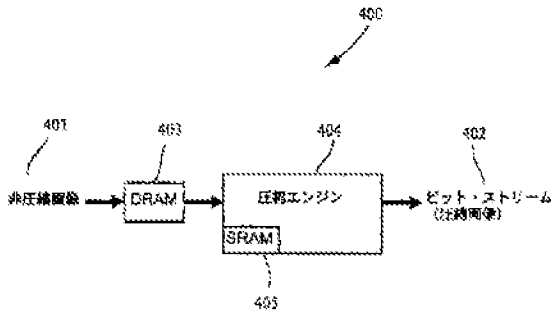
【図2】



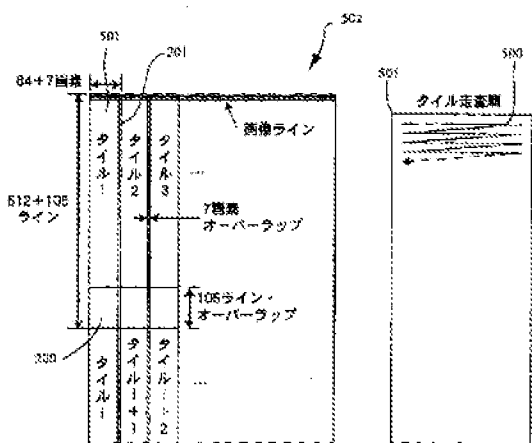
【図3】



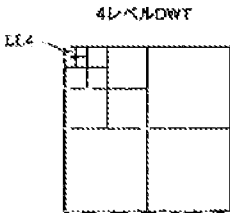
【図4】



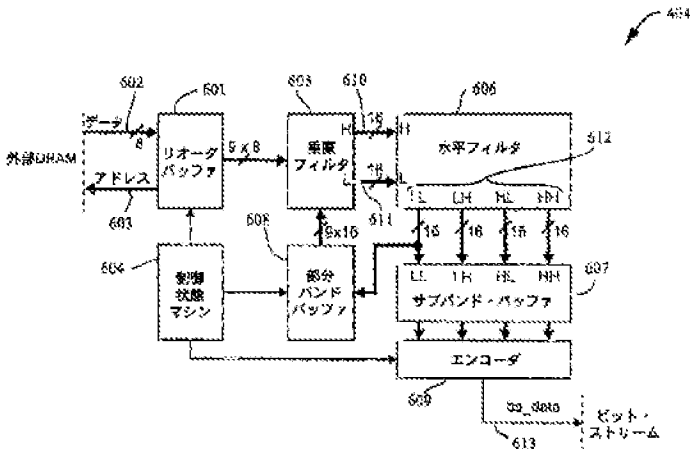
【図5】



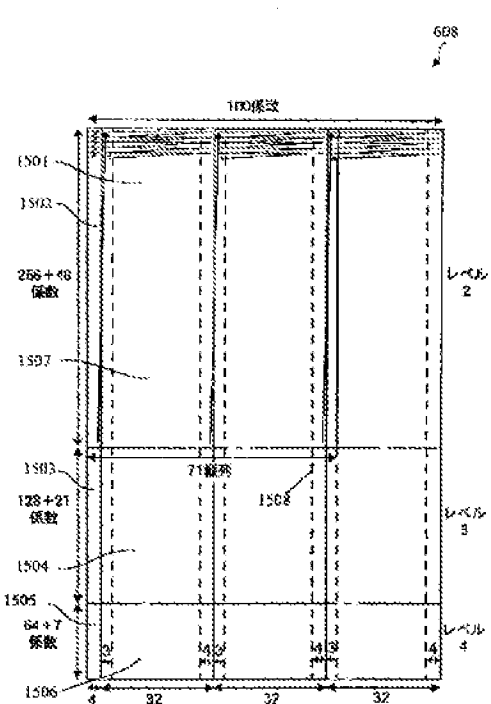
【図20】



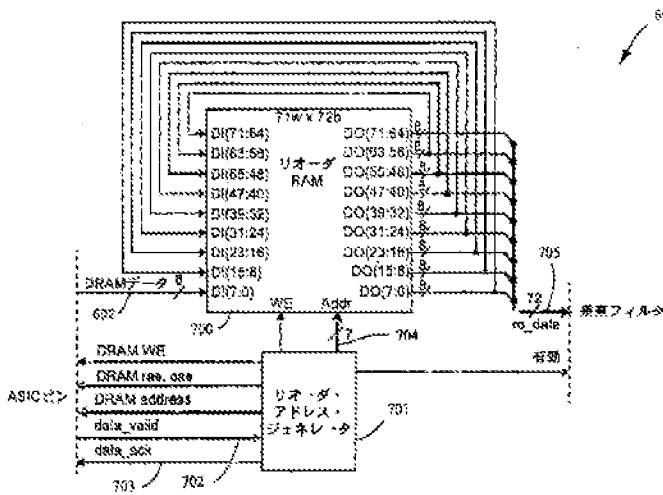
【図6】



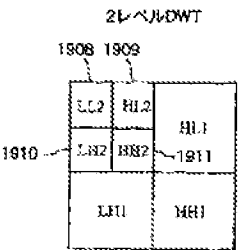
【図15】



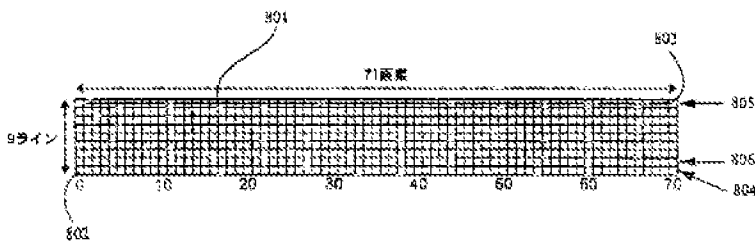
【図7】



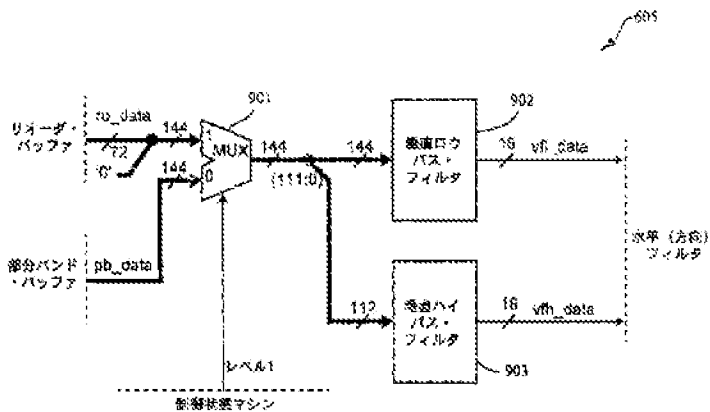
【図19】



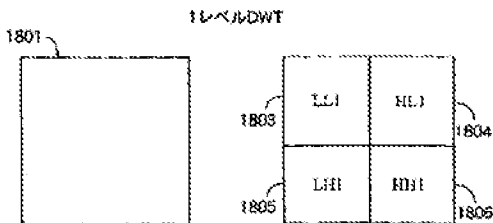
【図8】



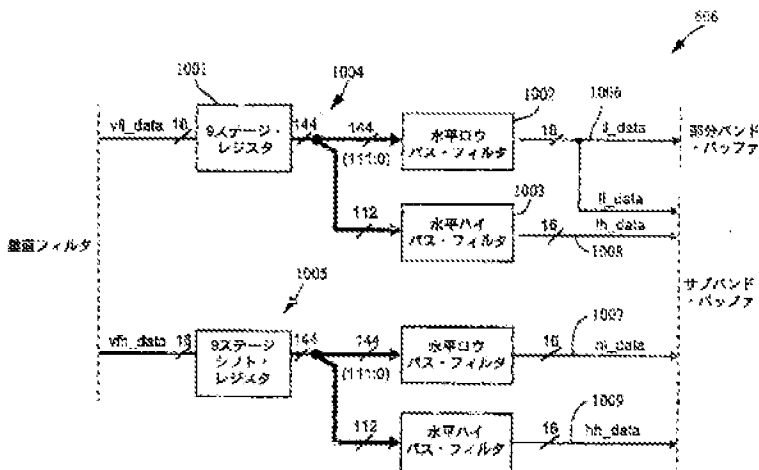
【図9】



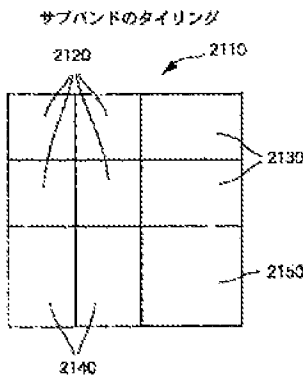
【図18】



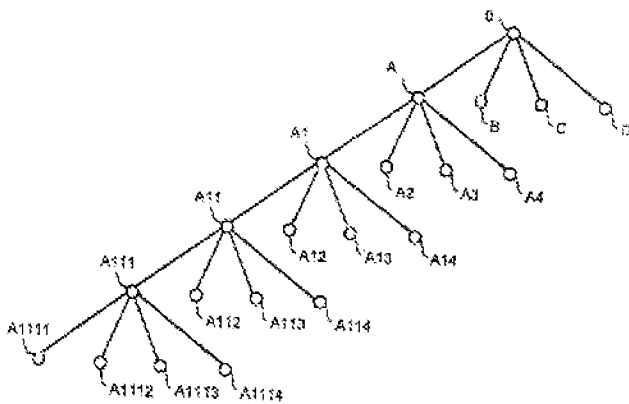
【図10】



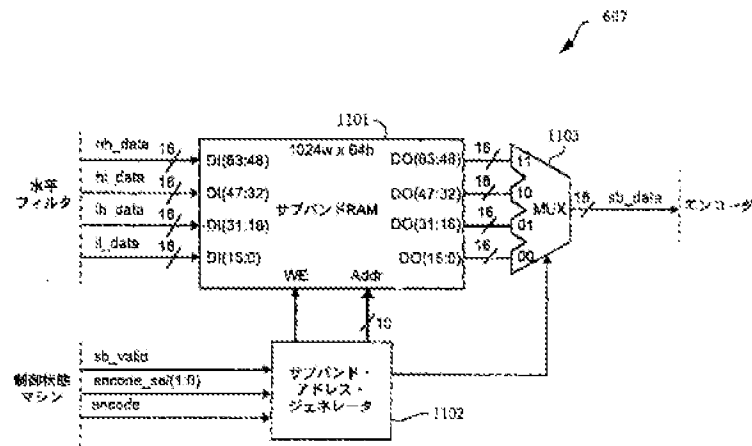
【図21】



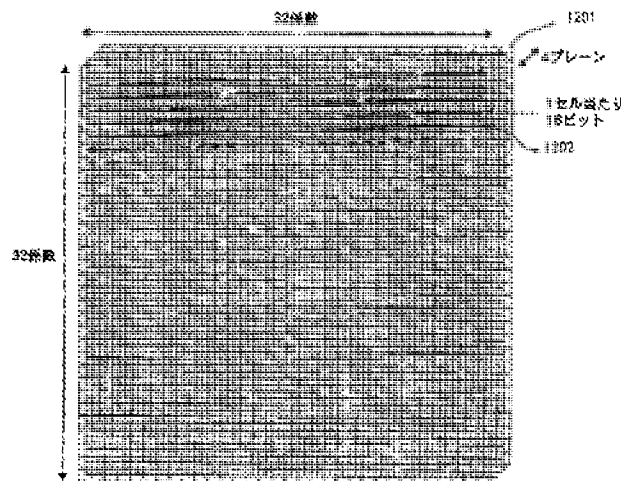
【図23】



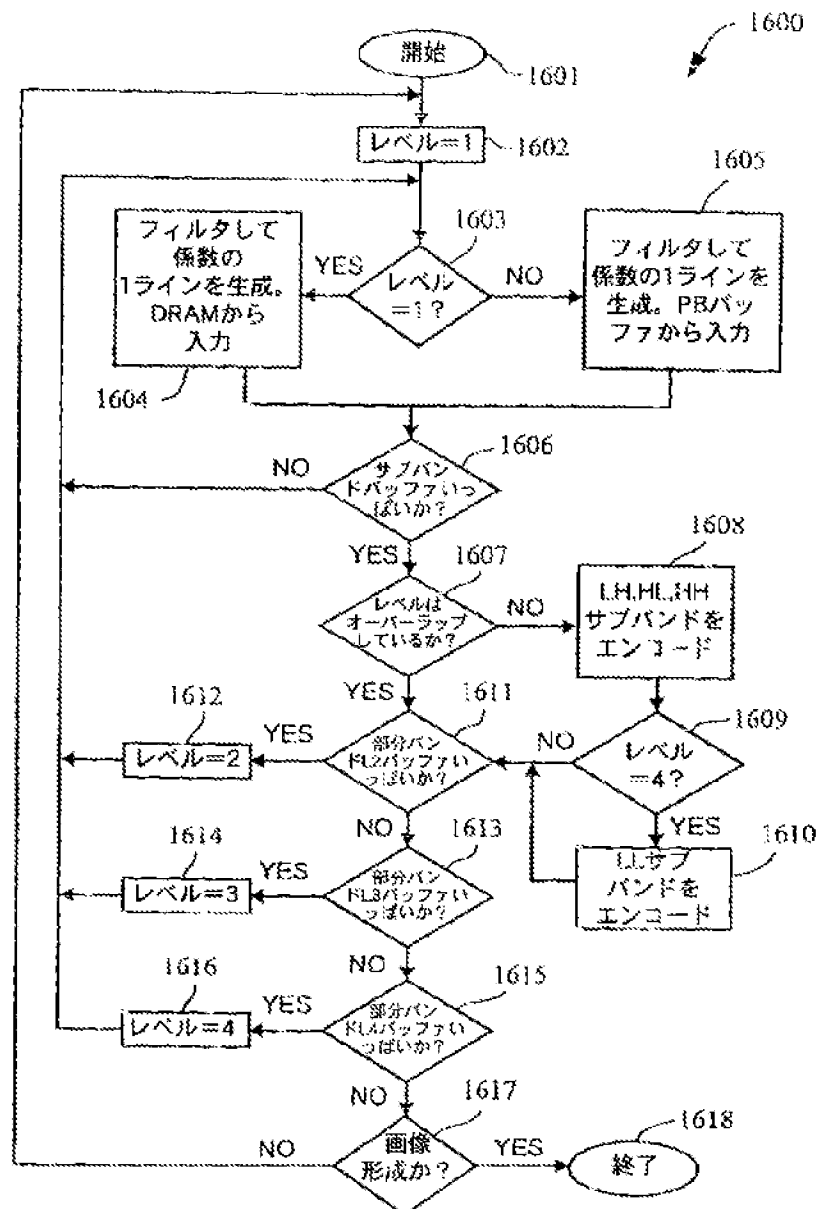
【図11】



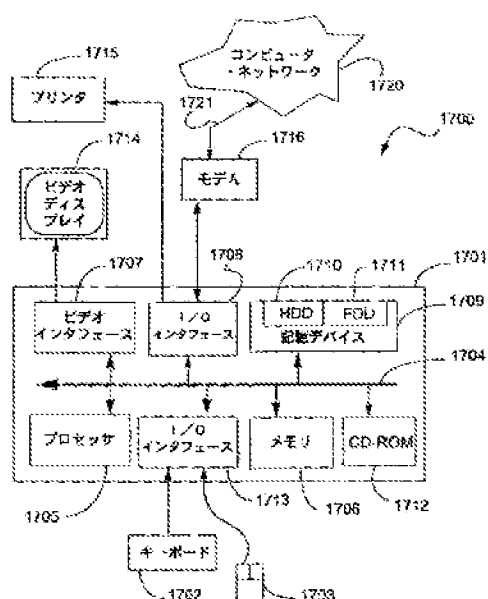
【図12】



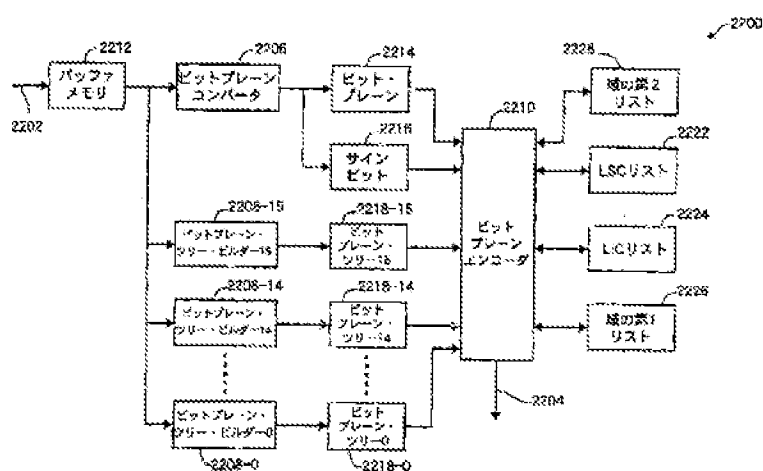
【図16】



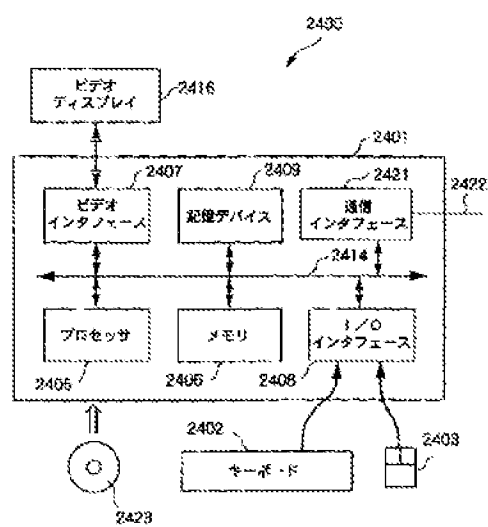
【例 17】



【例 22】



【図24】



フロントページの続き

(72)発明者 エリック マヤーニ
フランス国 レンヌーアタラント、セデッ
クス セッソニーセヴィニエ 35517、リ
ュ ドゥ ラ トラッシューランベール
キヤノン リサーチ センター フランス
エス、エー、内

(72)発明者 ジェームス フィリップ アンドリュ
ー オーストラリア国 2113 ニュー サウス
ウェールズ州、ノース ライド、トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド内

CONTRIBUTIONS TO INTERFRAME CODING

Marcos Faúndez-Zanuy, Francesc Vallverdú-Bayes, Francesc Tarrés-Ruiz.

Signal Theory & Communications Department.ETSETB. UPC.

C/ Gran Capitá ,s/n mòdul D5

08034 BARCELONA (SPAIN)

tel.:93-4016440

fax:93-4016447

e-mail:marcos@tsc.upc.es

ABSTRACT

Advanced motion models (4 or 6 parameters) are needed for a good representation of the motion experimented by the different objects contained in a sequence of images. If the image is split in very small blocks, then an accurate description of complex movements can be achieved with only 2 parameters. This alternative implies a large set of vectors per image.

We propose a new approach to reduce the number of vectors, using different block sizes as a function of the local characteristics of the image, without increasing the error accepted with the smallest blocks.

A second algorithm is proposed for an inter/intraframe coder.

I. INTRODUCTION

This paper describes complementary techniques for interframe coding, with the goal of reducing the total amount of information.

In many real scenes, the motion vector information is the same in big areas of the image, thus an efficient coding implies a variable block size. One possibility, is to adopt a variable block shape, segmenting the image in homogeneous parts, which has several problems:

1. Often it is difficult to segment the image accurately.
2. Segmentation implies large computational complexity.
3. The block shape coding requires a lot of bits and often, the transmission ratio is increased.

Another solution [3] implies the calculation of motion vectors with small and big size, and the selection of the big size when the error is similar to the accumulated error of the equivalent small blocks.

We propose the use of variable block size with a predetermined shape. The minimum and maximum block size is also fixed and studied in our paper.

II. PARAMETERS SELECTION

In our study, we have selected the following parameters:

- * error criterion: mean absolute frame difference (MAD).
- * motion model: two parameters (d_x, d_y), although this method is easily extensible to more complicated models.

- * motion estimation algorithm: modification of the conjugate direction algorithm [4], with maximum displacement allowed fixed to 7.
- * prediction process: writing method proposed in [4]. The motion vectors show the address in the image I_k where must be written the block of the image I_{k-1} .
- * lower block size: we have studied the cases 8x8 and 16x16.

The justification of the different parameters selected can be found in [4].

III. VECTOR POSTPROCESSING

The motion vectors obtained with the motion estimation algorithm, are coded with a quadtree structure (fig. 1) which is used to provide some important advantages:

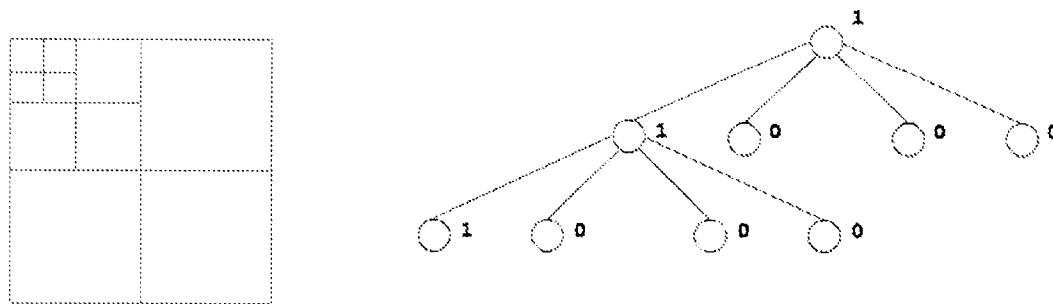


fig. 1 quadtree and codification for sizes from 8x8 to 64x64.

1. It is very simple to encode and decode.
2. It requires a few bits on its coding.
3. The compression ratio is increased about 3 times for a videoconferencing scenes.

This structure has been successfully used in different references ([1], [2]) and applications.

The quadtree structure is used to efficiently encode the motion information obtained with a modification of the conjugate direction method proposed in [4].

III.1 TREE CONSTRUCTION.

The block size selection is parallel to the tree construction. There are two ways to construct the tree:

- a) top-down: from the big size, we split it into four squares if the vectors contained are not identical. Every square is treated in the same way until the minimum size is reached or the square is considered homogeneous.
- b) bottom-up: from the smallest size, four adjacent blocks are grouped if the four vectors are identical. The process is repeated until the maximum size is reached or it is not possible the clustering.

We have selected the second alternative. If the clustering is realized only when the four vectors are identical, both methods obtain the same result.

Computational time, would be different. If the image contains big areas with identical vectors, the first alternative would be selected. If not, the second alternative is more suitable.

The compression ratio can be increased if the condition of equality between vectors is relaxed to "nearly equal" vectors. This alternative will increase the error in the postprocessing stage, that

otherwise is not affected. Sometimes, it will reduce the error, because it is similar to a smoothing function of the vectors, that can remove erroneous motion estimations.

IV. INTER/INTRAFRAME CODING.

For an inter/intraframe predictor, we propose the following structure in order to encode jointly the block shape and the inter/intraframe decision:

1. To implement the quadtree coding proposed in the last sections, grouping blocks coded with the same predictor.
2. Last level of the tree is used to indicate what kind of predictor has been used (inter or intra). It is interesting to observe that this last level was not coded in the original quadtree structure, because it indicates no division, and this is already known from the number of levels used. Also, in the terminal nodes different of the last level, two bits are needed. One bit indicates terminal node, and the other the inter/intraframe decision. For instance, if a 0 represents a terminal node, one alternative would be:
00: terminal interframe node. 01: terminal intraframe node.

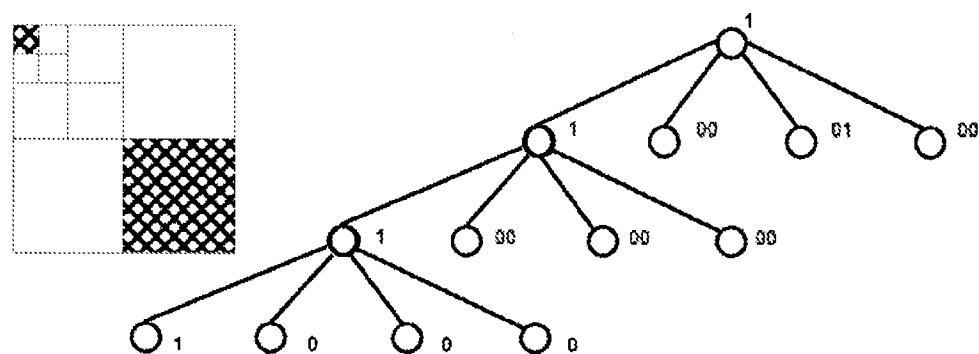


fig. 2 modified quadtree for inter/intraframe predictor. Last level and second bit of terminal nodes codifies the inter/intraframe decision. Dashed squares are intraframe coded zones.

Table 1 summarizes the theoretical compressions obtained with the proposed algorithms for the interframe and the inter/intraframe cases.

CASES	INTERFRAME (N=128)		INTER/INTRAFRAME	
	WITHOUT FLAG	WITH FLAG	WITHOUT FLAG	WITH FLAG
BETTER	100:7 (N:9)	100:8 (N:10)	100:12 (16:2)	100:19 (16:3)
WORST	100:104 (N:133)	100:86 (N:110)	100:131 (16:21)	100:94 (16:15)

table 1

For the case of inter/intraframe predictor we have only considered the reduction in the overhead

information (bit for inter/intra decision), because the total compression depends on the intraframe predictor used.

Worst case situation is evaluated when it is still useful the modified quadtree structure. A flag is also used to indicate if the quadtree codification is selected or not (in order to prevent the cases when the quadtree does not imply a reduction).

For the inter/intraframe predictor, the error obtained with the intraframe predictor can be penalized (P) to increase the compression ratio, because the compression is bigger with the interframe method (0.03 bits per pixel in the case of block size 16x16 pixels). This penalization would be adaptive as function of the contiguous blocks, increasing it when the cluster is dependent of the considered inter/intra decisor.

$$\text{condition of selection: } \text{error}_{\text{interframe}} < P \cdot \text{error}_{\text{intraframe}} \quad (P > 1)$$

V. RESULTS

A sample of the obtained results is the following:

For a 16x16 pels block size, the results obtained for the two first images of the secretary sequence are summarized in table 2.

The number of bits needed for quadtree coding is:

$$16(\text{roots}) + 8 \cdot 4(16^{\text{leaves}}) = 48 \text{ bits}$$

which added to the number of bits needed for coding the vector information results:

$$\frac{48}{8} + 8 + 15 + 68 = 97 \text{ bytes}$$

instead of the 256 bytes required originally. This implies a reduction of 100:37.9

BLOCK SIZE	Nº BLOCKS	EQ. SUBIMAGES
16x16	68	68
32X32	15	60
64X64	8	128
		256

Table 2.

From table 2 it is deduced that in this case half of the image is coded with a block size of 64x64 (8 blocks which represent 128 blocks of 16x16 size).

If the motion estimation is realized with blocks of 8x8 on the same images, 268 bits are needed for the tree coding, and 514 bytes for the vectors. Thus, the reduction is 100:53 (table 3).

BLOCK SIZE	Nº BLOCKS	EQ. SUBIMAGES
8x8	412	412
16x16	89	356
32X32	12	192
64X64	1	64
		1024

table 3.

Lower block size can be reduced to 4x4, but the modelation of the movement is not considerably better to the case 8x8 [5], and it is more difficult to group vectors, so it is not recommended to reduce the block size more than 8x8.

VI. OTHER POSSIBILITIES

These algorithms can be implemented with other tree structures, that let more flexibility in the block shape (for instance, rectangles), but more flexibility implies more bits for tree coding.

Other compression methods based on blocks, can utilize this structure to reduce the interblock redundancy, with a small adaptation. Interesting results have been obtained in the case of the block truncation coding [6], where the quadtree structure has been applied to postprocessing the information that must be encoded.

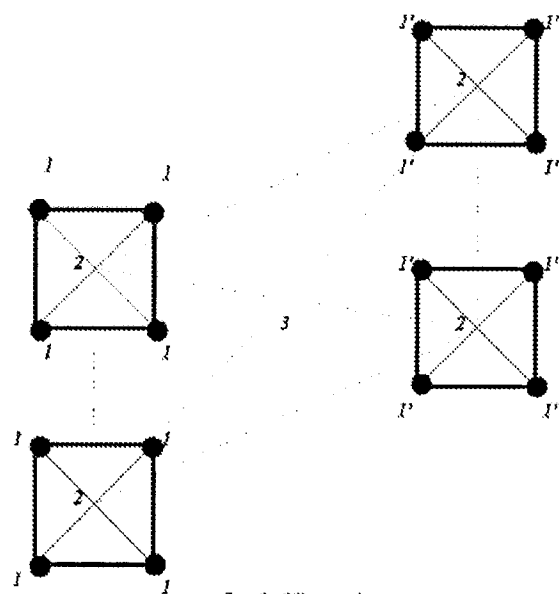


fig. 3 3D quadtree

M. Faúndez-Zanuy, F. Vallverdú, Bayes, F. Tarrés-Ruiz "Contributions to interframe coding". Workshop on image analysis & synthesis in image coding. October 1994. Berlin. pp. C3.1 to C3.6

The quadtree can be implemented in a 3D approach (see fig. 3), since vectors at the same location but different images (I_k and I_{k-1}) are highly correlated. This assumption is correct if the objects have a constant motion along several frames. In this case, in the first stage 16 vectors are clustered into 4 vectors (vectors 1 of image I_{k-1} , and 1' of I_k). In the next step, the four vectors (referred as 2) are clustered into one vector (named 3).

Tree construction is similar to the process explained in section III.1, although the vectors belong to different images. Other tree structures or clustering strategies can be applied in a similar way. This method can also be applied to other coding methods that work with image sequences.

VII. REFERENCES

- [1] J. Vaisey, A. Gersho "Image compression with variable block size segmentation" IEEE Transactions on Signal Processing Vol. 40 n.8, August 1992. pp. 2040-2060.
- [2] P. Strobach "Motion compensated regular decomposition coding of HDTV signals". Proceedings of the second international workshop on Signal Processing of HDTV. L'Aquila 1988, pp. 223-230. Elsevier publishers.
- [3] M. Kerdranvant "Hierarchical motion estimation & motion information encoding". Proceedings of the 3 int. Workshop on Signal Processing of HDTV II, 1990 pp. 101-108. Elsevier publishers.
- [4] M. Faúndez-Zanuy, F. Tarrés." A modification of the conjugate direction method for motion estimation" Proceedings EUSIPCO-92 Brussels, pp.1311-1314. Elsevier publishers.
- [5] M. Faúndez-Zanuy "Métodos predictivos y de compensación de movimiento en secuencias de imágenes". PFC ETSETB (UPC) July 1993.
- [6] X. Domingo "BTC per a compressió d'imatges". TFC EUETT La Salle. Oct. 1994



Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11
1st Meeting: Dresden, DE, 15-23 April, 2010

Document: JCTVC-A124

Title: Samsung's Response to the Call for Proposals on Video Compression Technology

Status: Input Document to JCT-VC

Purpose: Proposal

Author(s) or Contact(s): Ken McCann, ken@zetacast.com, +44 1962 625347
Woo-Jin Han, wjhan.han@samsung.com, +82 31 279 8831
Il-Koo Kim, ilkoo.kim@samsung.com
Jung-Hye Min, jh643.min@samsung.com
Elena Alshina, elena_a.alshina@samsung.com
Alexander Alshin, alexander_b.alshin@samsung.com
Tammy Lee, tammy.lee@samsung.com
Jianle Chen, jianle.chen@samsung.com
Vadim Seregin, v.seregin@samsung.com
Sunil Lee, sunil.lee@samsung.com
Yoon-Mi Hong, yoonmi.hong@samsung.com
Min-Su Cheon, minsu.cheon@samsung.com
Nikolay Shlyakhov, n.shlyakhov@samsung.com

Source: Samsung Electronics Co., Ltd. and British Broadcasting Corporation

Abstract

This proposal is Samsung's response to the Call for Proposals (CfP) on video compression technology, jointly issued by ITU-T SG16 Q.6 (VCEG) and ISO/IEC JTC1/SC29/WG11 (MPEG). It has been produced in collaboration with the British Broadcasting Corporation. The goal of this proposal is to provide a video compression technology which has significantly higher compression capability than the state-of-the-art H.264/AVC standard, especially for high-definition (HD) video contents. To achieve this goal, a number of new algorithmic tools are proposed covering several aspects of video compression technology. These include a general structure for representation of video content, inter/intra prediction, in-loop filtering, and entropy coding. When all the proposed algorithmic tools are used, the proposed video codec achieves approximately 40% bit-saving on average compared to H.264/AVC in both constraint set 1 and 2 configurations. The average decoding time for the proposed codec was measured to be between about 0.9 and 2.4 times that of JM17.0, depending on the computer HDD configuration.

Contents

Abstract.....	1
Glossary of terms used in this proposal	4
1 Introduction	5
2 Algorithm description	5
2.1 Overview	5
2.2 Unit definition.....	7
2.2.1 Coding unit (CU)	7
2.2.2 Prediction unit (PU)	8
2.2.3 Transform unit (TU).....	9
2.2.4 Relationship between CU, PU and TU.....	10
2.2.5 Size-independent syntax representation	10
2.3 Motion representation	11
2.3.1 Asymmetric motion partition (AMP).....	11
2.3.2 Advanced motion vector prediction (AMVP)	11
2.3.3 Improved skip and direct mode.....	12
2.3.4 DCT-based interpolation filter (DIP)	12
2.3.4.1 DCT basis decomposition	12
2.3.4.2 Example of filter coefficients.....	13
2.3.4.3 Low complexity optimization	14
2.3.5 High accuracy motion (HAM)	14
2.4 Intra-frame prediction	15
2.4.1 Arbitrary directional intra (ADI).....	16
2.4.2 Multi-parameter intra (MPI)	18
2.4.3 Color Component correlation based prediction (CCCP).....	19
2.4.4 Pixel based template matching (PTM)	20
2.5 Spatial transforms	20
2.5.1 Large transform (16x16, 32x32 and 64x64).....	20
2.5.2 Rotational transform (ROT).....	21
2.5.3 Logical transform (LOT)	22
2.5.4 Quantization	23
2.6 In-loop filtering.....	23
2.6.1 Deblocking filter	23
2.6.2 CU-synchronized adaptive loop filter (ALF)	23
2.6.3 Extreme correction (EXC) and band correction (BDC)	24
2.6.4 Content adaptive dynamic range (CADR)	24
2.7 Entropy coding.....	25
2.7.1 Syntax-based context-adaptive binary arithmetic coding (SBAC).....	25
2.7.2 Adaptive coefficient scanning (ACS)	26
3 Compression performance discussion	27
3.1 Objective versus subjective compression performance	27
3.2 Constraint set 1 configuration relative to Alpha anchor	27
3.3 Constraint set 2 configuration relative to Beta and Gamma anchors	28
4 Complexity analysis	30
4.1 Encoding time and measurement methodology	30
4.2 Decoding time and measurement methodology	31
4.3 Expected memory usage of encoder	33
4.4 Expected memory usage of decoder	33
4.5 Complexity characteristics of encoder motion estimation and motion segmentation selection	33
4.5.1 Motion estimation	33
4.5.2 Motion segmentation selection	34
4.6 Complexity characteristics of decoder motion compensation.....	34
4.6.1 Interpolation process	34
4.6.2 Number of motion compensations	34
4.7 Complexity characteristics of encoder intra-frame prediction type selection	35
4.8 Complexity characteristics of decoder intra-frame prediction operation	35
4.9 Complexity characteristics of encoder transforms and transform type selection	35

4.10	Complexity characteristics of decoder inverse transform operations	36
4.10.1	Large size DCT.....	36
4.10.2	ROT.....	37
4.11	Complexity characteristics of encoder quantization and quantization type selection	37
4.12	Complexity characteristics of decoder inverse quantization.....	37
4.13	Complexity characteristics of encoder in-loop filtering type selection.....	37
4.13.1	Adaptive loop filter.....	37
4.14	Complexity characteristics of decoder in-loop filtering operations	38
4.14.1	Deblocking filter.....	38
4.14.2	Adaptive loop filter.....	38
4.14.3	Extreme correction and band correction	38
4.15	Complexity characteristics of encoder entropy coding type selection	39
4.16	Complexity characteristics of decoder entropy decoding operation	39
4.16.1	Syntax-based context-adaptive binary arithmetic coding	39
4.16.2	Adaptive coefficient scanning	39
4.17	Degree of capability for encoder parallel processing.....	39
4.18	Degree of capability for decoder parallel processing.....	39
4.19	Requirement for encoder frame-level multi-pass processing.....	39
5	Algorithm characteristics	40
5.1	Random access characteristics.....	40
5.2	Delay characteristics	40
6	Software implementation description.....	40
7	Further improvements after submission.....	40
7.1	High accuracy SKIP and DIRECT mode.....	40
7.2	New boundary strength decision rule	40
7.3	Bi-directional prediction refinement.....	41
7.4	Constraint set 1 configuration relative to Alpha anchor (updated)	41
8	Closing remarks	41
9	Patent rights declaration(s).....	42
	References	42

Glossary of terms used in this proposal

ACS	adaptive coefficient scanning
ADI	arbitrary directional intra
ALF	adaptive loop filter
AMP	asymmetric motion partition
AMVP	advanced motion vector prediction
BDC	band correction
CADR	content-adaptive dynamic range
CCCP	color component correlation based prediction
CU	coding unit
DIF	DCT-based interpolation filter
EXC	extreme correction
HAM	high accuracy motion
IBDI	internal bit-depth increase
LCU	largest coding unit
LOT	logical transform
MPI	multi-parameter intra
PTM	pixel-based template matching
PU	prediction unit
ROT	rotational transform
SBAC	syntax-based context-adaptive binary arithmetic coder
SCU	smallest coding unit
SPS	sequence parameter set
TU	transform unit

1 Introduction

With the ever increasing popularity of high-definition (HD) video content, video compression technologies which can provide higher coding efficiency than existing video coding standards have received increased attention. Recently, ITU-T SG16 Q.6 (VCEG) and ISO/IEC JTC1/SC29/WG11 (MPEG) have decided to create a Joint Collaborative Team on Video Coding (JCT-VC), and have jointly issued a Call for Proposals (CFP) on video compression technology. This proposal is Samsung's response to the CFP, produced in collaboration with the BBC. It aims to provide a video compression technology which has significantly higher compression capability than the state-of-the-art H.264/AVC standard, especially for HD video content. To achieve this goal, a number of new algorithmic tools are proposed, covering many aspects of video compression technology. These include a general structure for the representation of video content, inter/intra prediction, in-loop filtering, and entropy coding.

For efficient and flexible representation of video content with various resolutions, the representation introduced in this proposal is based on a triplet of coding unit (CU), prediction unit (PU), and transform unit (TU). CU is the basic unit of compression; CU sizes larger than 16x16, e.g. 32x32, 64x64 or even 128x128 are supported in order to compress HD video content effectively. CU is broadly analogous to the concept of macroblock, but is much more flexible. PU is the unit of inter/intra prediction and there can be multiple PUs in a single CU. To code irregular image patterns effectively in a video frame, asymmetric motion partition (AMP) for inter prediction is also supported. TU is the unit of transform, and can include one or more PUs. In addition to conventional 4x4 and 8x8 transforms, 16x16, 32x32, and 64x64 fast integer transforms are supported for TUs.

To improve inter prediction performance, a number of novel algorithmic tools are proposed. Advanced motion vector prediction (AMVP) is proposed, for the efficient representation of motion information. Sub-pixel high accuracy motion (HAM), with resolution down to 1/12 pixel, is also supported along with new 1/4-pel based interpolation filters designed separately for each phase and replacing the concatenated half-band and bilinear interpolation architecture for H.264/AVC. All filters have been designed using a common DCT interpolation filter (DIF) method.

To improve the performance of intra prediction, four new intra coding tools are included: arbitrary directional intra (ADI), pixel based template matching (PTM), color component correlation based prediction (CCCP), and multi-parameter intra (MPI). A rotational transform (ROT) tool, which can adaptively compact the energy into low frequency components, is also proposed. Using these algorithmic tools, prediction patterns which cannot be generated from directional pattern extrapolation or simple calculation from neighboring pixels can be provided; this leads to reduced energy of the residual signal and hence to improved coding performance.

In this proposal, several in-loop filters are combined to reduce the distortion between the original and the reconstructed video frames. In addition to a modified H.264/AVC deblocking filter for the CU/PU/TU structure, an adaptive loop filter (ALF) is applied to reduce the average distortion of the reconstructed frame, and then extreme correction (EXC) is used to reduce the distortion for local regions. Content adaptive dynamic range adaptation (CADR) is also performed to mitigate the rounding effects. All of the in-loop filtering operations are performed adaptively with respect to the flexible unit representation.

A syntax-based context-adaptive binary arithmetic coder (SBAC) is proposed for the entropy coding algorithm. SBAC is an adaptive binary arithmetic coding method with context models, which can provide high coding efficiency for various syntax elements with varying statistical properties. In order to increase the efficiency of entropy coding of transform coefficients, adaptive coefficient scanning (ACS) is used, which explicitly specifies the scanning order for each TU.

When all the proposed algorithmic tools are used, the proposed video codec achieves approximately 40% bit-saving on average compared to H.264/AVC in both constraint set 1 and 2 configurations. Furthermore, the bit-savings tends to be larger when the spatial resolution of the video increases, which is a good property according to the major goal of the HVC standard.

2 Algorithm description

2.1 Overview

In this proposal, new algorithmic tools are presented to exploit the properties of high resolution video. Highlighted features of proposed codec are as follows.

- **Flexible size unit representation:** the proposal separately defines three block concepts: coding unit (CU), prediction unit (PU) and transform unit (TU). After the size of largest coding unit (LCU) and the hierarchical depth of CU have been defined, the overall structure of codec is characterized by the various sizes of CU, PU and TU in a recursive manner. This allows the proposed codec to be readily adapted for various kinds of content, applications, or devices that have different capabilities/resources.

- **Size-independent syntax representation:** while block level syntax such as coded block pattern and intra prediction mode are coded differently depending upon the block sizes in H.264/AVC, the proposed codec employs one common syntax representation for all CU sizes, which reduces complexity and improves clarity.
- **Support of large and asymmetric motion partitions:** larger PUs than 16x16 are supported. Asymmetric motion partition (AMP) is also supported, to increase the performance for irregular image patterns.
- **Support of higher motion accuracy than 1/4 pel with new interpolation filter:** high accuracy motion (HAM) is supported with a new DCT-based interpolation filter (DIF). Motion vector refinement is introduced to obtain high accuracy such as 1/12 pixel.
- **Support of large integer transforms:** in addition to conventional 4x4 and 8x8 transform, fast integer realizations of 16x16, 32x32 and 64x64 transforms are proposed.
- **New rotational transform:** a new supplementary rotational transform (ROT) is proposed to encode high energy residual information more efficiently.
- **New motion vector prediction method:** advanced motion vector prediction (AMVP) is utilized to find the best motion vector predictor among the various PU combinations
- **New in-loop filtering methods:** several in-loop filters are combined to reduce the reconstruction distortion. The H.264/AVC deblocking filter has been modified to make it suitable for the hierarchical CU/PU/TU structure. In addition, the CU-synchronized adaptive loop filter (ALF) minimizes the expected average distortion whilst the spatial filtering extreme correction (EXC) reduces the distortion in the specific regions which are important to visual perception. Also, content adaptive dynamic range (CADR) is performed to mitigate rounding effects and to increase the accuracy of intermediate calculation without increasing bit-depth.
- **New intra prediction methods:** to increase the performance of intra coding, four new intra tools are included: arbitrary directional intra (ADI), pixel based template matching (PTM), color component correlation based prediction (CCCP) and multi-parameter intra (MPI). Using these tools, prediction patterns can be provided which cannot be generated efficiently in the conventional way.
- **New entropy coding with explicit scan order signaling:** syntax-based binary arithmetic coder (SBAC) is proposed. In order to increase the efficiency of the entropy coding of transform coefficients, an explicit scan order is signaled from amongst pre-defined scan orders.

Figure 2-1 shows all building blocks to decode the input bit stream. White boxes indicate the newly introduced building blocks. From the following sections, each building block is explained in detail.

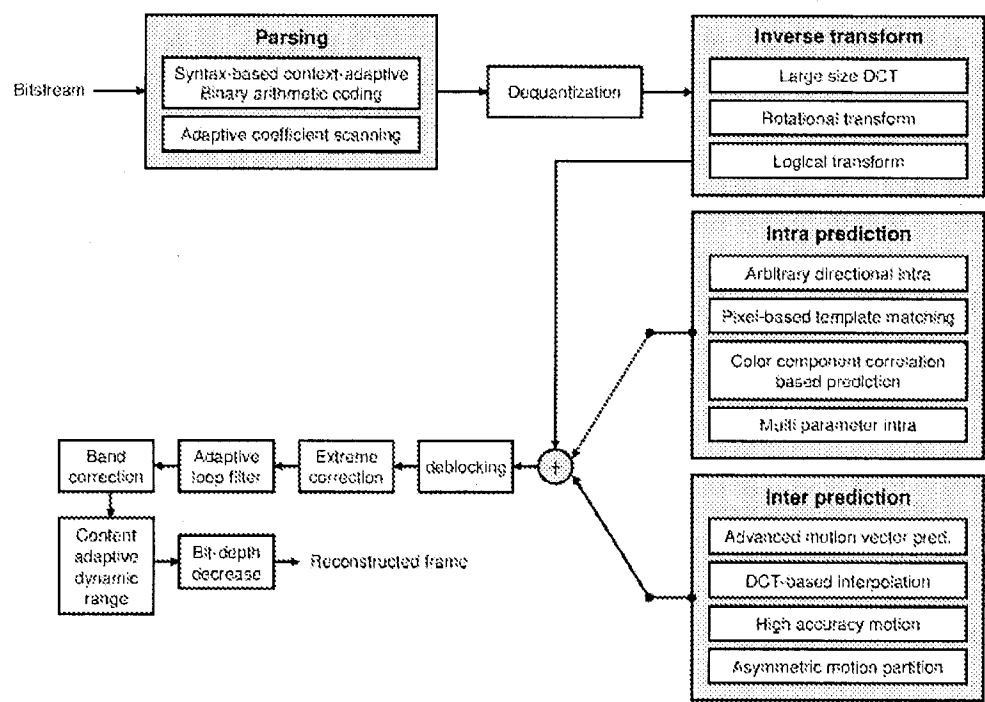


Figure 2-1 All building blocks to decode the input bit stream

2.2 Unit definition

2.2.1 Coding unit (CU)

In this proposal, the coding unit (CU) is defined as a basic unit which has a square shape. Although it has a similar role to the macroblock and sub-macroblock in H.264/AVC, the main difference lies in the fact that CU can have various sizes, with no distinction corresponding to its size. All processing except frame-based loop filtering is performed on a CU basis, including intra/inter prediction, transform, quantization and entropy coding. Two special terms are defined: the largest coding unit (LCU) and the smallest coding unit (SCU). For convenient implementation, LCU size and SCU size are limited to values which are a power of 2 and which are greater than or equal to 8.

It is assumed that a picture consists of non-overlapped LCUs. Since the CU is restricted to be a square shape, the CU structure within a LCU can be expressed in a recursive tree representation adapted to the picture. That is, CU is characterized by LCU size and the hierarchical depth in the LCU that the CU belongs to.

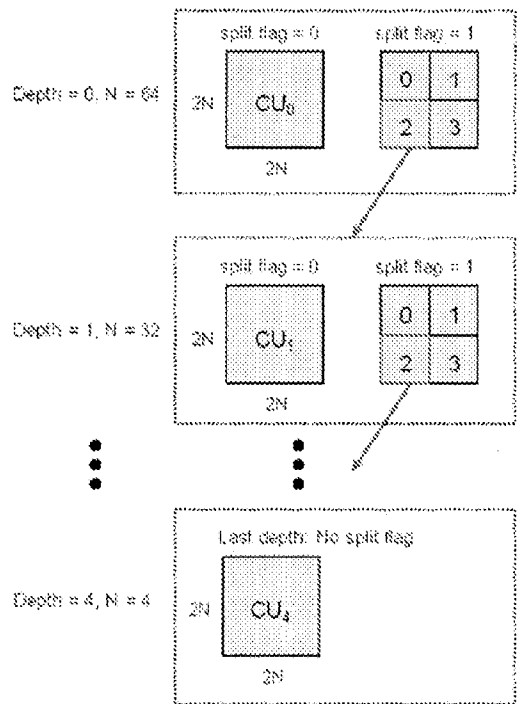


Figure 2-2 Illustration of recursive CU structure (LCU size = 128, maximum hierarchical depth = 5)

Figure 2-2 shows an example where LCU size is 128 and the maximum hierarchical depth is 5. The recursive structure is represented by a series of split flags. For CU_d , which has depth d and size $2N \times 2N$, the coding of CU is performed in the current depth when split flag is set to zero. When the split flag is set to 1, CU_d is split into 4 independent CU_{d+1} which have depth $(d+1)$ and size $N \times N$. In this case, CU_{d+1} is called a sub-CU of CU_d similar to a sub-macroblock in H.264/AVC. Unless the depth of sub-CU $(d+1)$ is equal to the maximum allowed depth (4 in this case), each CU_{d+1} is processed in a recursive manner. If the depth of sub-CU $(d+1)$ is equal to the maximum allowed depth, further splitting is not allowed. Note that a CU can be further split into PUs, however (see section 2.1.2).

The sizes of LCU and SCU are specified in the Sequence Parameter Set (SPS). The embedded information in the SPS is LCU size (s) and the maximum hierarchical depth (h) in a LCU. For example, if $s = 128$ and $h = 5$, then 5 kinds of CU sizes are possible: 128x128 (LCU), 64x64, 32x32, 16x16 and 8x8 (SCU). If $s = 16$ and $h = 2$, then 16x16 (LCU) and 8x8 (SCU) are possible; this is a similar block structure to H.264/AVC. Therefore, if the LCU size and maximum hierarchical depth are given, this defines the possible CU sizes which are allowed.

This kind of arbitrary unit representation provides several major benefits. The first benefit comes from the support of CU sizes greater than the conventional 16x16 macroblock. When the region of interest is homogeneous, a large CU can represent the region in a smaller number of symbols than is possible in the case of several small blocks. The evidence for this conclusion has already been presented by several papers both inside MPEG [1][2] and at external conferences [3][4]. Similar approaches have also been followed by recent proposals in MPEG and VCEG [5][6][7].

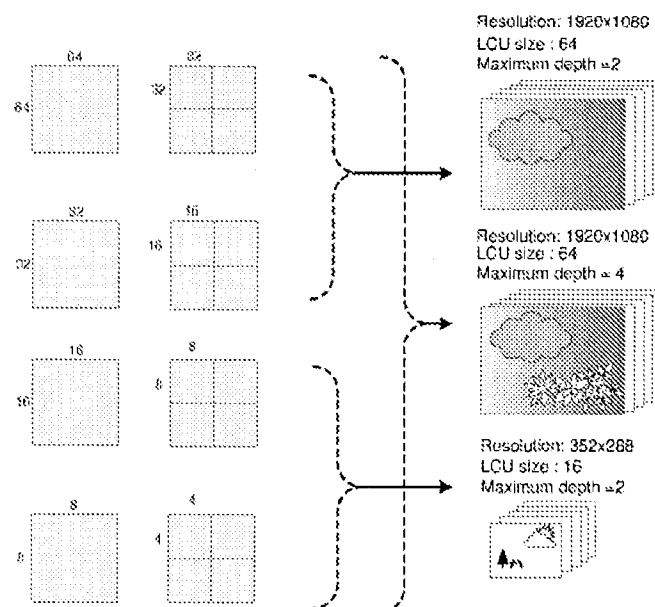


Figure 2-3 Example of LCU size and maximum depth combinations for various resolutions

Furthermore, supporting arbitrary LCU sizes enables the codec to be readily optimized for various content, applications and devices. Compared to the use of fixed size macroblock, support of various LCU sizes is one of the major properties of this proposal. It is especially useful for low resolution video services, which is still commonly used in the market. By choosing LCU size and maximum hierarchical depth appropriately, the hierarchical block structure can be optimized in a better way for the targeted application. In the later stage of the standardization, the range of LCU sizes could be specified in the Profiles and Levels section to match the requirements more specifically. Figure 2-3 shows examples of CU size combinations for several resolutions.

Finally, by eliminating the distinction between macroblock and sub-macroblock and using only one unit type, CU, the multi-level hierarchical structure can be specified in a very simple and elegant way: LCU size, maximum hierarchical depth and a series of split flags. Together with the proposed size-independent syntax representation, it's sufficient to specify syntax items of one general size for the remaining coding tools. This kind of consistency can greatly simplify the specification effort as well as the actual parsing process.

2.2.2 Prediction unit (PU)

Once the splitting process is done, prediction methods are specified for every CU which is not further split i.e. the leaf nodes of the CU hierarchical tree.

Coupled with CU, this proposal introduces a basic unit for the prediction mode: the prediction unit (PU). It should be noted that the PU is defined only for the last-depth CU and its size is limited to that of the CU.

Similar to conventional standards, we define two different terms to specify the prediction method: the prediction type and the PU splitting. The prediction type is one of the values among skip, intra or inter, which roughly describe the nature of the prediction method. After that, possible PU splittings are defined according to the prediction type.

Figure 2-4 shows possible PU splittings according to the different prediction types for a CU of size $2N \times 2N$. The PU for intra has 2 different possible splittings: $2N \times 2N$ (i.e. no split) and $N \times N$ (quarter split). The PU for inter has 8 different possible splittings: 4 symmetric splittings ($2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$) and 4 asymmetric splittings ($2N \times nU$, $2N \times nD$, $nL \times 2N$ and $nR \times 2N$). A skipped PU can only be $2N \times 2N$ i.e. the whole CU is skipped. The number N is derived from the size of the CU which the PU belongs to. For example, if the size of CU is 128×128 , then both 128×128 and 64×64 PUs for intra are possible. For inter prediction, 128×128 , 128×64 , 64×128 , 64×64 , 128×32 (for $2N \times nU$ and $2N \times nD$), 128×96 (for $2N \times nU$ and $2N \times nD$), 32×128 (for $nR \times 2N$ and $nL \times 2N$) and 96×128 (for $nR \times 2N$ and $nL \times 2N$) are possible.

All information related to prediction is signaled on a PU basis, for instance, the direction of intra prediction for intra or the motion vector difference and reference index for inter. Motion vector prediction and motion compensation are also performed on a PU basis.

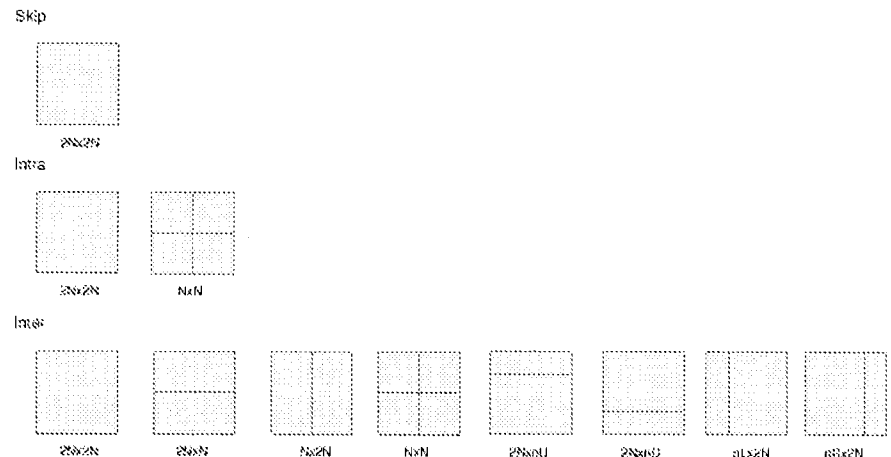


Figure 2-4 Example of possible PU splittings for skip, intra and inter prediction

2.2.3 Transform unit (TU)

In addition to the CU and PU definitions, the transform unit (TU) for transform and quantization is defined separately. It should be noted that the size of the TU may be larger than that of the PU, which is different from previous video standards, but it may not exceed the CU size. However, the TU size is not arbitrary and once a PU structure has been defined for a CU, only two TU partitions are possible. As a result, the size of TU in the CU is determined by the transform unit size flag. If the transform unit size flag is set to 0, the size of TU is the same as that of the CU which the TU belongs to. Otherwise, the TU size is set as $N \times N$ or $N/2 \times N/2$ according to the PU splitting. Figure 2-5 shows an example of TU when size of CU is $2N \times 2N$. According to transform unit size flag and PU splitting, the size of TU is specified to be $2N \times 2N$, $N \times N$ or $N/2 \times N/2$, as illustrated in the figure below. It should be noted that the case of zero transform unit size flag is designed for performing a transform of CU size regardless of PU splitting. This approach yields coding gains especially for coding semi-random residuals generated by complex texture areas.

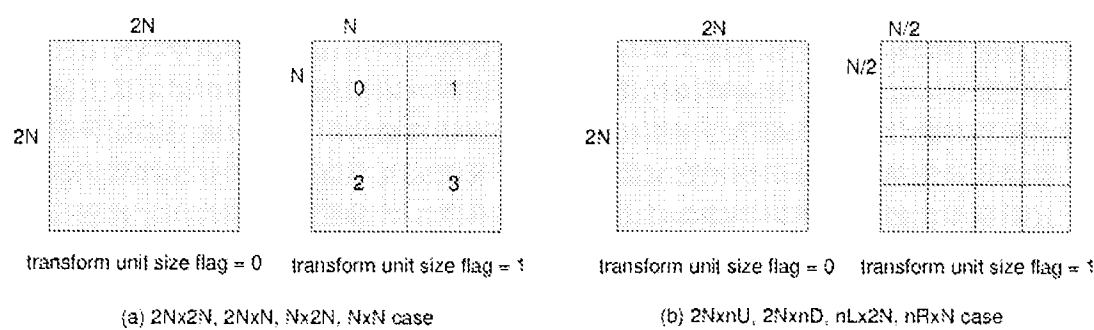


Figure 2-5 Example of transform unit (TU) according to the transform unit size flag

2.2.4 Relationship between CU, PU and TU

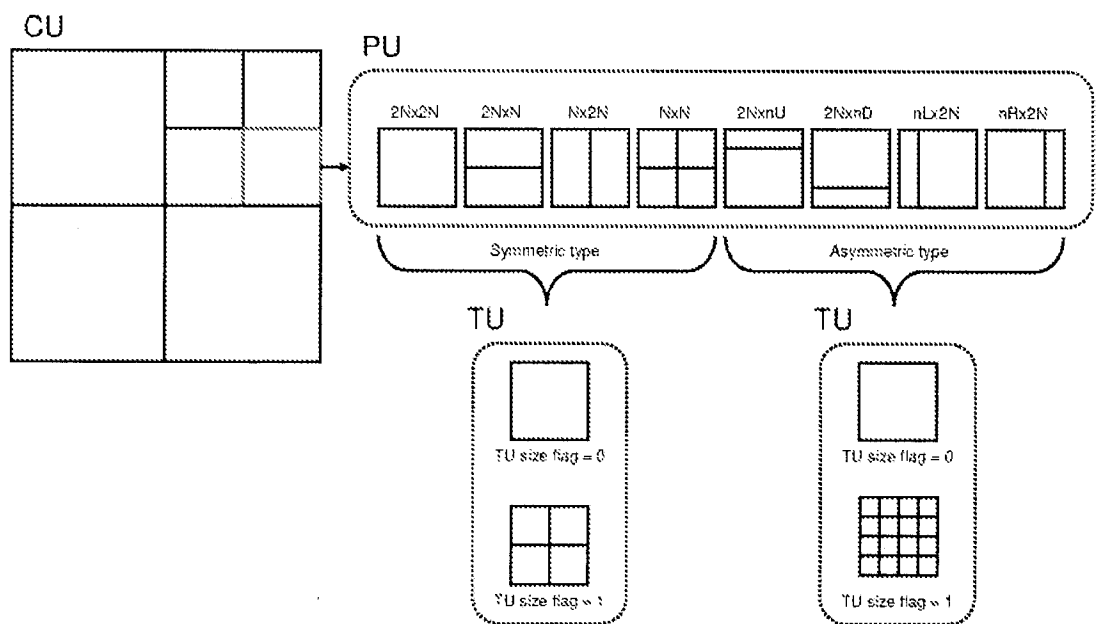


Figure 2-6 Relationship between CU, PU and TU

Figure 2-6 shows the relationship between the three different unit definitions, CU, PU and TU. For any size of CU given, a series of split flags is used to specify smaller size CUs inside the CU. For every CU which does not split any more, a PU is defined specifying how the prediction can be generated. Note that the PU need not be a square shape and virtually any kind of new PU splitting could be added without changing the overall structure. TU size is specified by TU size flag and PU splitting. If the TU size flag is zero, TU size is always equal to CU size whereas TU size is defined differently depending upon PU splitting when the TU size flag is one. In this latter case, the TU size is chosen so that the transform blocks do not cross PU block boundaries. Table 2-1 summarizes the possible combinations of CU, PU, and TU based on the value of split flag, prediction type, PU splitting and TU size.

Table 2-1 Possible combinations of CU, PU and TU

split flag = 0 (CUa processing, current depth = d)				split flag = 1 (NxN split)
Prediction type	PU splitting		TU size	
INTRA (only for 2Nx2N, NxN)	symmetric type	asymmetric type	TU size flag = 0	TU size flag = 1
INTER (for all PU splitting)	2Nx2N	2Nx2N	2Nx2N	NxN (symmetric type)
SKIP (only for 2Nx2N)	Nx2N	2Nx2N		N/2xN/2 (asymmetric type)
	NxN	nLx2N		
		nRx2N		
Recursive CU processing for CUa+1 whose size is NxN				

2.2.5 Size-independent syntax representation

In the proposed codec, the maximum depth for CU hierarchy is arbitrary and potentially larger than the maximum depth allowed in the H.264/AVC block hierarchy. Therefore, the proposal includes a size-independent syntax representation which specifies all syntax elements in a consistent way independent of the CU size. By contrast, in H.264/AVC block level syntax elements such as transform_8x8_mode_flag, coded_block_flag and intra_pred_modes are coded differently depending on whether the block size is 16, 8 or 4. Table 2-2 shows the recursive part of the CU syntax in tabular form.

Table 2-2 Recursive CU syntax in tabular form

read_CU (currDepth) {	Descriptor
split_flag	ue(v)
if (split_flag && currDepth < maxAllowedDepth) {	
for (culdx = 0; culdx < 4; culdx++)	
read_CU (currDepth+1)	
}	
else {	
CU_data (currDepth) /* read syntax elements for arbitrary size CU */	
}	
}	

As shown in the table, the splitting process for the CU can be specified recursively and all other syntax elements for the leaf CU are defined in the same way independent on the CU size. This kind of representation is very useful in terms of reduced parsing complexity and improved clarity if a large hierarchical depth is allowed.

2.3 Motion representation

In this proposal, motion fields are represented by the motion vectors obtained from each PU. As specified in Section 2.2.2, four kinds of asymmetric shape of motion partition are supported to enhance the accuracy of the motion field representation. For the motion accuracy, quarter-pel precision with 1/12 optional refinement is supported. Motion vector prediction is applied only for the quarter-pel accuracy level and multiple motion vector predictor candidates are considered to improve the prediction efficiency. We also propose the use of non-cascaded spatial filtering for every motion accuracy position to minimize the motion compensation complexity when higher motion vector accuracy is used whereas the previous standards typically use cascaded spatial filtering, such as well-known 6-tap with bi-linear interpolation for the quarter-pel motion accuracy.

2.3.1 Asymmetric motion partition (AMP)

The hierarchical block structure of the proposed video codec makes the representation of image patterns more accurate and efficient. To further improve the efficiency, asymmetric motion partition (AMP) is introduced by allowing asymmetric partitions of a block for inter prediction. Figure 2-7 shows an example of the AMP where a 64x64 block is asymmetrically partitioned into 64x16, 64x48, 16x64, or 48x16 blocks, instead of two 64x32 or 32x64 blocks. Note that the motion partitions shown in the figure correspond to 2Nx2N, 2Nx2N, nLx2N, and nRx2N, respectively. AMP improves the coding efficiency, since irregular image patterns, which otherwise would be constrained to being represented by a symmetric partition, can now be more efficiently represented without requiring further splitting.

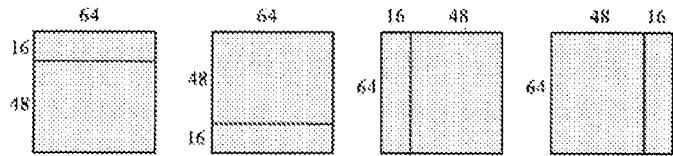


Figure 2-7 Example of asymmetric motion partitions for 64x64 block

2.3.2 Advanced motion vector prediction (AMVP)

The bit rate for encoding motion information is reduced by utilizing predictive coding of motion vectors. Conventional video coding standards use a simple shared rule between encoder and decoder. For example, H.264/AVC predicts its motion vector using a spatial median, based on neighboring blocks. The median is adapted depending on the availability of these neighboring blocks. A few competition-based schemes have also been proposed, which usually select the best predictor from a given set, and send the index of this predictor as side information. The set is typically composed of three neighboring motion vectors and a temporal motion vector, where these predictors are exhaustively compared [8]. However, such a competition-based scheme needs to consider more available neighboring motion vectors and temporal motion vectors when dealing with large block representation and a flexible structure.

In this proposal, an advanced motion vector prediction method is introduced which is adapted to large block representation and a flexible temporal structure. It allows the selection of the best predictor from a given set, where the set is composed of three spatial motion vectors, a median motion vector and a temporal motion vector.

In the predictor set, the three spatial motion vectors are chosen from among those above, to the left and from each applicable corner. In each direction, the first available motion vector is inspected among neighboring blocks and it is chosen as a spatial motion predictor once it is found. The corner blocks include the right top, the left top as well as the left below neighboring block depending on the current block position.

The temporal motion predictor is given by the nearest reference frame and is scaled according to temporal distance. For B pictures, the direction to get the reference frame is indicated for each slice, to provide the better temporal motion predictor among forward and backward reference frames.

The ordering of candidates in the set is important, to reduce the overhead of signaling the position of the best motion predictor in the predictor set. The ordering of the set is adapted depending on the current prediction mode to place the most probable motion predictor in the first position, since minimum overhead occurs if the first candidate is chosen as the best predictor.

The overhead of signaling overhead the index of the best predictor can be reduced further by minimizing the number of candidates in the set. Duplicated motion vectors are simply removed from the set and motion predictors which could not have been used in the encoder are also removed after possible candidates are determined from the transmitted MVD value [11]. If only one motion predictor is available, the best motion predictor is implied by the motion predictor without overhead.

2.3.3 Improved skip and direct mode

As the video resolution and the size of coding units increase, the provision of intermediate types of mode, between the simplest and the most complex mode, becomes useful to improve coding efficiency. In H.264/AVC, the simplest modes like SKIP or DIRECT mode send no motion information whatsoever whilst the other modes require both motion and texture information.

In this proposal, new skip and direct concepts are presented. Regardless of slice type, both skip and direct modes are supported; the difference between skip and direct is whether texture information is sent or not. The motion of skip and direct is inferred by the best motion vector predictor signaled by its position in the motion predictor set. Furthermore, uni-directional direct modes in B slices are supported in addition to the bi-directional direct mode.

2.3.4 DCT-based interpolation filter (DIF)

Motion compensation is the key factor for efficient video compression. Compensation for motion with fractional pel accuracy requires interpolation of the reference signal. A conventional approach uses a 6-tap Wiener filter for half-pel generation. For higher accuracy (for example 1/4 pel), first a Wiener filter is applied and then a bi-linear combination of integer and half pel values is used to provide 1/4 pel accuracy interpolation.

This proposal replaces the combination of Wiener and bilinear filters by directly providing the interpolated value at the desired fractional accuracy. More specifically, only one filter operation is needed to provide interpolated pixel to any pixel accuracy, thus motion compensation process can be simplified in the implementation point of view and complexity can be reduced for quarter-pel accuracy when a combination of 6-tap and bilinear filter is used. Furthermore, this tool supports the unified way to generate the interpolation filters supporting higher than 1/4 pel motion vector accuracy, e.g. 1/8 pel or even higher. More detailed explanation supporting higher motion vector accuracy than quarter-pel is provided in Section 0.

2.3.4.1 DCT basis decomposition

This section explains the motivation and conceptual realization of the new interpolation. Assume that the integer points values $\{p_i, i = -(M-1), \dots, M\}$ are known and that the value p in position $0 \leq a \leq 1$ is to be predicted.



Figure 2-8 Known integer pel values (shaded) and predicted fractional pel value p .

Assume that the values $\{p_i, i = -(M-1), \dots, M\}$ are integer points values of a smooth function $p(x)$. The forward DCT provides the transform coefficients set:

$$C_k = \frac{1}{M} \sum_{l=-M+1}^M p(l) \cos\left(\frac{(2l-1+2M)k\pi}{4M}\right), \quad (1)$$

The inverse DCT

$$p(x) = \frac{C_0}{2} + \sum_{k=1}^{2M-1} C_k \cos\left(\frac{\pi(2x-1+2M)k}{4M}\right) \quad (2)$$

returns absolutely exactly $p(x)$ for integer $x=-(M-1), -(M-2), \dots, M$.
A fractional point value can be calculated by using the corresponding shift as the basis function argument.

$$p(\alpha) = \frac{C_0}{2} + \sum_{k=1}^{2M-1} C_k \cos\left(\frac{\pi(2\alpha-1+2M)k}{4M}\right), \quad (3)$$

The sequential processing of forward DCT decomposition (2) and inverse DCT with a shifted argument for the basis functions (3) therefore leads to a new interpolation realization.
However, the actual usage of a forward-inverse transform pair is not desirable because of its high computational complexity. In this proposal, transform operation pairs and intermediate phase shift were merged into a single spatial domain filter.

2.3.4.2 Example of filter coefficients

The computational complexity of filtering depends on the number of filter taps: $2M$. For $M=3$ computational complexity is comparable to a 6-tap Wiener filter. A larger number of filter taps provides better interpolation quality. Several pre-calculated filter coefficients for each fractional position are given in the Table 2-3 and Table 2-4 for 6 tap and 12 tap, respectively. The filter coefficients are scaled by 256. A higher bit depth of filter coefficients leads to more accurate interpolation; for comparison, H.264/AVC uses 5 bits for filter coefficients.

Table 2-3 Filter coefficients of 6-tap DIF filter

α	$2M = 6$ (6-tap filter)
-1/12	{ -4, -19, 254, -19, 8, -2, }
1/12	{ 4, -16, 252, 22, -8, 2, }
1/6	{ 6, -28, 242, 48, -17, 5, }
1/4	{ 9, -37, 227, 75, -25, 7, }
2/6	{ 11, -42, 208, 103, -33, 9, }
5/12	{ 12, -44, 184, 132, -39, 11, }
1/2	{ 11, -43, 160, 160, -43, 11, }
7/12	{ 11, -39, 132, 184, -44, 12, }
2/3	{ 9, -33, 103, 208, -42, 11, }
3/4	{ 7, -25, 75, 227, -37, 9, }
5/6	{ 5, -17, 48, 242, -28, 6, }

Table 2-4 Filter coefficients for 12-tap DIF filter

α	$2M = 12$ (12-tap filter)
-1/12	{ 1, -3, -5, -10, 22, 253, -19, 10, -6, -4, -2, 1 }
1/12	{ -1, 3, -5, -9, -19, 253, 23, -10, -6, -4, 2, -1 }
1/6	{ -2, 5, -9, 16, -34, 244, 49, -21, 12, -7, 4, -1 }
1/4	{ -1, 6, -12, 21, -43, 229, 75, -30, 17, -10, 5, -1 }
2/6	{ -3, 8, -15, 26, -50, 211, 105, -40, 22, -13, 7, -2 }
5/12	{ -3, 9, -16, 28, -53, 188, 134, -47, 26, -15, 8, -3 }
1/2	{ -2, 7, -15, 28, -52, 162, 162, -52, 28, -15, 7, -2 }
7/12	{ -3, 8, -15, 26, -47, 134, 188, -53, 28, -16, 9, -3 }
2/3	{ -2, 7, -13, 22, -40, 105, 211, -50, 26, -15, 8, -3 }
3/4	{ -1, 5, -10, 17, -30, 75, 229, -43, 21, -12, 6, -1 }
5/6	{ -1, 4, -7, 12, -21, 49, 244, -34, 16, -9, 5, -2 }

2.3.4.3 Low complexity optimization

An efficient hardware implementation requires multiplication-free representation for all calculations. In order to minimize the number of arithmetical operations (shifts and additions), the filter coefficients in Table 2-3 and Table 2-4 can be slightly changed without significant change of performance. Table 2-5 and Table 2-6 contain optimized filter coefficients together with the required number of arithmetical operations. In the case of 6 tap filters half-pel interpolation, the DIF optimized coefficients are the same as the Wiener filter coefficients in H.264/AVC multiplied by 8. Hence, the computational complexity is also the same as for the half-pel interpolation filter in H.264/AVC. With the same computational complexity, the 6-tap DIF outperforms the Wiener + bi linear filter combination in H.264/AVC. The 12-tap DIF outperforms the Adaptive Interpolation Filter (AIF) without requiring any content adaptation.

Table 2-5 Optimized Filter coefficients for 6-tap DIF filter

α	$2M = 6$ (6-tap filter)
1/4	{ 8, -32, 224, 72, -24, 8 } (8 additions, 4 shifts)
1/2	{ 8, -40, 160, 160, -40, 8 } (6 additions, 3 shifts)
3/4	{ 8, -24, 72, 224, -32, 8 } (8 additions, 4 shifts)

Table 2-6 Optimized Filter coefficients for 12-tap DIF filter

α	$2M = 12$ (12-tap filter)
1/4	{ -1, 5, -12, 20, -40, 229, 76, -32, 16, -8, 4, -1 } (18 additions, 6 shifts)
1/2	{ -1, 8, -16, 24, -48, 161, 161, -48, 24, -16, 8, -1 } (15 additions, 4 shifts)
3/4	{ -1, 4, -8, 16, -32, 76, 229, -40, 20, -12, 5, -1 } (18 additions, 6 shifts)

2.3.5 High accuracy motion (HAM)

Generally speaking, if the motion accuracy for motion compensation is increased, smaller prediction error can be obtained. However, more accurate motion representation also generally leads to an increase in the bit-rate required to convey the motion information. The additional overhead mainly comes from two facts. Firstly, when the motion accuracy is increased the magnitude of motion vector is also increased. Secondly, motion vector prediction is more difficult since the range of motion vector values is increased. As a result, 1/8 pel motion accuracy (i.e. twice the 1/4 pel accuracy in H.264/AVC) may result in a reduction in the overall compression efficiency, except for particular sequences [9].

In this proposal, efficient representation is presented to utilize higher motion accuracy than 1/4 or 1/8 pel without any negative effects. Additional motion information beyond 1/4 pel accuracy is represented by simple refinement information instead of requiring a more accurate motion vector. More specifically, high accuracy motion is represented by MV (with lower motion accuracy, for example 1/4 pel) + refinement. The benefit of this representation is as follows. Firstly, the magnitude of the motion vector is not increased because the accuracy of motion vector is not changed, only the refinement. Secondly, since motion vector prediction is done without refinement information, there is no increase in the difficulty of motion vector prediction. Finally, the overhead is minimized since the refinement information is sent only when it is useful. For example, for each motion vector component, the first bit of refinement indicates whether the refinement is needed or not. If the bit is zero, it means that there is no more refinement information and next bit is not needed. If the bit is not zero, following bit indicates -1/12 or +1/12 refinement.

Although the refinement information can represent any accuracy, the current design uses 1/12 instead of 1/8 pel to give a more compact representation. If 1/8 pel accuracy is used instead of 1/12, there may be up to four different ways of representing the same position.

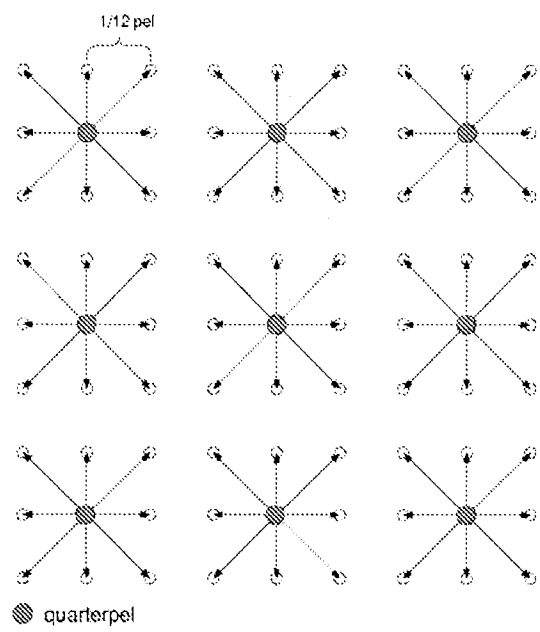


Figure 2-9 High accuracy motion grid

2.4 Intra-frame prediction

In this proposal, four intra tools are integrated to generate the final intra prediction results. The arbitrary directional intra (ADI) is the main intra prediction tool; it is an extension of the directional intra tool in H.264/AVC. The pixel based template matching (PTM) provides different prediction patterns which cannot be generated from directional pattern extrapolation or by simple calculation from neighboring pixels. The color component correlation based prediction (CCCP) tool improves the coding efficiency of chroma samples by exploiting the correlation between luma and chroma samples. The multi parameter intra (MPI) tool smoothes the luma predicted patterns while maintaining the meaningful details, which results in better coding efficiency from the transform. The MPI also provides variations to predicted patterns from ADI and PTM. Figure 2-10 shows an example of various target regions related to each intra prediction tool.

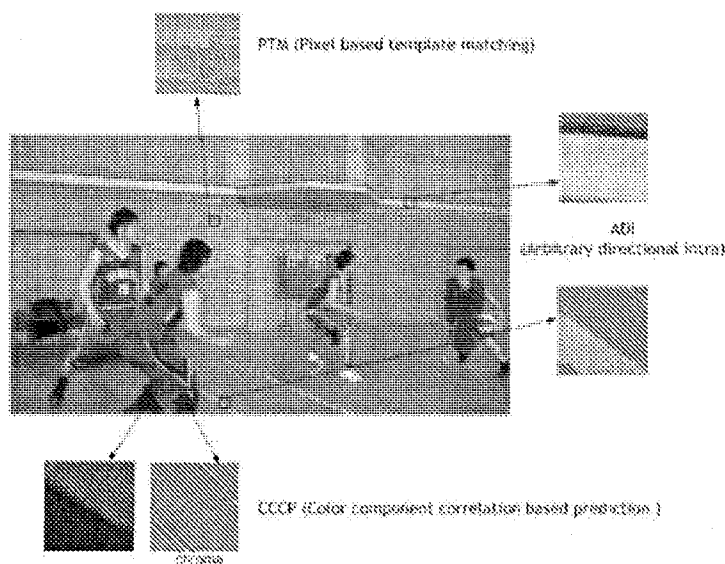


Figure 2-10 An example of target areas for various intra prediction tools

2.4.1 Arbitrary directional intra (ADI)

The proposed codec structure supports various sizes of coding units and transforms. This new codec structure provides large blocks and large transforms which are known to improve coding efficiency for HD resolutions. But if prediction methods are not designed accordingly, the benefits from the new codec structure cannot be maximized. Arbitrary directional intra (ADI) is designed to fit into this flexible codec structure.

In H.264/AVC intra coding, only 4 intra prediction modes are available for 16x16 blocks, since 16x16 blocks in lower resolution video generally represent either homogenous regions which can be described by the 4 prediction modes (vertical, horizontal, DC, plane) or else overly complicated patterns which cannot be predicted by a single prediction. But in HD sequences, even relatively large blocks (>=16x16) cover only a small part of objects or backgrounds, which can be described as homogeneous texture patterns with variations such as gradation or directional pattern extrapolation. Therefore additional performance improvement can be expected by applying directional prediction for these large blocks.

However, applying the same 9 directional predictions from conventional 4x4 intra prediction does not provide enough variation to describe the likely patterns in large blocks in HD sequences. Therefore, more prediction modes (up to 33) are provided in the ADI. Different block sizes are used for different image patterns for intra prediction; the number of available prediction modes for each block is adjusted according to the block size.

As in H.264/AVC intra prediction, the ADI generates prediction pixels by directional extrapolation or calculation using the nearest boundary pixels of the already decoded area. But in ADI, even boundary pixels from the left down region may be used as context pixels for prediction as depicted in Figure 2-11.

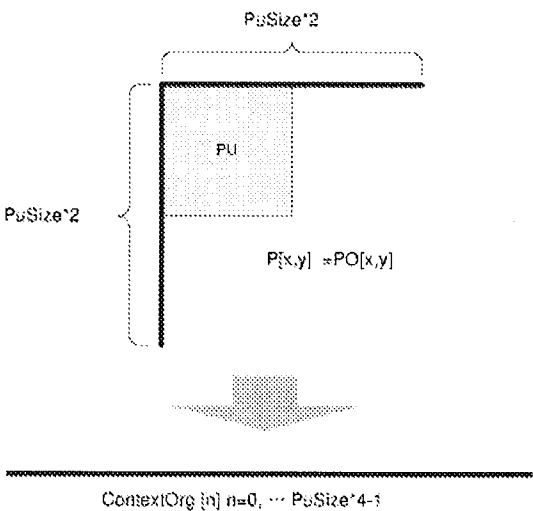


Figure 2-11 Example of context pixels for ADI

Whilst the 9 prediction modes are defined separately as Vertical, Horizontal, DC, Diagonal Down-Left, Diagonal Down-Right, Vertical-Right, Horizontal-Down, Vertical-Left, and Horizontal-Up in H.264/AVC, most of prediction modes in ADI are defined by integer pair information (dx, dy). The (dx, dy) pair represents the direction which each mode uses for context pixel extrapolation, as illustrated in Figure 2-12 below.

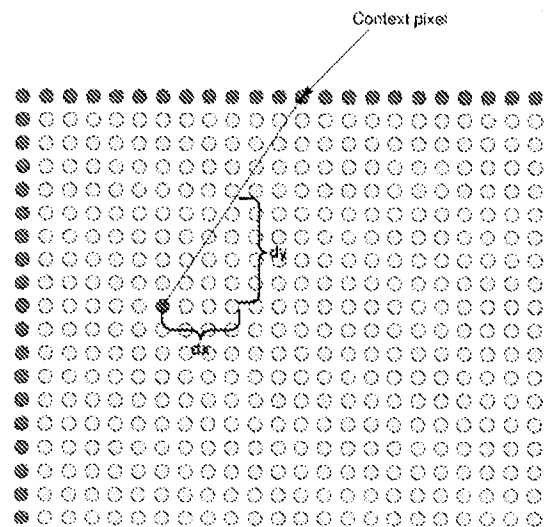


Figure 2-12 prediction methods defined by (dx, dy)

The available prediction modes for ADI are as follows.

Table 2-7 Available prediction modes for ADI

Mode description	Pred mode number
Vertical	0
Horizontal	1
DC	2
Plane	3
(dx_i, dx_y)	4 ~ 31
Bi-linear	32

The prediction modes used for each coding unit sizes can be found in the following table.

Table 2-8 Prediction modes used for each coding unit sizes

Prediction Unit size	Number of prediction modes	Used prediction modes
4x4	9	0~2, 4~9
8x8	9	0~2, 4~9
16x16	33	0~32
32x32	33	0~32
64x64	5	0~5
128x128	5	0~5

Depending on the prediction unit size, the number of filtering operations that may be applied to the context pixels varies, which provides different prediction patterns.

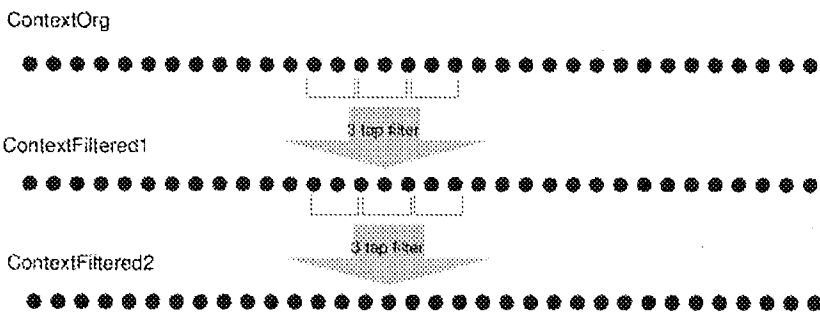


Figure 2-13 Example of context pixel filtering

For the chroma component, different prediction modes are applied for each prediction size. The prediction modes are chosen from 0: Vertical, 1: Horizontal, 2: CCCP mode, 3: Plane and 4: PTM mode.

2.4.2 Multi-parameter intra (MPI)

Directional intra prediction often results in non-smooth and unnatural patterns which are not transform-friendly. It is possible to reduce this problem by using a smoothing filter, but conventional smoothing filters tend to also remove meaningful details from the predicted results.

The purpose of MPI is to generate more natural patterns by applying different post processing filters to the prediction results from directional intra. Whilst conventional smoothing filters use NxN symmetric filters, MPI uses a 4 point filter for each pixel which maintains the directional patterns even after the smoothing. Figure 2-14 depicts the prediction signal after ADI and MPI is applied.

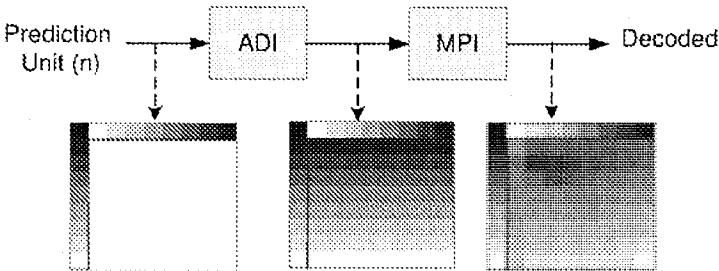


Figure 2-14 Combination of ADI and MPI

For each prediction unit, MPI post-processing is applied after the ADI predictions. If $MPI_index=0$, no MPI processing is applied. If $MPI_index=1$, then a 4 point based filter is applied as in the following equation and the figure (Figure 2-15) where $pred'$ represents the predicted results after MPI and $pred$ defines the predicted results from ADI or already reconstructed pixels.

$$pred'[x,y]=(pred[x,y]+pred[x-1,y]+pred[x,y-1]+pred[x,y+1]+2)>>2.$$

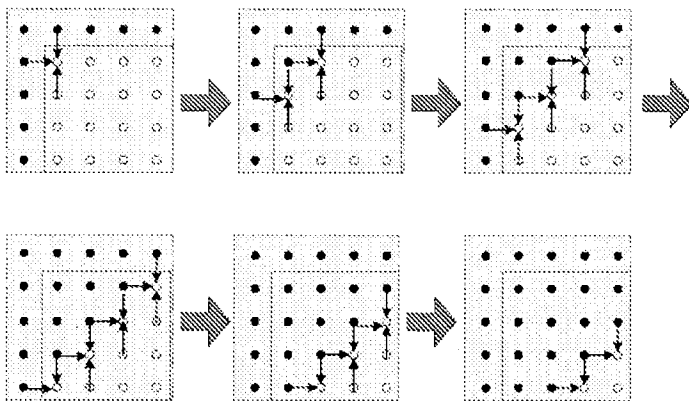


Figure 2-15 MPI process description

2.4.3 Color Component correlation based prediction (CCCP)

In most video codecs, chroma sample prediction is performed after the luma sample reconstruction. Therefore, any information inferred from reconstructed luma samples can be used for chroma sample prediction. The CCCP exploits the information of region partitions analyzed from the reconstructed luma samples to improve the coding efficiency of chroma samples.

In the CCCP, region partitions are estimated by performing segmentation on the reconstructed luma samples. The derived segmentation map is used to predict chroma samples. Based on the estimated segmentation map, different reconstructed samples in the neighbourhoods are used to predict each region.

The procedure of CCCP method is illustrated in Figure 2-16. Firstly, reconstructed luma samples are down-sampled by bi-linear filter to match the chroma block size for 4:2:0 chroma format case. Then, simple thresholding using the mean value method is exploited in order to generate the segmentation map. After that, two mean values for the chroma predictions according to the generated map are calculated and assigned to the inner part of chroma block. Finally, a 3x3 averaging filter is applied to smooth the prediction.

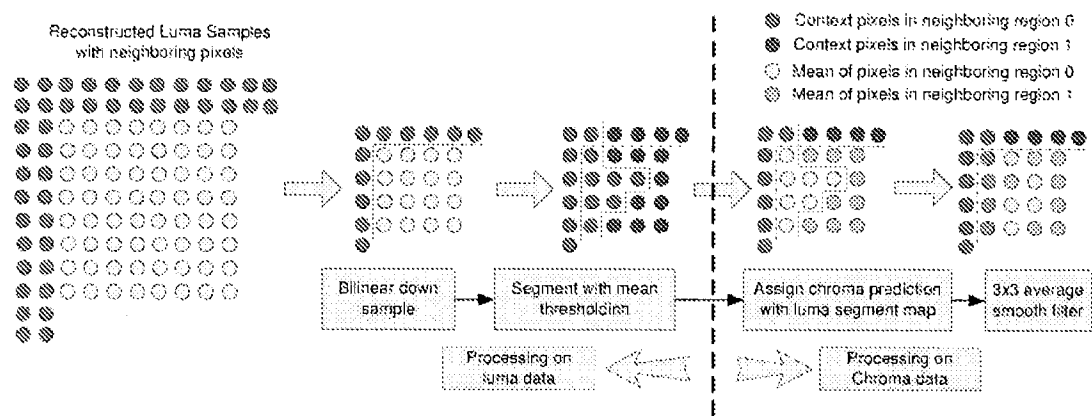
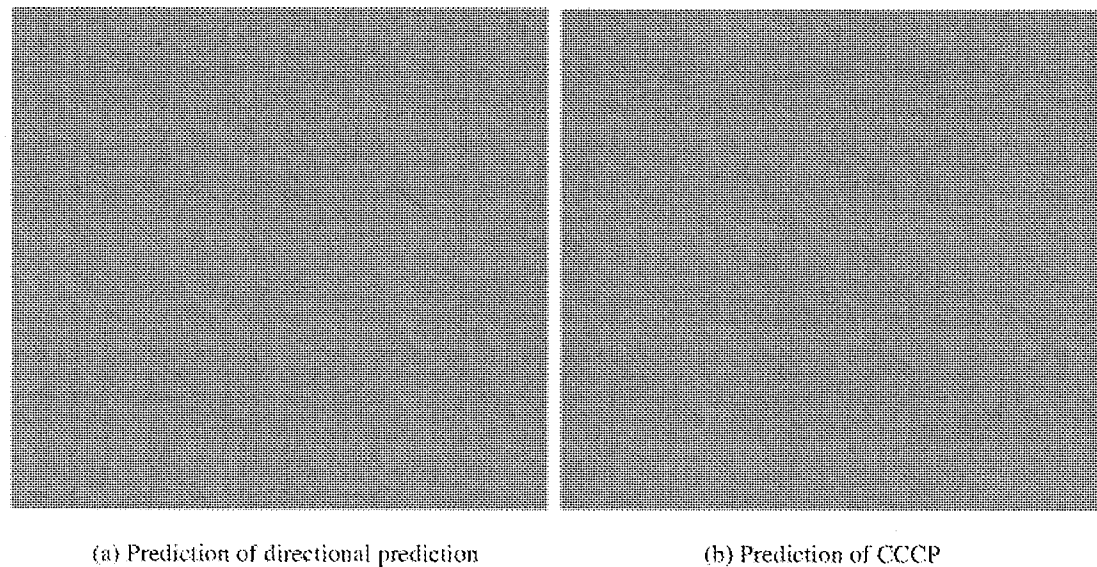


Figure 2-16 CCCP processing flow

Since CCCP utilizes already reconstructed luma blocks, it can generate complex shapes of object, which cannot be predicted well by the directional intra prediction technique. Figure 2-17 shows one typical example from the BasketballPass sequence.



(a) Prediction of directional prediction

(b) Prediction of CCCP

Figure 2-17 Visual prediction examples: (a) directional prediction (b) CCCP prediction

2.4.4 Pixel based template matching (PTM)

Whilst directional extrapolation based intra prediction tools are very efficient for regions with linear patterns, they show poor performance in regions with repeated regular patterns. Many previous intra prediction tools used some form of template matching to reduce this problem.

In this proposal, pixel based template matching is employed. Performing template matching on the pixel level results in different effects compared with previous techniques with a large template. A key difference is that already predicted samples in the same prediction unit are also used as template and search candidates for further pixel prediction.

The principle is illustrated in Figure 2-18 below. To predict a sample (PR), three neighbouring pixels (T0, T1 and T2) from above, left, and above-left are used as the template. After searching using this template, the pixel position with minimum template difference is found in the L-shaped search region (reconstructed or already predicted) of the current pixel. The chosen pixel (C) is set to the prediction value of the current pixel. Predicted pixels are also used as the template in the PTM.

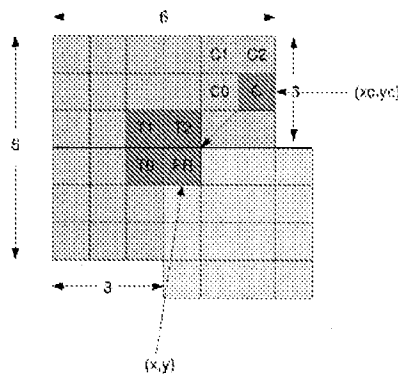


Figure 2-18 Pixel based template matching

2.5 Spatial transforms

2.5.1 Large transform (16x16, 32x32 and 64x64)

For smooth data, a large transform has several advantages such as better energy compaction and reduced quantization error. In HD sequences, most image patterns in a macroblock represent a small part of objects or backgrounds which can be described as homogeneous texture patterns with little variation. Therefore, the coding efficiency of high resolution video can be improved by the use of large transforms as well as large block sizes. By contrast, H.264/AVC supports only 4x4 and 8x8 transform sizes. In this proposal, larger transform sizes can be chosen (or set) for each coding unit (blocks or partitions). Transform sizes larger than the prediction unit can also be supported.

In this proposal, 3 additional sizes of transform are included: 16x16, 32x32 and 64x64. In large transform design, it is very important to minimize complexity. The transforms in this proposal are based on Chen's fast DCT algorithm [10]. Chen's algorithm is not the fastest one, but it has reduced implementation complexity due to the regular butterfly structure. Moreover, it is readily extensible to larger transform sizes.

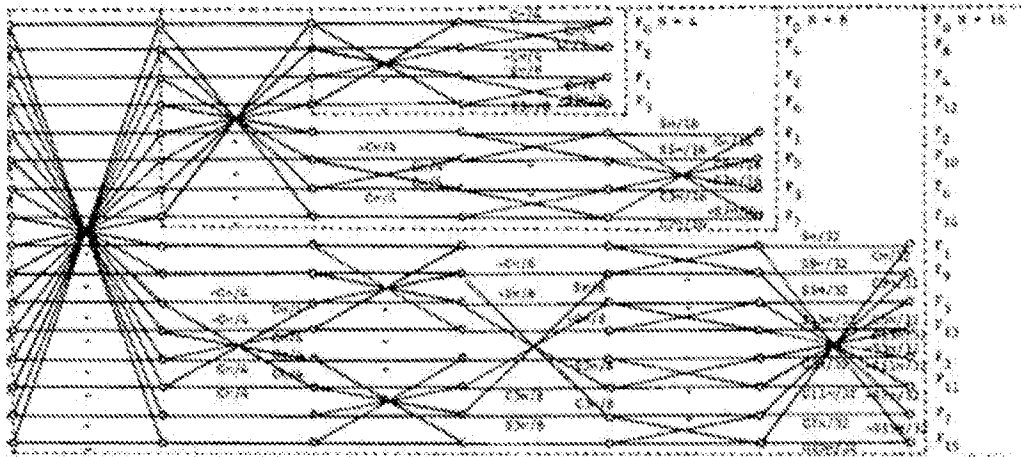


Figure 2-19 Signal flow graph of Chen's fast 16-point DCT transform

Figure 2-19 shows the signal flow graph of Chen's fast factorization of 16 point transform. In this figure, multiplication constants are represented by sinusoidal functions of specific angles, requiring floating point operations. To solve this problem, we scale and approximate the factors by fixed precision using pre-defined values, which can be calculated by cost effective shift operations. The approximated constants are shown in Table 4-10 the pre-defined precision value is 64 in this case. Here, a_k 's are approximated values of $\cos(k\pi/32)$ for $k=1,2,\dots,15$.

Table 2-9 Approximated constants for 16 point transform

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15
63/64	62/64	61/64	59/64	56/64	53/64	49/64	45/64	40/64	35/64	30/64	24/64	18/64	12/64	6/64

When the constant is approximated by the dyadic rational, the transform is no longer truly orthogonal. However, this minor non-orthogonality does not result in any perceptible negative effect on the compression performance, whilst the complexity can be significantly reduced.

2.5.2 Rotational transform (ROT)

DCT is the most widely used transform in block based video and image codecs. The input to the transform is the residual, i.e. the difference between the prediction and the original signal. But it is well-known that the DCT basis functions are not optimal for some types of residual signal. For example, the residual signal having strong diagonal components cannot be represented efficiently with the DCT basis vectors.

Typically, directional transform schemes are exploited to solve this problem since spatial domain rotation makes it difficult to maintain the original square shape of the block and generally requires many floating point operations. However, it's hard to develop directional transform cores, especially since many transforms of different sizes are needed. Instead of introducing completely new transform cores, ROT was designed as a 2nd transform that could be applied after the DCT operation.

The main idea of ROT is to change the coordination system of the transform basis, instead of direct rotation of the input source. For this purpose, the following matrices are defined as

$$R_{vertical}(\alpha_1, \alpha_2, \alpha_3) = \begin{pmatrix} \cos\alpha_1\cos\alpha_2 - \sin\alpha_1\cos\alpha_2\sin\alpha_3 & -\sin\alpha_1\cos\alpha_2 - \cos\alpha_1\cos\alpha_2\sin\alpha_3 & \sin\alpha_1\sin\alpha_3 & 0 \\ \cos\alpha_1\sin\alpha_2 + \sin\alpha_1\cos\alpha_2\cos\alpha_3 & -\sin\alpha_1\sin\alpha_2 + \cos\alpha_1\cos\alpha_2\cos\alpha_3 & -\sin\alpha_1\cos\alpha_3 & 0 \\ \sin\alpha_1\sin\alpha_2 & \cos\alpha_1\sin\alpha_2 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_{horizontal}(\alpha_4, \alpha_5, \alpha_6) = \begin{pmatrix} \cos\alpha_4\cos\alpha_5 - \sin\alpha_4\cos\alpha_5\sin\alpha_6 & -\sin\alpha_4\cos\alpha_5 - \cos\alpha_4\cos\alpha_5\sin\alpha_6 & \sin\alpha_4\sin\alpha_6 & 0 \\ \cos\alpha_4\sin\alpha_5 + \sin\alpha_4\cos\alpha_5\cos\alpha_6 & -\sin\alpha_4\sin\alpha_5 + \cos\alpha_4\cos\alpha_5\cos\alpha_6 & -\sin\alpha_4\cos\alpha_6 & 0 \\ \sin\alpha_4\sin\alpha_5 & \cos\alpha_4\sin\alpha_5 & \cos\alpha_4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $R_{vertical}$ and $R_{horizontal}$ represent the rotation matrices for horizontal and vertical directions with the rotation angles of $(\alpha_1, \alpha_2, \alpha_3)$ and $(\alpha_4, \alpha_5, \alpha_6)$, respectively. In this proposal, only four out of all possible rotation angles are quantized and used, in order to minimize the encoder-side complexity. Furthermore, since ROT is a second transform, not all of the DCT coefficients are processed. As shown in Figure 2-20, for the TUs of sizes larger than 4x4, only 8x8 low-frequency areas are rotated by multiplying by the ROT matrix since most coefficients are already compacted into these low-frequency areas. The ROT process is applied to all transform sizes but only for intra predicted residuals.

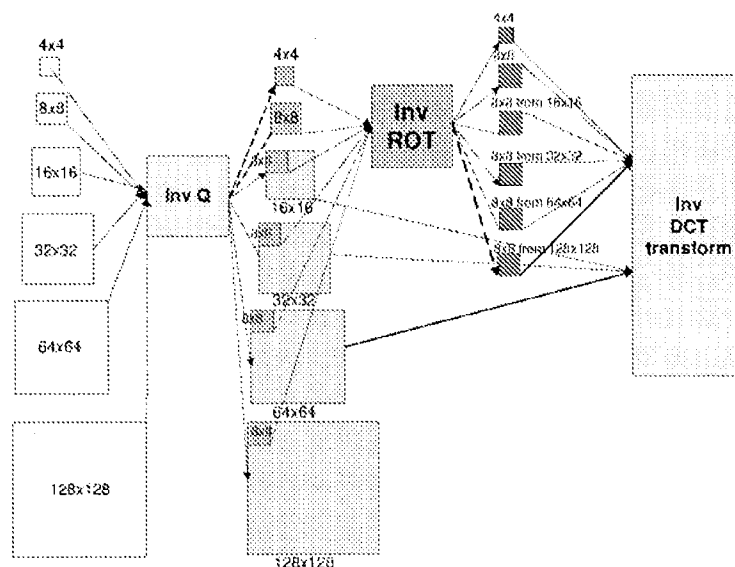


Figure 2-20 Overview of ROT process

2.5.3 Logical transform (LOT)

The LOT allows the input residual size to be bigger than the maximum physical transform size without multiple uses of transforms. If the maximum physical transform size is $N \times N$ and the residual to be transformed is of size $2N \times 2N$, the LOT process is applied to the residual. First, a $2N \times 2N$ 5-3 integer wavelet transform is performed and for the LL-band signal, then a normal DCT is applied. The LOT process is roughly equivalent to taking the $N \times N$ low-frequency components of $2N \times 2N$ DCT, thus it works well for the smooth regions which occur relatively frequently in high resolution video. It is beneficial since large transforms are typically selected for coding such a smooth region.

2.5.4 Quantization

The basic principle for the quantization and de-quantization coefficients for large transforms is the same as that used in H.264/AVC, i.e. a scalar quantizer with dead-zone. Exactly the same QP range and corresponding quantization step have been used in the proposed codec.

2.6 In-loop filtering

In this proposal, in-loop filter is a combination of several spatial processes: deblocking filtering, extreme correction, adaptive loop filtering, band correction and content adaptive dynamic range adaptation. Each process is explained in more detail within the following sections.



Figure 2-21 in-loop filter combination

2.6.1 Deblocking filter

As in H.264/AVC, a deblocking filter is employed in this proposal. The boundary strength decision is the same as in H.264/AVC, if the CU boundary is regarded as equivalent to a macroblock boundary. That is, coding modes of adjacent blocks, quantization step size, and the steepness of the luminance gradient between blocks are considered in the same way.

However, as this proposal supports variable sizes of CU, PU and TU while the boundary decision was designed for fixed sizes of macroblock and partition in H.264/AVC, a new boundary decision for this flexible structure has been developed. It is known that prediction units cause prediction error and transform units cause quantization error in lossy coding. These errors result in blocking artifacts. To compensate for these artifacts, we need to consider prediction boundaries and transform boundaries together. Unlike H.264/AVC, it is not guaranteed that the transform unit is always included in the prediction unit, thus the boundary for the deblocking process is chosen from the boundary of the smaller unit. The remaining process is identical to the deblocking filter in H.264/AVC, i.e. it uses the same filters and follows the same boundary strength decision.

However, the overall complexity is reduced in many cases, since the flexible hierarchical structure of the proposal often causes large blocks to be used, which reduces the total number of boundary pixels to be filtered in comparison with H.264/AVC.

2.6.2 CU-synchronized adaptive loop filter (ALF)

After extreme values are corrected, adaptive loop filtering is invoked. The purpose of this process is to further reduce the distortion between the original picture and the reconstructed picture caused by complex lossy coding. Filters minimizing the distortion for both luma and chroma components are calculated using the Wiener filter approach. After filters are applied in a frame or coding units, filter coefficients are explicitly sent in the bitstream. This adaptive loop filter increases reconstructed picture quality as well as reference picture quality for the next picture coding. Several adaptive loop filter schemes have been proposed: frame-based, block-based and quadtree-based [12] adaptive loop filters with different filtering control basis. The filtering control basis is usually obtained independently from the prediction structure, which results in a large amount of information being sent to the decoder and its inability to access prediction information in its process.

In this proposal, CU-synchronized ALF is presented. By utilizing the coding unit structure, the amount of control information to be sent is reduced. Due to the flexible coding unit structure from large to small block, coding unit boundary provides great flexibility to be used as control basis in the adaptive loop filter process. Since the optimal coding unit boundary obtained from previous coding stages reflects objects' boundaries having similar properties, like prediction and quantization error, a simple control scheme which infers filter control structure from the coding unit structure and the optimal control depth reduces the complexity of the optimal control map while achieving similar performance.

Once the control units for filtering are determined, a control flag indicating whether filtering is applied is sent to maximize filter performance and reduce complexity of filter application. Depending on the largest block size, the maximum depth, sequence property and coding parameters, the number of coding units can be too many to be used as filter control units itself. Thus the optimal control depth is sent to indicate the maximum depth (i.e., the smallest size) of a coding unit boundary to be used as filter control map.

2.6.3 Extreme correction (EXC) and band correction (BDC)

Both the adaptive loop filter and the deblocking filter can be considered as adaptive low-pass filtering. The deblocking filter works only for the block edge pixels whereas the adaptive loop filter reduces overall distortion by computing optimal filter coefficients. Both techniques are effective to reduce the distortion as well as to improve the subjective quality; however, they do not consider the local property of the contents, e.g. object edge. For this purpose, two further techniques, EXC and BDC, are proposed to reduce the distortion based on the pixel classification.

EXC tries to classify all pixels into several categories and compute correction value for each category based on the relationship between current pixel value and four neighboring pixels, left, above, right and bottom. Let P be the current pixel value. Table 2-10 shows the conditions used in the classification stage in EXC.

Table 2-10 Conditions used in the classification stage in EXC

Category	Condition	Note
0	P < 4 neighbors	Local min
1	P < 3 neighbors && P = 4 th neighbor	Object edge
2	P < 3 neighbors && P > 4 th neighbor	Object edge
3	P > 3 neighbors && P < 4 th neighbor	Object edge
4	P > 3 neighbors && P = 4 th neighbor	Object edge
5	P > 4 neighbors	Local max

For pixels in each category, the mean difference between the reconstructed signal and the original signal is computed and transmitted to the decoder. EXC is useful to correct the unusual values (category 0 and 5) or the pixels near the object edge (category 1, 2, 3 and 4).

To reduce the number of conditional operations, the following pseudo-code can be used to find the category:

```
ExtremeType = 0;
if (Rec[x][y] > Rec[x-1][y]) ExtremeType++;
if (Rec[x][y] < Rec[x-1][y]) ExtremeType--;
if (Rec[x][y] > Rec[x+1][y]) ExtremeType++;
if (Rec[x][y] < Rec[x+1][y]) ExtremeType--;
if (Rec[x][y] > Rec[x][y-1]) ExtremeType++;
if (Rec[x][y] < Rec[x][y-1]) ExtremeType--;
if (Rec[x][y] > Rec[x][y+1]) ExtremeType++;
if (Rec[x][y] < Rec[x][y+1]) ExtremeType--;
```

The resultant ExtremeType has distinct values of -4, -3, -2, 2, 3 and 4 for each category, respectively, thus 8 comparisons for each pixel are needed for the EXC.

In BDC, pixel classification is simply done according to the value ranges. The main motivation of this is to equalize two different PDF's of the reconstructed signal and the original signal. The p most significant bits of the pixels are used for the classification. For example, if p = 4, a total 16 bands are defined. For each band, the mean difference is computed and transmitted to the decoder.

2.6.4 Content adaptive dynamic range (CADR)

The CADR provides high accuracy for internal processes by exploiting given luma or chroma sample ranges. Unlike internal bit depth increase (IBDI) technique, CADR itself doesn't require bit depth increase. This makes it preferable for hardware implementations. In addition, CADR can be used with IBDI to further improve the possible accuracy.

The CADR process can be defined by two parameters: MaxOrg and MinOrg which represent the minimum and maximum values of the input source, respectively. In principle, those parameters can be estimated on a slice-by-slice basis to maximize the effect, but this proposal simply uses the safe range between [16, 235] from the definition in BT.709 in order to reduce the encoder complexity.

Once the parameters are given, new values F(x) can be simply generated by

F(x) = (2^{bit-depth} - 1) * (x - MinOrg) / (MaxOrg - MinOrg)

It should be noted that the conversion can be simply implemented by table look-up. Figure 2-22 illustrates a mapping function F(x) when the range of [16, 235] is used.

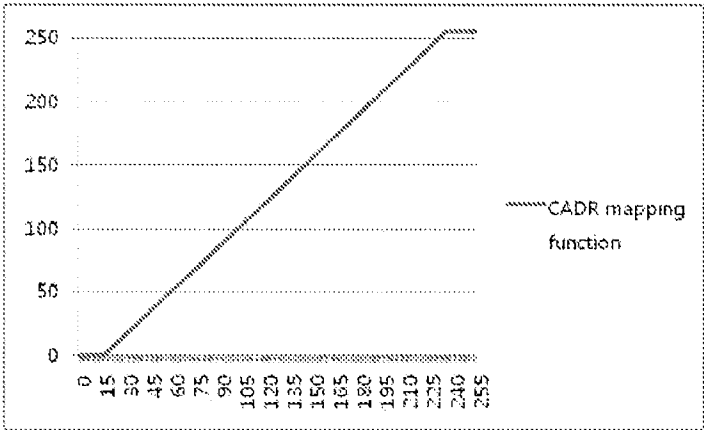


Figure 2-22 Example of a mapping function of CADR

Table 2-11 shows an example of the performance related to the various combinations of IBDI and CADR, using a one second segment of the Class D sequences. These results show that both tools are useful, although CADR has the advantage of requiring only 8-bit bit-depth.

Table 2-11 Performance of various combinations of IBDI and CADR (1s, GOP 8)

Sequence	CADR range	IBDI off, CADR on BD-rate (%)	IBDI (4-bit) on, CADR off BD-rate (%)	IBDI (4-bit) on CADR on BD-rate (%)
BasketballPass	[16, 235]	1.86	1.28	3.92
BQSquare	[16, 235]	4.34	1.86	5.90
BlowingBubbles	[16, 235]	2.09	1.27	3.74
RaceHorses	[16, 235]	1.72	1.20	3.49

2.7 Entropy coding

2.7.1 Syntax-based context-adaptive binary arithmetic coding (SBAC)

Entropy coding, which improves the coding performance by exploiting the statistical redundancy in the syntax elements of a video sequence, is an essential element of all current video codecs, including this proposal. Variable-length Coding (VLC) methods and their context-adaptive variants, e.g. the Context-Adaptive Variable-Length Coding (CAVLC) of H.264/AVC, have been widely used in the past due to their low complexity and ease of implementation. However, VLC-based methods are inherently sub-optimal in terms of coding performance, and it is hard to make them adapt to non-stationary symbols. Arithmetic coding is known to outperform VLC-based methods and can be well adapted to the varying statistical properties of the input symbols. However, its high complexity has hindered its adoption in video coding standards. Recently, due to the needs for higher coding efficiency and the advances in available computing power of codec platforms, arithmetic coding methods, e.g. Syntax-based Arithmetic Coding (SAC) of H.263 and Context-Adaptive Binary Arithmetic Coding (CABAC) of H.264/AVC, are adopted and widely used in the state-of-the-art video codecs.

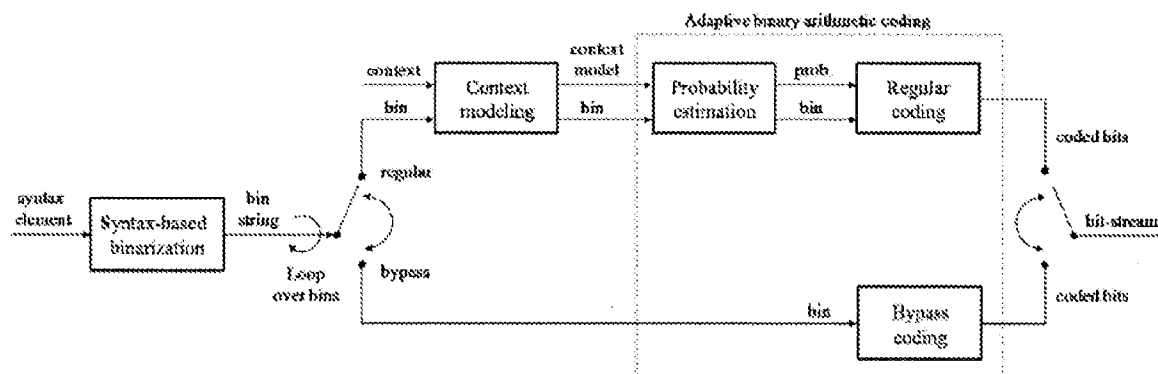


Figure 2-23 Block diagram of SBAC encoder

In this proposal, Syntax-based context-adaptive binary arithmetic coding (SBAC) is used for the entropy coding of syntax elements. SBAC is an adaptive binary arithmetic coding method with context modeling, which can provide high coding efficiency for various syntax elements with varying statistical properties. The coding core of SBAC is based on the arithmetic coding method specified in Annex D of JPEG still image coding standard, although it has been re-designed to meet the requirements of the proposal. Furthermore, a number of features desirable for video codec, e.g. bypass coding mode to reduce complexity, are also supported.

As shown in Figure 2-23, the encoding process of SBAC is composed of four major steps – syntax-based binarization, context modeling, probability estimation, and binary arithmetic coding. Even though the overall structure of SBAC is similar to that of the well-known CABAC, the details of each step are different from those of CABAC, and these differences lead to better coding performance.

- 1) **Syntax-based binarization** – to limit complexity and avoid context dilution problems, SBAC uses a binary arithmetic coding core. Since only binary values can be coded using SBAC, any non-binary syntax elements such as motion vector difference values must be binarized. For each syntax element, SBAC chooses an appropriate binarization method which can provide the best possible coding performance. For example, for binarization of the partition size, a modified unary code taking into account the geometry of the partition has been designed. Note that, when a given syntax element is binary, the binarization process is skipped.
- 2) **Context modeling** – the performance of an arithmetic coding method is dependent on the accuracy of the probability model that the coder uses. SBAC uses a context model, which is a probability model for one or more binary values to be coded, to provide an accurate estimate of the probability for each binary value. For each bin of each syntax element, a set of one or more context models is provided, and one of them is selected based on a given context and the bin value currently being coded. The rule for the selection of the context models is defined for each bin of each syntax element, and designed to provide the most relevant probability estimate to the coding engine.
- 3) **Probability estimation** – in SBAC, each context model consists of the most probable symbol (MPS) and the probability of the least probable symbol (P_{LPS}), and they are updated at each coding process based on the previous state of the context model and the coded binary value. For accurate and fast estimation, a finite state machine (FSM) based method is used for probability estimation and update.
- 4) **Binary arithmetic coding** – the binary arithmetic coding engine of SBAC is based on the arithmetic coding method specified in Annex D of the JPEG still image coding standard. Given a binary value to be coded and the corresponding context model, the binary arithmetic coding engine performs entropy coding based on its internal parameters. To improve the coding performance and meet the requirements of the video codec, the coding engine has been re-designed. For example, the table for probability estimation has been completely modified to be more accurate and adapt faster. The multiplication-free design has also been modified to improve performance while maintaining low complexity. A bypass coding mode is used for syntax elements with approximately equi-probable property, such as sign of motion vector difference. Note that, when the bypass coding mode is used, context modeling and probability estimation are skipped.

2.7.2 Adaptive coefficient scanning (ACS)

There are several elements in the codec which should be adapted to maximize the effect of large transforms. The transform coefficient scanning method is one of them. Since large transforms frequently have very sparse coefficient distributions, a fixed zigzag based scanning method often results in long coefficient scanning lists including many zeroes. The Adaptive Coefficient Scanning technique (ACS) was designed to improve the coding performance of coefficients of large transforms.

For each transform unit, the ACS index is parsed. Based on the ACS index, the scanning method is chosen from 0: conventional zigzag scan, 1: horizontal scan, 2: vertical scan. The ACS index is only necessary when there are non-DC coefficients.

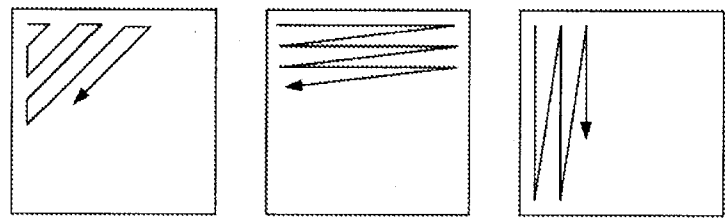


Figure 2-24 Zig-zag, horizontal and vertical scanning

3 Compression performance discussion

3.1 Objective versus subjective compression performance

The coding tools in this proposal represent an extension of the type of tools (prediction + DCT-based transform + entropy coding) used in H.264/AVC. As a result, there is not expected to be any really major difference with H.264/AVC in the way in which subjective and objective assessments compare.

One exception to this may be in the subjective visibility of blockiness, where the use of a large transform in the proposal can be expected to generate less blocky images, since the total number of edge pixels is decreased.

3.2 Constraint set 1 configuration relative to Alpha anchor

For satisfying constraint set 1, a typical hierarchical-B structure of GOP size 8 was used as specified in the joint CFP document [13], thus the number of frames required to decode a given frame is exactly the same as the Alpha anchor. Further information for test configuration is as follows with bold marks indicating differences compared to the Alpha anchor.

- Hierarchical B pictures **IbBbBbBbP** (8) coding structure – each picture uses 2 reference pictures for forward prediction and 2 reference pictures for backward prediction.
 - Open GOP structuring with an intra picture every 24, 32, 48 and 64 pictures for 24 fps, 30 fps, 50 and 60 fps sequences, respectively
 - QP scaling: QP (I picture), QP+1 (P picture), QP+2 (first B layer), QP+3 (second B layer), QP+4 (third B layer)
 - Same random access capability to Alpha anchor
- Flat quantization weighting matrices
- RD optimization enabled
- RDOQ enabled (fast mode, NUM=1)
- Adaptive rounding disabled
- **Weighted prediction disabled**
- Fast motion estimation (range 128x128)
- **Picture-level multi-loop encoding is disabled**
- **Largest block size of 128x128 is used except for Class D which uses 64x64**
- **12-tap DIF for luma and 6-tap DIF for chroma are used**

Table 3-1 shows the summarized performance of the proposed codec relative to the Alpha anchor. The average compression performance improvement over all sequences is 39.49%. Class B (1080p resolution) achieves 43.29% bit saving on average, with BQTerrace as the best sequence, showing 54.06% performance improvement. RaceHorses in Class D shows the lowest performance improvement, at 27.08%.

Table 3-1 Summarized performance relative to Alpha anchor

Class	Resolution	Seq. No	Seq. Name	BD-rate (%)
A	4K	S01	Traffic	40.59
		S02	PeopleOnStreet	29.84
B	1080p	S03	Kimono	44.42
		S04	ParkScene	32.96
		S05	Cactus	40.78
		S06	BasketballDrive	44.21
		S07	BQTerrace	54.06
C	WVGA	S08	BasketballDrill	43.09
		S09	BQMall	39.69
		S10	PartyScene	41.13
		S11	RaceHorses	35.86
D	WQVGA	S12	BasketballPass	30.55
		S13	BQSquare	54.19
		S14	BlowingBubbles	33.97
		S15	RaceHorses	27.08
Summary				
A				35.21
B				43.29
C				39.94
D				36.45
All				39.49

3.3 Constraint set 2 configuration relative to Beta and Gamma anchors

For satisfying constraint set 2, only the first frame is coded as intra frame and backward temporal prediction is prohibited in rest of frames, whilst maintaining the same display and coding order as the anchors. A hierarchical P structure with GOP of 4 was used, however IPPP structure was used to keep the bit-rate fluctuation as small as possible when the performance difference between two temporal structures is not significant. Detailed information for the test configuration is as follows with bold marks of differences compared the Beta anchor and Gamma anchor.

- No random access refresh requirement (a single I frame as the first picture)
- Hierarchical P pictures IpPp (GOP size of 4, with 3 temporal P picture layers, and no backward reference for inter prediction) coding structure – each picture uses at most 4 reference pictures for inter prediction
 - QP scaling: QP (I picture), QP+1 (first P layer), QP+4 (second P layer), QP+5 (third P layer)
 - Same structural delay constraints to Beta anchor
- **IPPP coding structure IPPP – no non-referenced frames are used**
 - QP scaling: QP (I picture), QP+1 (first P layer), QP+2 (second P layer), QP+3 (third P layer)
 - **Same structural delay constraints to Gamma anchor**
- Flat quantization weighting matrices
- RD Optimization enabled
- RDOQ enabled (fast mode, NUM=1)
- Adaptive rounding disabled
- **Weighted prediction disabled**
- Fast motion estimation. (range 128x128)
- **Picture-level multi-loop encoding is disabled**
- **Largest block size of 128x128 is used except Class D which uses 64x64**
- **12-tap DIF for luma and 6-tap DIF for chroma are used**

Table 3-2 shows the summarized performance of the proposed codec relative to the Beta anchor sequences. Average performance improvement over all sequences is 39.48% which is almost the same improvement as for Constraint Set 1. In particular, Class B (1080p resolution) achieves 43.66% bit saving on average. The best performance improvement over a Class is obtained with Class E, at 47.74%. BQTerrace shows the greatest performance improvement for a single sequence, at 60.62%. RaceHorses in Class D shows the lowest performance improvement, with 21.59% bit saving.

Table 3-2 Summarized performance relative to Beta anchors

Class	Resolution	Seq. No	Seq. Name	BD-rates (%)	Temporal structure
B	1080p	S03	Kimono	44.45	IPPP
		S04	ParkScene	31.18	Hier-P
		S05	Cactus	34.74	Hier-P
		S06	BasketballDrive	47.29	IPPP
		S07	BQTerrace	60.62	Hier-P
C	WVGA	S08	BasketballDrill	36.82	Hier-P
		S09	BQMall	37.06	IPPP
		S10	PartyScene	40.45	Hier-P
		S11	RaceHorses	27.99	IPPP
D	WQVGA	S12	BasketballPass	26.93	IPPP
		S13	BQSquare	51.87	Hier-P
		S14	BlowingBubbles	28.24	Hier-P
		S15	RaceHorses	21.59	IPPP
E	720p	S16	Vidyo1	48.38	Hier-P
		S17	Vidyo3	46.26	Hier-P
		S18	Vidyo4	47.79	Hier-P
Summary					
B				43.66	
C				35.58	
D				32.16	
E				47.48	
All				39.48	

Table 3-3 shows the summarized performance of the proposed codec relative to the Gamma anchor sequences. The average performance over all sequences is about 15% higher than for the Beta anchors.

Table 3-3 Summarized performance relative to Gamma anchors

Class	Resolution	Seq. No	Seq. Name	BD-rate (%)
B	1080p	S03	Kimono	58.01
		S04	ParkScene	47.92
		S05	Cactus	52.48
		S06	BasketballDrive	59.15
		S07	BQTerrace	72.87
C	WVGA	S08	BasketballDrill	54.37
		S09	BQMall	50.75
		S10	PartyScene	59.80
		S11	RaceHorses	36.97
D	WQVGA	S12	BasketballPass	37.70
		S13	BQSquare	74.25
		S14	BlowingBubbles	49.17
		S15	RaceHorses	28.62
E	720p	S16	Vidyo1	60.98
		S17	Vidyo3	59.55
		S18	Vidyo4	62.63
Summary				
B				58.09
C				50.47
D				47.44
E				61.05
All				54.08

4 Complexity analysis

4.1 Encoding time and measurement methodology

For encoding time measurement, a clustered system with multiple CPUs and RAMs was used. Each CPU has at most similar clock-speed with AMD Opteron 2.6GHz. The 32 bit executable is generated by GCC 4.12. The execution time is measured on RedHat Linux using intrinsic API functions of Linux.

Table 4-1 Results of encoding time measurement for constraint set 1

Class	Resolution	Seq. No	Seq. Name	Enc. Time (h/QP)
A	4K	S01	Traffic	60.42
		S02	PeopleOnStreet	67.55
B	1080p	S03	Kimono	76.53
		S04	ParkScene	58.28
		S05	Cactus	113.01
		S06	BasketballDrive	113.01
		S07	BQTerrace	96.00
C	WVGA	S08	BasketballDrill	24.02
		S09	BQMall	28.46
		S10	PartyScene	19.82
		S11	RaceHorses	24.68
D	WQVGA	S12	BasketballPass	5.94
		S13	BQSquare	4.63
		S14	BlowingBubbles	3.83
		S15	RaceHorses	4.47
Summary				
A				63.99
B				91.37
C				24.24
D				4.72

Table 4-2 Results of encoding time measurement for constraint set 2

Class	Resolution	Seq. No	Seq. Name	Enc. Time (h/QP)
B	1080p	S03	Kimono	51.41
		S04	ParkScene	43.49
		S05	Cactus	97.98
		S06	BasketballDrive	90.07
		S07	BQTerrace	104.21
C	WVGA	S08	BasketballDrill	19.69
		S09	BQMall	17.91
		S10	PartyScene	14.23
		S11	RaceHorses	18.06
D	WQVGA	S12	BasketballPass	3.91
		S13	BQSquare	6.19
		S14	BlowingBubbles	2.46
		S15	RaceHorses	2.87
E	720p	S16	Vidyo1	27.06
		S17	Vidyo3	25.36
		S18	Vidyo4	28.63
Summary				
B				77.43
C				17.47
D				3.86
E				27.02

4.2 Decoding time and measurement methodology

For the decoding time measurements, the test platform is a PC based on Intel Core2 Quad CPU Q9400 (2.66GHz) and 2GB RAM. The 32-bit executables of JM17.0 and the proposed codec were generated by Intel compiler 11.0.072. The execution time was measured on Windows XP Professional SP3 using intrinsic API functions of Windows. As recommended in the CIP document and the reflector, YUV writing was enabled. However, JM 17.0 exhibits unusual decoding performance depending on HDD configuration. It was observed that decoding time on a SCSI drive was very much increased compared to a SATA drive, presumably due to optimization for fast disk access. The results in this section are therefore presented for both drive configurations.

Only one core was utilized for measuring the decoding time while other cores remain empty. Table 4-3 and 4-4 show the decoding time for the submitted bitstreams under constraint set 1 for both disk configurations. Tables 4-5 and 4-6 do the same for constraint set 2.

As shown in the tables, with a fast SATA drive, the average decoding time of the proposed codec is 2.4 times that of JM17.0 for constraint set 1 and 2.3 times that of JM17.0 for constraint set 2. With a SCSI drive, the average decoding time is approximately 0.9 times that of JM17.0 for both configurations.

The proposal software has a high level of maturity, but it has not received the same level of optimization as JM 17.0. It is probably safe to conclude that the intrinsic algorithmic complexity of the proposed codec is, at worst, no more than moderately higher than H.264/AVC.

Table 4-3 Results of decoding time measurement for constraint set 1 (SATA drive)

Class	Resolution	Seq. No	Seq. Name	JM17.0 Time (sec/QP)	JM17.0 Fps	Prop. Time (sec/QP)	Prop. Fps	Rel. Time (Prop/JM)
A	4K	S01	Traffic	30.17	5.00	85.29	1.76	2.84
		S02	PeopleOnStreet	29.75	5.10	94.51	1.60	3.18
B	1080p	S03	Kimono	33.12	7.31	75.56	3.19	2.28
		S04	ParkScene	32.01	7.58	71.24	3.41	2.23
		S05	Cactus	53.75	9.36	117.49	4.31	2.19
		S06	BasketballDrive	65.67	9.19	165.20	3.65	2.52
		S07	BQTerrace	77.73	7.74	164.48	3.67	2.12
C	WVGA	S08	BasketballDrill	11.34	44.41	20.03	25.52	1.77
		S09	BQMall	13.92	43.16	24.47	25.12	1.76
		S10	PartyScene	11.59	43.28	23.12	22.17	1.99
		S11	RaceHorses	7.88	38.21	17.30	17.99	2.20
D	WQVGA	S12	BasketballPass	2.85	180.62	6.68	77.32	2.35
		S13	BQSquare	3.42	177.30	8.44	72.03	2.47
		S14	BlowingBubbles	2.56	200.07	6.08	86.18	2.38
		S15	RaceHorses	2.06	151.43	5.37	58.24	2.61
Summary								
A				29.96	5.05	89.90	1.68	3.01
B				52.46	8.23	118.79	3.64	2.26
C				11.18	42.27	21.23	22.70	1.93
D				2.72	177.36	6.64	73.44	2.45
ALL								2.41

Table 4-4 Results of decoding time measurement for constraint set 1 (SCSI drive)

Class	Resolution	Seq. No	Seq. Name	JM17.0 Time (Sec/QP)	JM17.0 Fps	Prop. Time (sec/QP)	Prop. Fps	Rel. Time (Prop/JM)
A	4K	S01	Traffic	137.04	1.09	83.39	1.80	0.61
		S02	PeopleOnStreet	136.91	1.10	91.85	1.64	0.67
B	1080p	S03	Kimono	114.98	2.09	74.25	3.25	0.65
		S04	ParkScene	112.53	2.13	70.84	3.43	0.63
		S05	Cactus	224.86	2.23	118.67	4.25	0.53
		S06	BasketballDrive	238.26	2.52	162.21	3.71	0.68
		S07	BQTerrace	288.85	2.08	161.88	3.73	0.56
C	WVGA	S08	BasketballDrill	29.21	17.12	21.33	23.66	0.73
		S09	BQMall	35.42	16.95	25.42	23.95	0.72
		S10	PartyScene	30.00	16.68	22.76	22.50	0.76
		S11	RaceHorses	18.31	16.41	17.47	17.63	0.95
D	WQVGA	S12	BasketballPass	4.67	107.80	6.57	78.12	1.41
		S13	BQSquare	5.55	109.05	8.09	75.20	1.46
		S14	BlowingBubbles	4.76	106.71	6.12	83.46	1.28
		S15	RaceHorses	3.17	95.63	5.19	59.78	1.64
Summary								
A				136.98	1.10	87.62	1.72	0.64
B				195.89	2.21	117.57	3.67	0.61
C				28.23	16.79	21.75	21.93	0.79
D				4.54	104.80	6.49	74.14	1.45
ALL								0.87

Table 4-5 Results of decoding time measurement for constraint set 2 (SATA drive)

Class	Resolution	Seq. No	Seq. Name	JM17.0 Time (sec/QP)	JM17.0 Fps	Prop. Time (sec/QP)	Prop. Fps	Rel. Time (Prop/JM)
B	1080p	S03	Kimono	28.01	8.66	77.23	3.13	2.77
		S04	ParkScene	26.64	9.15	65.41	3.74	2.46
		S05	Cactus	48.85	10.28	120.74	4.23	2.47
		S06	BasketballDrive	54.90	11.10	169.84	3.56	3.09
		S07	BQTerrace	63.94	9.42	164.61	3.66	2.57
C	WVGA	S08	BasketballDrill	10.54	47.60	22.47	22.80	2.13
		S09	BQMall	13.74	43.95	26.01	23.61	1.89
		S10	PartyScene	10.56	48.21	20.12	25.24	1.90
		S11	RaceHorses	6.89	44.17	19.68	15.86	2.86
D	WQVGA	S12	BasketballPass	2.48	207.22	7.06	72.87	2.85
		S13	BQSquare	3.18	192.31	7.63	80.19	2.40
		S14	BlowingBubbles	2.58	199.97	5.65	90.48	2.19
		S15	RaceHorses	1.98	159.36	6.12	50.96	3.09
E	720p	S16	Vidyo1	24.58	24.59	41.08	14.87	1.67
		S17	Vidyo3	23.30	25.75	39.56	15.37	1.70
		S18	Vidyo4	26.19	23.14	40.68	15.00	1.55
Summary								
B				44.47	9.72	119.57	3.66	2.67
C				10.43	45.98	22.07	21.88	2.20
D				2.55	189.72	6.62	73.68	2.63
E				24.69	24.50	40.44	15.08	1.64
ALL								2.29

Table 4-6 Results of decoding time measurement for constraint set 2 (SCSI drive)

Class	Resolution	Seq. No	Seq. Name	JM17.0 Time (sec/QP)	JM17.0 Fps	Prop. Time (sec/QP)	Prop. Fps	Rel. Time (Prop/JM)
B	1080p	S03	Kimono	108.53	2.21	75.92	3.18	0.70
		S04	ParkScene	107.89	2.23	64.35	3.79	0.60
		S05	Cactus	227.47	2.20	119.09	4.29	0.52
		S06	BasketballDrive	231.37	2.59	165.10	3.66	0.71
		S07	BQTerrace	276.88	2.17	161.49	3.73	0.58
C	WVGA	S08	BasketballDrill	27.90	17.95	22.84	22.22	0.82
		S09	BQMall	34.12	17.60	26.08	23.41	0.76
		S10	PattyScene	28.62	17.51	20.92	24.16	0.73
		S11	RaceHorses	17.77	16.93	19.32	16.07	1.09
D	WQVGA	S12	BasketballPass	4.47	113.16	6.85	75.20	1.53
		S13	BQSquare	5.30	114.04	7.62	79.50	1.44
		S14	BlowingBubbles	4.65	108.85	5.87	85.85	1.26
		S15	RaceHorses	3.17	96.43	5.78	54.11	1.82
E	720p	S16	Vidyo1	96.25	6.24	46.36	12.99	0.48
		S17	Vidyo3	94.33	6.36	45.16	13.35	0.48
		S18	Vidyo4	94.97	6.32	44.80	13.42	0.47
Summary								
B				190.43	2.28	117.19	3.73	0.62
C				27.10	17.50	22.29	21.47	0.85
D				4.40	108.12	6.53	73.67	1.51
E				95.18	6.31	45.44	13.25	0.48
ALL								0.87

4.3 Expected memory usage of encoder

The required memory of the proposed encoder is comparable to H.264/AVC. Although the size of several temporal buffers is increased according to the LCU size and the maximum depth, it is not a major impact compared to the reference memory size.

4.4 Expected memory usage of decoder

The required memory of the proposed decoder is comparable to H.264/AVC except the adaptive loop filter. Since the adaptive loop filter is applied to the reconstruction signal on the frame-basis, one additional frame buffer storing the reconstruction signal before applying the adaptive loop filter is required. However, because the number of filter taps is limited, the required buffer size can be reduced significantly with some increases of the buffer management complexity.

4.5 Complexity characteristics of encoder motion estimation and motion segmentation selection

4.5.1 Motion estimation

In this proposal, motion vector accuracy normally ranges from integer to 1/4 pel with the addition of eight 1/12 pel accuracy positions. Therefore, in principle it covers an area 9 times as large as H.264/AVC motion grid.

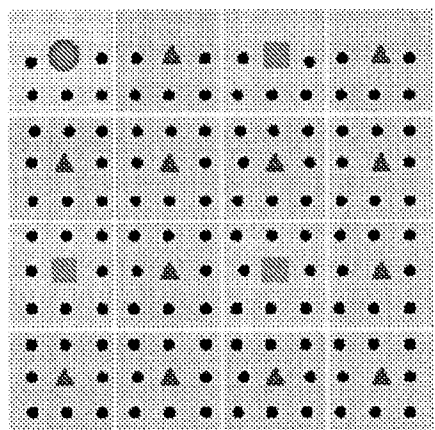


Figure 4-1 Fractional pel grid

However, even though its coverage is much larger than H.264/AVC, the complexity of motion estimation is only marginally increased because motion estimation for 1/12 pel accuracy can be realized by 9 refinements around the best 1/4 accuracy position. Luma SAD was used for the integer motion search and luma hadamard for the fractional motion search.

4.5.2 Motion segmentation selection

Motion segmentation (or motion partition) is done by the typical rate-distortion optimization process. SSE was used to compute the distortion for motion segmentation. Compared to H.264/AVC, four shapes of asymmetric motion partition are added and the supported hierarchy level is increased. As a result, encoder complexity in this implementation is higher than H.264/AVC. However, large block search costs can be ameliorated in real implementations by well-known techniques (phase correlation, hierarchical block matching, etc.).

4.6 Complexity characteristics of decoder motion compensation

4.6.1 Interpolation process

As already stated in Section 2.3.4.3, the proposed DIF is a non-cascaded interpolation scheme, i.e. only one spatial filtering is performed for all possible positions, which can be implemented by a multiplication-free design. Thus, the 6-tap DIF has the smaller number of operations compared to H.264/AVC since pixels at the quarter-pel position can be generated by one 6-tap filtering operation whereas H.264/AVC uses the combination of 6-tap and bi-linear filters.

The 12-tap DIF filter requires two times as many operations as the 6-tap DIF, however it provides similar performance to the well-known adaptive interpolation filter (AIF) scheme whilst it has much less complexity and is more suitable for hardware implementation.

High accuracy motion (HAM) also uses one spatial filtering to get 1/12 accuracy position, so the complexity characteristic is similar to the DIF. It's interesting to see that the complexity of DIF is virtually the same at any arbitrary accuracy point. This property allows the design of motion compensation processes based on arbitrary motion accuracy without increasing the complexity according to the supported motion accuracy.

4.6.2 Number of motion compensations

Many operations such as motion vector access, motion vector predictor computation and reference position computation are performed once for each motion compensation unit. Therefore, the average number of motion compensations is very important in reducing this kind of complexity. Table 4-7 shows the average number of motion compensations collected from the Class B streams of constraint set 1. It is very clear that the proposed method requires only 1/3 of the number of motion compensations compared to the Alpha anchor, mainly due to the use of large motion compensation units, which naturally results in the complexity reduction. It also means that the memory bandwidth required to access the motion information can be reduced significantly.

Table 4-7 Number of motion compensations in Class B streams

Sequence	Alpha anchor	Proposed method	Ratio
Kimono	1896948	577465	30.4%
Parkscene	2056280	811951	39.4%
Cactus	4134668	1420699	34.3%
BasketballDrive	3865489	1159876	30.0%
BQTerrace	5198921	1488036	28.6%
Average	-	-	32.5%

4.7 Complexity characteristics of encoder intra-frame prediction type selection

For ADI, the number of directions depends upon the PU splitting. For the sizes of 4x4 and 8x8, 9 directions are used whereas 33 directions are supported for the sizes of 16x16 and 32x32. For the sizes of 64x64 and 128x128, only 5 directions are used. Once the PU size is given, RD comparison is performed for all possible directions.

In addition to the ADI predictions, the encoder performs pixel-based template matching and compares the result with the best ADI mode in terms of coding efficiency.

After ADI and pixel based template matching have determined the best luma prediction mode, the MPI decision is made. This determines whether the best performance is gained by applying the 4 pixel smoothing filter or not.

For chroma samples, the number of prediction modes is determined by the best prediction modes for luma samples. If PTM mode is chosen, or the best ADI modes are larger than DC (2), 5 prediction modes are supported. Otherwise, only 4 modes are used for chroma samples. For chroma predictions, the CCCP process replaces the DC prediction. Although CCCP performs region segmentation for luma samples, the process is not computationally demanding compared with other segmentation based algorithms, since it uses a simple thresholding approach.

4.8 Complexity characteristics of decoder intra-frame prediction operation

Where an ADI mode is signaled, the decoder performs the appropriate directional prediction by extrapolating reconstructed pixels based on the appropriate (dx, dy) direction. As a result, prediction of significant numbers of pixels in large blocks can be achieved by copying from referenced pixels.

While ADI is generally applied for most prediction units, pixel based template matching and CCCP are selected for only a small portion of prediction units. Therefore, although more complex, their effects on total decoder complexity are not as large as ADI.

The complexity of the pixel based template matching can be described in a similar way to motion estimation. Full integer pel search is applied in the block. For each pixel the search range is up to 27 pixels and the size of template is 3 pixels (refer to Figure 2-18).

For luma samples with MPI mode =1, the 4 pixel based smoothing operation is applied for each pixel after the first intra prediction stage.

For chroma samples which have DC prediction mode, CCCP is applied. The CCCP consists of 1) bilinear down-sampling of reconstructed luma 2) calculation of mean values of down-sampled luma samples, 2) thresholding by mean value 3) assigning the context pixels based on the thresholding results, 4) smoothing by 3x3 averaging filter. Averaging by 3x3 filter can be efficiently implemented by 3-tap averaging in horizontal direction following by 3-tap averaging in vertical direction. Also, the result of summation can be shared for neighbor pixels, resulting in further complexity reduction.

4.9 Complexity characteristics of encoder transforms and transform type selection

For a given PU splitting, there could be two kinds of TU sizes. Between the two candidates, the best TU size is determined to be the one which produces smaller rate-distortion costs. The complexity of the transform itself is described in Section 4.10 in detail.

For intra CU, ROT is followed by the DCT. In the process to determine the best TU size, the optimal ROT matrix between the 4 possible matrices is determined to be the one which provides the smallest RD cost. To minimize the related complexity for finding the best ROT matrix, the ROT search is invoked only if the best prediction mode is intra-type. The complexity of ROT itself is described in Section 4.10.2.

4.10 Complexity characteristics of decoder inverse transform operations

4.10.1 Large size DCT

In this section, an analysis of the complexity of inverse transforms based on the number of operations is presented. First, the number of operations required for a 16x16 block is summarized in Table 4-8. In this table, + and >> mean addition and shift operation. For a 16x16 block, sixteen 4x4 transforms or four 8x8 transforms are required to process entire block.

Table 4-8 Number of operations per 16x16 block

Transform size	Operation	Number / 16x16	Total operations
4x4	+	256	320
	>>	64	
8x8	+	256	336
	>>	80	
16x16	+	148	288 (with 88 mult.)
	>>	52	
	*	88	

To make the transform multiplication-free, simple conversions rule can be applied. Since the constant is dyadic rational, it can be factorized into the addition and/or subtraction of several dyadic rationals which have 1 as a numerator. For example, 3/8 can be factorized into 1/4 + 1/8 = (1 + (1<<1))>>3. So in this case 1 multiplication and 1 shift can be calculated by only 2 shifts and one addition without loss. Table 4-9 shows the multiplication-free results.

Table 4-9 Number of operations per 16x16 block, multiplication-free 16x16 transform

Transform size	Operation	Number / 16x16	Total operations
4x4	+	256	320
	>>	64	
8x8	+	256	336
	>>	80	
16x16	+	300	536
	>>	236	

The 32x32 transform and 64x64 transform are constructed in the similar manner to the 16x16 transform described above. The number of operations per 64x64 block is summarized in Table 4-10.

Table 4-10 Number of operations per 64x64 block

Transform size	Operation	Number / 64x64	Total operations	Ratio vs. 8x8
4x4	+	4096	5120	0.95
	>>	1024		
8x8	+	4096	5376	1.00
	>>	1280		
16x16	+	2368	4608 (with 1408 mult.)	-
	>>	832		
	*	1408		
16x16 multi. free (multi. free)	+	4800	8576	1.60
	>>	3776		
32x32	+	1552	3008 (with 928 mult.)	-
	>>	528		
	*	928		
32x32 (multi. free)	+	3872	6968	1.30
	>>	3096		
64x64	+	964	1872 (with 584 mult.)	-
	>>	324		
	*	584		
64x64 (multi. free)	+	2832	5200	0.97
	>>	2368		

Note that the total number of operations per 64x64 block may become smaller when the transform size becomes larger, as shown in Table 4-10. In particular, the 64x64 multiplication-free transform requires only a similar number

of operations as a 4x4 integer transform. Although 16x16 and 32x32 transforms require about 60% and 30% more operations respectively compared to the conventional 8x8 integer transform, the overall impact on the decoder complexity is not large since the portion of the inverse transform complexity among total decoder complexity is generally limited.

4.10.2 ROT

The ROT operation is two matrix multiplication operations. To minimize the complexity burden, ROT is applied only for the low-frequency elements which size is limited to 8x8 for any transform size. The total number of operations required for the inverse ROT of size 4x4 and 8x8 is represented in Table 4-11.

Table 4-11 Number of operations for inverse ROT

ROT size	Operation	Number	Total operations
4x4	+	12	30 (with 18 mult.)
	>>	0	
	*	18	
8x8	+	48	120 (with 72 mult.)
	>>	0	
	*	72	

4.11 Complexity characteristics of encoder quantization and quantization type selection

Only one type of quantization process is used in the proposed codec, i.e. no quantization matrix is used. In addition, the RDOQ process only uses one fixed QP value according to the anchor configuration.

4.12 Complexity characteristics of decoder inverse quantization

There is no specific difference compared to the H.264/AVC since they share the same scalar dequantizer scheme.

4.13 Complexity characteristics of encoder in-loop filtering type selection

4.13.1 Adaptive loop filter

For the luma component, there are two kinds of filter application control: a frame-based one and a unit-based one. Given the fixed tap which is 5x5 in the proposed codec's encoder, the optimal control structure is first estimated. For all possible control depths where a control depth is from 0 to the max CU depth - 1, the unique control map is generated. For example, if the max CU depth is set to 5, then there are only five control maps to be tested. For every unit in a control map, the control flag is determined by comparison of the original distortion and the new distortion where the new distortion is set to the difference between the original signal and the filtered reconstruction signal. The Wiener filter is re-estimated several times with units which prefer to be filtered. In this proposal, the Wiener filter is re-estimated up to 3 times by the encoder configuration. Because filtering is determined for each CU or merged CU, the correlation can be reused if it is calculated and stored for each smallest CU to reduce the complexity of the Wiener filter estimation. Once the best control map, control flags and the filter are selected for unit-based control, its RD cost is compared to the RD cost of frame-based control.

The optimal filter length is estimated with the selected control units and flags in the previous stage. In this proposal, there are three filter taps: 5x5, 7x7 and 9x9. For low complexity, this process is omitted in the proposed encoder when a slice type is non-referenced B.

For chroma components, filtering is only frame-based. Furthermore, its process is not invoked for non-referenced B, in order to reduce encoder complexity. Each chroma component can be applied or not independently and only one filter is shared for both chroma components. If one chroma component is decided not to be applied with the initially estimated filter, then the Wiener filter for the other chroma component is re-designed. For simplicity, only one filter tap is supported for chroma components and that is 5x5.

The optimal Wiener filter is usually estimated in floating type. It loses the accuracy when it is approximated and quantized to the limited integer type. But to reduce the overhead of filter coefficients, quantization is necessary. As many Wiener filters are estimated for a slice with multiple loop tests or different conditions, as there are many chances to find the quantized filter optimally to compensate for quantization loss. But consequently, it increases encoder complexity as a result of the many tests. As the control structure has the freedom to be generated in an arbitrary shape, it is important to limit the number of test cases in the encoder to limit the complexity of the task. In this proposal, it is proposed to use the CU boundary to generate a control map. The CU boundary is already a

clustered prediction error region and quantization error region, and furthermore, due to its hierarchical representation, the proposed adaptive loop filter can control the tradeoff between filter efficiency and filter overhead. So with relatively few trials, the encoder can produce an efficient filter and the control map.

4.14 Complexity characteristics of decoder in-loop filtering operations

4.14.1 Deblocking filter

Although we use the same process defined in the H.264/AVC, the use of large block sizes can effectively reduce the complexity of the deblocking filter. Table 4-12 shows the average number of deblocked pixels in Class B sequences for constraint set 1. As shown in the table, the number of deblocked pixels can be reduced by 41% in average, which means that the complexity of the deblocking filter can be reduced significantly in the proposed scheme.

Table 4-12 Average number of deblocked pixels in Class B of constraint set 1

Sequence	Number of deblocked pixels		Relative ratio
	Alpha anchor	Proposed	
Kimono1	145951397	70604005	0.48
ParkScene	86871747	54583037	0.63
Cactus	205011157	126507089	0.62
BasketballDrive	316322552	157577813	0.50
BQTerrace	132100148	96465482	0.73
Average	177251400	101147485	0.59

4.14.2 Adaptive loop filter

5x5 tap, 7x7 tap, and 9x9 tap Wiener filters with offset are used. Complexity for coding, parsing and applying the filters is reduced by exploiting the filter symmetries as follows [12] (each number corresponds to a tap value):

Table 4-13 Position of filter coefficients

(a) 9x9 tap									
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	26	
27	28	29	30	31	32	33	34	35	
36	37	38	39	40	39	38	37	36	
35	34	33	32	31	30	29	28	27	
26	25	24	23	22	21	20	19	18	
17	16	15	14	13	12	11	10	9	
8	7	6	5	4	3	2	1	0	

(b) 7x7 tap						
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	23	22	21
20	19	18	17	16	15	14
13	12	11	10	9	8	7
6	5	4	3	2	1	0

(c) 5x5 tap				
0	1	2	3	4
5	6	7	8	9
10	11	12	11	10
9	8	7	6	5
4	3	2	1	0

Since FIR filter is used with no data dependency, it makes parallelization easier. Once it is determined to use the adaptive loop filter, it is applied to the whole frame or controlled by a unit. When the adaptive loop filtering is controlled by a unit, it is applied, or not, according to a binary partition map. This map is derived from the coding unit boundaries to reduce metadata and complexity. The partition map can be rendered coarse or fine by a filter control depth parameter which is signaled, and which lies between the lowest and the highest coding unit depth. Only coding unit boundaries whose depth is less than or equal to the filter control depth are used in deriving the partition map.

4.14.3 Extreme correction and band correction

Because the positions of reference pixels used for extreme correction are only those above, below, to the left and to the right, and the filter is FIR with no data dependency, this process is highly parallelizable. Extreme band correction needs no reference pixel, and so it can be merged into the adaptive loop filter. Both the filters used in extreme correction and extreme band correction work by replacing the target value by the respective value determined by some rule, which can be realized by a simple table lookup to reduce complexity.

4.15 Complexity characteristics of encoder entropy coding type selection

This proposal uses one entropy coding method, SBAC, thus there is no need to select entropy coding type.

4.16 Complexity characteristics of decoder entropy decoding operation

4.16.1 Syntax-based context-adaptive binary arithmetic coding

Overall, the complexity of SBAC is comparable to CABAC, since SBAC is also a context-adaptive binary arithmetic coding method. SBAC uses binarization techniques and bypass techniques similar to CABAC in complexity. The core binary coding engine uses a small FSM for probability updates, as does H.264/AVC. SBAC also uses a small number of shifts and additions for interval rescaling, unlike the LUT method in H.264/AVC JM implementation, but this is unlikely to affect relative performance.

4.16.2 Adaptive coefficient scanning

When the horizontal or vertical scanning methods are chosen as the best scanning method, the mapping process from the coefficient array index to the zigzag scanning index is not necessary. Thus, the complexity for the coefficient indexing can be reduced and also the memory bandwidth to access the zigzag mapping table can be saved.

4.17 Degree of capability for encoder parallel processing

Most of the parallel processing approaches for H.264/AVC encoders are based on the separate estimation of different level of units: GOP, slice or group of macroblocks. There is no particular problem in using typical H.264/AVC schemes for the proposed codec.

4.18 Degree of capability for decoder parallel processing

The proposed scheme provides the same level of parallelism compared to H.264/AVC. All inter-predicted units can be processed in parallel and each slice can be decoded independently. However, there is a significant benefit with the use of large size blocks for the processors supporting vector processing, e.g. single instruction multiple data (SIMD). Widely used SIMD techniques support the use of 128 bit register and even larger size register is now considered such as Intel advanced vector extension instructions (AVX) of 256 bit register.

Table 4-14 Average width of processing unit for various classes in constraint set 1

Class	Alpha anchor		Proposed		Ratio	
	MC	Intra	MC	Intra	MC	Intra
A	15.6	9.1	53.4	15.3	3.4	1.7
B	15.7	10.0	62.5	22.4	4.0	2.4
C	15.5	9.0	55.2	14.5	3.6	1.6
D	14.4	7.1	30.4	9.5	2.1	1.3
ALL					3.3	1.7

To maximize the usefulness of SIMD operations, the number of simultaneously processed pixels should be large. From the simple analysis as shown in Table 4-14, the average width of the motion compensation unit is 3.3 times longer than the Alpha anchor case and 1.7 times for the intra prediction unit. It means that the usefulness of larger size register can be improved significantly in the motion compensation and intra prediction processes in the proposed scheme.

4.19 Requirement for encoder frame-level multi-pass processing

There is no particular need to perform encoder frame-level multi-pass processing for the proposed scheme.

5 Algorithm characteristics

5.1 Random access characteristics

As described in Section 3.2, the constraint set 1 anchor GOP conditions condition were also satisfied by this proposal. Therefore, the number of frames required to decode any given frame is exactly same as the Alpha anchor, which means that the same random access interval of 1.1 sec can be achieved.

5.2 Delay characteristics

As described in Section 3.3, we have used both the typical hierarchical-P structure of GOP 4 and the IPPP structure to generate the bitstreams for constraint set 2. On the decoder-side, no particular reordering of pictures is necessary. Thus the structural delay characteristics of the proposal are exactly the same as that of the given anchors which satisfies constraint set 2 conditions.

To estimate the delay caused by bit-rate fluctuation, RVM was proposed in the Kyoto meeting. Table 5-1 summarizes the RVM values of the anchor and the proposed codec for each sequence class.

Table 5-1 RVM summary of the anchors and the proposed codec

Class	Resolution	Beta	Gamma	Proposed		Proposed (lambda fixed)	
		RVM	RVM	RVM	BD-rate (%)	RVM	BD-rate (%)
B	1080p	1.22	0.36	1.74	43.66	1.12	31.19
C	WVGA	1.01	0.34	1.70	35.58		
D	WQVGA	1.12	0.38	1.63	32.16		
E	720p	1.38	0.44	2.10	47.48		
ALL	-	1.17	0.38	1.77	39.48		

As shown in the table, RVM values of the proposal are increased compared to both anchors. We found that this effect comes from the fact that the compression efficiency is increased significantly higher in B-slices by the proposed methods than in I- or P-slices. This increases the variance of the bit-rates, resulting in an increased RVM measurement. However, this situation can be avoided by a simple change of the lambda values of B-slices in the encoder. Fortunately, this kind of change does not result in any significant performance drop. In the test of Class D sequences, less than a 1% performance drop is observed when the RVM of the proposed codec is fixed to have the same value of Beta anchor as shown in Table 5-1.

6 Software implementation description

The proposed algorithms were implemented using standard C++ programming language.

7 Further improvements after submission

After CFP submission, we finalized to test three additional coding tools for constraint set 1 condition. In this section, they are briefly introduced, for information only.

7.1 High accuracy SKIP and DIRECT mode

In the CFP submission, quarter-pel accuracy motion vector was used in SKIP and DIRECT mode, however, it's easy to use high accuracy of 1/12 pel in SKIP and DIRECT mode since all neighbouring blocks already use 1/12 pel accuracy. Simple enabling of this feature improves coding efficiency especially for steady motion sequences.

7.2 New boundary strength decision rule

In the CFP submission, conventional H.264/AVC deblocking rules were used except for small modifications according to the CU/PU/TU structure. However, due to the addition of other in-loop filters and large block size units, boundary strength decision rule should be tuned differently. For the first attempt, we add weakest boundary strength as BS5 and changed the decision rule to use coding unit size.

7.3 Bi-directional prediction refinement

In the CFP submission, typical bi-directional prediction between blocks was used. However, it's possible to improve the prediction quality when we have two predictors, one from L0 and the other from L1. We try to use the small region of L0 prediction signal to find the best corresponding region in L1 prediction signal. This process can be implemented iteratively to use L1 region for finding best-matched L0 region. After certain number of iterations, typical bi-directional prediction is performed between the corresponding regions. It is similar to decoder-side motion vector derivation technique using the prediction block of the other temporal direction as the template.

Currently, 3x3 size is used as the region and only +1, 0, -1 position refinement in 1/12 pel accuracy for each horizontal and vertical direction is allowed to minimize the complexity. Since this process should be repeated in the decoder-side, we also developed non-iterative method to compute the refinement value based on the theory of optical-flow.

7.4 Constraint set 1 configuration relative to Alpha anchor (updated)

As you can see Table 7-1, average 41.58% bit-saving can be achieved for constraint set 1 configuration. Additional gain compared to the submitted version is about 2.1%.

Table 7-1 Summarized performance relative to Alpha anchor (updated)

Class	Resolution	Seq. No	Seq. Name	BD-rate (%)	Additional gain (%)
A	4K	S01	Traffic	42.99	2.40
		S02	PeopleOnStreet	31.20	1.36
B	1080p	S03	Kimono	45.80	1.37
		S04	ParkScene	36.09	3.13
		S05	Cactus	43.28	2.49
		S06	BasketballDrive	45.11	0.90
		S07	BQTerrace	55.96	1.90
C	WVGA	S08	BasketballDrill	44.56	1.47
		S09	BQMall	41.94	2.25
		S10	PartyScene	43.36	2.23
		S11	RaceHorses	37.11	1.25
D	WQVGA	S12	BasketballPass	33.45	2.90
		S13	BQSquare	56.92	2.73
		S14	BlowingBubbles	36.95	2.98
		S15	RaceHorses	28.95	1.86
Summary					
A				37.09	1.88
B				45.25	1.96
C				41.74	1.80
D				39.07	2.62
All				41.58	2.09

8 Closing remarks

In the previous sections, all aspects of the proposed codec, including detailed algorithm descriptions, performance results and complexity analysis, have been discussed. On average, the proposed codec showed about 40% performance improvement over the Alpha and Beta anchor sequences. The complexity of the proposed encoder and decoder is similar to that of current standards.

In conclusion, we think that enough information has been provided to demonstrate the validity of this proposal and the reasons for it to be considered as a strong candidate to be used for the Test Model that will be used as the basis of the Core Experiments in the next phase of HVC standardization.

9 Patent rights declaration(s)

British Broadcasting Corporation (BBC) may have IPR relating to the technology described in 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 | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

Samsung may have IPR relating to the technology described in 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 | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

References

- [1] K. H. Lee, E. Alshina, J. H. Park, W. J. Han and J. H. Min, "Technical considerations for Ad Hoc Group on New Challenges in Video Coding Standardization," ISO/IEC MPEG 85th meeting, M15580, Hannover, Germany, 2008.
- [2] E. Alshina, K. H. Lee, W. J. Han and J. H. Park, "Technical considerations for Ad Hoc Group on New Challenges in Video Coding Standardization," ISO/IEC MPEG 86th meeting, M15899, Busan, Korea, 2008.
- [3] S. Naito, A. Matsumura and A. Koike, "Efficient coding scheme for super high definition video based on extending H.264 high profile", VCIP 2006, Jan. 2006.
- [4] S. Ma and C.-C. Jay Kuo, "High-definition video coding with super-macroblocks", VCIP 2007, Jan 2007.
- [5] P. Chen, Y. Ye and M. Karczewicz, "Video Coding Using Extended Block Sizes", VCEG-AJ23, Oct 2008.
- [6] Tomonobu Yoshino, Sei Naito and Shigeyuki Sakazawa, "Preliminary response for Draft Call for Evidence on High Performance Video Coding", M16082, Feb 2009
- [7] Shun-ichi Sekiguchi and Shuichi Yamagishi, "On coding efficiency with extended block size for UHDTV", VCEG-C91, Jan 2009
- [8] Joel Jung and Guillaume Laroche, "Competition-Based Scheme for Motion Vector Selection and Coding", VCEG-AC06, July 2006
- [9] Thomas Wedi, "Hybrid Video Coding Based on High-Resolution Displacement Vectors", VCIP, Jan. 2001
- [10] W-H. Chen, C. Harrison Smith and S. C. Fralick, "A Fast computational Algorithm for the Discrete Cosine Transform ", IEEE Trans. Communications, No. 9, Sept. 1977
- [11] S. D. Kim and J. B. Ra, "An efficient motion vector coding scheme based on minimum bitrate prediction," IEEE Trans. on Image. Proc, vol. 8, no. 8, pp. 1117–1120, Aug. 1999
- [12] Takeshi Chujoh, Naofumi Wada and Goki Yasuda, "Quadtree-based adaptive loop filter", VCEG-C181, Jan. 2009
- [13] "Joint call for proposals on video compression technology," ISO/IEC MPEG 91st meeting, N11113, Kyoto, Japan, Apr. 2010.

BIGGER BLOCK MOTION PARTITION

1 Background

Most of the popular video coding standards such as H.263, MPEG-2 and MPEG-4, H.264/AVC utilize block-based video coding techniques. An image or a frame of a video is partitioned into multiple macroblocks and each macroblock can be further partitioned. Figure 1 shows macroblock partition used in H.264. The outer rectangles refer to the samples in a macroblock or sub-macroblock, respectively. The rectangles refer to the partitions. For macroblocks in an I frame, they are encoded by using the prediction from their spatial neighbors while for macroblocks in a P or B frame, they are encoded by using either the prediction from their spatial neighbors (spatial prediction) or areas in other frames (temporal prediction). Video coding standards define syntax elements to represent coding information. For example, for every macrobock H.264 defines *mb_type* to represent the way of macroblock partition and the method of prediction (spatial or temporal), *mvd* with some other syntax elements to represent the motion vector information, and *coded_block_pattern* with some other syntax elements to represent residual information after prediction.

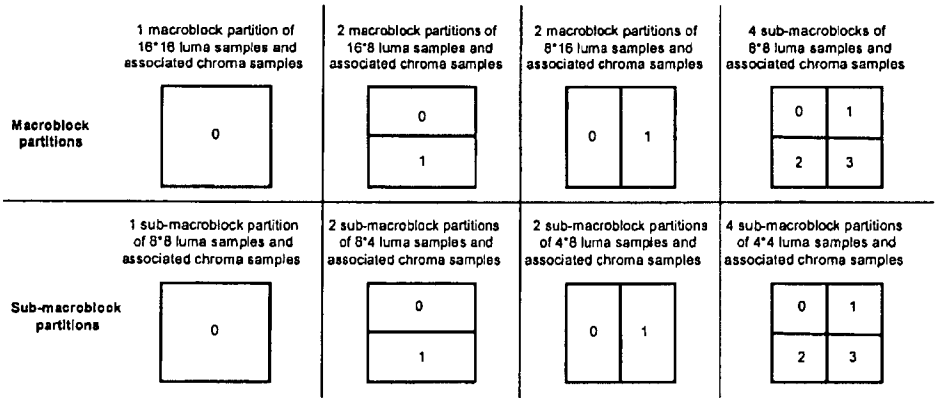


Figure 1 Macroblock partitions, sub-macroblock partitions in H.264.

2 Describe the problem solved and the advantage(s) of the invention

One of the reasons we can compress video is the existence of spatial and temporal redundancy. The higher resolution and frame rate, the more redundancy we can utilize. 1280x720, 60 frame/s video is spatially 36 times as large as and temporally 4 times as fast (sampled) as 176x144, 15 frame/s

video, which means much more redundancy. But the current macroblock size of 16x16 limits the capability to utilize the redundancy as resolution and frame rate increase. In this invention, bigger macroblock size is introduced to take care of this issue. In the following macroblock size up to 64x64 is introduced primarily for educational purpose. It could be easily adjusted to even higher size.

3 Describe how the invention solves the stated problem and achieves the stated advantage(s)

3.1 64x64 Macroblock partition

We increase macroblock size to 64x64 and introduce corresponding macroblock modes to accommodate. Macroblocks or sub-macroblocks may be partitioned as shown in Figure 2.

<div>1 macroblock partition of 64*64 luma samples and associated chroma samples</div> <div>level 0 Macroblock partitions</div> <div>0</div>	<div>2 macroblock partitions of 64*32 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>2 macroblock partitions of 32*64 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>4 sub- macroblocks of 32*32 luma samples and associated chroma samples</div> <div>0</div> <div>1</div> <div>2</div> <div>3</div>
<div>1 sub- macroblock partition of 32*32 luma samples and associated chroma samples</div> <div>level 1 Macroblock partitions</div> <div>0</div>	<div>2 sub- macroblock partitions of 32*16 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>2 sub- macroblock partitions of 16*32 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>4 sub- macroblocks of 16*16 luma samples and associated chroma samples</div> <div>0</div> <div>1</div> <div>2</div> <div>3</div>
<div>1 sub- macroblock partition of 16*16 luma samples and associated chroma samples</div> <div>level 2 macroblock partitions</div> <div>0</div>	<div>2 sub- macroblock partitions of 16*8 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>2 sub- macroblock partitions of 8*16 luma sample and associated chroma samples</div> <div>0</div> <div>1</div>	<div>4 sub- macroblock partitions of 8*8 luma samples and associated chroma samples</div> <div>0</div> <div>1</div> <div>2</div> <div>3</div>
<div>1 sub- macroblock partition of 8*8 luma samples and associated chroma samples</div> <div>level 3 macroblock partitions</div> <div>0</div>	<div>2 sub- macroblock partitions of 8*4 luma samples and associated chroma samples</div> <div>0</div> <div>1</div>	<div>2 sub- macroblock partitions of 4*8 luma sample and associated chroma samples</div> <div>0</div> <div>1</div>	<div>4 sub- macroblock partitions of 4*4 luma samples and associated chroma samples</div> <div>0</div> <div>1</div> <div>2</div> <div>3</div>

Figure 2 64x64 Macroblock partitions, sub-macroblock partitions.

There are four levels of partition. If four 32x32 blocks result from level 0 partition, each 32x32 partition can be further partitioned as shown in the 2nd row of Figure 2. If four 16x16 blocks result from level 1 partition, each 16x16 partition can be further partitioned as shown in the 3rd row of Figure 2. If four 8x8 blocks results from level 2 partition, each 8x8 block can be further partitioned as shown in the 4th row of Figure 2. By having these 4 levels of partition, both big homogeneous area and fine sporadic change can be adaptively represented within this framework.

To have more flexibility to encode a macroblock, each partition can choose its own way of further partition and each final partition can choose either temporal prediction or spatial prediction. For example, in a 32x32 macroblock with 4 16x16 partitions, some 16x16 partitions might use spatial prediction and the others use temporal prediction.

3.2 New syntax

Context adaptive arithmetic coding is used to encode information such as macroblock type, coded block pattern, motion vector and transform coefficients. The context is based on neighboring macroblocks.

3.2.1 Macroblock type

For 64x64 macroblock, we introduce a new syntax mb64_type to indicate the level 0 macroblock partition shown in Figure 2

<div>1 macroblock partition of 64*64 luma samples and associated chroma samples</div> <div>level 0 Macroblock partitions</div> <div><div>0</div></div>	<div>2 macroblock partitions of 64*32 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>2 macroblock partitions of 32*64 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>4 sub- macroblocks of 32*32 luma samples and associated chroma samples</div> <div><div><div><div>0</div><div>1</div><div>2</div><div>3</div></div></div></div>
<div>1 sub- macroblock partition of 32*32 luma samples and associated chroma samples</div> <div>level 1 Macroblock partitions</div> <div><div>0</div></div>	<div>2 sub- macroblock partitions of 32*16 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>2 sub- macroblock partitions of 16*32 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>4 sub- macroblocks of 16*16 luma samples and associated chroma samples</div> <div><div><div><div>0</div><div>1</div><div>2</div><div>3</div></div></div></div>
<div>1 sub- macroblock partition of 16*16 luma samples and associated chroma samples</div> <div>level 2 macroblock partitions</div> <div><div>0</div></div>	<div>2 sub- macroblock partitions of 16*8 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>2 sub- macroblock partitions of 8*16 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>4 sub- macroblock partitions of 8*8 luma samples and associated chroma samples</div> <div><div><div><div>0</div><div>1</div><div>2</div><div>3</div></div></div></div>
<div>1 sub- macroblock partition of 8*8 luma samples and associated chroma samples</div> <div>level 3 macroblock partitions</div> <div><div>0</div></div>	<div>2 sub- macroblock partitions of 8*4 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>2 sub- macroblock partitions of 4*8 luma samples and associated chroma samples</div> <div><div><div>0</div><div>1</div></div></div>	<div>4 sub- macroblock partitions of 4*4 luma samples and associated chroma samples</div> <div><div><div><div>0</div><div>1</div><div>2</div><div>3</div></div></div></div>

. If this macroblock has four 32x32 partitions, for each partition, we will introduce a syntax mb32_type. If one 32x32 block has four 16x16 partitions, for each partition, we will use the current H.264 16x16 macroblock type syntax to indicate level 2 and level 3 partition.

The (macro)block types corresponding to level 0 and level 1 partition for P and B slices are specified in Table 1 and Table 2 with N being 64 or 32 and M=N/2. Since spatial prediction might be more efficient than temporal prediction sometimes, intra modes are also necessary. There are 2 different methods to design intra modes. The first is still using the existing h264 intra modes. By grouping 4 intra blocks into a bigger block, we can generate a bigger intra block. For example, 4 intra blocks each having the size of 16x16 will form a 32x32 intra block. Each 16x16 block can select its own way of spatial prediction. Therefore, we might end up with the combination shown in Figure 3.

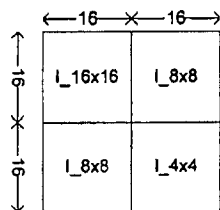


Figure 3 The grouping of intra blocks.

The second method of designing intra modes is generating the spatial predictive 32x32 blocks based on their neighboring pixels directly and generating the spatial predictive 64x64 block based on their neighboring pixels directly. Since the spatial prediction is done at a larger scale compared to 16x16 intra blocks, there may be some bit rate saving. Intra modes generated by these two different methods can be put into a single candidate pool and compete to each other for the best rate-distortion performance.

Table 1 block type values for P slices

mbN_type	Name of mbN_type	NumMbPart (mbN_type)	MbPartPredMode (mbN_type, 0)	MbPartPredMode (mbN_type, 1)	MbPartWidth (mbN_type)	MbPartHeight (mbN_type)
0	P_L0_NxN	1	Pred_L0	Na	N	N
1	P_L0_L0_NxM	2	Pred_L0	Pred_L0	N	M
2	P_L0_L0_MxN	2	Pred_L0	Pred_L0	M	N
3	PN_MxM	4	na	na	M	M
inferred	PN_Skip	1	Pred_L0	na	N	N

Table 2 block type values for B slices

mbN_type	Name of mbN_type	NumMbPart (mbN_type)	MbPartPredMode (mbN_type, 0)	MbPartPredMode (mbN_type, 1)	MbPartWidth (mbN_type)	MbPartHeight (mbN_type)
0	B_Direct_NxN	Na	Direct	na	N	N
1	B_L0_NxN	1	Pred_L0	na	N	N
2	B_L1_NxN	1	Pred_L1	na	N	N
3	B_Bi_NxN	1	BiPred	na	N	N
4	B_L0_L0_NxM	2	Pred_L0	Pred_L0	N	M
5	B_L0_L0_MxN	2	Pred_L0	Pred_L0	M	N
6	B_L1_L1_NxM	2	Pred_L1	Pred_L1	N	M
7	B_L1_L1_MxN	2	Pred_L1	Pred_L1	M	N
8	B_L0_L1_NxM	2	Pred_L0	Pred_L1	N	M
9	B_L0_L1_MxN	2	Pred_L0	Pred_L1	M	N
10	B_L1_L0_NxM	2	Pred_L1	Pred_L0	N	M
11	B_L1_L0_MxN	2	Pred_L1	Pred_L0	M	N
12	B_L0_Bi_NxM	2	Pred_L0	BiPred	N	M
13	B_L0_Bi_MxN	2	Pred_L0	BiPred	M	N
14	B_L1_Bi_NxM	2	Pred_L1	BiPred	N	M
15	B_L1_Bi_MxN	2	Pred_L1	BiPred	M	N
16	B_Bi_L0_NxM	2	BiPred	Pred_L0	N	M
17	B_Bi_L0_MxN	2	BiPred	Pred_L0	M	N
18	B_Bi_L1_NxM	2	BiPred	Pred_L1	N	M
19	B_Bi_L1_MxN	2	BiPred	Pred_L1	M	N
20	B_Bi_Bi_NxM	2	BiPred	BiPred	N	M
21	B_Bi_Bi_MxN	2	BiPred	BiPred	M	N
22	BN_MxM	4	na	na	M	M
inferred	BN_Skip	Na	Direct	na	M	M

3.2.2 MB64x64 coded block pattern

For a 64x64 macroblock, we introduce a 1 bit syntax cbp64 to indicate whether the whole 64x64 macroblock has nonzero coefficients or not. 1 means there is

at least one nonzero coefficient, otherwise all coefficients are zero.

If cbp64 is 1, for each 32x32 block we introduce another 1 bit syntax cbp32 to indicate whether the whole 32x32 block has nonzero coefficients or not. 1 means there is at least one nonzero coefficient, otherwise all coefficients are zero. If a 32x32 block has nonzero coefficients, for each 16x16 block we use the following syntax to indicate its status.

```

    if ( motion partition bigger than 8x8 )
    {
        lumacbp16
        if ( lumacbp16 != 0 )
        {
            transform_size_flag
            if ( transform_size_flag == 2 )
            {
                if ( mb16_type == P_16x8 || mb16_type == P_8x16 )
                    luma16x8_cbp
                else
                    chroma_cbp
            }
            else
                h264_cbp
        }
        else
            chroma_cbp
    }
    else
        h264_cbp

```

Here we introduce several new syntax. lumacbp16 is a 1 bit flag indicating whether the whole 16x16 luma block has nonzero coefficients or not. 1 means there is at least 1 nonzero coefficient. transform_size_flag will indicate the transform being used, 4x4 transform, 8x8 transform, 16x16 transform (for motion partition equal to or bigger than 16x16), 16x8 transform (only for P_16x8), or 8x16 transform (only for P_8x16). luma16x8_cbp is a 2 bit number with each bit indicating whether one of the two partitions of P_16x8 or P_8x16 has nonzero coefficients or not. chroma_cbp is the same as CodedBlockPatternChroma in H.264. h264_cbp is the same as the cbp defined in H.264.

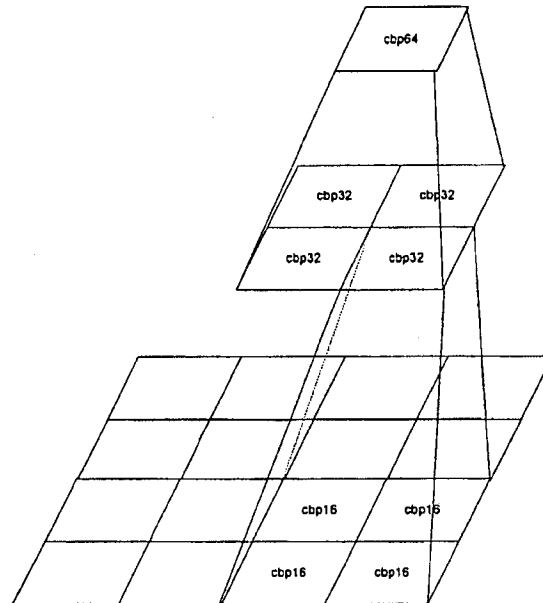


Figure 4 cbp hierarchy.

3.2.3 MB64x64 quantization parameter (mb64_delta_qp)

mb64_delta_qp can change the value of luminance quantization parameter, QP_Y , in the 64x64 macroblock layer. The decoded value of mb64_qp_delta shall be in the range of -26 to +25, inclusive. Mb64_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

$$QP_Y = (QP_{Y,PREV} + mb64_qp_delta + 52) \% 52$$

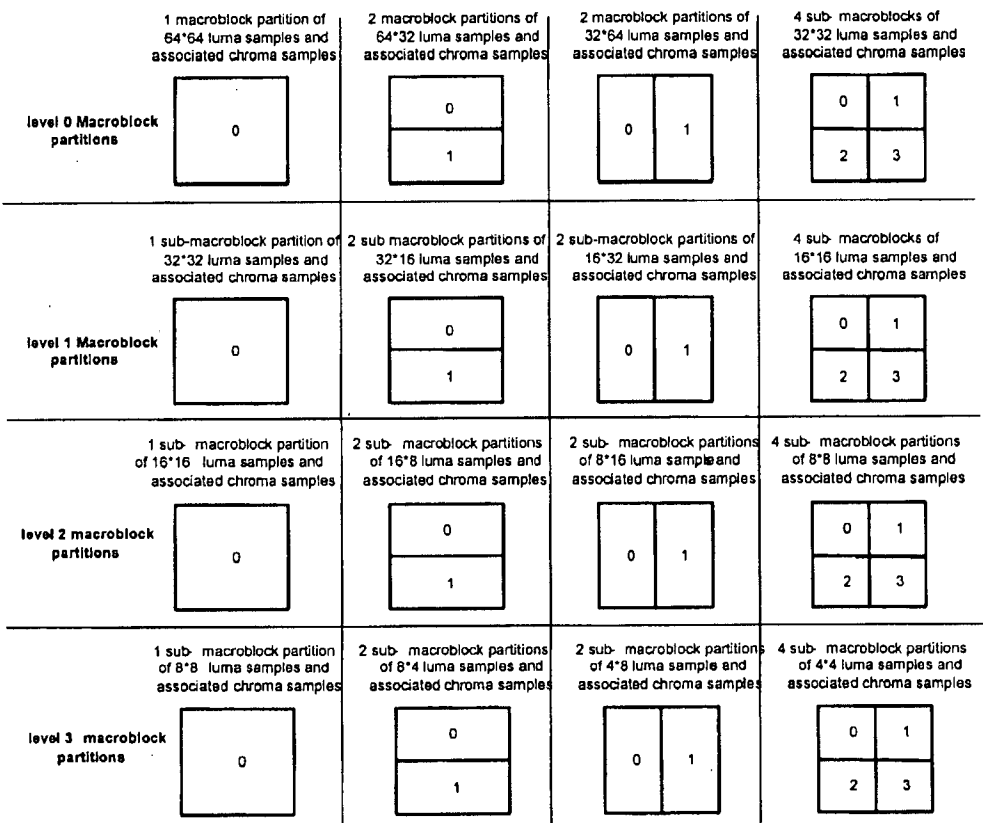
where $QP_{Y,PREV}$ is the luminance quantization parameter, QP_Y , of the previous 64x64 macroblock in decoding order in the current slice. For the first 64x64 macroblock in the slice $QP_{Y,PREV}$ is initially set equal to Slice QP sent in the slice header.

To have finer quality/rate control, we can introduce partition-based quantization parameters inside a 64x64 macroblock.

3.3 Mode decision

To find out the best method to partition a 64x64 macroblock and the best encoding method (spatial or temporal prediction) for each partition, lagrangian multiplier is used. We calculate the cost for a 64x64 macroblock using single partition, two 64x32 partitions, two 32x64 partitions and four 32x32 partitions respectively. The cost function is defined as $C = D + \lambda R$, where D is the distortion and R is the rate. When we calculate the cost for four 32x32

partitions, we start with the uper left 32x32 partition and find the cost associated with the best partition method and encoding method. Then we find out the cost of the other three 32x32 partition by the order indicated by the number inside each 32x32 partition in Figure 2. In order to find the best partition method and encoding method of a 32x32 partition, we compare the cost of single partition, two 32x16 partitions, two 16x32 partitions and four 16x16 partitions. When we calculate the cost for four 16x16 partition, we find the best partition method and encoding method for each 16x16 in the order indicated in Figure 2



3.4 Frame or slice level rate-distortion optimization

Since extra levels of hierarchy in the macroblock partition might cost extra bits, we introduce a frame level logic to decide the maximum macroblock size. This information could be transmitted in the picture parameter set or the slice header. For this purpose, we might use 3 pass encoding. During the first pass, only conventional 16x16 macroblocks are used and during the second pass, only 32x32 macroblocks are used. In the 3rd pass, 64x64 macroblock is used.

After each encoding passes, we collect bit rate R and distortion D . By comparing the cost function which is equal to $D + \lambda R$ of these 3 encoding passes, we will select the best one. This process can also be applied at slice level.

VIDEO CODING WITH LARGE MACROBLOCKS

[0001] This provisional application relates to U.S. Provisional Application Nos. 61/102,787 filed on October 3, 2008, and 61/144,357 filed on January 13, 2009, both of which are incorporated herein by reference in their entireties.

TECHNICAL FIELD

[0002] This disclosure relates to digital video coding and, more particularly, block-based video coding.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, video gaming devices, video game consoles, cellular or satellite radio telephones, and the like. Digital video devices implement video compression techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263 or ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), and extensions of such standards, to transmit and receive digital video information more efficiently.

[0004] Video compression techniques perform spatial prediction and/or temporal prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video frame or slice may be partitioned into macroblocks. Each macroblock can be further partitioned. Macroblocks in an intra-coded (I) frame or slice are encoded using spatial prediction with respect to neighboring macroblocks. Macroblocks in an inter-coded (P or B) frame or slice may use spatial prediction with respect to neighboring macroblocks in the same frame or slice or temporal prediction with respect to other reference frames.

SUMMARY

[0005] In general, this disclosure describes techniques for encoding digital video data using large macroblocks. Large macroblocks are larger than macroblocks generally prescribed by

existing video encoding standards. Most video encoding standards prescribe the use of a macroblock in the form of a 16x16 array of pixels. In accordance with this disclosure, an encoder and decoder may utilize large macroblocks that are greater than 16x16 pixels in size. As examples, a large macroblock may have a 32x32, 64x64, or larger array of pixels.

[0006] Video coding relies on spatial and/or temporal redundancy to support compression of video data. Video frames generated with higher spatial resolution and/or higher frame rate may support more redundancy. The use of large macroblocks, as described in this disclosure, may permit a video coding technique to utilize larger degrees of redundancy produced as spatial resolution and/or frame rate increase. In accordance with this disclosure, video coding techniques may utilize a variety of features to support coding of large macroblocks.

[0007] As described in this disclosure, a large macroblock coding technique may partition a large macroblock into partitions, and use different partition sizes and different coding modes, e.g., different spatial (I) or temporal (P or B) modes, for selected partitions. As another example, a coding technique may utilize hierarchical coded block pattern (CBP) values to efficiently identify coded macroblocks and partitions having non-zero coefficients within a large macroblock. As a further example, a coding technique may compare rate-distortion metrics produced by coding using large and small macroblocks to select a macroblock size producing more favorable results.

[0008] In one example, the disclosure provides a method comprising encoding, with a video encoder, a video block having a size of more than 16x16 pixels, generating block-type syntax information that indicates the size of the block, and generating a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

[0009] In another example, the disclosure provides an apparatus comprising a video encoder configured to encode a video block having a size of more than 16x16 pixels, generate block-type syntax information that indicates the size of the block, and generate a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

[0010] In another example, the disclosure provides a computer-readable medium encoded with instructions to cause a video encoding apparatus to encode, with a video encoder, a

video block having a size of more than 16x16 pixels, generate block-type syntax information that indicates the size of the block, and generate a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

[0011] In an additional example, the disclosure provides a method comprising receiving, with a video decoder, an encoded video block having a size of more than 16x16 pixels, receiving block-type syntax information that indicates the size of the encoded block, receiving a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient, and

decoding the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

[0012] In a further example, the disclosure provides an apparatus comprising a video decoder configured to receive an encoded video block having a size of more than 16x16 pixels, receive block-type syntax information that indicates the size of the encoded block, receive a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient, and decode the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

[0013] In another example, the disclosure provides a computer-readable medium comprising instructions to cause a video decoder to receive an encoded video block having a size of more than 16x16 pixels, receive block-type syntax information that indicates the size of the encoded block, receive a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient, and decode the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

[0014] In another example, the disclosure provides a method comprising receiving, with a video encoder, a video block having a size of more than 16x16 pixels, partitioning the block into partitions, encoding one of the partitions using a first encoding mode, encoding another of the partitions using a second encoding mode different from the first encoding

mode, and generating block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.

[0015] In an additional example, the disclosure provides an apparatus comprising a video encoder configured to receive a video block having a size of more than 16x16 pixels, partition the block into partitions, encode one of the partitions using a first encoding mode, encode another of the partitions using a second encoding mode different from the first encoding mode, generate block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.

[0016] In another example, the disclosure provides a computer-readable medium encoded with instructions to cause a video encoder to receive a video block having a size of more than 16x16 pixels, partition the block into partitions, encode one of the partitions using a first encoding mode, encode another of the partitions using a second encoding mode different from the first encoding mode, and generate block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.

[0017] In a further example, the disclosure provides a method comprising receiving, with a video decoder, a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode, receiving block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions, and decoding the video block based on the block-type syntax information.

[0018] In another example, the disclosure provides an apparatus comprising a video decoder configured to receive a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode, receive block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions, and decode the video block based on the block-type syntax information.

[0019] In an additional example, the disclosure provides a computer-readable medium encoded with instructions to cause a video decoder to receive, with a video decoder, a video

block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode, receive block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions, and decode the video block based on the block-type syntax information.

[0020] In another example, the disclosure provides a method comprising receiving, with a digital video encoder, a video coding unit, determining a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels, determining a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels, encoding the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric, and encoding the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.

[0021] In an additional example, the disclosure provides an apparatus comprising a video encoder configured to receive a video coding unit, determine a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels, determine a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels, encode the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric, encode the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.

[0022] In another example, the disclosure provides a computer-readable medium encoded with instructions to cause a video encoder to receive a video coding unit, determine a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels, determine a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels, encode the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric, and encode the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.

[0023] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0024] FIG. 1 is a block diagram illustrating an example video encoding and decoding system that encodes and decodes digital video data using large macroblocks.

[0025] FIG. 2 is a block diagram illustrating an example of a video encoder that implements techniques for coding large macroblocks.

[0026] FIG. 3 is a block diagram illustrating an example of a video decoder that implements techniques for coding large macroblocks.

[0027] FIG. 4A is a conceptual diagram illustrating partitioning among various levels of a large macroblock.

[0028] FIG. 4B is a conceptual diagram illustrating assignment of different coding modes to different partitions a large macroblock.

[0029] FIG. 5 is a conceptual diagram illustrating a hierarchical view of various levels of a large macroblock.

[0030] FIG. 6 is a flowchart illustrating an example method for setting a coded block pattern (CBP) value of a 64x64 pixel large macroblock.

[0031] FIG. 7 is a flowchart illustrating an example method for setting a CBP value of a 32x32 pixel partition of a 64x64 pixel large macroblock.

[0032] FIG. 8 is a flowchart illustrating an example method for setting a CBP value of a 16x16 pixel partition of a 32x32 pixel partition of a 64x64 pixel large macroblock.

[0033] FIG. 9 is a flowchart illustrating an example method for determining a two-bit luma16x8_CBP value.

[0034] FIG. 10 is a block diagram illustrating an example arrangement of a 64x64 pixel large macroblock.

[0035] FIG. 11 is a flowchart illustrating an example method for calculating optimal partitioning and encoding methods for an NxN pixel large video block.

[0036] FIG. 12 is a block diagram illustrating an example 64x64 pixel macroblock with various partitions and selected encoding methods for each partition.

[0037] FIG. 13 is a flowchart illustrating an example method for determining an optimal size of a macroblock for encoding a frame of a video sequence.

[0038] FIG. 14 is a block diagram illustrating an example wireless communication device including a video encoder/decoder (CODEC) that codes digital video data using large macroblocks.

[0039] FIG. 15 is a block diagram illustrating an example array representation of a hierarchical CBP representation for a large macroblock.

[0040] FIG. 16 is a block diagram illustrating an example tree structure corresponding to the hierarchical CBP representation of FIG. 15.

DETAILED DESCRIPTION

[0041] The disclosure describes techniques for encoding and decoding digital video data using large macroblocks. Large macroblocks are larger than macroblocks generally prescribed by existing video encoding standards. Most video encoding standards prescribe the use of a macroblock in the form of a 16x16 array of pixels. In accordance with this disclosure, an encoder and/or a decoder may utilize large macroblocks that are greater than 16x16 pixels in size. As examples, a large macroblock may have a 32x32, 64x64, or possibly larger array of pixels.

[0042] In some cases, large macroblocks may support improvements in coding efficiency and/or reductions in data transmission overhead while maintaining or possibly improving image quality. For example, the use of large macroblocks may permit a video encoder and/or decoder to take advantage of increased redundancy provided by video data generated with increased spatial resolution (e.g., 1280x720 or 1920x1080 pixels per frame) and/or increased frame rate (e.g., 30 or 60 frames per second).

[0043] As an illustration, a digital video sequence with a spatial resolution of 1280x720 pixels per frame and a frame rate of 60 frames per second is spatially 36 times larger than and temporally 4 times faster than a digital video sequence with a spatial resolution of 176x144 pixels per frame and a frame rate of 15 frames per second. With increased macroblock size, a video encoder and/or decoder can better exploit increased spatial and/or temporal redundancy to support compression of video data.

[0044] Also, by using larger macroblocks, a smaller number of blocks may be encoded for a given frame or slice, reducing the amount of overhead information that needs to be transmitted. In other words, larger macroblocks may permit a reduction in the overall number of macroblocks coded per frame or slice. With a reduced number of macroblocks, for example, the cumulative amount of coding information such as syntax information, motion vector data, and the like can be reduced.

[0045] In this disclosure, the size of a macroblock generally refers to the number of pixels contained in the macroblock, e.g., 64x64, 32x32, 16x16, or the like. Hence, a large macroblock (e.g., 64x64 or 32x32) may be large in the sense that it contains a larger number of pixels than a 16x16 macroblock. However, the area defined by the vertical and horizontal dimensions of a large macroblock, i.e., as a fraction of the area defined by the vertical and horizontal dimensions of a video frame, may or may not be larger than the area of a conventional 16x16 macroblock. In some examples, the area of the large macroblock may be the same or similar to a conventional 16x16 macroblock. However, the large macroblock has a higher spatial resolution characterized by a higher number and higher spatial density of pixels within the macroblock.

[0046] The size of the macroblock may be configured based at least in part on the number of pixels in the frame, i.e., the spatial resolution in the frame. If the frame has a higher number of pixels, a large macroblock can be configured to have a higher number of pixels. As an illustration, a video encoder may be configured to utilize a 32x32 pixel macroblock for a 1280x720 pixel frame displayed at 30 frames per second. As another illustration, a video encoder may be configured to utilize a 64x64 pixel macroblock for a 1280x720 pixel frame displayed at 60 frames per second.

[0047] Each macroblock encoded by an encoder may require data that describes one or more characteristics of the macroblock. The data may indicate, for example, macroblock type data to represent the size of the macroblock, the way in which the macroblock is partitioned, and the coding mode (spatial or temporal) applied to the macroblock and/or its partitions. In addition, the data may include motion vector difference (mvd) data along with other syntax elements that represents motion vector information for the macroblock and/or its partitions. Also, the data may include a coded block pattern (CBP) value along with other syntax elements to represent residual information after prediction. The

macroblock type data may be provided in a single macroblock header for the large macroblock.

[0048] As mentioned above, by utilizing a large macroblock, the encoder may reduce the number of macroblocks per frame or slice, and thereby reduce the amount of net overhead that needs to be transmitted for each frame or slice. Also, by utilizing a large macroblock, the total number of macroblocks may decrease for a particular frame or slice, which may reduce blocky artifacts in video displayed to a user.

[0049] Video coding techniques described in this disclosure may utilize one or more features to support coding of large macroblocks. For example, a large macroblock may be partitioned into smaller partitions. Different coding modes, e.g., different spatial (I) or temporal (P or B) coding modes, may be applied to selected partitions within a large macroblock. Also, a hierarchical coded block pattern (CBP) values can be utilized to efficiently identify coded macroblocks and partitions having non-zero transform coefficients representing residual data. In addition, rate-distortion metrics may be compared for coding using large and small macroblock sizes to select a macroblock size producing favorable results.

[0050] Use of different coding modes for different partitions in a large macroblock may be referred to as mixed mode coding of large macroblocks. Instead of coding a large macroblock uniformly such that all partitions have the same intra- or inter-coding mode, a large macroblock may be coded such that some partitions have different coding modes, such as different intra-coding modes (c.g., I_16x16, I_8x8, I_4x4) or intra- and inter-coding modes.

[0051] If a large macroblock is divided into two or more partitions, for example, at least one partition may be coded with a first mode and another partition may be coded with a second mode. In some cases, the first mode may be a first I mode and the second mode may be a second I mode, different from the first I mode. In other cases, the first mode may be an I mode and the second mode may be a P or B mode. Hence, in some examples, a large macroblock may include one or more temporally (P or B) coded partitions and one or more spatially (I) coded partitions, or one or more spatially coded partitions with different I modes.

[0052] One or more hierarchical coded block pattern (CBP) values may be used to efficiently describe whether any partitions in a large macroblock have at least one non-zero transform coefficient and, if so, which partitions. The transform coefficients encode residual data for the large macroblock. A large macroblock level CBP bit indicates whether any partitions in the large macroblock includes a non-zero, quantized coefficient. If not, there is no need to consider whether any of the partitions has a non-zero coefficient, as the entire large macroblock is known to have no non-zero coefficients. In this case, a predictive macroblock can be used to decode the macroblock without residual data.

[0053] Alternatively, if the macroblock-level CBP value indicates that at least one partition in the large macroblock has a non-zero coefficient, then partition-level CBP values can be analyzed to identify which of the partitions includes at least one non-zero coefficient. The decoder then may retrieve appropriate residual data for the partitions having at least one non-zero coefficient, and decode the partitions using the residual data and predictive block data. In some cases, one or more partitions may have non-zero coefficients, and therefore include partition-level CBP values with the appropriate indication. Both the large macroblock and at least some of the partitions may be larger than 16x16 pixels.

[0054] To select macroblock sizes yielding favorable rate-distortion metrics, rate-distortion metrics may be analyzed for both large macroblocks (e.g., 32x32 or 64x64) and small macroblocks (e.g., 16x16). For example, an encoder may compare rate-distortion metrics between 16x16 macroblocks, 32x32 macroblocks, and 64x64 macroblocks for a coded unit, such as a frame or a slice. The encoder may then select the macroblock size that results in the best rate-distortion and encode the coded unit using the selected macroblock size, i.e., the macroblock size with the best rate-distortion.

[0055] The selection may be based on encoding the frame or slice in three passes, e.g., a first pass using 16x16 pixel macroblocks, a second pass using 32x32 pixel macroblocks, and a third pass using 64x64 pixel macroblocks, and comparing rate-distortion metrics for each pass. In this manner, an encoder may optimize rate-distortion by varying the macroblock size and selecting the macroblock size that results in the optimal rate-distortion for a given coding unit, such as a slice or frame. The encoder may further transmit syntax information for the coded unit, e.g., as part of a frame header or a slice header, that identifies the size of the macroblocks used in the coded unit.

[0056] For some video frames or slices, large macroblocks may present substantial bit rate savings and thereby produce the best rate-distortion results, given relatively low distortion. For other video frames or slices, however, smaller macroblocks may present less distortion, outweighing bit rate in the rate-distortion cost analysis. Hence, in different cases, 64x64, 32x32 or 16x16 may be appropriate for different video frames or slices, e.g., depending on video content and complexity.

[0057] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 10 that may utilize techniques for encoding/decoding digital video data using a large macroblock, i.e., a macroblock that contains more pixels than a 16x16 macroblock. As shown in FIG. 1, system 10 includes a source device 12 that transmits encoded video to a destination device 14 via a communication channel 16. Source device 12 and destination device 14 may comprise any of a wide range of devices. In some cases, source device 12 and destination device 14 may comprise wireless communication devices, such as wireless handsets, so-called cellular or satellite radiotelephones, or any wireless devices that can communicate video information over a communication channel 16, in which case communication channel 16 is wireless. The techniques of this disclosure, however, which concern use of a large macroblock comprising more pixels than macroblocks prescribed by conventional video encoding standards, are not necessarily limited to wireless applications or settings. For example, these techniques may apply to over-the-air television broadcasts, cable television transmissions, satellite television transmissions, Internet video transmissions, encoded digital video that is encoded onto a storage medium, or other scenarios. Accordingly, communication channel 16 may comprise any combination of wireless or wired media suitable for transmission of encoded video data.

[0058] In the example of FIG. 1, source device 12 may include a video source 18, video encoder 20, a modulator/demodulator (modem) 22 and a transmitter 24. Destination device 14 may include a receiver 26, a modem 28, a video decoder 30, and a display device 32. In accordance with this disclosure, video encoder 20 of source device 12 may be configured to apply one or more of the techniques for using, in a video encoding process, a large macroblock having a size that is larger than a macroblock size prescribed by conventional video encoding standards. Similarly, video decoder 30 of destination device 14 may be configured to apply one or more of the techniques for using, in a video decoding process, a

macroblock size that is larger than a macroblock size prescribed by conventional video encoding standards.

[0059] The illustrated system 10 of FIG. 1 is merely one example. Techniques for using a large macroblock as described in this disclosure may be performed by any digital video encoding and/or decoding device. Source device 12 and destination device 14 are merely examples of such coding devices in which source device 12 generates coded video data for transmission to destination device 14. In some examples, devices 12, 14 may operate in a substantially symmetrical manner such that each of devices 12, 14 include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between video devices 12, 14, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0060] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, an/or a video feed from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In some cases, if video source 18 is a video camera, source device 12 and destination device 14 may form so-called camera phones or video phones. As mentioned above, however, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless or wired applications. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 20. The encoded video information may then be modulated by modem 22 according to a communication standard, and transmitted to destination device 14 via transmitter 24. Modem 22 may include various mixers, filters, amplifiers or other components designed for signal modulation. Transmitter 24 may include circuits designed for transmitting data, including amplifiers, filters, and one or more antennas.

[0061] Receiver 26 of destination device 14 receives information over channel 16, and modem 28 demodulates the information. Again, the video encoding process may implement one or more of the techniques described herein to use a large macroblock, e.g., larger than 16x16, for inter (i.e., temporal) and/or intra (i.e., spatial) encoding of video data. The video decoding process performed by video decoder 30 may also use such techniques during the decoding process. The information communicated over channel 16 may include

syntax information defined by video encoder 20, which is also used by video decoder 30, that describes the large macroblocks, as discussed in greater detail below. Display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a cathode ray tube (CRT), a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0062] In the example of FIG. 1, communication channel 16 may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines, or any combination of wireless and wired media. Communication channel 16 may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. Communication channel 16 generally represents any suitable communication medium, or collection of different communication media, for transmitting video data from source device 12 to destination device 14, including any suitable combination of wired or wireless media. Communication channel 16 may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from source device 12 to destination device 14.

[0063] Video encoder 20 and video decoder 30 may operate according to a video compression standard, such as the ITU-T H.264 standard, alternatively described as MPEG-4, Part 10, Advanced Video Coding (AVC). The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples include MPEG-2 and ITU-T H.263. Although not shown in FIG. 1, in some aspects, video encoder 20 and video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0064] The ITU-T H.264/MPEG-4 (AVC) standard was formulated by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the product of a collective partnership known as the Joint Video Team (JVT). In some aspects, the techniques described in this disclosure may be applied to devices that generally conform to the H.264 standard. The H.264 standard is described in ITU-T

Recommendation H.264, Advanced Video Coding for generic audiovisual services, by the ITU-T Study Group, and dated March, 2005, which may be referred to herein as the H.264 standard or H.264 specification, or the H.264/AVC standard or specification. The Joint Video Team (JVT) continues to work on extensions to H.264/MPEG-4 AVC.

[0065] Video encoder 20 and video decoder 30 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. Each of video encoder 20 and video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective camera, computer, mobile device, subscriber device, broadcast device, set-top box, server, or the like.

[0066] A video sequence typically includes a series of video frames. Video encoder 20 operates on video blocks within individual video frames in order to encode the video data. A video block may correspond to a macroblock or a partition of a macroblock. A video block may further correspond to a partition of a partition. The video blocks may have fixed or varying sizes, and may differ in size according to a specified coding standard or in accordance with the techniques of this disclosure. Each video frame may include a plurality of slices. Each slice may include a plurality of macroblocks, which may be arranged into partitions, also referred to as sub-blocks. As an example, the ITU-T H.264 standard supports intra prediction in various block sizes, such as 16 by 16, 8 by 8, or 4 by 4 for luma components, and 8x8 for chroma components, as well as inter prediction in various block sizes, such as 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 for luma components and corresponding scaled sizes for chroma components. In this disclosure, "x" and "by" may be used interchangeably to refer to the pixel dimensions of the block, e.g., 16x16 pixels or 16 by 16 pixels. In general, a 16x16 block will have 16 pixels in a vertical direction and 16 pixels in a horizontal direction. The pixels in a block may be arranged in rows and columns.

[0067] Block sizes that are less than 16 by 16 may be referred to as partitions of a 16 by 16 macroblock. The techniques of this disclosure describe intra- and inter-coding for macroblocks larger than the conventional 16x16 pixel macroblock, such as 32x32 pixel

macroblocks, 64x64 pixel macroblocks, or larger macroblocks. Video blocks may comprise blocks of pixel data in the pixel domain, or blocks of transform coefficients in the transform domain, e.g., following application of a transform such as a discrete cosine transform (DCT), an integer transform, a wavelet transform, or a conceptually similar transform to the residual video block data representing pixel differences between coded video blocks and predictive video blocks. In some cases, a video block may comprises blocks of quantized transform coefficients in the transform domain.

[0068] Smaller video blocks can provide better resolution, and may be used for locations of a video frame that include high levels of detail. In general, macroblocks and the various partitions, sometimes referred to as sub-blocks, may be considered to be video blocks. In addition, a slice may be considered to be a plurality of video blocks, such as macroblocks and/or sub-blocks. Each slice may be an independently decodable unit of a video frame. Alternatively, frames themselves may be decodable units, or other portions of a frame may be defined as decodable units. The term “coded unit” or “coding unit” may refer to any independently decodable unit of a video frame such as an entire frame, a slice of a frame, or another independently decodable unit defined according to applicable coding techniques.

[0069] Following intra-predictive or inter-predictive coding to produce predictive data and residual data, and following any transforms (such as the 4x4 or 8x8 integer transform used in H.264/AVC or a discrete cosine transform DCT) to produce transform coefficients, quantization of transform coefficients may be performed. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the coefficients. The quantization process may reduce the bit depth associated with some or all of the coefficients. For example, an n -bit value may be rounded down to an m -bit value during quantization, where n is greater than m .

[0070] Following quantization, entropy coding of the quantized data may be performed, e.g., according to content adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), or another entropy coding methodology. A processing unit configured for entropy coding, or another processing unit, may perform other processing functions, such as zero run length coding of quantized coefficients and/or generation of syntax information such as CBP values, macroblock type, coding mode, or the like.

[0071] According to various techniques of this disclosure, video encoder 20 may use a macroblock that is larger than that prescribed by conventional video encoding standards to encode digital video data. In one example, video encoder 20 may encode, with a video encoder, a video block having a size of more than 16x16 pixels, generate block-type syntax information that indicates the size of the block, and generate a CBP value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient. The macroblock block-type syntax information may be provided in a macroblock header for the large macroblock. The macroblock block-type syntax information may indicate an address of the macroblock in a frame or slice, a type of coding mode applied to the macroblock, a quantization value for the macroblock, any motion vector information for the macroblock and a CBP value for the macroblock.

[0072] In another example, video encoder 20 may receive a video block having a size of more than 16x16 pixels, partitioning the block into partitions, encode one of the partitions using a first encoding mode, encode another of the partitions using a second encoding mode different from the first encoding mode, and generate block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.

[0073] In an additional example, video encoder 20 may receive a video coding unit, such as a frame or slice, determine a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels, determine a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels, encode the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric, and encode the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.

[0074] In one example, video decoder 30 may receive an encoded video block having a size of more than 16x16 pixels, receive block-type syntax information that indicates the size of the encoded block, receive a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient, and decode the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

[0075] In another example, video decoder 30 may receive a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is intra-encoded and another of the partitions is intra-encoded, receive block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions, and decode the video block based on the block-type syntax information.

[0076] FIG. 2 is a block diagram illustrating an example of a video encoder 50 that may implement techniques for using a large macroblock consistent with this disclosure. Video encoder 50 may correspond to video encoder 20 of source device 12, or a video encoder of a different device. Video encoder 50 may perform intra- and inter-coding of blocks within video frames, including large macroblocks, or partitions or sub-partitions of large macroblocks. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in video within a given video frame. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames of a video sequence.

[0077] Intra-mode (I-mode) may refer to any of several spatial based compression modes and inter-modes such as prediction (P-mode) or bi-directional (B-mode) may refer to any of several temporal-based compression modes. The techniques of this disclosure may be applied both during inter-coding and intra-coding. In some cases, techniques of this disclosure may also be applied to encoding non-video digital pictures. That is, a digital still picture encoder may utilize the techniques of this disclosure to intra-code a digital still picture using large macroblocks in a manner similar to encoding intra-coded macroblocks in video frames in a video sequence.

[0078] As shown in FIG. 2, video encoder 50 receives a current video block within a video frame to be encoded. In the example of FIG. 2, video encoder 50 includes motion compensation unit 35, motion estimation unit 36, intra prediction unit 37, mode select unit 39, reference frame store 34, summer 48, transform unit 38, quantization unit 40, and entropy coding unit 46. For video block reconstruction, video encoder 50 also includes inverse quantization unit 42, inverse transform unit 44, and summer 51. A deblocking filter (not shown in FIG. 2) may also be included to filter block boundaries to remove blockiness

artifacts from reconstructed video. If desired, the deblocking filter would typically filter the output of summer 51.

[0079] During the encoding process, video encoder 50 receives a video frame or slice to be coded. The frame or slice may be divided into multiple video blocks, including large macroblocks. Motion estimation unit 36 and motion compensation unit 35 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference frames to provide temporal compression. Intra prediction unit 37 performs intra-predictive coding of the received video block relative to one or more neighboring blocks in the same frame or slice as the block to be coded to provide spatial compression.

[0080] Mode select unit 39 may select one of the coding modes, intra or inter, e.g., based on error results, and provides the resulting intra- or inter-coded block to summer 48 to generate residual block data and to summer 51 to reconstruct the encoded block for use as a reference frame. In accordance with the techniques of this disclosure, the video block to be coded may comprise a macroblock that is larger than that prescribed by conventional coding standards, i.e., larger than a 16x16 pixel macroblock. For example, the large video block may comprise a 64x64 pixel macroblock or a 32x32 pixel macroblock.

[0081] Motion estimation unit 36 and motion compensation unit 35 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a predictive block within a predictive reference frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit). A predictive block is a block that is found to closely match the block to be coded, in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics.

[0082] A motion vector may also indicate displacement of a partition of a large macroblock. In one example with respect to a 64x64 pixel macroblock with a 32x64 partition and two 32x32 partitions, a first motion vector may indicate displacement of the 32x64 partition, a second motion vector may indicate displacement of a first one of the 32x32 partitions, and a third motion vector may indicate displacement of a second one of the 32x32 partitions, all relative to corresponding partitions in a reference frame. Such partitions may also be considered video blocks, as those terms are used in this disclosure.

Motion compensation may involve fetching or generating the predictive block based on the motion vector determined by motion estimation. Again, motion estimation unit 36 and motion compensation unit 35 may be functionally integrated.

[0083] Motion estimation unit 36 calculates a motion vector for the video block of an inter-coded frame by comparing the video block to video blocks of a reference frame in reference frame store 34. Motion compensation unit 35 may also interpolate sub-integer pixels of the reference frame, e.g., an I-frame or a P-frame. The ITU H.264 standard refers to reference frames as “lists.” Therefore, data stored in reference frame store 34 may also be considered lists. Motion estimation unit 36 compares blocks of one or more reference frames (or lists) from reference frame store 34 to a block to be encoded of a current frame, e.g., a P-frame or a B-frame. When the reference frames in reference frame store 34 include values for sub-integer pixels, a motion vector calculated by motion estimation unit 36 may refer to a sub-integer pixel location of a reference frame. Motion estimation unit 36 sends the calculated motion vector to entropy coding unit 46 and motion compensation unit 35. The reference frame block identified by a motion vector may be referred to as a predictive block. Motion compensation unit 35 calculates error values for the predictive block of the reference frame.

[0084] Motion compensation unit 35 may calculate prediction data based on the predictive block. Video encoder 50 forms a residual video block by subtracting the prediction data from motion compensation unit 35 from the original video block being coded. Summer 48 represents the component or components that perform this subtraction operation. Transform unit 38 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform coefficient values. Transform unit 38 may perform other transforms, such as those defined by the H.264 standard, which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform unit 38 applies the transform to the residual block, producing a block of residual transform coefficients. The transform may convert the residual information from a pixel value domain to a transform domain, such as a frequency domain.

[0085] Quantization unit 40 quantizes the residual transform coefficients to further reduce bit rate. The quantization process may reduce the bit depth associated with some or all of the coefficients. In one example, quantization unit 40 may establish a different degree of quantization for each 64x64 pixel macroblock according to a luminance quantization parameter, referred to in this disclosure as QP_Y . Quantization unit 40 may further modify the luminance quantization parameter used during quantization of a 64x64 macroblock based on a quantization parameter modifier, referred to herein as “MB64_delta_QP,” and a previously encoded 64x64 pixel macroblock.

[0086] Each 64x64 pixel large macroblock may comprise an individual MB64_delta_QP value, in the range between -26 and +25, inclusive. In general, video encoder 50 may establish the MB64_delta_QP value for a particular block based on a desired bitrate for transmitting the encoded version of the block. The MB64_delta_QP value of a first 64x64 pixel macroblock may be equal to the QP value of a frame or slice that includes the first 64x64 pixel macroblock, e.g., in the frame/slice header. QP_Y for a current 64x64 pixel macroblock may be calculated according to the formula:

$$QP_Y = (QP_{Y,PREV} + MB64_delta_QP + 52) \% 52$$

where $QP_{Y,PREV}$ refers to the QP_Y value of the previous 64x64 pixel macroblock in the decoding order of the current slice/frame, and where “%” refers to the modulo operator such that $N \% 52$ returns a result between 0 and 51, inclusive, corresponding to the remainder value of N divided by 52. For a first macroblock in a frame/slice, $QP_{Y,PREV}$ may be set equal to the frame/slice QP sent in the frame/slice header.

[0087] In one example, quantization unit 40 presumes that the MB64_delta_QP value is equal to zero when a MB64_delta_QP value is not defined for a particular 64x64 pixel macroblock, including “skip” type macroblocks, such as P_Skip and B_Skip macroblock types. In some examples, additional delta_QP values may be defined for finer grain quantization control of partitions within a 64x64 pixel macroblock, such as MB32_delta_QP values for each 32x32 pixel partition of a 64x64 pixel macroblock. In some examples, each partition of a 64x64 macroblock may be assigned an individual quantization parameter. Using an individualized quantization parameter for each partition

may result in more efficient quantization of a macroblock, e.g., to better adjust quantization for a non-homogeneous area, instead of using a single QP for a 64x64 macroblock.

[0088] Following quantization, entropy coding unit 46 entropy codes the quantized transform coefficients. For example, entropy coding unit 46 may perform content adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), or another entropy coding technique. Following the entropy coding by entropy coding unit 46, the encoded video may be transmitted to another device or archived for later transmission or retrieval. The coded bitstream may include entropy coded residual transform coefficient blocks, motion vectors for such blocks, MB64_delta_QP values for each 64x64 pixel macroblock, and other syntax elements including, for example, macroblock-type identifier values, QP_Y values, coded block pattern (CBP) values, values that identify a partitioning method of a macroblock or sub-block, and transform size flag values, as discussed in greater detail below. In the case of context adaptive binary arithmetic coding, context may be based on neighboring macroblocks.

[0089] In some cases, entropy coding unit 46 or another unit of video encoder 50 may be configured to perform other coding functions, in addition to entropy coding. For example, entropy coding unit 46 may be configured to determine the CBP values for the large macroblocks and partitions. Entropy coding unit 46 may apply a hierarchical CBP scheme to provide a CBP value for a large macroblock that indicates whether any partitions in the macroblock include non-zero transform coefficient values and, if so, other CBP values to indicate whether particular partitions within the large macroblock have non-zero transform coefficient values. Also, in some cases, entropy coding unit 46 may perform run length coding of the coefficients in a large macroblock or subpartition. In particular, entropy coding unit 46 may apply a zig-zag scan or other scan pattern to scan the transform coefficients in a macroblock or partition and encode runs of zeros for further compression. Entropy coding unit 46 also may construct header information with appropriate syntax elements for transmission in the encoded video bitstream.

[0090] Inverse quantization unit 42 and inverse transform unit 44 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block in the pixel domain, e.g., for later use as a reference block. Motion compensation unit 35 may calculate a reference block by adding the residual block to a predictive block of one of the

frames of reference frame store 34. Motion compensation unit 35 may also apply one or more interpolation filters to the reconstructed residual block to calculate sub-integer pixel values. Summer 51 adds the reconstructed residual block to the motion compensated prediction block produced by motion compensation unit 35 to produce a reconstructed video block for storage in reference frame store 34. The reconstructed video block may be used by motion estimation unit 36 and motion compensation unit 35 as a reference block to inter-code a block in a subsequent video frame. The large macroblock may comprise a 64x64 pixel macroblock, a 32x32 pixel macroblock, or other macroblock that is larger than the size prescribed by conventional video coding standards.

[0091] FIG. 3 is a block diagram illustrating an example of a video decoder 60, which decodes a video sequence that is encoded in the manner described in this disclosure. The encoded video sequence may include encoded macroblocks that are larger than the size prescribed by conventional video encoding standards. For example, the encoded macroblocks may be 32x32 pixel or 64x64 pixel macroblocks. In the example of FIG. 3, video decoder 60 includes an entropy decoding unit 52, motion compensation unit 54, intra prediction unit 55, inverse quantization unit 56, inverse transformation unit 58, reference frame store 62 and summer 64. Video decoder 60 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 50 (FIG. 2). Motion compensation unit 54 may generate prediction data based on motion vectors received from entropy decoding unit 52.

[0092] Entropy decoding unit 52 entropy-decodes the received bitstream to generate quantized coefficients and syntax elements (e.g., motion vectors, CBP values, QP_Y values, transform size flag values, MB64_delta_QP values). Entropy decoding unit 52 may parse the bitstream to identify syntax information in coded units such as frames, slices and/or macroblock headers. Syntax information for a coded unit comprising a plurality of macroblocks may indicate the size of the macroblocks, e.g., 16x16 pixels, 32x32 pixels, 64x64 pixels, or other larger sized macroblocks. The syntax information for a block is forwarded from entropy coding unit 52 to either motion compensation unit 54 or intra-prediction unit 55, e.g., depending on the coding mode of the block.

[0093] Motion compensation unit 54 may use motion vectors received in the bitstream to identify a prediction block in reference frames in reference frame store 62. Intra prediction

unit 55 may use intra prediction modes received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 56 inverse quantizes, i.e., de-quantizes, the quantized block coefficients provided in the bitstream and decoded by entropy decoding unit 52. The inverse quantization process may include a conventional process, e.g., as defined by the H.264 decoding standard. The inverse quantization process may also include use of a quantization parameter QP_Y calculated by encoder 50 for each 64x64 macroblock to determine a degree of quantization and, likewise, a degree of inverse quantization that should be applied.

[0094] Inverse transform unit 58 applies an inverse transform, e.g., an inverse DCT, an inverse integer transform, or a conceptually similar inverse transform process, to the transform coefficients in order to produce residual blocks in the pixel domain. Motion compensation unit 54 produces motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used for motion estimation with sub-pixel precision may be included in the syntax elements. Motion compensation unit 54 may use interpolation filters as used by video encoder 50 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 54 may determine the interpolation filters used by video encoder 50 according to received syntax information and use the interpolation filters to produce predictive blocks.

[0095] Motion compensation unit 54 uses some of the syntax information to determine sizes of macroblocks used to encode frame(s) of the encoded video sequence, partition information that describes how each macroblock of a frame of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (or lists) for each inter-encoded macroblock or partition, and other information to decode the encoded video sequence.

[0096] Summer 64 sums the residual blocks with the corresponding prediction blocks generated by motion compensation unit 54 or intra-prediction unit to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in reference frame store 62, which provides reference blocks for subsequent motion compensation and also produces decoded video for presentation on a display device (such as device 32 of FIG. 1).

The decoded video blocks may each comprise a 64x64 pixel macroblock, 32x32 pixel macroblock, or other larger-than-standard macroblock. Some macroblocks may include partitions with a variety of different partition sizes.

[0097] FIG. 4A is a conceptual diagram illustrating example partitioning among various partition levels of a large macroblock. Blocks of each partition level include a number of pixels corresponding to the particular level. Four partitioning patterns are also shown for each level, where a first partition pattern includes the whole block, a second partition pattern includes two horizontal partitions of equal size, a third partition pattern includes two vertical partitions of equal size, and a fourth partition pattern includes four equally-sized partitions. One of the partitioning patterns may be chosen for each partition at each partition level.

[0098] In the example of FIG. 4A, level 0 corresponds to a 64x64 pixel macroblock partition of luma samples and associated chroma samples. Level 1 corresponds to a 32x32 pixel block of luma samples and associated chroma samples. Level 2 corresponds to a 16x16 pixel block of luma samples and associated chroma samples, and level 3 corresponds to an 8x8 pixel block of luma samples and associated chroma samples.

[0099] In other examples, additional levels could be introduced to utilize larger or smaller numbers of pixels. For example, level 0 could begin with a 128x128 pixel macroblock, a 256x256 pixel macroblock, or other larger-sized macroblock. The highest-numbered level, in some examples, could be as fine-grain as a single pixel, i.e., a 1x1 block. Hence, from the lowest to highest levels, partitioning may be increasingly sub-partitioned, such that the macroblock is partitioned, partitions are further partitioned, further partitions are still further partitioned, and so forth. In some instances, partitions below level 0, i.e., partitions of partitions, may be referred to as sub-partitions.

[00100] When a block at one level is partitioned using four equally-sized sub-blocks, any or all of the sub-blocks may be partitioned according to the partition patterns of the next level. That is, for an $N \times N$ block that has been partitioned at level x into four equally sized sub-blocks $(N/2) \times (N/2)$, any of the $(N/2) \times (N/2)$ sub-blocks can be further partitioned according to any of the partition patterns of level $x+1$. Thus, a 32x32 pixel sub-block of a 64x64 pixel macroblock at level 0 can be further partitioned according to any of the patterns shown in FIG. 4A at level 1, e.g., 32x32, 32x16 and 32x16, 16x32 and 16x32, or

16x16, 16x16, 16x16 and 16x16. Likewise, where four 16x16 pixel sub-blocks result from a 32x32 pixel sub-block being partitioned, each of the 16x16 pixel sub-blocks can be further partitioned according to any of the patterns shown in FIG. 4A at level 2. Where four 8x8 pixel sub-blocks result from a 16x16 pixel sub-block being partitioned, each of the 8x8 pixel sub-blocks can be further partitioned according to any of the patterns shown in FIG. 4A at level 3.

[00101] Using the example four levels of partitions shown in FIG. 4A, large homogeneous areas and fine sporadic changes can be adaptively represented by an encoder implementing the framework and techniques of this disclosure. For example, video encoder 50 may determine different partitioning levels for different macroblocks, as well as coding modes to apply to such partitions, e.g., based on rate-distortion analysis. Also, as described in greater detail below, video encoder 50 may encode at least some of the final partitions differently, using spatial (P-encoded or B-encoded) or temporal (I-encoded) prediction, e.g., based on rate-distortion metric results or other considerations.

[00102] Instead of coding a large macroblock uniformly such that all partitions have the same intra- or inter-coding mode, a large macroblock may be coded such that some partitions have different coding mode. For example, some (at least one) partitions may be coded with different intra-coding modes (e.g., I_16x16, I_8x8, I_4x4) relative to other (at least one) partitions in the same macroblock. Also, some (at least one) partitions may be intra-coded while other (at least one) partitions in the same macroblock are inter-coded.

[00103] For example, video encoder 50 may, for a 32x32 block with four 16x16 partitions, encode some of the 16x16 partitions using spatial prediction and other 16x16 partitions using temporal prediction. As another example, video encoder 50 may, for a 32x32 block with four 16x16 partitions, encode one or more of the 16x16 partitions using a first prediction mode (e.g., one of I_16x16, I_8x8, I_4x4) and one or more other 16x16 partitions using a different spatial prediction mode (e.g., one of I_16x16, I_8x8, I_4x4).

[00104] FIG. 4B is a conceptual diagram illustrating assignment of different coding modes to different partitions a large macroblock. In particular, FIG. 4B illustrates assignment of an I_16x16 intra-coding mode to an upper left 16x16 block of a large 32x32 macroblock, I_8x8 intra-coding modes to upper right and lower left 16x16 blocks of the large 32x32 macroblock, and an I_4x4 intra-coding mode to a lower right 16x16 block of

the large 32x32 macroblock. In some cases, the coding modes illustrated in FIG. 4B may be H.264 intra-coding modes for luma coding.

[00105] In the manner described, each partition can be further partitioned on a selective basis, and each final partition can be selectively coded using either temporal prediction or spatial prediction, and using selected temporal or spatial coding modes. Consequently, it is possible to code a large macroblock with mixed modes such that some partitions in the macroblock are intra-coded and other partitions in the same macroblock are inter-coded, or some partitions in the same macroblock are coded with different intra-coding modes or different inter-coding modes.

[00106] Video encoder 50 may further define each partition according to a macroblock type. The macroblock type may be included as a syntax element in an encoded bitstream, e.g., as a syntax element in a macroblock header. In general, the macroblock type may be used to identify how the macroblock is partitioned, and the respective methods or modes for encoding each of the partitions of the macroblock, as discussed above. Methods for encoding the partitions may include not only intra- and inter-coding, but also particular modes of intra-coding (e.g., I_16x16, I_8x8, I_4x4) or inter-coding (e.g., P_ or B_ 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4).

[00107] As discussed with respect to the example of Table 1 below in greater detail for P-blocks and with respect to the example of Table 2 below for B-blocks, partition level 0 blocks may be defined according to an MB64_type syntax element, representative of a macroblock with 64x64 pixels. Similar type definitions may be formed for any MB[N]_type, where [N] refers to a block with NxN pixels. When an NxN block has four partitions of size (N/2)x(N/2), as shown in the last column on FIG. 4A, each of the four partitions may receive their own type definitions, e.g., MB[N/2]_type. For example, for a 64x64 pixel block (of type MB64_type) with four 32x32 pixel partitions, video encoder 50 may introduce an MB32_type for each of the four 32x32 pixel partitions. These macroblock type syntax elements may assist decoder 60 in decoding large macroblocks and various partitions of large macroblocks, as described in this disclosure.

[00108] In general, each MB[N]_type definition may represent, for a corresponding type, a number of pixels in a block of the corresponding type (e.g., 64x64), a reference frame (or reference list) for the block, a number of partitions for the block, the size of each

partition of the block, how each partition is encoded (e.g., intra or inter and particular modes), and the reference frame (or reference list) for each partition of the block when the partition is inter-coded. For 16x16 and smaller blocks, video encoder 50 may, in some examples, use conventional type definitions as the types of the blocks, such as types specified by the H.264 standard. In other examples, video encoder 50 may applied newly defined block types for 16x16 and smaller blocks.

[00109] Video encoder 50 may evaluate both conventional inter- or intra-coding methods using normal macroblock sizes and partitions, such as methods prescribed by ITU H.264, and inter- or intra-coding methods using the larger macroblocks and partitions described by this disclosure, and compare the rate-distortion characteristics of each approach to determine which method results in the best rate-distortion performance. Video encoder 50 then may select, and apply to the block to be coded, the best coding approach, including inter- or intra-mode, macroblock size (large, larger or normal), and partitioning, based on optimal or acceptable rate-distortion results for the coding approach. As an illustration, video encoder 50 may select the use of 64x64 macroblocks, 32x32 macroblocks or 16x16 macroblocks to encode a particular frame or slice based on rate-distortion results produced when the video encoder uses such macroblock sizes.

[00110] In general, two different approaches may be used to design intra modes using large macroblocks. As one example, during intra-coding, spatial prediction may be performed for a block based on neighboring blocks directly. In accordance with the techniques of this disclosure, video encoder 50 may generate spatial predictive 32x32 blocks based on their neighboring pixels directly and generate spatial predictive 64x64 blocks based on their neighboring pixels directly. In this manner, spatial prediction may be performed at a larger scale compared to 16x16 intra blocks. Therefore, these techniques may, in some examples, result in some bit rate savings, e.g., with a smaller number of blocks or partitions per frame or slice.

[00111] As another example, video encoder 50 may group four NxN blocks together to generate an $(N*2) \times (N*2)$ block, and then encode the $(N*2) \times (N*2)$ block. Using existing H.264 intra-coding modes, video encoder 50 may group four intra-coded blocks together, thereby forming a large intra-coded macroblock. For example, four intra-coded blocks, each having a size of 16x16, can be grouped together to form a large, 32x32 intra-coded

block. Video encoder 50 may encode each of the four corresponding NxN blocks using a different encoding mode, e.g., I_16x16, I_8x8, or I_4x4 according to H.264. In this manner, each 16x16 block can be assigned its own mode of spatial prediction by video encoder 50, e.g., to promote favorable encoding results.

[00112] Video encoder 50 may design intra modes according to either of the two different methods discussed above, and analyze the different methods to determine which approach provides better encoding results. For example, video encoder 50 may apply the different intra mode approaches, and place them in a single candidate pool to allow them to compete with each other for the best rate-distortion performance. Using a rate-distortion comparison between the different approaches, video encoder 50 can determine how to encode each partition and/or macroblock. In particular, video encoder 50 may select the coding modes that produce the best rate-distortion performance for a given macroblock, and apply those coding modes to encode the macroblock.

[00113] FIG. 5 is a conceptual diagram illustrating a hierarchical view of various partition levels of a large macroblock. FIG. 5 also represents the relationships between various partition levels of a large macroblock as described with respect to FIG. 4A. Each block of a partition level, as illustrated in the example of FIG. 5, may have a corresponding coded block pattern (CBP) value. The CBP values form part of the syntax information that describes a block or macroblock. In one example, the CBP values are each one-bit syntax values that indicate whether or not there are any nonzero transform coefficient values in a given block following transform and quantization operations.

[00114] In some cases, a prediction block may be very close in pixel content to a block to be coded such that all of the residual transform coefficients are quantized to zero, in which case there may be no need to transmit transform coefficients for the coded block. Instead, the CBP value for the block may be set to zero to indicate that the coded block includes no non-zero coefficients. Alternatively, if a block includes at least one non-zero coefficient, the CBP value may be set to one. Decoder 60 may use CBP values to identify residual blocks that are coded, i.e., with one or more non-zero transform coefficients, versus blocks that are not coded, i.e., including no non-zero transform coefficients.

[00115] In accordance with some of the techniques described in this disclosure, an encoder may assign CBP values to large macroblocks hierarchically based on whether those

macroblocks, including their partitions, have at least one non-zero coefficient, and assign CBP values to the partitions to indicate which partitions have non-zero coefficients. Hierarchical CBP for large macroblocks can facilitate processing of large macroblocks to quickly identify coded large macroblocks and uncoded large macroblocks, and permit identification of coded partitions at each partition level for the large macroblock to determine whether it is necessary to use residual data to decode the blocks.

[00116] In one example, a 64x64 pixel macroblock at level zero may include syntax information comprising a CBP64 value, e.g., a one-bit value, to indicate whether the entire 64x64 pixel macroblock, including any partitions, has non-zero coefficients or not. In one example, video encoder 50 “sets” the CBP64 bit, e.g., to a value of “1,” to represent that the 64x64 pixel macroblock includes at least one non-zero coefficient. Thus, when the CBP64 value is set, e.g., to a value of “1,” the 64x64 pixel macroblock includes at least one non-zero coefficient somewhere in the macroblock. In another example, video encoder 50 “clears” the CBP64 value, e.g., to a value of “0,” to represent that the 64x64 pixel macroblock has all zero coefficients. Thus, when the CBP64 value is cleared, e.g., to a value of “0,” the 64x64 pixel macroblock is indicated as having all zero coefficients. Macroblocks with CBP64 values of “0” do not generally require transmission of residual data in the bitstream, whereas macroblocks with CBP64 values of “1” generally require transmission of residual data in the bitstream for use in decoding such macroblocks.

[00117] A 64x64 pixel macroblock that has all zero coefficients need not include CBP values for partitions or sub-blocks thereof. That is, because the 64x64 pixel macroblock has all zero coefficients, each of the partitions also necessarily has all zero coefficients. On the contrary, a 64x64 pixel macroblock that includes at least one non-zero coefficient may further include CBP values for the partitions at the next partition level. For example, a CBP64 with a value of one may include additional syntax information in the form of a one-bit value CBP32 for each 32x32 partition of the 64x64 block. That is, in one example, each 32x32 pixel partition (such as the four partition blocks of level 1 in FIG. 5) of a 64x64 pixel macroblock is assigned a CBP32 value as part of the syntax information of the 64x64 pixel macroblock. As with the CBP64 value, each CBP32 value may comprise a bit that is set to a value of one when the corresponding 32x32 pixel block has at least one

non-zero coefficient and that is cleared to a value of zero when the corresponding 32x32 pixel block has all zero coefficients.

[00118] In this manner, a 64x64 pixel macroblock that has all zero coefficients may use a single bit to represent the fact that the macroblock has all zero coefficients, whereas a 64x64 pixel macroblock with at least one non-zero coefficient may include CBP syntax information comprising at least five bits, a first bit to represent that the 64x64 pixel macroblock has a non-zero coefficient and four additional bits, each representative of whether a corresponding one of four 32x32 pixel partitions of the macroblock includes at least one non-zero coefficient. In some examples, when the first three of the four additional bits are zero, the fourth additional bit may not be included, which the decoder may interpret as the last partition being one. That is, the encoder may determine that the last bit has a value of one when the first three bits are zero and when the bit representative of the higher level hierarchy has a value of one. For example, a prefix of a CBP64 value of "10001" may be shortened to "1000," as the first bit indicates that at least one of the four partitions has non-zero coefficients, and the next three zeros indicate that the first three partitions have all zero coefficients. Therefore, a decoder may deduce that it is the last partition that includes a non-zero coefficient, without the explicit bit informing the decoder of this fact, e.g., from the bit string "1000." That is, the decoder may interpret the CBP64 prefix "1000" as "10001."

[00119] Likewise, a one-bit CBP32 may be set to a value of "1" when the 32x32 pixel partition includes at least one non-zero coefficient, and to a value of "0" when all of the coefficients have a value of zero. If a 32x32 pixel partition has a CBP value of 1, then partitions of that 32x32 partition at the next partition level may be assigned CBP values to indicate whether the respective partitions include any non-zero coefficients. Hence, the CBP values may be assigned in a hierarchical manner at each partition level until there are no further partition levels or no partitions including non-zero coefficients.

[00120] In the above manner, encoders and/or decoders may utilize hierarchical CBP values to represent whether a large macroblock (e.g., 64x64 or 32x32) and partitions thereof include at least one non-zero coefficient or all zero coefficients. Accordingly, an encoder may encode a large macroblock of a coded unit of a digital video stream, such that the macroblock block comprises greater than 16x16 pixels, generate block-type syntax

information that identifies the size of the block, generate a CBP value for the block, such that the CBP value identifies whether the block includes at least one non-zero coefficient, and generate additional CBP values for various partition levels of the block, if applicable.

[00121] In one example, the hierarchical CBP values may comprise an array of bits (e.g., a bit vector) whose length depends on the values of the prefix. The array may further represent a hierarchy of CBP values, such as a tree structure, as shown in FIG. 5. The array may represent nodes of the tree in a breadth-first manner, where each node corresponds to a bit in the array. When a node of the tree has a bit that is set to “1,” in one example, the node has four branches (corresponding to the four partitions), and when the bit is cleared to “0,” the node has no branches.

[00122] In this example, to identify the values of the nodes that branch from a particular node X , an encoder and/or a decoder may determine the four consecutive bits starting at node Y that represent the nodes that branch from node x by calculating:

$$y = \left(4 * \sum_{i=0}^x tree[i] \right) - 3$$

where $tree[]$ corresponds to the array of bits with a starting index of 0, i is an integer index into the array $tree[]$, x corresponds to the index of node X in $tree[]$, and y corresponds to the index of node Y that is the first branch-node of node X . The three subsequent array positions (i.e., $y+1$, $y+2$, and $y+3$) correspond to the other branch-nodes of node X .

[00123] An encoder, such as video encoder 50 (FIG. 2), may assign CBP values for 16x16 pixel partitions of the 32x32 pixel partitions with at least one non-zero coefficient using existing methods, such as methods prescribed by ITU H.264 for setting CBP values for 16x16 blocks, as part of the syntax of the 64x64 pixel macroblock. The encoder may also select CBP values for the partitions of the 32x32 pixel partitions that have at least one non-zero coefficient based on the size of the partitions, a type of block corresponding to the partitions (e.g., chroma block or luma block), or other characteristics of the partitions. Example methods for setting a CBP value of a partition of a 32x32 pixel partition are discussed in further detail with respect to FIGS. 8 and 9.

[00124] FIGS. 6–9 are flowcharts illustrating example methods for setting various coded block pattern (CBP) values in accordance with the techniques of this disclosure. Although the example methods of FIGS. 6–9 are discussed with respect to a 64x64 pixel macroblock, it should be understood that similar techniques may apply for assigning hierarchical CBP values for other sizes of macroblocks. Although the examples of FIGS. 6–9 are discussed with respect to video encoder 50 (FIG. 2), it should be understood that other encoders may employ similar methods to assign CBP values to larger-than-standard macroblocks. Likewise, decoders may utilize similar, albeit reciprocal, methods for interpreting the meaning of a particular CBP value for a macroblock. For example, if an inter-coded macroblock received in the bitstream has a CBP value of “0,” the decoder may receive no residual data for the macroblock and may simply produce a predictive block identified by a motion vector as the decoded macroblock, or a group of predictive blocks identified by motion vectors with respect to partitions of the macroblock.

[00125] FIG. 6 is a flowchart illustrating an example method for setting a CBP64 value of an example 64x64 pixel macroblock. Similar methods may be applied for macroblocks larger than 64x64. Initially, video encoder 50 receives a 64x64 pixel macroblock (100). Motion estimation unit 36 and motion compensation unit 35 may then generate one or more motion vectors and one or more residual blocks to encode the macroblock, respectively. The output of transform unit 38 generally comprises an array of residual transform coefficient values for an intra-coded block or a residual block of an inter-coded block, which array is quantized by quantization unit 40 to produce a series of quantized transform coefficients.

[00126] Entropy coding unit 46 may provide entropy coding and other coding functions separate from entropy coding. For example, in addition to CAVLC, CABAC, or other entropy coding functions, entropy coding unit 46 or another unit of video encoder 50 may determine CBP values for the large macroblocks and partitions. In particular, entropy coding unit 46 may determine the CBP64 value for a 64x64 pixel macroblock by first determining whether the macroblock has at least one non-zero, quantized transform coefficient (102). When entropy coding unit 46 determines that all of the transform coefficients have a value of zero (“NO” branch of 102), entropy coding unit 46 clears the CBP64 value for the 64x64 macroblock, e.g., resets a bit for the CBP64 value to “0” (104).

When entropy coding unit 46 identifies at least one non-zero coefficient (“YES” branch of 102) for the 64x65 macroblock, entropy coding unit 46 sets the CBP64 value, e.g., sets a bit for the CBP64 value to “1” (106).

[00127] When the macroblock has all zero coefficients, entropy coding unit 46 does not need to establish any additional CBP values for the partitions of the macroblock, which may reduce overhead. In one example, when the macroblock has at least one non-zero coefficient, however, entropy coding unit 46 proceeds to determine CBP values for each of the four 32x32 pixel partitions of the 64x64 pixel macroblock (108). Entropy coding unit 46 may utilize the method described with respect to FIG. 7 four times, once for each of the four partitions, to establish four CBP32 values, each corresponding to a different one of the four 32x32 pixel partitions of the 64x64 macroblock. In this manner, when a macroblock has all zero coefficients, entropy coding unit 46 may transmit a single bit with a value of “0” to indicate that the macroblock has all zero coefficients, whereas when the macroblock has at least one non-zero coefficient, entropy coding unit 46 may transmit five bits, one bit for the macroblock and four bits, each corresponding to one of the four partitions of the macroblock. In addition, when a partition includes at least one non-zero coefficient, residual data for the partition may be sent in the encoded bitstream. As with the example of the CBP64 discussed above, when the first three of the four additional bits are zero, the fourth additional bit may not be necessary, because the decoder may determine that it has a value of one. Thus in some examples, the encoder may only send three zeros, i.e., “000,” rather than three zeros and a one, i.e., “0001.”

[00128] FIG. 7 is a flowchart illustrating an example method for setting a CBP32 value of a 32x32 pixel partition of a 64x64 pixel macroblock. Initially, for the next partition level, entropy coding unit 46 receives a 32x32 pixel partition of the macroblock (110), e.g., one of the four partitions referred to with respect to FIG. 6. Entropy coding unit 46 then determines a CBP32 value for the 32x32 pixel partition by first determining whether the partition includes at least one non-zero coefficient (112). When entropy coding unit 46 determines that all of the coefficients for the partition have a value of zero (“NO” branch of 112), entropy coding unit 46 clears the CBP32 value, e.g., resets a bit for the CBP32 value to “0” (114). When entropy coding unit 46 identifies at least one non-zero

coefficient of the partition ("YES" branch of 112), entropy coding unit 46 sets the CBP32 value, e.g., sets a bit for the CBP32 value to a value of "1" (116).

[00129] In one example, when the partition has all zero coefficients, entropy coding unit 46 does not establish any additional CBP values for the partition. When a partition includes at least one non-zero coefficient, however, entropy coding unit 46 determines CBP values for each of the four 16x16 pixel partitions of the 32x32 pixel partition of the macroblock. Entropy coding unit 46 may utilize the method described with respect to FIG. 8 to establish four CBP16 values each corresponding to one of the four 16x16 pixel partitions.

[00130] In this manner, when a partition has all zero coefficients, entropy coding unit 46 may set a bit with a value of "0" to indicate that the partition has all zero coefficients, whereas when the partition has at least one non-zero coefficient, entropy coding unit 46 may include five bits, one bit for the partition and four bits each corresponding to a different one of the four sub-partitions of the partition of the macroblock. Hence, each additional partition level may present four additional CBP bits when the partition in the preceding partition level had at least one nonzero transform coefficient value. As one example, if a 64x64 macroblock has a CBP value of 1, and four 32x32 partitions have CBP values of 1, 0, 1 and 1, respectively, the overall CBP value up to that point is 11011. Additional CBP bits may be added for additional partitions of the 32x32 partitions, e.g., into 16x16 partitions.

[00131] FIG. 8 is a flowchart illustrating an example method for setting a CBP16 value of a 16x16 pixel partition of a 32x32 pixel partition of a 64x64 pixel macroblock. For certain 16x16 pixel partitions, video encoder 50 may utilize CBP values as prescribed by a video coding standard, such as ITU H.264, as discussed below. For other 16x16 partitions, video encoder 50 may utilize CBP values in accordance with other techniques of this disclosure. Initially, as shown in FIG. 8, entropy coding unit 46 receives a 16x16 partition (120), e.g., one of the 16x16 partitions of a 32x32 partition described with respect to FIG. 7.

[00132] Entropy coding unit 46 may then determine whether a motion partition for the 16x16 pixel partition is larger than an 8x8 pixel block (122). In general, a motion partition describes a partition in which motion is concentrated. For example, a 16x16 pixel

partition with only one motion vector may be considered a 16x16 motion partition. Similarly, for a 16x16 pixel partition with two 8x16 each having one motion vector, each of the two 8x16 partitions may be considered 8x16 motion partitions. In any case, when the motion partition is not larger than an 8x8 pixel block ("NO" branch of 122), entropy coding unit 46 assigns a CBP value to the 16x16 pixel partition in the same manner as prescribed by ITU H.264 (124), in the example of FIG. 8.

[00133] When there exists a motion partition for the 16x16 pixel partition that is larger than an 8x8 pixel block ("YES" branch of 122), entropy coding unit 46 constructs and sends a lumacbp16 value (125) using the steps following step 125. In the example of FIG. 8, to construct the lumacbp16 value, entropy coding unit 46 determines whether the 16x16 pixel luma component of the partition has at least one non-zero coefficient (126). When the 16x16 pixel luma component has all zero coefficients ("NO" branch of 126), entropy coding unit 46 assigns the CBP16 value according to the Coded Block Pattern Chroma portion of ITU H.264 (128), in the example of FIG. 8.

[00134] When entropy coding unit 46 determines that the 16x16 pixel luma component has at least one non-zero coefficient ("YES" branch of 126), entropy coding unit 46 determines a transform-size flag for the 16x16 pixel partition (130). The transform-size flag generally indicates a transform being used for the partition. The transform represented by the transform-size flag may include one of a 4x4 transform, an 8x8 transform, a 16x16 transform, a 16x8 transform, or an 8x16 transform. The transform-size flag may comprise an integer value that corresponds to an enumerated value that identifies one of the possible transforms. Entropy coding unit 46 may then determine whether the transform-size flag represents that the transform size is greater than or equal to 16x8 (or 8x16) (132).

[00135] When the transform-size flag does not indicate that the transform size is greater than or equal to 16x8 (or 8x16) ("NO" branch of 132), entropy coding unit 46 assigns a value to CBP16 according to ITU H.264 (134), in the example of FIG. 8. When the transform-size flag indicates that the transform size is greater than or equal to 16x8 (or 8x16) ("YES" branch of 132), entropy coding unit 46 then determines whether a type for the 16x16 pixel partition is either two 16x8 or two 8x16 pixel partitions (136).

[00136] When the type for the 16x16 pixel partition is not two 16x8 and not two 8x16 pixel partitions (“NO” branch of 138), entropy coding unit 46 assigns the CBP16 value according to the Chroma Coded Block Partition prescribed by ITU H.264 (140), in the example of FIG. 8. When the type for the 16x16 pixel partition is either two 16x8 or two 8x16 pixel partitions (“YES” branch of 136), entropy coding unit 46 also uses the Chroma Coded Block Pattern prescribed by ITU H.264, but in addition assigns the CBP16 value a two-bit luma16x8_CBP value (142), e.g., according to the method described with respect to FIG. 9.

[00137] FIG. 9 is a flowchart illustrating an example method for determining a two-bit luma16x8_CBP value. Entropy coding unit 46 receives a 16x16 pixel partition that is further partitioned into two 16x8 or two 8x16 pixel partitions (150). Entropy coding unit 46 generally assigns each bit of luma16x8_CBP according to whether a corresponding sub-block of the 16x16 pixel partition includes at least one non-zero coefficient.

[00138] Entropy coding unit 46 determines whether a first sub-block of the 16x16 pixel partition has at least one non-zero coefficient to determine whether the first sub-block has at least one non-zero coefficient (152). When the first sub-block has all zero coefficients (“NO” branch of 152), entropy coding unit 46 clears the first bit of luma16x8_CBP, e.g., assigns luma16x8_CBP[0] a value of “0” (154). When the first sub-block has at least one non-zero coefficient (“YES” branch of 152), entropy coding unit 46 sets the first bit of luma16x8_CBP, e.g., assigns luma16x8_CBP[0] a value of “1” (156).

[00139] Entropy coding unit 46 also determines whether a second sub-partition of the 16x16 pixel partition has at least one non-zero coefficient (158). When the second sub-partition has all zero coefficients (“NO” branch of 158), entropy coding unit 46 clears the second bit of luma16x8_CBP, e.g., assigns luma16x8_CBP[1] a value of “0” (160). When the second sub-block has at least one non-zero coefficient (“YES” branch of 158), entropy coding unit 46 then sets the second bit of luma16x8_CBP, e.g., assigns luma16x8_CBP[1] a value of “1” (162).

[00140] The following pseudocode provides one example implementation of the methods described with respect to FIGS. 8 and 9:

```

    if (motion partition bigger than 8x8)
    {

```

```

lumacbp16
if (lumacbp16 != 0 )
{
    transform_size_flag
    if (transform_size_flag == TRANSFORM_SIZE_GREATER_THAN_16x8)
    {
        if ((mb16_type == P_16x8) OR (mb16_type == P_8x16))
        {
            luma16x8_cbp
            chroma_cbp
        }
        else
            chroma_cbp
    }
    else
        h264_cbp
}
else
    chroma_cbp
}
else
    h264_cbp

```

[00141] In the pseudocode, “lumacbp16” corresponds to an operation of appending a one-bit flag indicating whether an entire 16x16 luma block has nonzero coefficients or not. When “lumacbp16” equals one, there is at least one nonzero coefficient. The function “Transform_size_flag” refers to a calculation performed having a result that indicates the transform being used, e.g., one of a 4x4 transform, 8x8 transform, 16x16 transform (for motion partition equal to or bigger than 16x16), 16x8 transform (for P_16x8), or 8x16 transform (for P_8x16). TRANSFORM_SIZE_GREATER_THAN_16x8 is an enumerated value (e.g., “2”) that is used to indicate that a transform size is greater than or equal to 16x8

or 8x16. The result of the transform_size_flag is incorporated into the syntax information of the 64x64 pixel macroblock.

[00142] “Luma16x8_cbp” refers to a calculation that produces a two-bit number with each bit indicating whether one of the two partitions of P_16x8 or P_8x16 has nonzero coefficients or not. The two-bit number resulting from luma16x8_cbp is incorporated into the syntax of the 64x64 pixel macroblock. “Chroma_cbp” may be calculated in the same manner as the CodedBlockPatternChroma as prescribed by ITU H.264. The calculated chroma_cbp value is incorporated into the syntax information of the 64x64 pixel macroblock. The function h264_cbp may be calculated in the same way as the CBP defined in ITU H.264. The calculated H264_cbp value is incorporated into the syntax information of the 64x64 pixel macroblock.

[00143] In general, a method according to FIGS. 6–9 may include encoding, with a video encoder, a video block having a size of more than 16x16 pixels, generating block-type syntax information that indicates the size of the block, and generating a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

[00144] FIG. 10 is a block diagram illustrating an example arrangement of a 64x64 pixel macroblock. The macroblock of FIG. 10 comprises four 32x32 partitions, labeled A, B, C, and D in FIG. 10. As discussed with respect to FIG. 4A, in one example, a block may be partitioned in any one of four ways: the entire block (64x64) with no sub-partitions, two equal-sized horizontal partitions (32x64 and 32x64), two equal-sized vertical partitions (64x32 and 64x32), or four equal-sized square partitions (32x32, 32x32, 32x32 and 32x32).

[00145] In the example of FIG. 10, the whole block partition comprises each of blocks A, B, C, and D; a first one of the two equal-sized horizontal partitions comprises A and B, while a second one of the two equal-sized horizontal partitions comprises C and D; a first one of the two equal-sized vertical partitions comprises A and C, while a second one of the two equal-sized vertical partitions comprises B and D; and the four equal-sized square partitions correspond to one of each of A, B, C, and D. Similar partition schemes can be used for any size block, e.g., larger than 64x64 pixels, 32x32 pixels, 16x16 pixels, 8x8 pixels, or other sizes of video blocks.

[00146] When a video block is intra-coded, various methods may be used for partitioning the video block. Moreover, each of the partitions may be intra-coded differently, i.e., with a different mode, such as different intra-modes. For example, a 32x32 partition, such as partition A of FIG. 10, may be further partitioned into four equal-sized blocks of size 16x16 pixels. As one example, ITU H.264 describes three different methods for intra-encoding a 16x16 macroblock, including intra-coding at the 16x16 level, intra-coding at the 8x8 level, and intra-coding at the 4x4 level. However, ITU H.264 prescribes encoding each partition of a 16x16 macroblock using the same intra-coding mode. Therefore, according to ITU H.264, if one sub-block of a 16x16 macroblock is to be intra-coded at the 4x4 level, every sub-block of the 16x16 macroblock must be intra-coded at the 4x4 level.

[00147] An encoder configured according to the techniques of this disclosure, on the other hand, may apply a mixed mode approach. For intra-coding, for example, a large macroblock may have various partitions encoded with different coding modes. As an illustration, in a 32x32 partition, one 16x16 partition may be intra-coded at the 4x4 pixel level, while other 16x16 partitions may be intra-coded at the 8x8 pixel level, and one 16x16 partition may be intra-coded at the 16x16 level, e.g., as shown in FIG. 4B.

[00148] When a video block is to be partitioned into four equal-sized sub-blocks for intra-coding, the first block to be intra-coded may be the upper-left block, followed by the block immediately to the right of the first block, followed by the block immediately beneath the first block, and finally followed by the block beneath and to the right of the first block. With reference to the example block of FIG. 10, the order of intra-coding would proceed from A to B to C and finally to D. Although FIG. 10 depicts a 64x64 pixel macroblock, intra-coding of a partitioned block of a different size may follow this same ordering.

[00149] When a video block is to be inter-coded as part of a P-frame or P-slice, the block may be partitioned into any of the four above-described partitions, each of which may be separately encoded. That is, each partition of the block may be encoded according to a different encoding mode, either intra-encoded (I-coded) or inter-encoded with reference to a single reference frame/slice/list (P-coded). Table 1, below, summarizes inter-encoding information for each potential partition of a block of size NxN. Where Table 1 refers to

“M,” $M = N/2$. In Table 1 below, L0 refers to “list 0,” i.e., the reference frame/slice/list. When deciding how to best partition the NxN block, an encoder, such as video encoder 50, may analyze rate-distortion cost information for each MB_N_type (i.e., each type of partition) based on a Lagrange multiplier, as discussed in greater detail with respect to FIG. 11, selecting the lowest cost as the best partition method.

TABLE 1

MB_N_type	Name of MB_N_type	# of parts	Prediction Mode part 1	Prediction Mode part 2	Part width	Part height
0	P_L0_NxN	1	Pred_L0	N/A	N	N
1	P_L0_L0_NxM	2	Pred_L0	Pred_L0	N	M
2	P_L0_L0_MxN	2	Pred_L0	Pred_L0	M	N
3	PN_MxM	4	N/A	N/A	M	M
inferred	PN_Skip	1	Pred_L0	N/A	N	N

[00150] In Table 1 above, elements of the column “MB_N_type” are keys for each type of partition of an NxN block. Elements of the column “Name of MB_N_type” are names of different partitioning types of an NxN block. “P” in the name refers to the block being inter-coded using P-coding, i.e., with reference to a single frame/slice/list. “L0” in the name refers to the reference frame/slice/list, e.g., “list 0,” used as reference frames or slices for P coding. “NxN” refers to the partition being the whole block, “NxM” refers to the partition being two partitions of width N and height M, “MxN” refers to the partition being two partitions of width M and height N, “MxM” refers to the partition being four equal-sized partitions each with width M and height M.

[00151] In Table 1, PN_Skip implies that the block was “skipped,” e.g., because the block resulting from coding had all zero coefficients. Elements of the column “Prediction Mode part 1” refer to the reference frame/slice/list for sub-partition 1 of the partition, while elements of the column “Prediction Mode part 2” refer to the reference frame/slice/list for sub-partition 2 of the partition. Because P_L0_NxN has only a single partition, the corresponding element of “Prediction Mode part 2” is “N/A,” as there is no second sub-partition. For PN_MxM, there exist four partition blocks that may be separately encoded.

Therefore, both prediction mode columns for PN_MxM include “N/A.” PN_Skip, as with P_L0_NxN, has only a single part, so the corresponding element of column “Prediction Mode part 2” is “N/A.”

[00152] Table 2, below, includes similar columns and elements to those of Table 1. However, Table 2 corresponds to various encoding modes for an inter-coded block using bi-directional prediction (B-encoded). Therefore, each partition may be encoded by either or both of a first frame/slice/list (L0) and a second frame/slice/list (L1). “BiPred” refers to the corresponding partition being predicted from both L0 and L1. In Table 2, column labels and values are similar in meaning to those used in Table 1.

TABLE 2

MB_N_type	Name of MB_N_type	# of parts	Prediction Mode part 1	Prediction Mode part 2	Part width	Part height
0	B_Direct_NxN	Na	Direct	na	N	N
1	B_L0_NxN	1	Pred_L0	na	N	N
2	B_L1_NxN	1	Pred_L1	na	N	N
3	B_Bi_NxN	1	BiPred	na	N	N
4	B_L0_L0_NxM	2	Pred_L0	Pred_L0	N	M
5	B_L0_L0_MxN	2	Pred_L0	Pred_L0	M	N
6	B_L1_L1_NxM	2	Pred_L1	Pred_L1	N	M
7	B_L1_L1_MxN	2	Pred_L1	Pred_L1	M	N
8	B_L0_L1_NxM	2	Pred_L0	Pred_L1	N	M
9	B_L0_L1_MxN	2	Pred_L0	Pred_L1	M	N
10	B_L1_L0_NxM	2	Pred_L1	Pred_L0	N	M
11	B_L1_L0_MxN	2	Pred_L1	Pred_L0	M	N
12	B_L0_Bi_NxM	2	Pred_L0	BiPred	N	M
13	B_L0_Bi_MxN	2	Pred_L0	BiPred	M	N
14	B_L1_Bi_NxM	2	Pred_L1	BiPred	N	M
15	B_L1_Bi_MxN	2	Pred_L1	BiPred	M	N
16	B_Bi_L0_NxM	2	BiPred	Pred_L0	N	M
17	B_Bi_L0_MxN	2	BiPred	Pred_L0	M	N
18	B_Bi_L1_NxM	2	BiPred	Pred_L1	N	M

MB_ N_type	Name of MB_N_type	# of parts	Prediction Mode part 1	Prediction Mode part 2	Part width	Part height
19	B_Bi_L1_MxN	2	BiPred	Pred_L1	M	N
20	B_Bi_Bi_NxM	2	BiPred	BiPred	N	M
21	B_Bi_Bi_MxN	2	BiPred	BiPred	M	N
22	BN_MxM	4	na	na	M	M
inferred	BN_Skip	Na	Direct	na	M	M

[00153] FIG. 11 is a flowchart illustrating an example method for calculating optimal partitioning and encoding methods for an NxN pixel video block. In general, the method of FIG. 11 comprises calculating the cost for each different encoding method (e.g., various spatial or temporal modes) as applied to each different partitioning method shown in, e.g., FIG. 4A, and selecting the combination of encoding mode and partitioning method with the best rate-distortion cost for the NxN pixel video block. Cost can be generally calculated using a Lagrange multiplier with rate and distortion values, such that the rate-distortion cost = distortion + λ * rate, where distortion represents error between an original block and a coded block and rate represents the bit rate necessary to support the coding mode. In some cases, rate and distortion may be determined on a macroblock, partition, slice or frame level.

[00154] Initially, video encoder 50 receives an NxN video block to be encoded (170). For example, video encoder 50 may receive a 64x64 large macroblock or a partition thereof, such as, for example, a 32x32 or 16x16 partition, for which video encoder 50 is to select an encoding and partitioning method. Video encoder 50 then calculates the cost to encode the NxN block (172) using a variety of different coding modes, such as different intra- and inter-coding modes. To calculate the cost to spatially encode the NxN block, video encoder 50 may calculate the distortion and the bitrate needed to encode the NxN block with a given coding mode, and then calculate cost = distortion_(Mode, NxN) + λ * rate_(Mode, NxN). Video encoder 50 may encode the macroblock using the specified coding technique and determine the resulting bit rate cost and distortion. The distortion may be determined based on a pixel difference between the coded macroblock and the original macroblock, e.g., based on a sum of absolute difference (SAD) metric, sum of square difference (SSD) metric, or other pixel difference metric.

[00155] Video encoder 50 may then partition the $N \times N$ block into two equally-sized non-overlapping horizontal $N \times (N/2)$ partitions. Video encoder 50 may calculate the cost to encode each of the partitions using various coding modes (176). For example, to calculate the cost to encode the first $N \times (N/2)$ partition, video encoder 50 may calculate the distortion and the bitrate to encode the first $N \times (N/2)$ partition, and then calculate $\text{cost} = \text{distortion}_{(\text{Mode}, \text{FIRST PARTITION}, N \times (N/2))} + \lambda * \text{rate}_{(\text{Mode}, \text{FIRST PARTITION}, N \times (N/2))}$.

[00156] Video encoder 50 may then partition the $N \times N$ block into two equally-sized non-overlapping vertical $(N/2) \times N$ partitions. Video encoder 50 may calculate the cost to encode each of the partitions using various coding modes (178). For example, to calculate the cost to encode the first one of the $(N/2) \times N$ partitions, video encoder 50 may calculate the distortion and the bitrate to encode the first $(N/2) \times N$ partition, and then calculate $\text{cost} = \text{distortion}_{(\text{Mode}, \text{FIRST PARTITION}, (N/2) \times N)} + \lambda * \text{rate}_{(\text{Mode}, \text{FIRST PARTITION}, (N/2) \times N)}$. Video encoder 50 may perform a similar calculation for the cost to encode the second one of the $(N/2) \times N$ macroblock partitions.

[00157] Video encoder 50 may then partition the $N \times N$ block into four equally-sized non-overlapping $(N/2) \times (N/2)$ partitions. Video encoder 50 may calculate the cost to encode the partitions using various coding modes (180). To calculate the cost to encode the $(N/2) \times (N/2)$ partitions, video encoder 50 may first calculate the distortion and the bitrate to encode the upper-left $(N/2) \times (N/2)$ partition and find the cost thereof as $\text{cost}_{(\text{Mode}, \text{UPPER-LEFT}, (N/2) \times (N/2))} = \text{distortion}_{(\text{Mode}, \text{UPPER-LEFT}, (N/2) \times (N/2))} + \lambda * \text{rate}_{(\text{Mode}, \text{UPPER-LEFT}, (N/2) \times (N/2))}$. Video encoder 50 may similarly calculate the cost of each $(N/2) \times (N/2)$ block in the order: (1) upper-left partition, (2) upper-right partition, (3) bottom-left partition, (4) bottom-right partition. Video encoder 50 may, in some examples, make recursive calls to this method on one or more of the $(N/2) \times (N/2)$ to calculate the cost of partitioning and separately encoding each of the $(N/2) \times (N/2)$ partitions further, e.g., as $(N/2) \times (N/4)$ partitions, $(N/4) \times (N/2)$ partitions, and $(N/4) \times (N/4)$ partitions.

[00158] Next, video encoder 50 may determine which combination of partitioning and encoding mode produced the best, i.e., lowest, cost in terms of rate and distortion (182). For example, video encoder 50 may compare the best cost of encoding two adjacent $(N/2) \times (N/2)$ partitions to the best cost of encoding the $N \times (N/2)$ partition comprising the two adjacent $(N/2) \times (N/2)$ partitions. When the aggregate cost of encoding the two adjacent

$(N/2) \times (N/2)$ partitions exceeds the cost to encode the $N \times (N/2)$ partition comprising them, video encoder 50 may select the lower-cost option of encoding the $N \times (N/2)$ partition. In general, video encoder 50 may apply every combination of partitioning method and encoding mode for each partition to identify a lowest cost partitioning and encoding method. In some cases, video encoder 50 may be configured to evaluate a more limited set of partitioning and encoding mode combinations.

[00159] Upon determining the best, e.g., lowest cost, partitioning and encoding methods, video encoder 50 may encode the $N \times N$ macroblock using the best-cost determined method (184). In some cases, the result may be a large macroblock having partitions that are coded using different coding modes. The ability to apply mixed mode coding to a large macroblock, such that different coding modes are applied to different partitions in the large macroblock, may permit the macroblock to be coded with reduced cost.

[00160] In some examples, method for coding with mixed modes may include receiving, with video encoder 50, a video block having a size of more than 16×16 pixels, partitioning the block into partitions, encoding one of the partitions with a first encoding mode, encoding another of the partitions with a second coding mode different from the first encoding mode, and generating block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.

[00161] FIG. 12 is a block diagram illustrating an example 64×64 pixel large macroblock with various partitions and different selected encoding methods for each partition. In the example of FIG. 12, each partition is labeled with one of an "I," "P," or "B." Partitions labeled "I" are partitions for which an encoder has elected to utilize intra-coding, e.g., based on rate-distortion evaluation. Partitions labeled "P" are partitions for which the encoder has elected to utilize single-reference inter-coding, e.g., based on rate-distortion evaluation. Partitions labeled "B" are partitions for which the encoder has elected to utilize bi-predicted inter-coding, e.g., based on rate-distortion evaluation. In the example of FIG. 12, different partitions within the same large macroblock have different coding modes, including different partition or sub-partition sizes and different intra- or inter-coding modes.

[00162] The large macroblock is a macroblock identified by a macroblock syntax element that identifies the macroblock type, e.g., `mb64_type` or `mb32_type`, for a given coding standard such as an extension of the H.264 coding standard. The macroblock type syntax element may be provided as a macroblock header syntax element in the encoded video bitstream. The I-, P- and B-coded partitions illustrated in FIG. 12 may be coded according to different coding modes, e.g., intra- or inter-prediction modes with various block sizes, including large block size modes for large partitions greater than 16x16 in size or H.264 modes for partitions that are less than or equal to 16x16 in size.

[00163] In one example, an encoder, such as video encoder 50, may use the example method described with respect to FIG. 11 to select various encoding modes and partition sizes for different partitions and sub-partitions of the example large macroblock of FIG. 12. For example, video encoder 50 may receive a 64x64 macroblock, execute the method of FIG. 11, and produce the example macroblock of FIG. 12 with various partition sizes and coding modes as a result. It should be understood, however, that selections for partitioning and encoding modes may result from application of the method of FIG. 11, e.g., based on the type of frame from which the macroblock was selected and based on the input macroblock upon which the method is executed. For example, when the frame comprises an I-frame, each partition will be intra-encoded. As another example, when the frame comprises a P-frame, each partition may either be intra-encoded or inter-coded based on a single reference frame (i.e., without bi-prediction).

[00164] The example macroblock of FIG. 12 is assumed to have been selected from a bi-predicted frame (B-frame) for purposes of illustration. In other examples, where a macroblock is selected from a P-frame, video encoder 50 would not encode a partition using bi-directional prediction. Likewise, where a macroblock is selected from an I-frame, video encoder 50 would not encode a partition using inter-coding, either P-encoding or B-encoding. However, in any case, video encoder 50 may select various partition sizes for different portions of the macroblock and elect to encode each partition using any available encoding mode.

[00165] In the example of FIG. 12, it is assumed that a combination of partition and mode selection based on rate-distortion analysis has resulted in one 32x32 B-coded partition, one 32x32 P-coded partition, one 16x32 I-coded partition, one 32x16 B-coded

partition, one 16x16 P-coded partition, one 16x8 P-coded partition, one 8x16 P-coded partition, one 8x8 P-coded partition, one 8x8 B-coded partition, one 8x8 I-coded partition, and numerous smaller sub-partitions having various coding modes. The example of FIG. 12 is provided for purposes of conceptual illustration of mixed mode coding of partitions in a large macroblock, and should not necessarily be considered representative of actual coding results for a particular large 64x64 macroblock.

[00166] FIG. 13 is a flowchart illustrating an example method for determining an optimal size of a macroblock for encoding a frame or slice of a video sequence. Although described with respect to selecting an optimal size of a macroblock for a frame, a method similar to that described with respect to FIG. 13 may be used to select an optimal size of a macroblock for a slice. Likewise, although the method of FIG. 13 is described with respect to video encoder 50, it should be understood that any encoder may utilize the example method of FIG. 13 to determine an optimal (e.g., least cost) size of a macroblock for encoding a frame of a video sequence. In general, the method of FIG. 13 comprises performing an encoding pass three times, once for each of a 16x16 macroblock, a 32x32 macroblock, and a 64x64 macroblock, and a video encoder may calculate rate-distortion metrics for each pass to determine which macroblock size provides the best rate-distortion.

[00167] Video encoder 50 may first encode a frame using 16x16 pixel macroblocks during a first encoding pass (190), e.g., using a function encode (frame, MB16_type), to produce an encoded frame F_{16} . After the first encoding pass, video encoder 50 may calculate the bit rate and distortion based on the use of 16x16 pixel macroblocks as R_{16} and D_{16} , respectively (192). Video encoder 50 may then calculate a rate-distortion metric in the form of the cost of using 16x16 pixel macroblocks C_{16} using the Lagrange multiplier $C_{16} = D_{16} + \lambda * R_{16}$ (194). Coding modes and partition sizes may be selected for the 16x16 pixel macroblocks, for example, according to the H.264 standard.

[00168] Video encoder 50 may then encode the frame using 32x32 pixel macroblocks during a second encoding pass (196), e.g., using a function encode (frame, MB32_type), to produce an encoded frame F_{32} . After the second encoding pass, video encoder 50 may calculate the bit rate and distortion based on the use of 32x32 pixel macroblocks as R_{32} and D_{32} , respectively (198). Video encoder 50 may then calculate a rate-distortion metric in the form the cost of using 32x32 pixel macroblocks C_{32} using the

Lagrange multiplier $C_{32} = D_{32} + \lambda * R_{32}$ (200). Coding modes and partition sizes may be selected for the 32x32 pixel macroblocks, for example, using rate and distortion evaluation techniques as described with reference to FIGS. 11 and 12.

[00169] Video encoder 50 may then encode the frame using 64x64 pixel macroblocks during a third encoding pass (202), e.g., using a function `encode(frame, MB64_type)`, to produce an encoded frame F_{64} . After the third encoding pass, video encoder 50 may calculate the bit rate and distortion based on the use of 64x64 pixel macroblocks as R_{64} and D_{64} , respectively (204). Video encoder 50 may then calculate a rate-distortion metric in the form the cost of using 64x64 pixel macroblocks C_{64} using the Lagrange multiplier $C_{64} = D_{64} + \lambda * R_{64}$ (206). Coding modes and partition sizes may be selected for the 64x64 pixel macroblocks, for example, using rate and distortion evaluation techniques as described with reference to FIGS. 11 and 12.

[00170] Next, video encoder 50 may determine which of the metrics C_{16} , C_{32} , and C_{64} is lowest for the frame (208). Video encoder 50 may elect to use the frame encoded with the macroblock size that resulted in the lowest cost (210). Thus, for example, when C_{16} is lowest, video encoder 50 may forward frame F_{16} , encoded with the 16x16 macroblocks as the encoded frame in a bitstream for storage or transmission to a decoder. When C_{32} is lowest, video encoder 50 may forward F_{32} , encoded with the 32x32 macroblocks. When C_{64} is lowest, video encoder 50 may forward F_{64} , encoded with the 64x64 macroblocks.

[00171] In other examples, video encoder 50 may perform the encoding passes in any order. For example, video encoder 50 may begin with the 64x64 macroblock encoding pass, perform the 32x32 macroblock encoding pass second, and end with the 16x16 macroblock encoding pass. Also, similar methods may be used for encoding other coded units comprising a plurality of macroblocks, such as slices with different sizes of macroblocks. For example, video encoder 50 may apply a method similar to that of FIG. 13 for selecting an optimal macroblock size for encoding slices of a frame, rather than the entire frame.

[00172] Video encoder 50 may also transmit an identifier of the size of the macroblocks for a particular coded unit (e.g., a frame or a slice) in the header of the coded unit. In accordance with the method of FIG. 13, a method may include receiving, with a

digital video encoder, a coded unit of a digital video stream, calculating a first rate-distortion metric corresponding to a rate-distortion for encoding the coded unit using a first plurality of blocks each comprising 16x16 pixels, calculating a second rate-distortion metric corresponding to a rate-distortion for encoding the coded unit using a second plurality of blocks each comprising greater than 16x16 pixels, and determining which of the first rate-distortion metric and the second rate-distortion metric is lowest for the coded unit. The method may further include, when the first rate-distortion metric is determined to be lowest, encoding the coded unit using the first plurality of blocks, and when the second rate-distortion metric is determined to be lowest, encoding the coded unit using the second plurality of blocks.

[00173] FIG. 14 is a block diagram illustrating an example wireless communication device 230 including a video encoder/decoder CODEC 234 that may encode and/or decode digital video data using the larger-than-standard macroblocks, using any of a variety of the techniques described in this disclosure. In the example of FIG. 14, wireless communication device 230 includes video camera 232, video encoder-decoder (CODEC) 234, modulator/demodulator (modem) 236, transceiver 238, processor 240, user interface 242, memory 244, data storage device 246, antenna 248, and bus 250.

[00174] The components included in wireless communication device 230 illustrated in FIG. 14 may be realized by any suitable combination of hardware, software and/or firmware. In the illustrated example, the components are depicted as separate units. However, in other examples, the various components may be integrated into combined units within common hardware and/or software. As one example, memory 244 may store instructions executable by processor 240 corresponding to various functions of video CODEC 234. As another example, video camera 232 may include a video CODEC that performs the functions of video CODEC 234, e.g., encoding and/or decoding video data.

[00175] In one example, video camera 232 may correspond to video source 18 (FIG. 1). In general, video camera 232 may record video data captured by an array of sensors to generate digital video data. Video camera 232 may send raw, recorded digital video data to video CODEC 234 for encoding and then to data storage device 246 via bus 250 for data storage. Processor 240 may send signals to video camera 232 via bus 250 regarding a mode in which to record video, a frame rate at which to record video, a time at which to

end recording or to change frame rate modes, a time at which to send video data to video CODEC 234, or signals indicating other modes or parameters.

[00176] User interface 242 may comprise one or more interfaces, such as input and output interfaces. For example, user interface 242 may include a touch screen, a keypad, buttons, a screen that may act as a viewfinder, a microphone, a speaker, or other interfaces. As video camera 232 receives video data, processor 240 may signal video camera 232 to send the video data to user interface 242 to be displayed on the viewfinder.

[00177] Video CODEC 234 may encode video data from video camera 232 and decode video data received via antenna 248, transceiver 238, and modem 236. Video CODEC 234 additionally or alternatively may decode previously encoded data received from data storage device 246 for playback. Video CODEC 234 may encode and/or decode digital video data using macroblocks that are larger than the size of macroblocks prescribed by conventional video encoding standards. For example, video CODEC 234 may encode and/or decode digital video data using a large macroblock comprising 64x64 pixels or 32x32 pixels. The large macroblock may be identified with a macroblock type syntax element according to a video standard, such as an extension of the H.264 standard.

[00178] Video CODEC 234 may perform the functions of either or both of video encoder 50 (FIG. 2) and/or video decoder 60 (FIG. 3), as well as any other encoding/decoding functions or techniques as described in this disclosure. For example, CODEC 234 may partition a large macroblock into a variety of differently sized, smaller partitions, and use different coding modes, e.g., spatial (I) or temporal (P or B), for selected partitions. Selection of partition sizes and coding modes may be based on rate-distortion results for such partition sizes and coding modes. CODEC 234 also may utilize hierarchical coded block pattern (CBP) values to identify coded macroblocks and partitions having non-zero coefficients within a large macroblock. In addition, in some examples, CODEC 234 may compare rate-distortion metrics for large and small macroblocks to select a macroblock size producing more favorable results for a frame, slice or other coding unit.

[00179] A user may interact with user interface 242 to transmit a recorded video sequence in data storage device 246 to another device, such as another wireless communication device, via modem 236, transceiver 238, and antenna 248. The video sequence may be encoded according to an encoding standard, such as MPEG-2, MPEG-3,

MPEG-4, H.263, H.264, or other video encoding standards, subject to extensions or modifications described in this disclosure. For example, the video sequence may also be encoded using larger-than-standard macroblocks, as described in this disclosure. Wireless communication device 230 may also receive an encoded video segment and store the received video sequence in data storage device 246.

[00180] Macroblocks of the received, encoded video sequence may be larger than macroblocks specified by conventional video encoding standards. To display an encoded video segment in data storage device 246, such as a recorded video sequence or a received video segment, video CODEC 234 may decode the video sequence and send decoded frames of the video segment to user interface 242. When a video sequence includes audio data, video CODEC 234 may decode the audio, or wireless communication device 230 may further include an audio codec (not shown) to decode the audio. In this manner, video CODEC 234 may perform both the functions of an encoder and of a decoder.

[00181] Memory 244 of wireless communication device 230 of FIG. 14 may be encoded with computer-readable instructions that cause processor 240 and/or video CODEC 234 to perform various tasks, in addition to storing encoded video data. Such instructions may be loaded into memory 244 from a data storage device such as data storage device 246. For example, the instructions may cause processor 240 to perform the functions described with respect to video CODEC 234.

[00182] FIG. 15 is a block diagram illustrating an example hierarchical coded block pattern (CBP) 260. The example of CBP 260 generally corresponds to a portion of the syntax information for a 64x64 pixel macroblock. In the example of FIG. 15, CBP 260 comprises a CBP64 value 262, four CBP32 values 264, 266, 268, 270, and four CBP16 values 272, 274, 276, 278. Each block of CBP 260 may include one or more bits. In one example, when CBP64 value 262 is a bit with a value of "1," indicating that there is at least one non-zero coefficient in the large macroblock, CBP 260 includes the four CBP32 values 264, 266, 268, 270 for four 32x32 partitions of the large 64x64 macroblock, as shown in the example of FIG. 15.

[00183] In another example, when CBP64 value 262 is a bit with a value of "0," CBP 260 may consist only of CBP64, as a value of "0" may indicate that the block corresponding to CBP 260 has all zero-valued coefficients. Hence, all partitions of that

block likewise will contain all zero-valued coefficients. In one example, when a CBP64 is a bit with a value of “1,” and one of the CBP32 values for a particular 32x32 partition is a bit with a value of “1,” the CBP32 value for the 32x32 partition has four branches, representative of CBP16 values, e.g., as shown with respect to CBP32 value 266. In one example, when a CBP32 value is a bit with a value of “0,” the CBP32 does not have any branches. In the example of FIG. 15, CBP 260 may have a five-bit prefix of “10100,” indicating that the CBP64 value is “1,” and that one of the 32x32 partitions has a CBP32 value of “1,” with subsequent bits corresponding to the four CBP16 values 272, 274, 276, 278 corresponding to 16x16 partitions of the 32x32 partition with the CBP 32 value of “1.” Although only a single CBP32 value is shown as having a value of “1” in the example of FIG. 15, in other examples, two, three or all four 32x32 partitions may have CBP32 values of “1,” in which case multiple instances of four 16x16 partitions with corresponding CBP16 values would be required.

[00184] In the example of FIG. 15, the four CBP16 values 272, 274, 276, 278 for the four 16x16 partitions may be calculated according to various methods, e.g., according to the methods of FIGS. 8 and 9. Any or all of CBP16 values 272, 274, 276, 278 may include a “luma_cbp16” value, a transform_size_flag, and/or a luma16x8_cbp. CBP16 values 272, 274, 276, 278 may also be calculated according to a CBP value as defined in ITU H.264 or as a CodedBlockPatternChroma in ITU H.264, as discussed with respect to FIGS. 8 and 9. In the example of FIG. 15, assuming that CBP16 278 has a value of “1,” and the other CBP 16 values 272, 274, 276 have values of “0,” the nine-bit CBP value for the 64x64 macroblock would be “101000001,” where each bit corresponds to one of the partitions at a respective level in the CBP/partition hierarchy.

[00185] FIG. 16 is a block diagram illustrating an example tree structure 280 corresponding to CBP 260 (FIG. 15). CBP64 node 282 corresponds to CBP64 value 262, CBP32 nodes 284, 286, 288, 290 each correspond to respective ones of CBP32 values 264, 266, 268, 270, and CBP16 nodes 292, 294, 296, 298 each correspond to respective ones of CBP16 values 272, 274, 276, 278. In this manner, a coded block pattern value as defined in this disclosure may correspond to a hierarchical CBP. Each node yielding another branch in the tree corresponds to a respective CBP value of “1.” In the examples of FIGS. 15 and 16, CBP64 282 and CBP32 286 both have values of “1,” and yield further partitions with

possible CBP values of “1,” i.e., where at least one partition at the next partition level includes at least one non-zero transform coefficient value.

[00186] The techniques described in this disclosure may be implemented, at least in part, in hardware, software, firmware or any combination thereof. For example, various aspects of the described techniques may be implemented within one or more processors, including one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), or any other equivalent integrated or discrete logic circuitry, as well as any combinations of such components. The term “processor” or “processing circuitry” may generally refer to any of the foregoing logic circuitry, alone or in combination with other logic circuitry, or any other equivalent circuitry. A control unit comprising hardware may also perform one or more of the techniques of this disclosure.

[00187] Such hardware, software, and firmware may be implemented within the same device or within separate devices to support the various operations and functions described in this disclosure. In addition, any of the described units, modules or components may be implemented together or separately as discrete but interoperable logic devices. Depiction of different features as modules or units is intended to highlight different functional aspects and does not necessarily imply that such modules or units must be realized by separate hardware or software components. Rather, functionality associated with one or more modules or units may be performed by separate hardware or software components, or integrated within common or separate hardware or software components.

[00188] The techniques described in this disclosure may also be embodied or encoded in a computer-readable medium, such as a computer-readable storage medium, containing instructions. Instructions embedded or encoded in a computer-readable medium may cause a programmable processor, or other processor, to perform the method, e.g., when the instructions are executed. Computer readable storage media may include random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM), flash memory, a hard disk, a CD-ROM, a floppy disk, a cassette, magnetic media, optical media, or other computer readable media.

[00189] Various examples have been described. These and other examples are within the scope of the following claims.

CLAIMS: SET A

1. A method comprising:
 - encoding, with a video encoder, a video block having a size of more than 16x16 pixels;
 - generating block-type syntax information that indicates the size of the block; and
 - generating a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.
2. The method of claim 1, further comprising:
 - setting the coded block pattern value to zero when the encoded block does not include at least one non-zero coefficient;
 - setting the coded block pattern value to one when the encoded block includes at least one non-zero coefficient; and
 - generating an encoded video bitstream comprising encoded video data for the encoded block, the block-type syntax information, and the coded block pattern value when the coded block pattern value is one.
3. The method of claim 1, further comprising:
 - when the encoded block does not include at least one non-zero coefficient, generating a single bit for the coded block pattern value and setting the value of the single bit to zero; and
 - when the encoded block includes at least one non-zero coefficient, generating a first bit of the coded block pattern value and setting the first bit to one, generating four partition bits for the coded block pattern value, each of the four partition bits corresponding to a different one of four equally-sized partitions of the encoded block, and setting the four partition bits to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient.

4. The method of claim 3, further comprising, when the encoded block includes at least one non-zero coefficient, identifying a 16x16 pixel partition of the encoded block, generating a partition coded block pattern value for the 16x16 pixel partition as a portion of the generated coded block pattern value and appending the partition coded block pattern to the coded block pattern value following the four partition bits.
5. The method of claim 4, further comprising, when the encoded block includes at least one non-zero coefficient, generating a coded block pattern value comprises generating a first luma_16x8 bit and a second luma_16x8 bit, the first luma_16x8 bit representing whether a 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient and the second luma_16x8 bit representing whether a second 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient.
6. The method of claim 4, wherein the video block has a size of at least 64x64 pixels.
7. The method of claim 1, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

8. An apparatus comprising a video encoder configured to encode a video block having a size of more than 16x16 pixels, generate block-type syntax information that indicates the size of the block, and generate a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

9. The apparatus of claim 8, wherein video encoder is configured to set the coded block pattern value to zero when the encoded block does not include at least one non-zero coefficient, set the coded block pattern value to one when the encoded block includes at least one non-zero coefficient, and generate an encoded video bitstream comprising encoded video data for the encoded block, the block-type syntax information, and the coded block pattern value when the coded block pattern value is one.

10. The apparatus of claim 8, wherein the video encoder is configured to:
when the encoded block does not include at least one non-zero coefficient, generate a single bit for the coded block pattern value and setting the value of the single bit to zero;
and
when the encoded block includes at least one non-zero coefficient, generate a first bit of the coded block pattern value and setting the first bit to one, generate four partition bits for the coded block pattern value, each of the four partition bits corresponding to a different one of four equally-sized partitions of the encoded block, and set the four partition bits to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient.

11. The apparatus of claim 10 wherein the video encoder is configured to, when the encoded block includes at least one non-zero coefficient, identify a 16x16 pixel partition of the encoded block, generate a partition coded block pattern value for the 16x16 pixel partition as a portion of the generated coded block pattern value, and append the partition coded block pattern value to the coded block pattern value following the four partition bits.

12. The apparatus of claim 11, wherein the video is configured to, when the encoded block includes at least one non-zero coefficient, generate a coded block pattern value comprises generating a first luma_16x8 bit and a second luma_16x8 bit, the first luma_16x8 bit representing whether a 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient and the second luma_16x8 bit representing whether a second 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient.
13. The apparatus of claim 8, wherein the video block has a size of at least 64x64 pixels.
14. The apparatus of claim 8, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.
15. The apparatus of claim 8, wherein the apparatus comprises a wireless communication device.
16. An apparatus comprising:
means for encoding, with a video encoder, a video block having a size of more than 16x16 pixels;
means for generating block-type syntax information that indicates the size of the block; and

means for generating a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

17. The apparatus of claim 16, further comprising:

means for setting the coded block pattern value to zero when the encoded block does not include at least one non-zero coefficient;

means for setting the coded block pattern value to one when the encoded block includes at least one non-zero coefficient; and

means for generating an encoded video bitstream comprising encoded video data for the encoded block, the block-type syntax information, and the coded block pattern value when the coded block pattern value is one.

18. The apparatus of claim 16, further comprising:

when the encoded block does not include at least one non-zero coefficient, means for generating a single bit for the coded block pattern value and setting the value of the single bit to zero; and

when the encoded block includes at least one non-zero coefficient, means for generating a first bit of the coded block pattern value and setting the first bit to one, means for generating four partition bits for the coded block pattern value, each of the four partition bits corresponding to a different one of four equally-sized partitions of the encoded block, and means for setting the four partition bits to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient.

19. The apparatus of claim 18, further comprising, when the encoded block includes at least one non-zero coefficient, means for identifying a 16x16 pixel partition of the encoded block, means for generating a partition coded block pattern value for the 16x16 pixel partition as a portion of the generated coded block pattern value and means for appending the partition coded block pattern value to the coded block pattern value following the four partition bits.

20. The apparatus of claim 19, further comprising, when the encoded block includes at least one non-zero coefficient, means for generating a coded block pattern value comprises generating a first luma_16x8 bit and a second luma_16x8 bit, the first luma_16x8 bit representing whether a 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient and the second luma_16x8 bit representing whether a second 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient.

21. The apparatus of claim 16, wherein the video block has a size of at least 64x64 pixels.

22. The apparatus of claim 16, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

23. A computer-readable medium encoded with instructions to cause a video encoding apparatus to:

 encode, with a video encoder, a video block having a size of more than 16x16 pixels;

 generate block-type syntax information that indicates the size of the block; and

 generate a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient.

24. The computer-readable medium of claim 23, further comprising instructions to cause the video encoding apparatus to:

set the coded block pattern value to zero when the encoded block does not include at least one non-zero coefficient;

set the coded block pattern value to one when the encoded block includes at least one non-zero coefficient; and

generate an encoded video bitstream comprising encoded video data for the encoded block, the block-type syntax information, and the coded block pattern value when the coded block pattern value is one.

25. The computer-readable medium of claim 23, further comprising instructions to cause the video encoding apparatus to:

when the encoded block does not include at least one non-zero coefficient, generate a single bit for the coded block pattern value and setting the value of the single bit to zero; and

when the encoded block includes at least one non-zero coefficient, generate a first bit of the coded block pattern value and setting the first bit to one, generate four partition bits for the coded block pattern value, each of the four partition bits corresponding to a different one of four equally-sized partitions of the encoded block, and set the four partition bits to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient.

26. The computer-readable medium of claim 25, further comprising further comprising instructions to cause the video encoding apparatus to, when the encoded block includes at least one non-zero coefficient, identify a 16x16 pixel partition of the encoded block, generate a partition coded block pattern value for the 16x16 pixel partition as a portion of the generated coded block pattern value and append the partition coded block pattern value to the coded block pattern value following the four partition bits.

27. The computer-readable medium of claim 26, further comprising instructions to cause the video encoding apparatus to, when the encoded block includes at least one non-zero coefficient, generate for the coded block pattern value a first luma_16x8 bit and a second luma_16x8 bit, the first luma_16x8 bit representing whether a 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient and the second luma_16x8 bit representing whether a second 16x8 partition of the 16x16 partition of the block includes at least one non-zero coefficient.

28. The computer-readable medium of claim 23, wherein the video block has a size of at least 64x64 pixels.

29. The computer-readable medium of claim 23, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

30. A method comprising:
receiving, with a video decoder, an encoded video block having a size of more than 16x16 pixels;
receiving block-type syntax information that indicates the size of the encoded block;
receiving a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient; and
decoding the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

31. The method of claim 30, wherein the coded block pattern value is zero when the encoded block does not include at least one non-zero coefficient, and the coded block pattern value is one when the encoded block includes at least one non-zero coefficient, the method further comprising receiving encoded video data for the encoded block when the coded block pattern value is one.

32. The method of claim 30, wherein the coded block pattern value is a single bit equal to zero when the encoded block does not include at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, the coded block pattern includes a first bit equal to one and four partition bits, each of the partition bits corresponding to a different one of four equally-sized partitions of the encoded block, wherein the four partition bits are set to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient, and wherein the coded block pattern value includes, when the encoded block includes at least one non-zero coefficient, a partition coded block pattern value for a 16x16 pixel partition appended to the coded block pattern value following the four partition bits.

33. The method of claim 30, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

34. The method of claim 30, wherein the video block has a size of at least 64x64 pixels.

35. An apparatus comprising a video decoder configured to:
- receive, with a video encoder, an encoded video block having a size of more than 16x16 pixels;
 - receive block-type syntax information that indicates the size of the encoded block;
 - receive a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient; and
 - decode the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.
36. The apparatus of claim 35, wherein the coded block pattern value is zero when the encoded block does not include at least one non-zero coefficient, and the coded block pattern value is one when the encoded block includes at least one non-zero coefficient, the video decoder being configured to receive encoded video data for the encoded block when the coded block pattern value is one.
37. The apparatus of claim 35, wherein the coded block pattern value is a single bit equal to zero when the encoded block does not include at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, the coded block pattern includes a first bit equal to one and four partition bits, each of the partition bits corresponding to a different one of four equally-sized partitions of the encoded block, wherein the four partition bits are set to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient, and wherein the coded block pattern value includes, when the encoded block includes at least one non-zero coefficient, a partition coded block pattern value for a 16x16 pixel partition appended to the coded block pattern value following the four partition bits.

38. The apparatus of claim 35, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

39. The apparatus of claim 35, wherein the video block has a size of at least 64x64 pixels.

40. The apparatus of claim 35, wherein the apparatus comprises a wireless communication device.

41. An apparatus comprising:
means for receiving, with a video encoder, an encoded video block having a size of more than 16x16 pixels;
means for receiving block-type syntax information that indicates the size of the encoded block;
means for receiving a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient; and
means for decoding the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.

42. The apparatus of claim 41, wherein the coded block pattern value is zero when the encoded block does not include at least one non-zero coefficient, and the coded block pattern value is one when the encoded block includes at least one non-zero coefficient, the

apparatus further comprising means for receiving encoded video data for the encoded block when the coded block pattern value is one.

43. The apparatus of claim 41, wherein the coded block pattern value is a single bit equal to zero when the encoded block does not include at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, the coded block pattern includes a first bit equal to one and four partition bits, each of the partition bits corresponding to a different one of four equally-sized partitions of the encoded block, wherein the four partition bits are set to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient, and wherein the coded block pattern value includes, when the encoded block includes at least one non-zero coefficient, a partition coded block pattern value for a 16x16 pixel partition appended to the coded block pattern value following the four partition bits.

44. The apparatus of claim 41, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

45. The apparatus of claim 41, wherein the video block has a size of at least 64x64 pixels.

46. A computer-readable medium comprising instructions to cause a video decoder to:
- receive, with a video encoder, an encoded video block having a size of more than 16x16 pixels;
 - receive block-type syntax information that indicates the size of the encoded block;
 - receive a coded block pattern value for the encoded block, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient; and
 - decode the encoded block based on the block-type syntax information and the coded block pattern value for the encoded block.
47. The computer-readable medium of claim 46, wherein the coded block pattern value is zero when the encoded block does not include at least one non-zero coefficient, and the coded block pattern value is one when the encoded block includes at least one non-zero coefficient, the method further comprising receiving encoded video data for the encoded block when the coded block pattern value is one.
48. The computer-readable medium of claim 46, wherein the coded block pattern value is a single bit equal to zero when the encoded block does not include at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, the coded block pattern includes a first bit equal to one and four partition bits, each of the partition bits corresponding to a different one of four equally-sized partitions of the encoded block, wherein the four partition bits are set to respective values representing whether the corresponding one of the four equally-sized partitions includes at least one non-zero coefficient, and wherein the coded block pattern value includes, when the encoded block includes at least one non-zero coefficient, a partition coded block pattern value for a 16x16 pixel partition appended to the coded block pattern value following the four partition bits.

49. The computer-readable medium of claim 46, wherein the coded block pattern value indicates whether the encoded block includes at least one non-zero coefficient and, when the encoded block includes at least one non-zero coefficient, indicates whether any first-level partitions of the encoded block include at least one non-zero coefficient, and, when at least one of the first-level partitions of the encoded block includes at least one non-zero coefficient, indicates whether any second-level partitions of the first-level partitions include at least one non-zero coefficient, and wherein the coded block pattern value includes a pattern of bits corresponding to the encoded block, the first-level partitions of the encoded block, and the second-level partitions of the first-level partitions.

50. The computer-readable medium of claim 46, wherein the video block has a size of at least 64x64 pixels.

CLAIMS: SET B

1. A method comprising:
receiving, with a video encoder, a video block having a size of more than 16x16 pixels;
partitioning the block into partitions;
encoding one of the partitions using a first encoding mode;
encoding another of the partitions using a second encoding mode different from the first encoding mode; and
generating block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.
2. The method of claim 1, further comprising generating an encoded video bitstream comprising coded data for the block and the block-type syntax information.
3. The method of claim 1, further comprising selecting the encoding modes used to encode the partitions based on an encoding cost analysis.
4. The method of claim 3, further comprising selecting the partitioning of the block based on the encoding cost analysis, wherein the encoding cost analysis includes an analysis of bit rate and distortion for encoding modes and the partitions.
5. The method of claim 1, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
6. The method of claim 1, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.

7. The method of claim 1, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
8. The method of claim 1, wherein the video block has a size of at least 64x64 pixels.
9. An apparatus comprising a video encoder configured to:
 - receive a video block having a size of more than 16x16 pixels;
 - partition the block into partitions;
 - encode one of the partitions using a first encoding mode;
 - encode another of the partitions using a second encoding mode different from the first encoding mode; and
 - generate block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.
10. The apparatus of claim 9, wherein the video encoder is configured to generate an encoded video bitstream comprising coded data for the block and the block-type syntax information.
11. The apparatus of claim 9, wherein the video encoder is configured to select the encoding modes used to encode the partitions based on an encoding cost analysis.
12. The apparatus of claim 11, wherein the video encoder is configured to select the partitioning of the block based on the encoding cost analysis, wherein the encoding cost analysis includes an analysis of bit rate and distortion for encoding modes and the partitions.
13. The apparatus of claim 9, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.

14. The apparatus of claim 9, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
15. The apparatus of claim 9, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
16. The apparatus of claim 9, wherein the video block has a size of at least 64x64 pixels.
17. The apparatus of claim 9, wherein the apparatus comprises a wireless communication device.
18. An apparatus comprising:
 - means for receiving a video block having a size of more than 16x16 pixels;
 - means for partitioning the block into partitions;
 - means for encoding one of the partitions using a first encoding mode;
 - means for encoding another of the partitions using a second encoding mode different from the first encoding mode; and
 - means for generating block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.
19. The apparatus of claim 18, further comprising means for generating an encoded video bitstream comprising coded data for the block and the block-type syntax information.
20. The apparatus of claim 18, further comprising means for selecting the encoding modes used to encode the partitions based on an encoding cost analysis.

21. The apparatus of claim 20, further comprising means for selecting the partitioning of the block based on the encoding cost analysis, wherein the encoding cost analysis includes an analysis of bit rate and distortion for intra- and inter-encoding of the partitions.
22. The apparatus of claim 18, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
23. The apparatus of claim 18, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
24. The apparatus of claim 18, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
25. The apparatus of claim 18, wherein the video block has a size of at least 64x64 pixels.
26. A computer-readable medium encoded with instructions to cause a video encoder to:
- receive a video block having a size of more than 16x16 pixels;
 - partition the block into partitions;
 - encode one of the partitions using a first encoding mode;
 - encode another of the partitions using a second encoding mode different from the first encoding mode; and
 - generate block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions.
27. The computer-readable medium of claim 26, further comprising instructions to cause the video encoder to generate an encoded video bitstream comprising coded data for the block and the block-type syntax information.

28. The computer-readable medium of claim 26, further comprising instructions to cause the video encoder to select the encoding modes used to encode the partitions based on an encoding cost analysis.
29. The computer-readable medium of claim 28, further comprising instructions to cause the video encoder to select the partitioning of the block based on the encoding cost analysis wherein the encoding cost analysis includes an analysis of bit rate and distortion for intra- and inter-encoding of the partitions.
30. The computer-readable medium of claim 26, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
31. The computer-readable medium of claim 26, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
32. The computer-readable medium of claim 26, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
33. The computer-readable medium of claim 26, wherein the video block has a size of at least 64x64 pixels.
34. A method comprising:
receiving, with a video decoder, a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode;

receiving block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions; and decoding the video block based on the block-type syntax information.

35. The method of claim 34, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
36. The method of claim 34, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
37. The method of claim 34, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
38. The method of claim 34, wherein the video block has a size of at least 64x64 pixels.
39. An apparatus comprising a video decoder configured to:
 - receive a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode;
 - receive block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions; and
 - decode the video block based on the block-type syntax information.
40. The apparatus of claim 39, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.

41. The apparatus of claim 39, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
42. The apparatus of claim 39, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
43. The apparatus of claim 39, wherein the video block has a size of at least 64x64 pixels.
44. The apparatus of claim 38, wherein the apparatus comprises a wireless communication device.
45. An apparatus comprising:
means for receiving a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode;
means for receiving block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions;
and
means for decoding the video block based on the block-type syntax information.
46. The apparatus of claim 45, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
47. The apparatus of claim 45, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.

48. The apparatus of claim 45, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
49. The apparatus of claim 45, wherein the video block has a size of at least 64x64 pixels.
50. A computer-readable medium encoded with instructions to cause a video decoder to:
- receive, with a video decoder, a video block having a size of more than 16x16 pixels, wherein the block is partitioned into partitions, one of the partitions is encoded with a first encoding mode and another of the partitions is encoded with a second encoding mode different from the first encoding mode;
 - receive block-type syntax information that indicates the size of the block and identifies the partitions and the encoding modes used to encode the partitions; and
 - decode the video block based on the block-type syntax information.
51. The computer-readable medium of claim 50, wherein the first encoding mode includes an intra-encoding mode and the second encoding mode includes an inter-encoding mode.
52. The computer-readable medium of claim 50, wherein the first encoding mode includes a first intra-encoding mode and the second encoding mode includes a second intra-encoding mode different from the first intra-encoding mode.
53. The computer-readable medium of claim 50, wherein the first encoding mode includes a first inter-encoding mode and the second encoding mode includes a second inter-encoding mode different from the first inter-encoding mode.
54. The computer-readable medium of claim 50, wherein the video block has a size of at least 64x64 pixels.

CLAIMS: SET C

1. A method comprising:
 - receiving, with a digital video encoder, a video coding unit;
 - determining a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels;
 - determining a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels;
 - encoding the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric; and
 - encoding the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.
2. The method of claim 1, further comprising generating an encoded video bitstream comprising block-type syntax information indicating the size of the blocks used to encode the video coding unit.
3. The method of claim 1, wherein the second blocks have sizes of at least 64x64 pixels.
4. The method of claim 1, further comprising:
 - determining a third rate-distortion metric for encoding the video coding unit using third video blocks with sizes greater than the sizes of the first video blocks and less than the sizes of the second video blocks; and
 - encoding the video coding unit using the second video blocks when the third rate-distortion metric is less than the first rate-distortion metric and the second rate-distortion metric.
5. The method of claim 4, wherein the third blocks have sizes of at least 32x32 pixels.

6. The method of claim 1, wherein the video coding unit comprises one of a frame or a slice of a video sequence.
7. An apparatus comprising a video encoder configured to:
 - receive a video coding unit;
 - determine a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels;
 - determine a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels;
 - encode the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric; and
 - encode the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.
8. The apparatus of claim 7, wherein the video encoder is configured to generate an encoded video bitstream comprising block-type syntax information indicating the size of the blocks used to encode the video coding unit.
9. The apparatus of claim 7, wherein the second blocks have sizes of at least 64x64 pixels.
10. The apparatus of claim 7, wherein the video encoder is configured to:
 - determine a third rate-distortion metric for encoding the video coding unit using third video blocks with sizes greater than the sizes of the first video blocks and less than the sizes of the second video blocks; and
 - encode the video coding unit using the second video blocks when the third rate-distortion metric is less than the first rate-distortion metric and the second rate-distortion metric.
11. The apparatus of claim 10, wherein the third blocks have sizes of at least 32x32 pixels.

12. The apparatus of claim 7, wherein the video coding unit comprises one of a frame or a slice of a video sequence.
13. An apparatus comprising:
 - means for receiving, with a digital video encoder, a video coding unit;
 - means for determining a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels;
 - means for determining a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels;
 - means for encoding the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric; and
 - means for encoding the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.
14. The apparatus of claim 13, further comprising means for generating an encoded video bitstream comprising block-type syntax information indicating the size of the blocks used to encode the video coding unit.
15. The apparatus of claim 13, wherein the second blocks have sizes of at least 64x64 pixels.
16. The apparatus of claim 15, further comprising:
 - means for determining a third rate-distortion metric for encoding the video coding unit using third video blocks with sizes greater than the sizes of the first video blocks and less than the sizes of the second video blocks; and
 - means for encoding the video coding unit using the second video blocks when the third rate-distortion metric is less than the first rate-distortion metric and the second rate-distortion metric.

17. The apparatus of claim 16, wherein the third blocks have sizes of at least 32x32 pixels.
18. The apparatus of claim 13, wherein the video coding unit comprises one of a frame or a slice of a video sequence.
19. A computer-readable medium encoded with instructions to cause a video encoder to:
 - receive a video coding unit;
 - determine a first rate-distortion metric for encoding the video coding unit using first video blocks with sizes of 16x16 pixels;
 - determine a second rate-distortion metric for encoding the video coding unit using second video blocks with sizes of more than 16x16 pixels;
 - encode the video coding unit using the first video blocks when the first rate-distortion metric is less than second rate-distortion metric; and
 - encode the video coding unit using the second video blocks when the second rate-distortion metric is less than the first rate-distortion metric.
20. The computer-readable medium of claim 19, further comprising instructions to cause the video encoder to generate an encoded video bitstream comprising block-type syntax information indicating the size of the blocks used to encode the video coding unit.
21. The computer-readable medium of claim 19, wherein the second blocks have sizes of at least 64x64 pixels.

22. The computer-readable medium of claim 19, further comprising instructions to cause the video encoder to:
 - determine a third rate-distortion metric for encoding the video coding unit using third video blocks with sizes greater than the sizes of the first video blocks and less than the sizes of the second video blocks; and
 - encode the video coding unit using the second video blocks when the third rate-distortion metric is less than the first rate-distortion metric and the second rate-distortion metric.
23. The computer-readable medium of claim 22, wherein the third blocks have sizes of at least 32x32 pixels.
24. The computer-readable medium of claim 19, wherein the video coding unit comprises one of a frame or a slice of a video sequence.

ABSTRACT

Techniques are described for encoding and decoding digital video data using macroblocks that are larger than the macroblocks prescribed by convention video encoding and decoding standards. For example, the techniques include encoding and decoding a video stream using macroblocks comprising greater than 16x16 pixels, for example, 64x64 pixels. In one example, an apparatus includes a video encoder that identifies a block of a coded unit of the digital video stream, wherein the block comprises greater than 16x16 pixels, partitions the block into a first partition and a second partition, encodes the first partition of the block using an intra-prediction encoding mode, encodes the second partition of the block using an inter-prediction encoding mode, and generates a block-type identifier that identifies the first partition as being encoded using the intra-prediction encoding mode and the second partition as being encoded using the inter-prediction encoding mode.

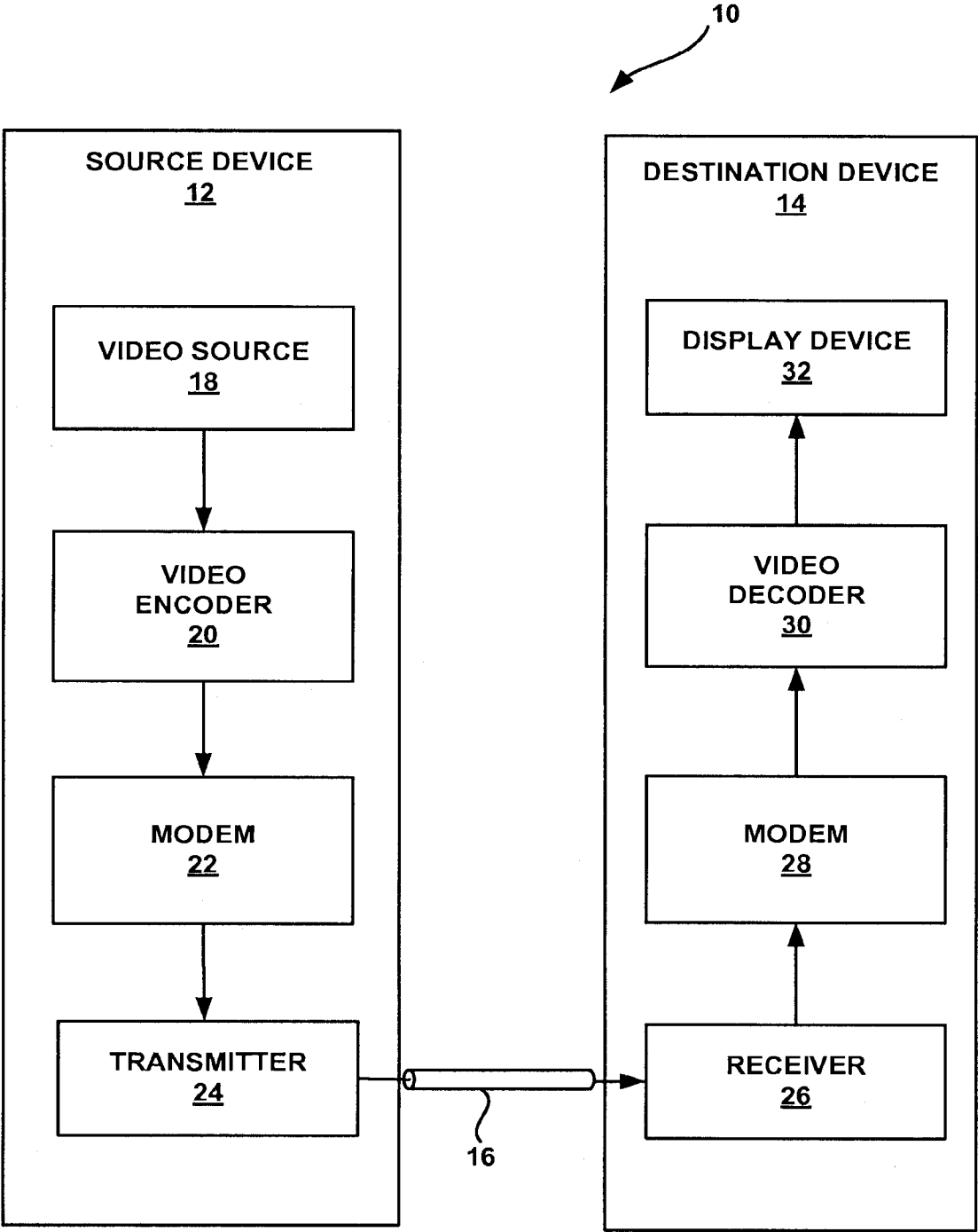


FIG. 1

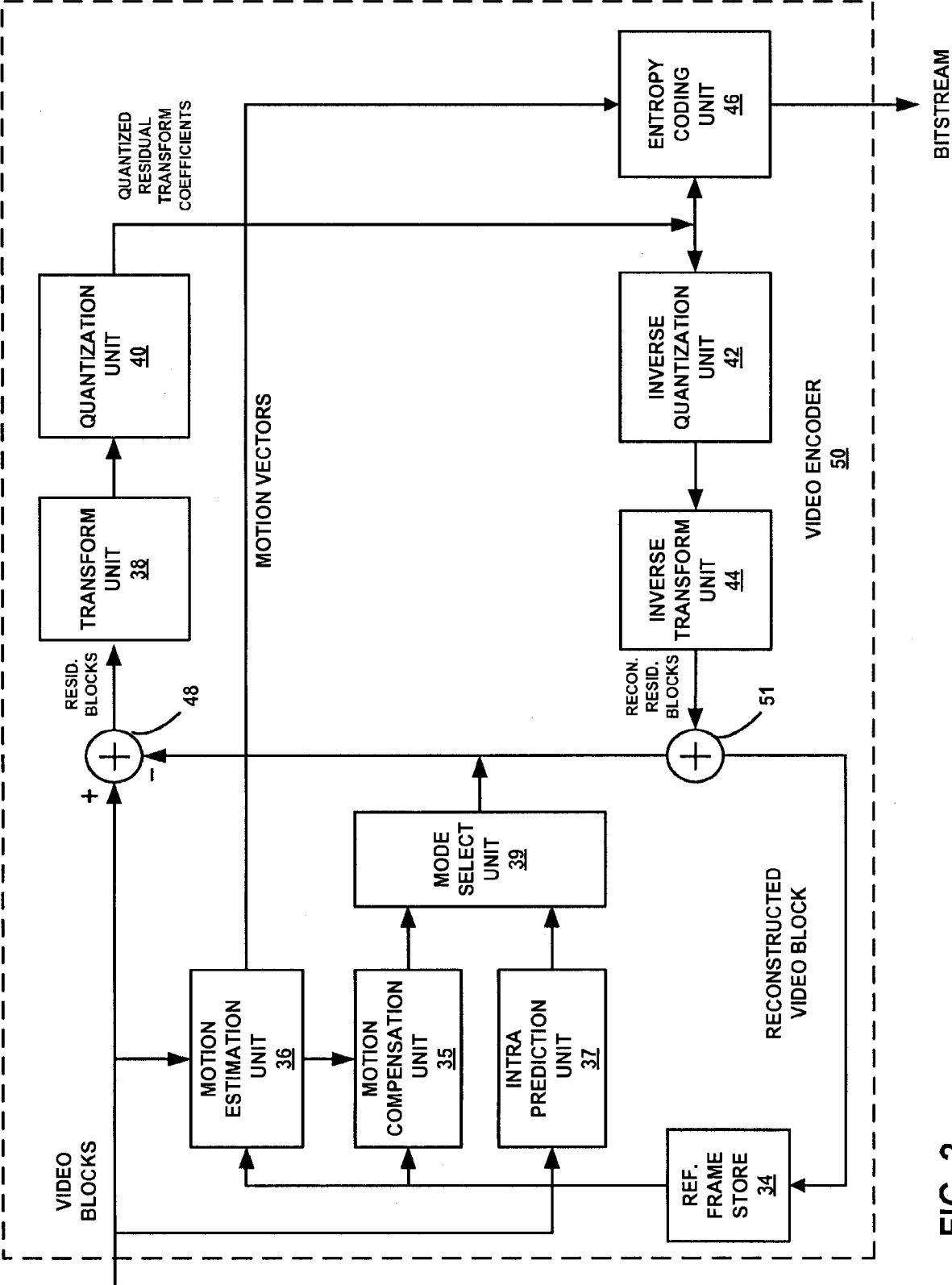


FIG. 2

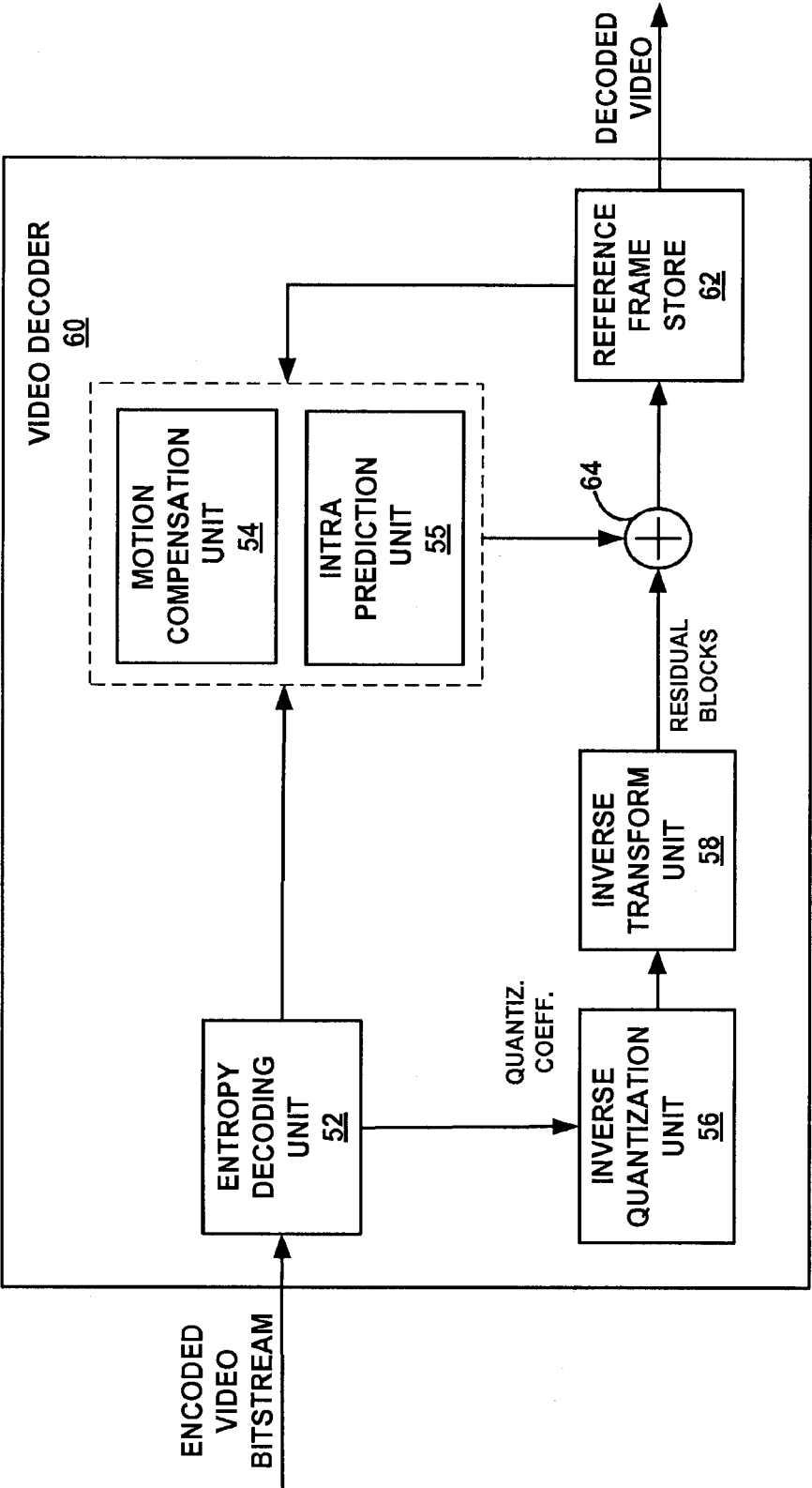


FIG. 3

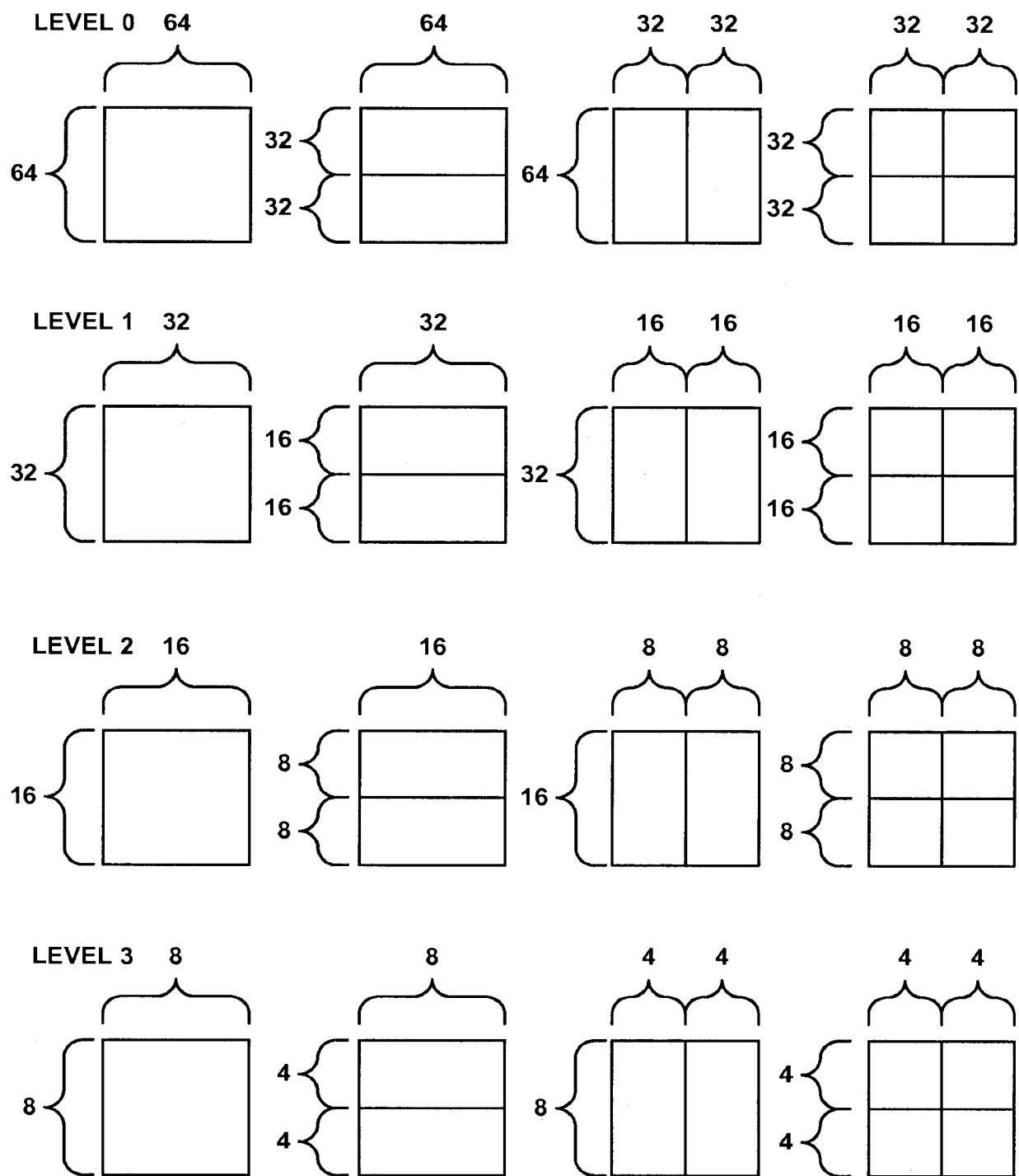


FIG. 4

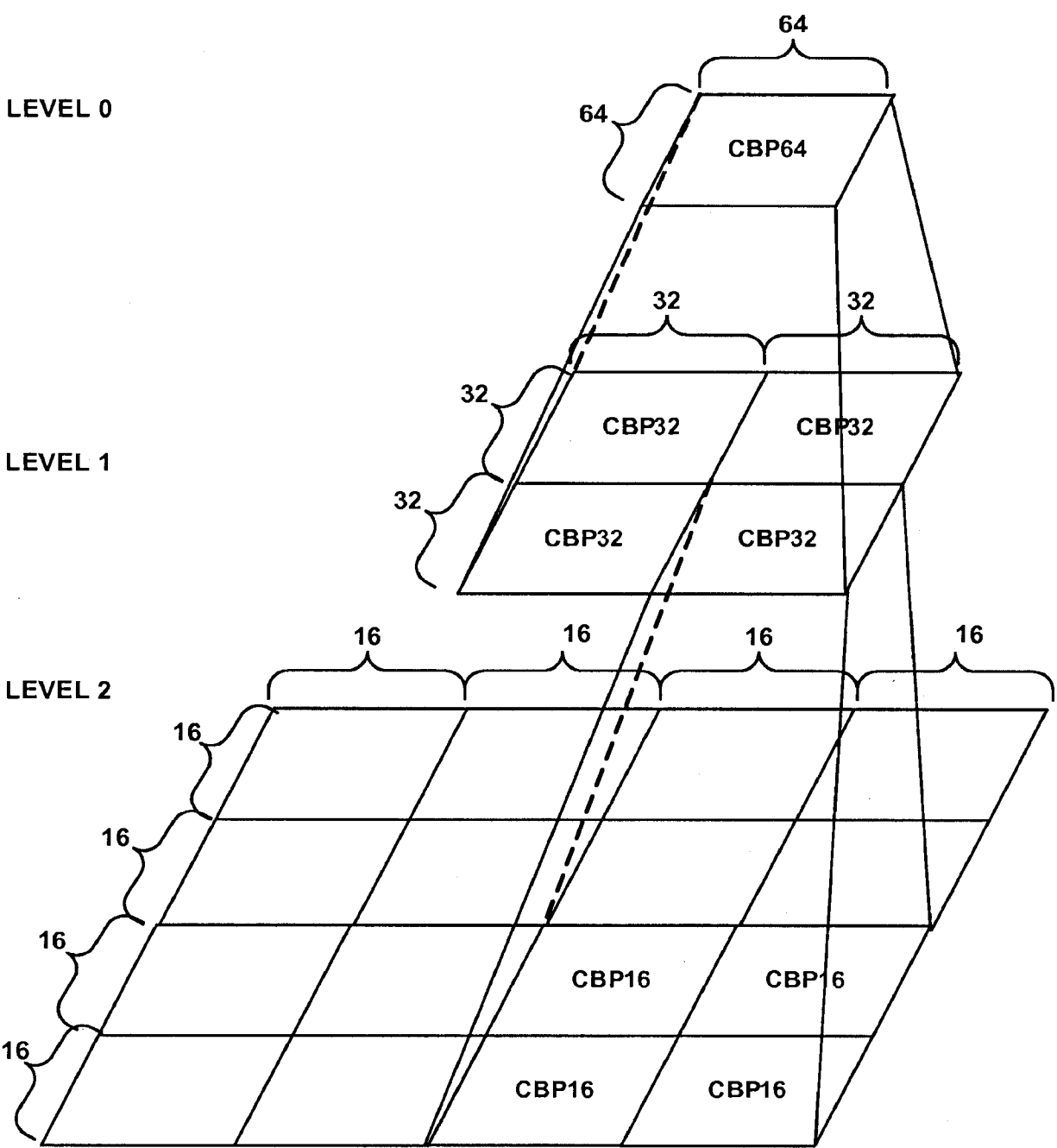


FIG. 5

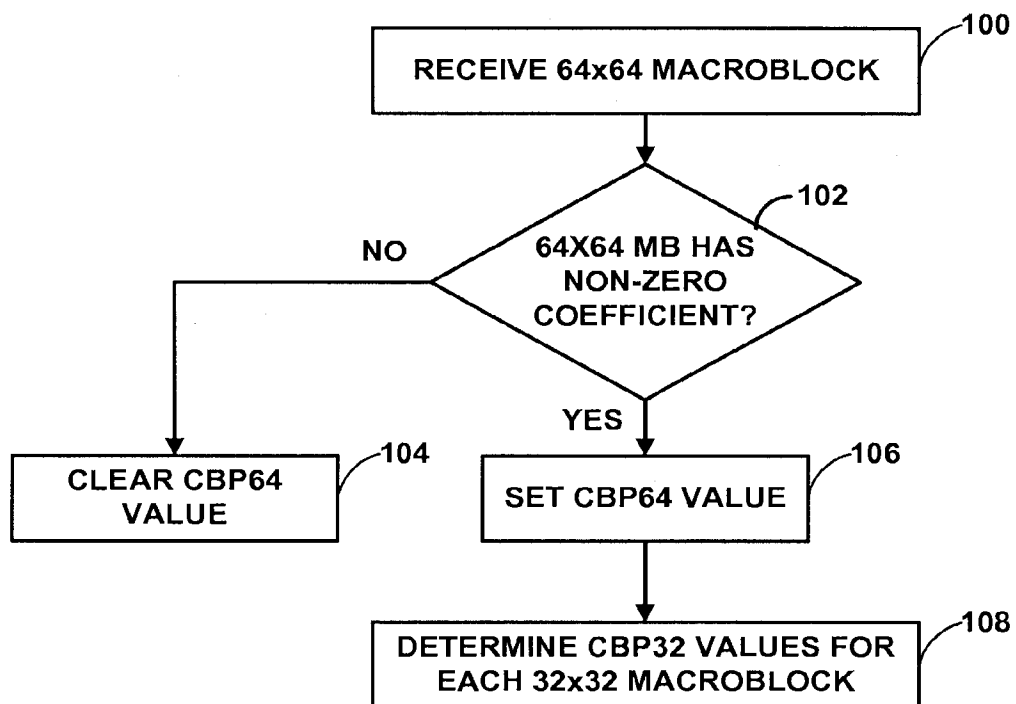


FIG. 6

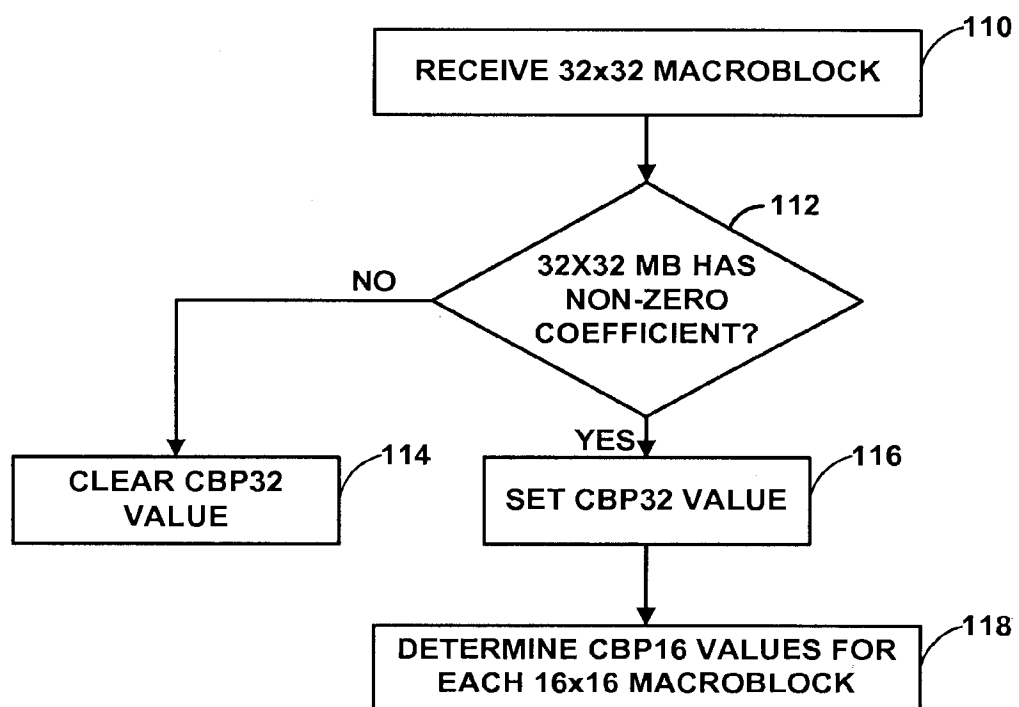


FIG. 7

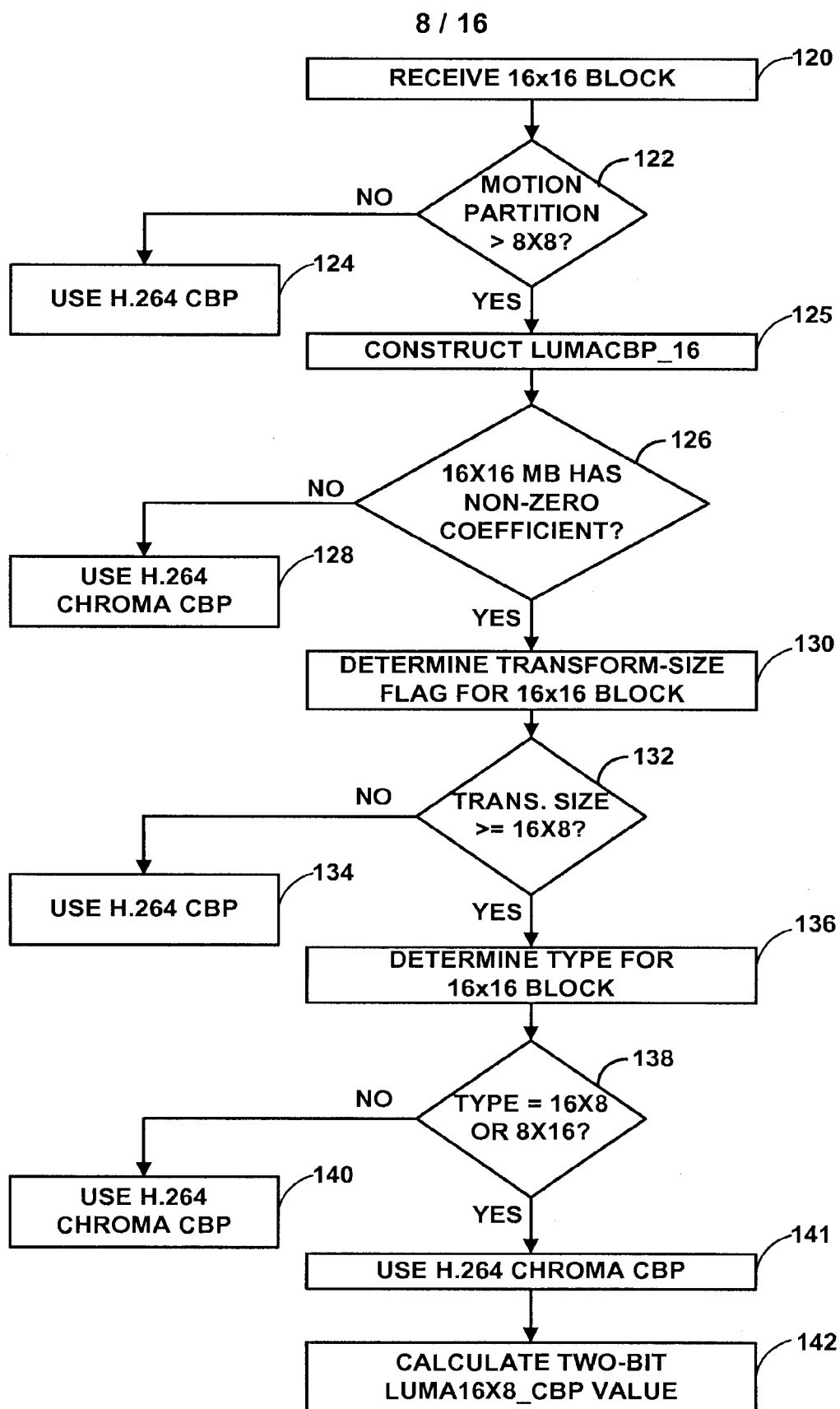


FIG. 8

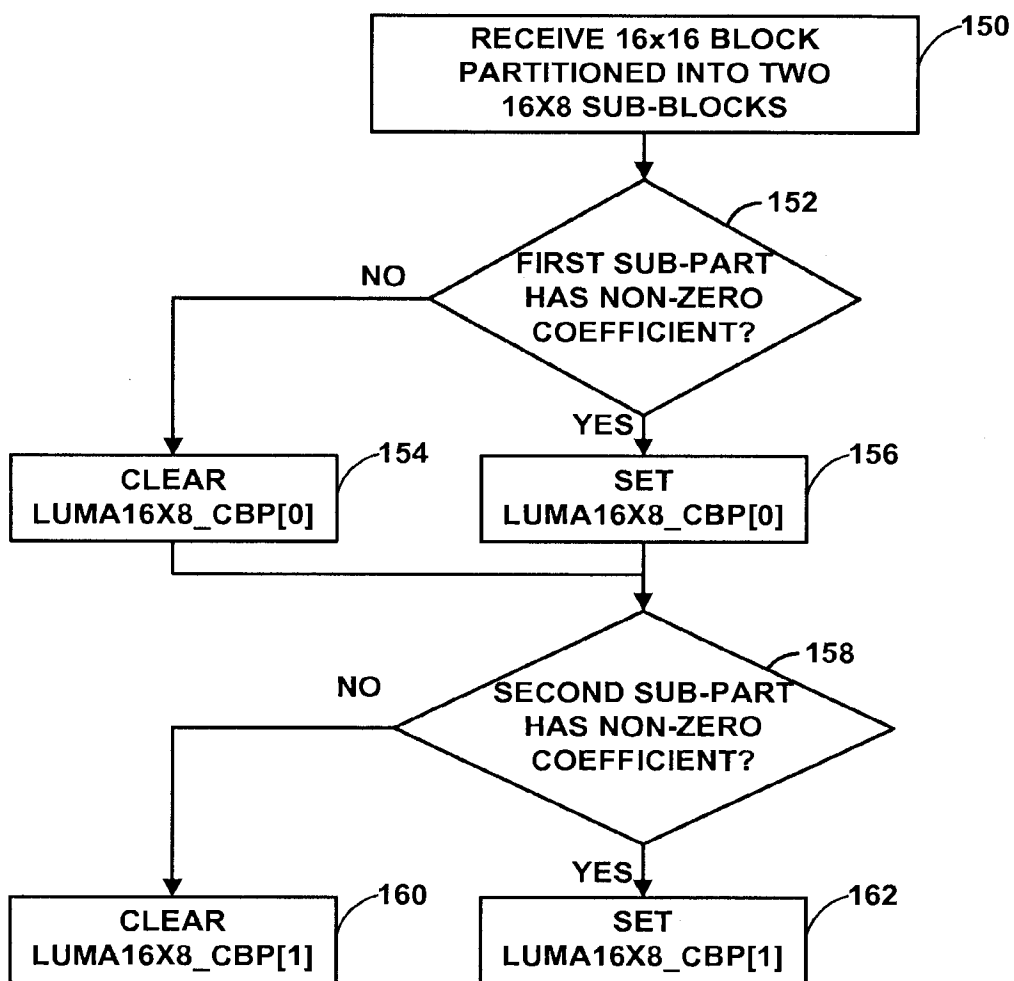


FIG. 9

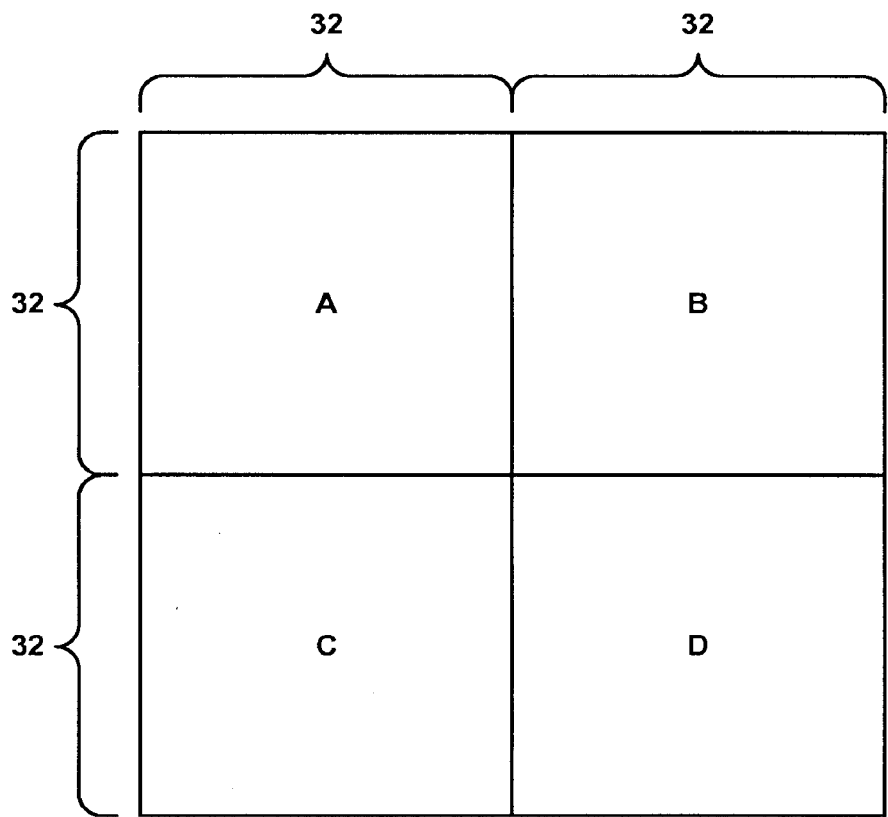


FIG. 10

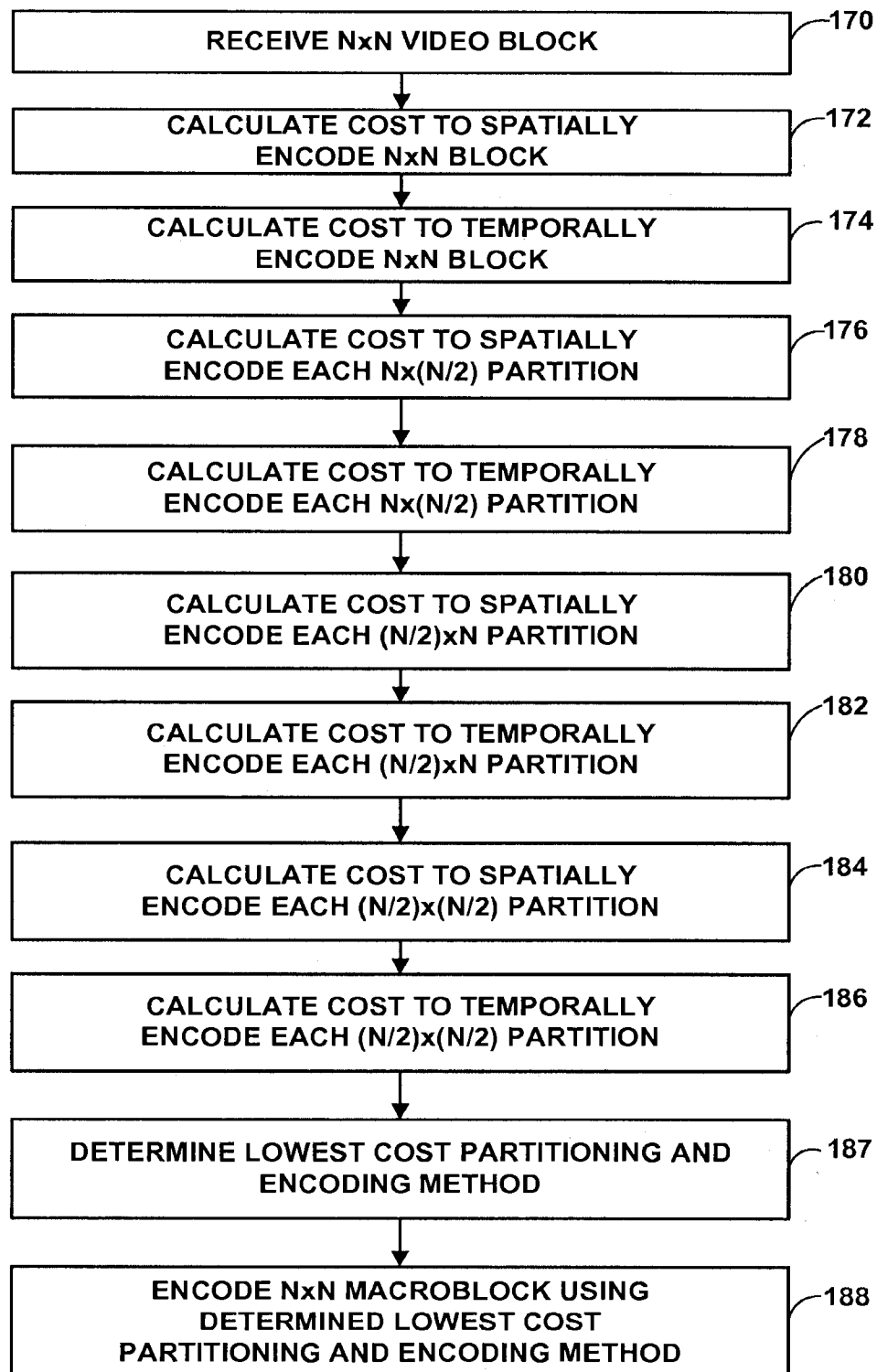


FIG. 11

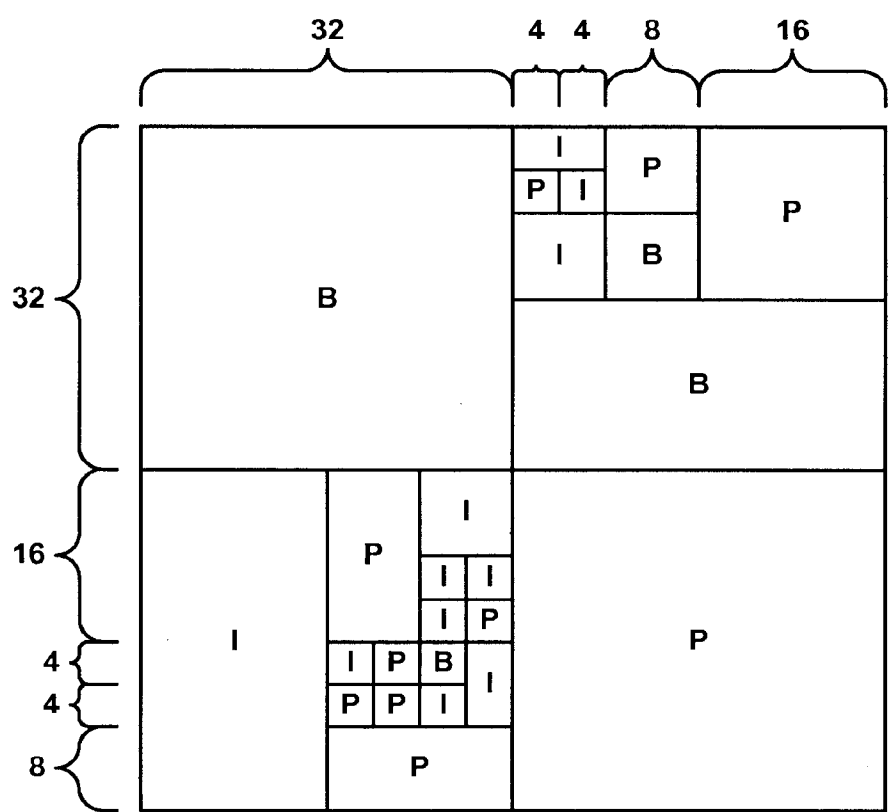


FIG. 12

13 / 16

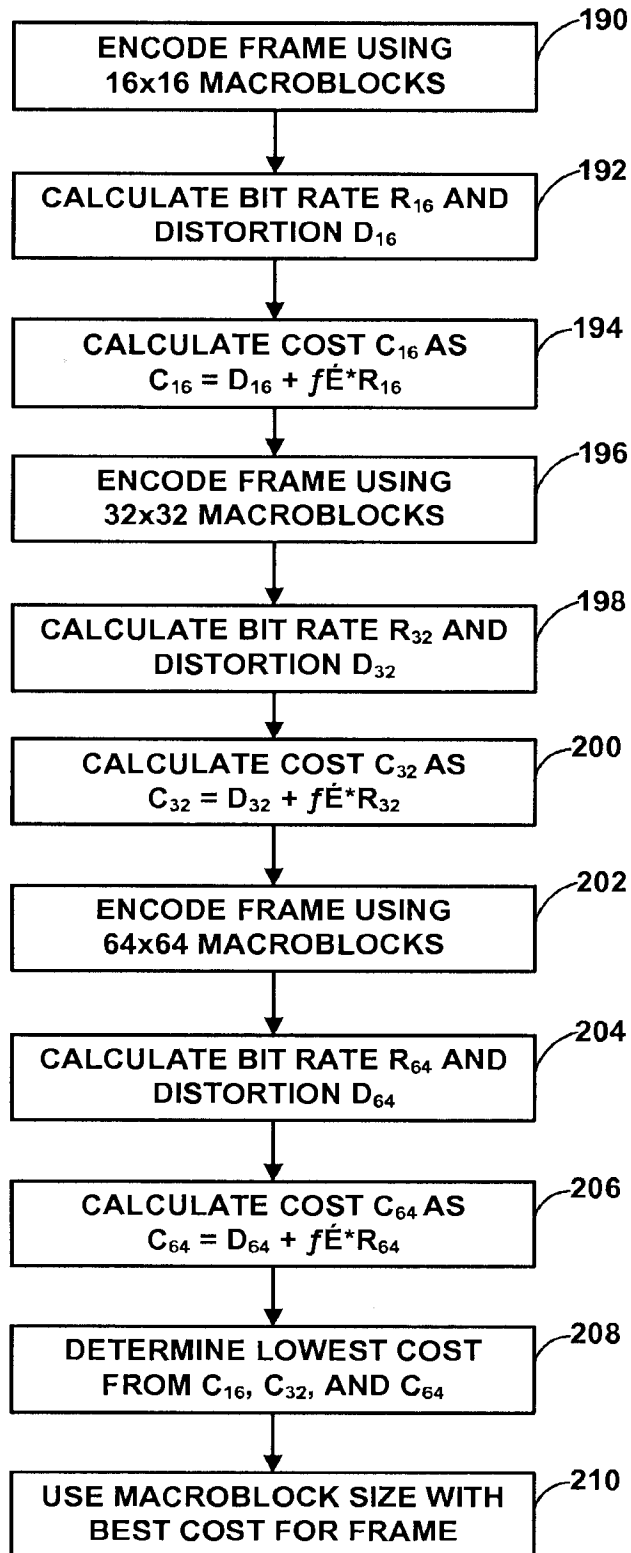


FIG. 13

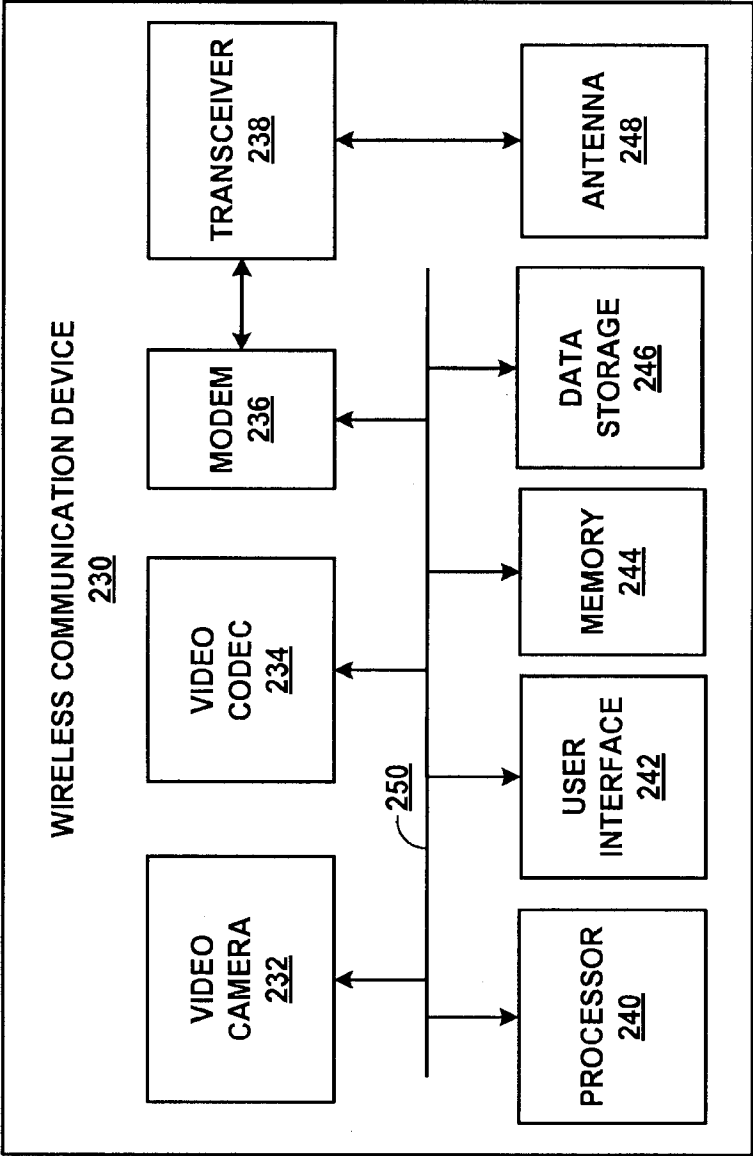


FIG. 14

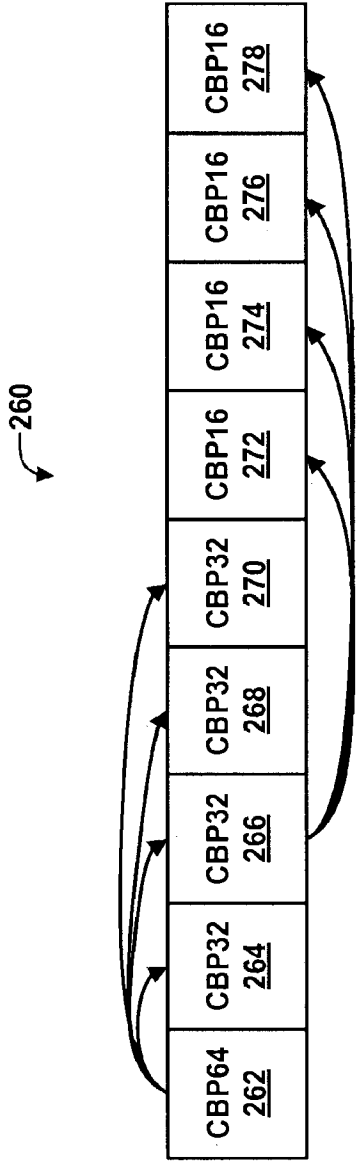


FIG. 15

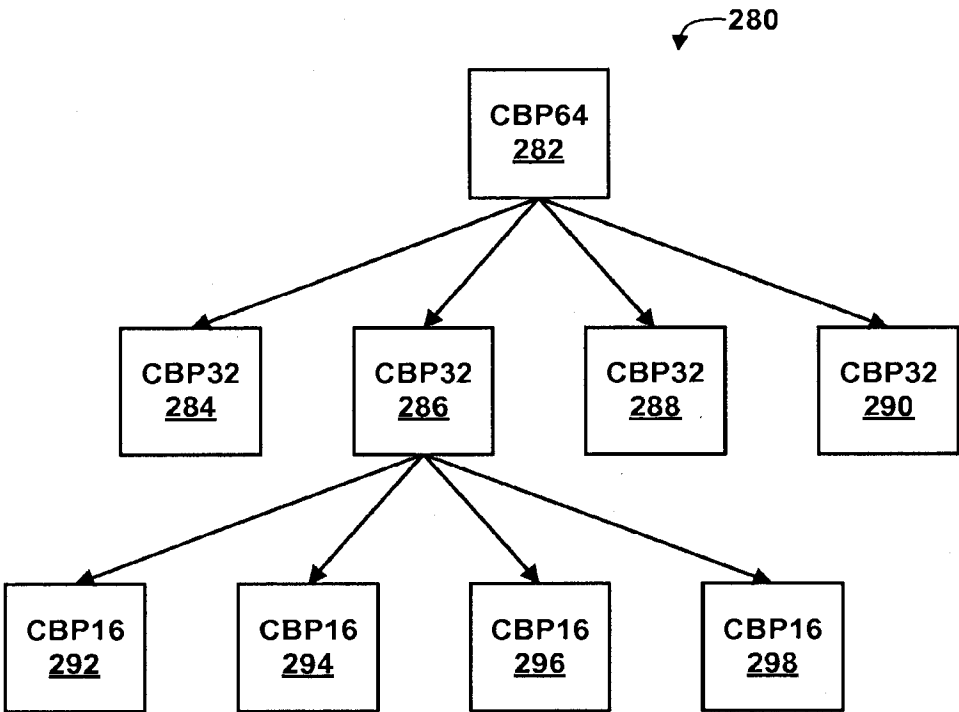


FIG. 16

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: TYLER SULLIVAN

Title: 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

A copy of reference number 85 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 86-90 listed on the attached Form PTO-1449 are enclosed herewith.

REFERENCES IN A LANGUAGE OTHER THAN ENGLISH

The following documents are not in the English language. Accordingly, an English language counterpart application has been provided as indicated below.

Foreign Language Document	Cite No.	English Language Counterpart	Cite No.
JP 2000-308060 A	86	US 6978048 B1	85

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: Nov. 30, 2023

/Brian J. Hubbard/
Brian J. Hubbard
Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
215-558-5740



UNITED STATES
PATENT AND TRADEMARK OFFICE

ELECTRONIC PAYMENT RECEIPT

APPLICATION # 17/351,649	RECEIPT DATE / TIME 12/01/2023 03:38:34 PM Z ET	ATTORNEY DOCKET # ICVW_2009P00073 US02
-----------------------------	--	---

Title of Invention

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

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2654	FILED BY	Melissa Inzaris
PATENT CENTER #	63460253	AUTHORIZED BY	Brian Hubbard
CUSTOMER #	15800	FILING DATE	06/18/2021
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Yunfei Zheng

Payment Information

PAYMENT METHOD CARD / 1628	PAYMENT TRANSACTION ID E2023B1F40349300	PAYMENT AUTHORIZED BY Melissa Inzaris
-------------------------------	--	--

FEE CODE	DESCRIPTION	ITEM PRICE(\$)	QUANTITY	ITEM TOTAL(\$)
1820	REQUEST FOR CONTINUED EXAMINATION (RCE)- 2ND AND SUBSEQUENT REQUEST (SEE 37 CFR 1.114)	2000.00	1	2000.00
			TOTAL AMOUNT:	\$2,000.00

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 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.

Notice of Allowability	Application No. 17/351,649	Applicant(s) Zheng et al.	
	Examiner Tyler W Sullivan	Art Unit 2487	AIA (FITF) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to RCE filed 12/1/2023.
☐ A declaration(s)/affidavit(s) under 37 CFR 1.130(b) was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are See Continuation Sheet . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)

2. ☒ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date 12/1/2023.

3. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material _____.

4. ☐ Interview Summary (PTO-413),
Paper No./Mail Date. _____.

5. ☐ Examiner's Amendment/Comment

6. ☒ Examiner's Statement of Reasons for Allowance

7. ☐ Other _____.

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

Continuation of 3. The allowed claim(s) is/are: 20-22,24-27 and 29-39

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after allowance or after an Office action under *Ex Parte Quayle*, 25 USPQ 74, 453 O.G. 213 (Comm'r Pat. 1935). Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, prosecution in this application has been reopened pursuant to 37 CFR 1.114. Applicant's submission filed on December 1st, 2023 has been entered.

Priority

Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, 365(c), or 386(c) is acknowledged (US Provisional Application 61/222,177 filed July 1st, 2009).

Information Disclosure Statement

The information disclosure statement (IDS) submitted on December 1st, 2023 were filed before the mailing date of the Notice of Allowance (this Office Action). The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the Examiner.

The information disclosure statements (IDS) submitted on June 18th, 2021 and October 26th, 2021 were filed before the mailing date of the First Action on the Merits (mailed August 18th, 2022). The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the Examiner.

Due to the excessively lengthy Information Disclosure Statement submitted by applicant, the examiner has given only a cursory review of the listed references. In accordance with MPEP 609.04(a), applicant is encouraged to provide a concise explanation of why the information is being submitted and how it is understood to be relevant. Concise explanations (especially those which point out the relevant pages and lines) are helpful to the Office, particularly where documents are lengthy and complex and applicant is aware of a section that is highly relevant to patentability or where a large number of documents are submitted and applicant is aware that one or more are highly relevant to patentability. Applicant is required to comply with this statement for any non-English language documents. See 37 CFR § 1.56 Duty to Disclose Information Material to Patentability.

Terminal Disclaimer

The terminal disclaimer filed on August 25th, 2023 disclaiming the terminal portion of any patent granted on this application which would extend beyond the expiration date of **US Patent #11,082,697** has been reviewed and is accepted. The terminal disclaimer has been recorded.

Allowable Subject Matter

Claims 20 – 22, 24 – 27, and 29 – 39 are allowed.

The following is an examiner's statement of reasons for allowance: Claim 20 is taken as the representative claim. Claim 20 in view of the previous PTAB decision recites a negative limitation (the large block is not further partition) which while the prior art renders obvious, the prior art does not render obvious signaling the intra prediction mode used for the large block. Even in the case of a 32x32 or 32x16 block, Chen and Moriya do not disclose an intra prediction mode for the whole 32x32 block (e.g. the 32 x 32 Intra DC mode proposed by the Applicant or unifying the 4 16x16 partition modes together) and rely on partitioning the larger block into smaller prediction units with basic / base block sizes for intra prediction. When the limitation is viewed as a whole, the prior art fails to contemplate intra modes as claimed by the Applicant in view of the video codecs being improved.

Claims 25, 32, and 36 cite similar allowable Subject Matter as independent claim 20 and thus are similarly Allowed.

Dependent claims 21 – 22, 24, 26 – 27, 29 – 35, and 37 – 39 depend on allowed independent claims and thus are similarly Allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Po, et al. (US PG PUB 2009/0219991 A1 referred to as “Po” throughout) teaches in Figures 3 – 4 and Paragraphs 8 at least the decisions to merge / split blocks with hierarchy of intra prediction modes / splits.

References previously cited against the claims: Chen, et al. (US PG PUB 2010/0086029 A1 referred to as “Chen” throughout in which citations will come from US Provisional Applications 61/166,631 (referred to as “Chen ‘631” throughout) and 61/144,357 (referred to as “Chen ‘631” throughout)) [The US PG PUB is cited in Applicant's June 18th, 2021 IDS as US PG PUB Item #24 – The Examiner will cite from the US Provisional References where “Chen” refers to the collective of “Chen ‘631” and “Chen ‘357”]; Moriya, et al. (US PG PUB 2009/0046781 A1 referred to as “Moriya” throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #20]; Moriya, et al. (US PG PUB 2008/0123947 A1 referred to as “Moriya 47” throughout) [Cited in Applicant's June 18th, 2021 IDS as US PG PUB Items #16 and 17].

References cited in previous updated search including Interference Search (from the prior Allowance on September 1st, 2023): Chen, et al. (US PG PUB 2010/0086030 A1 referred to as “Chen” throughout) teaches in Paragraphs 120 – 125 considerations for intra predicting a large block not further split, but without description of unifying the intra modes for the 16 x 16 blocks into a 32 x 32 mode for the large block (each partitioned intra block is treated separately).

Additional Reference based on updated search in the December 1st, 2023 RCE filed: Chen, et al. (US PG PUB 2011/0090967 A1 referred to as “Chen 67” throughout) in which Figures 7 and 23A illustrate

a 32 x 32 block not further partitioned to be intra predicted; however, the PCT filing date is not applicable as the WIPO publication was filed and published in Korean thus no 102(e) filing date is available thus is reference is NOT prior art.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tyler W Sullivan whose telephone number is (571)270-5684. The examiner can normally be reached IFP.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Czekaj can be reached on (571)-272-7327. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: <https://patentcenter.uspto.gov>. Visit <https://www.uspto.gov/patents/apply/patent-center> for more information about Patent Center and <https://www.uspto.gov/patents/docx> for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

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

<i>Notice of References Cited</i>	Application/Control No. 17/351,649		Applicant(s)/Patent Under Reexamination Zheng et al.	
	Examiner Tyler W Sullivan		Art Unit 2487	Page 1 of 1

U.S. PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date YYYY-MM-DD	Name	CPC Classification	US Classification
*	A	US-20090219991-A1	2009-09-03	Po; Lai-Man	H04N19/176	375/E7.126
*	B	US-20100086030-A1	2010-04-08	Chen; Peisong	H04N19/147	375/E7.243
*	C	US-20110090967-A1	2011-04-21	Chen; Jianle	H04N19/129	375/E7.125
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS						
*		Document Number Country Code-Number-Kind Code	Date YYYY-MM-DD	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS		
*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

Substitute for 1449/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	17/351,649
				Filing Date	June 18, 2021
				First Named Inventor	Yunfei Zheng
				Art Unit	2487
				Examiner Name	TYLER SULLIVAN
Sheet	1	of	1	Attorney Docket Number	ICVW_2009P00073 US02

U.S. PATENTS						
Examiner Initial*	Cite No	Publication Number	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)
/TWS/	85	6978048	B1	2005-12-20	Higginbottom et al.	

FOREIGN PATENT DOCUMENTS								
Examiner Initials*	Cite No	Foreign Document Number	Country Code	Kind Code	Publication Date	Name of Patentee or Applicant of cited Document	Specific Part (if noted)	T
/TWS/	86	2000-308060	JP	A	2000-11-02	Canon KK		

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	
/TWS/	87	FAUNDEZ-ZANUY et al., "Contributions to Interframe Coding", Workshop on image analysis & synthesis in image coding, Berlin, October 1994, 6 pages		
	88	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, April 15-23, 2010, pp. 1-42		
	89	CHEN, et al., "US Provisional Patent Application No. 61/144,357", January 13, 2009, 10 pages		
	90	CHEN, et al., "US Provisional Patent Application No. 61/166,631", April 3, 2009, 97 pages		

EXAMINER SIGNATURE			
Examiner Signature	/TYLER W. SULLIVAN/	Date Considered	12/06/2023
*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.			

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
InterDigital VC Holdings, Inc.

Confirmation No.: 2654

Application No.: 17/351,649

Group Art Unit: 2487

Filing Date: June 18, 2021

Examiner: TYLER SULLIVAN

Title: 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

A copy of reference number 85 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 86-90 listed on the attached Form PTO-1449 are enclosed herewith.

REFERENCES IN A LANGUAGE OTHER THAN ENGLISH

The following documents are not in the English language. Accordingly, an English language counterpart application has been provided as indicated below.

/TWS/

Foreign Language Document	Cite No.	English Language Counterpart	Cite No.
JP 2000-308060 A	86	US 6978048 B1	85

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /TWS/

/TYLER W. SULLIVAN/
12/06/2023

Please charge any deficiency or credit any overpayment to Deposit Account No. 50-5519.

Date: Nov. 30, 2023

/Brian J. Hubbard/
Brian J. Hubbard
Registration No. 45,873

CONDO ROCCIA KOPTIW LLP
1800 JFK Blvd., Suite 1700
Philadelphia, PA 19103
215-558-5740



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

NOTICE OF ALLOWANCE AND FEE(S) DUE

15800 7590 12/18/2023
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

EXAMINER

SULLIVAN, TYLER

ART UNIT PAPER NUMBER

2487

DATE MAILED: 12/18/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654

TITLE OF INVENTION: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	03/18/2024

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.
If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.
If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".
For purposes of this notice, small entity fees are 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via the USPTO patent electronic filing system.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

15800759012/18/2023

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

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via the USPTO patent electronic filing system or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

17/351,64906/18/2021Yunfei ZhengICVW_2009P00073 US022654

TITLE OF INVENTION: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	03/18/2024

EXAMINER	ART UNIT	CLASS-SUBCLASS
SULLIVAN, TYLER	2487	375-240020

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,

1

(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

2

3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required)

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

☐ Electronic Payment via the USPTO patent electronic filing system☐ Enclosed check☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No.

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized SignatureDate

Typed or printed nameRegistration No.



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	12/18/2023	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	
Suite 1700			PAPER NUMBER	
Philadelphia, PA 19103			2487	
DATE MAILED: 12/18/2023				

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. 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 30 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. 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.

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. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013).

<https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf>

Routine uses of the information in this record may include disclosure to:

- 1) law enforcement, in the event that the system of records indicates a violation or potential violation of law;
- 2) a federal, state, local, or international agency, in response to its request;
- 3) a contractor of the USPTO having need for the information in order to perform a contract;
- 4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record;
- 5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record;
- 6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations;
- 7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals;
- 8) 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));
- 9) the Office of Personnel Management (OPM) for personnel research purposes; and
- 10) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.



UNITED STATES
PATENT AND TRADEMARK OFFICE

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/351,649	02/09/2024 11:54:59 AM Z ET	ICVW_2009P00073 US02

Title of Invention

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

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2654	FILED BY	Cassandra Katz
PATENT CENTER #	64271677	FILING DATE	06/18/2021
CUSTOMER #	15800	FIRST NAMED INVENTOR	Yunfei Zheng
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Brian Hubbard

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
ICVW_2009P00073US02_IsueFeeTransmittal.pdf	1	Issue Fee Payment (PTO-85B)	1679 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
ICVW_2009P00073US02_IssueFeeTransmittal.pdf	66A36E5367768B2D9EC300B174AF95C11A2CF4627F2582CA811FDD5551FD88E114C13F9616CB79A261CDF689113424279D6E1D88C707BF8A2FFD84012D5F4B8C

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 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.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via the USPTO patent electronic filing system.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

15800 7590 12/18/2023
Condo Roccia Koptiw LLP
1800 JFK Boulevard
Suite 1700
Philadelphia, PA 19103

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via the USPTO patent electronic filing system or by facsimile to (571) 273-2885, on the date below.

	(Typed or printed name)
	(Signature)
	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

17/351.649 06/18/2021 Yunfei Zheng ICVW_2009P00073 US02 2654

TITLE OF INVENTION: METHODS AND APPARATUS FOR SIGNALING INTRA PREDICTION FOR LARGE BLOCKS FOR VIDEO ENCODERS AND DECODERS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	\$1200	\$0.00	\$0.00	\$1200	03/18/2024

EXAMINER	ART UNIT	CLASS-SUBCLASS
SULLIVAN, TYLER	2487	375-240020

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363). ☐ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached. ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.	2. For printing on the patent front page, list (1) The names of up to 3 registered patent attorneys or agents OR, alternatively, (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 1. <u>CONDO ROCCIA KOPTIW LLP</u> 2. _____ 3. _____
---	--

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE InterDigital VC Holdings, Inc.	(B) RESIDENCE: (CITY and STATE OR COUNTRY) Wilmington, Delaware
--	--

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☒ Issue Fee ☐ Publication Fee (if required)

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

☒ Electronic Payment via the USPTO patent electronic filing system ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. 50-5519

5. Change in Entity Status (from status indicated above)

☐ Applicant certifying micro entity status. See 37 CFR 1.29

☐ Applicant asserting small entity status. See 37 CFR 1.27

☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature <u>/Brian J. Hubbard/</u>	Date <u>Feb. 9, 2024</u>
Typed or printed name <u>Brian J. Hubbard</u>	Registration No. <u>45,873</u>



UNITED STATES
PATENT AND TRADEMARK OFFICE

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/351,649	02/09/2024 11:54:59 AM Z ET	ICVW_2009P00073 US02

Title of Invention

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

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2654	FILED BY	Cassandra Katz
PATENT CENTER #	64271677	AUTHORIZED BY	Brian Hubbard
CUSTOMER #	15800	FILING DATE	06/18/2021
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Yunfei Zheng

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1628	E202429B56059530	Cassandra Katz
PRE-AUTHORIZED ACCOUNT	PRE-AUTHORIZED CATEGORY	
505519	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)	

FEE CODE	DESCRIPTION	ITEM PRICE(\$)	QUANTITY	ITEM TOTAL(\$)
1501	UTILITY ISSUE FEE	1200.00	1	1200.00
			TOTAL AMOUNT:	\$1,200.00

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 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.



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

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	------------	------------	---------------------	------------------

17/351,649	03/19/2024	11936876	ICVW_2009P00073 US02	2654
------------	------------	----------	----------------------	------

15800	7590	02/28/2024		
-------	------	------------	--	--

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

ISSUE NOTIFICATION

The projected patent number and issue date are specified above. The patent will issue electronically. The electronically issued patent is the official patent grant pursuant to 35 U.S.C. § 153. The patent may be accessed on or after the issue date through Patent Center at <https://patentcenter.uspto.gov/>. The patent will be available in both the public and the private sides of Patent Center. Further assistance in electronically accessing the patent, or about Patent Center, is available by calling the Patent Electronic Business Center at 1-888-217-9197.

The USPTO is implementing electronic patent issuance with a transition period, during which period the USPTO will mail a ceremonial paper copy of the electronic patent grant to the correspondence address of record. Additional copies of the patent (i.e., certified and presentation copies) may be ordered for a fee from the USPTO’s Certified Copy Center at <https://certifiedcopycenter.uspto.gov/index.html>. The Certified Copy Center may be reached at (800)972-6382.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Center (<https://patentcenter.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Patents Stakeholder Experience (OPSE), Stakeholder Support Division (SSD) at (571)-272-4200.

INVENTOR(s) (Please see PATENT CENTER site <https://patentcenter.uspto.gov> for additional inventors):

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

APPLICANT(s) (Please see PATENT CENTER site <https://patentcenter.uspto.gov> for additional applicants):

InterDigital VC Holdings, Inc., Wilmington, DE;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit [SelectUSA.gov](https://selectusa.gov).



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/351,649	06/18/2021	Yunfei Zheng	ICVW_2009P00073 US02	2654
15800	7590	03/19/2024	EXAMINER	
Condo Roccia Koptiw LLP			SULLIVAN, TYLER	
1800 JFK Boulevard			ART UNIT	PAPER NUMBER
Suite 1700			2487	
Philadelphia, PA 19103			MAIL DATE	DELIVERY MODE
			03/19/2024	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

APPLICATION NO.	ISSUE DATE	PATENT NO.
17/351,649	19-Mar-2024	11936876

Condo Roccia Koptiw LLP
1800 JFK Boulevard
Philadelphia, PA 19103

EGRANT NOTIFICATION

Your electronic patent grant (eGrant) is now available, which can be accessed via Patent Center at <https://patentcenter.uspto.gov>

The electronic patent grant is the official patent grant under 35 U.S.C. 153. For more information, please visit <https://www.uspto.gov/electronicgrants>

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 17/351,649				
APPLICATION AS FILED - PART I										
(Column 1)		(Column 2)		SMALL ENTITY		OR	OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A			N/A	320		
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A		N/A			N/A	700		
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A			N/A	800		
TOTAL CLAIMS (37 CFR 1.16(j))	20	minus 20 =	*			OR	x 100 =	0.00		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	4	minus 3 =	* 1				x 480 =	480		
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).							0.00		
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))								0.00		
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL			TOTAL	2300		
APPLICATION AS AMENDED - PART II										
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))							OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))							OR		
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.										

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875						Application or Docket Number 17/351,649				
APPLICATION AS FILED - PART I										
(Column 1)		(Column 2)		SMALL ENTITY		OR	OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA		RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A			N/A	320		
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A		N/A			N/A	700		
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A			N/A	800		
TOTAL CLAIMS (37 CFR 1.16(j))	20	minus 20 =	*			OR	x 100 =	0.00		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	4	minus 3 =	* 1				x 480 =	480		
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).							0.00		
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))								0.00		
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL			TOTAL	2300		
APPLICATION AS AMENDED - PART II										
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)		RATE(\$)	ADDITIONAL FEE(\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.										

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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 17/351,649		Filing Date 06/18/2021		☐ To be Mailed	
ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO									
APPLICATION AS FILED - PART I									
		(Column 1)		(Column 2)					
FOR		NUMBER FILED		NUMBER EXTRA		RATE (\$)		FEE (\$)	
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A		N/A		N/A			
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))		N/A		N/A		N/A			
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A		N/A		N/A			
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 = *				x \$100 =			
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 = *				x \$480 =			
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).							
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))									
* If the difference in column 1 is less than zero, enter "0" in column 2.						TOTAL			
APPLICATION AS AMENDED - PART II									
		(Column 1)		(Column 2)		(Column 3)			
AMENDMENT	01/18/2023	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	* 18	Minus	** 20	= 0	x \$100 =		0	
	Independent (37 CFR 1.16(h))	* 4	Minus	*** 4	= 0	x \$480 =		0	
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE		0	
		(Column 1)		(Column 2)		(Column 3)			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=	x \$0 =			
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x \$0 =			
	☐ Application Size Fee (37 CFR 1.16(s))								
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
						TOTAL ADD'L FEE			
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.						LIE			
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".						/JACQUELYN L WILLIAMS/			
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".									
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.									

This collection of information is required by 37 CFR 1.16. 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 12 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.

PTO/SB/06 (09-11)

Approved for use through 1/31/2014. 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 displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875			Application or Docket Number 17/351,649		Filing Date 06/18/2021		☐ To be Mailed	
ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO								
APPLICATION AS FILED - PART I								
		(Column 1)	(Column 2)					
FOR		NUMBER FILED	NUMBER EXTRA		RATE (\$)		FEE (\$)	
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))		N/A	N/A		N/A			
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))		N/A	N/A		N/A			
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))		N/A	N/A		N/A			
TOTAL CLAIMS (37 CFR 1.16(i))		minus 20 = *		x \$100 =				
INDEPENDENT CLAIMS (37 CFR 1.16(h))		minus 3 = *		x \$480 =				
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))		If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))								
* If the difference in column 1 is less than zero, enter "0" in column 2.					TOTAL			
APPLICATION AS AMENDED - PART II								
		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT	06/22/2023	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))	* 18	Minus	** 20	= 0	x \$100 =		0
	Independent (37 CFR 1.16(h))	* 4	Minus	*** 4	= 0	x \$480 =		0
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
						TOTAL ADD'L FEE		0
		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)		ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))	*	Minus	**	=	x \$0 =		
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x \$0 =		
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
						TOTAL ADD'L FEE		
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.						LIE		
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".						/LAMONT N MCLAUCHLIN/		
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".								
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.								

This collection of information is required by 37 CFR 1.16. 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 12 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.

Derwent Innovation Patent Export, 2025-08-04 02:53:54 +0000

Search results for: PNS=(US11936876);

Collections searched: DWPI; ARIPO grant; Argentina UM; Argentina app; Armenia grant; Australia UM; Australia app; Australia grant; Austria UM; Austria app; Austria grant; Belarus UM; Belarus grant; Belgium app; Belgium grant; Brazil UM; Brazil app; Brazil grant; Bulgaria UM; Bulgaria app; Bulgaria grant; Canada app; Canada grant; China, Mainland UM; China, Mainland app; China, Mainland grant; Colombia app; Costa Rica UM; Costa Rica app; Croatia UM; Croatia app; Croatia grant; Cuba app; Cuba grant; Czech Republic UM; Czech Republic app; Czech Republic grant; Czechoslovakia grant; Denmark UM; Denmark app; Denmark grant; Estonia UM; Estonia app; Estonia grant; Eurasian (EAPO) grant; Europe (EP) app; Europe (EP) grant; Finland app; Finland grant; France app; France grant; Georgia grant; German Democratic Republic app; German Democratic Republic grant; Germany UM; Germany app; Germany grant; Greece app; Greece grant; Gulf Cooperation Council grant; Hong Kong UM; Hong Kong app; Hong Kong grant; Hungary grant; Iceland grant; India app; India grant; Indonesia UM; Indonesia app; Ireland UM; Ireland app; Ireland grant; Israel grant; Italy grant; Italy grant; Italy grant; Italy grant; Japan UM; Japan app; Japan grant; Korea UM; Korea app; Korea grant; Latvia grant; Lithuania grant; Luxembourg app; Luxembourg grant; Malaysia app; Malaysia grant; Mexico app; Mexico grant; Moldova UM; Moldova grant; Monaco grant; Mongolia UM; Mongolia grant; Morocco app; Morocco grant; Netherlands app; Netherlands grant; New Zealand grant; Norway app; Norway grant; OAPI grant; PCT (WO) app; Philippines UM; Philippines app; Philippines grant; Poland UM; Poland grant; Portugal UM; Portugal app; Portugal grant; Romania UM; Romania app; Romania granted; Russia UM; Russia app; Russia grant; Saudi Arabia app; Saudi Arabia grant; Serbia UM; Serbia app; Serbia grant; Singapore app; Singapore grant; Slovakia UM; Slovakia app; Slovakia grant; Slovenia UM; Slovenia grant; South Africa app; Spain UM; Spain app; Spain grant; Sweden app; Sweden grant; Switzerland app; Switzerland grant; Taiwan UM; Taiwan appl; Taiwan grant; Thailand UM; Thailand app; Thailand grant; Tunisia app; Turkey UM; Turkey app; UK app; UK grant; USSR grant; Ukraine UM; Ukraine grant; United States app; United States grant; Uruguay app; Vietnam UM; Vietnam app; Vietnam grant; Other

Table of Contents

1. US11936876B2	Methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders
-----------------	---

Family 1/1

1 record(s) per family

Record 1/1 US11936876B2 Methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders

Title: Methods and apparatus for signaling intra prediction for large blocks for video encoders and decoders

Title - DWPI: Intra prediction signaling apparatus for large blocks for video encoders and decoders has video encoder for encoding picture data for at least one large block in picture by signaling intra prediction for large block

Priority Number: US2009222177P | WO2010US1862A | US13381538A | US15806893A

Priority Date: 2009-07-01 | 2010-06-29 | 2011-12-29 | 2017-11-08

Application Number: US17351649A

Application Date: 2021-06-18

Publication Date: 2024-03-19

IPC Class Table:

IPC	Section	Class	Subclass	Class Group	Subgroup
H04N0019136	H	H04	H04N	H04N0019	H04N0019136
H04N001911	H	H04	H04N	H04N0019	H04N001911
H04N0019147	H	H04	H04N	H04N0019	H04N0019147
H04N0019157	H	H04	H04N	H04N0019	H04N0019157
H04N0019159	H	H04	H04N	H04N0019	H04N0019159
H04N0019176	H	H04	H04N	H04N0019	H04N0019176
H04N0019196	H	H04	H04N	H04N0019	H04N0019196
H04N001946	H	H04	H04N	H04N0019	H04N001946
H04N0019463	H	H04	H04N	H04N0019	H04N0019463
H04N0019593	H	H04	H04N	H04N0019	H04N0019593
H04N001970	H	H04	H04N	H04N0019	H04N001970
H04N001996	H	H04	H04N	H04N0019	H04N001996

IPC Class Table - DWPI:

IPC - DWPI	Section - DWPI	Class - DWPI	Subclass - DWPI	Class Group - DWPI	Subgroup - DWPI
H04N001911	H	H04	H04N	H04N0019	H04N001911
H04N0019119	H	H04	H04N	H04N0019	H04N0019119
H04N0019136	H	H04	H04N	H04N0019	H04N0019136
H04N0019147	H	H04	H04N	H04N0019	H04N0019147
H04N0019157	H	H04	H04N	H04N0019	H04N0019157
H04N0019159	H	H04	H04N	H04N0019	H04N0019159
H04N0019176	H	H04	H04N	H04N0019	H04N0019176
H04N0019196	H	H04	H04N	H04N0019	H04N0019196
H04N001946	H	H04	H04N	H04N0019	H04N001946
H04N0019463	H	H04	H04N	H04N0019	H04N0019463
H04N0019593	H	H04	H04N	H04N0019	H04N0019593
H04N001970	H	H04	H04N	H04N0019	H04N001970
H04N001996	H	H04	H04N	H04N0019	H04N001996
H04N000726	H	H04	H04N	H04N0007	H04N000726
H04N000732	H	H04	H04N	H04N0007	H04N000732
H04N	H	H04	H04N	-	-

Assignee/Applicant: InterDigital VC Holdings Inc.,Wilmington,DE,US

JP F Terms:

JP FI Codes:

Assignee - Original: InterDigital VC Holdings Inc.

Any CPC Table:

Type	CPC - Current - Invention	CPC - Current - Additional	Version	Office
Current	H04N 19/136	-	20141101	EP; US
Current	H04N 19/147		20141101	EP; US
Current	H04N 19/196		20141101	EP; US
Current	H04N 19/463		20141101	EP; US
Current	H04N 19/96		20141101	EP; US
Current	H04N 19/11	-	20141101	EP; KR; US
Current	H04N 19/176		20141101	EP; KR; US

Current	H04N 19/70		20141101	EP; KR; US
Current	H04N 19/119	-	20141101	KR
Current	H04N 19/157	-	20141101	US
Current	H04N 19/159		20141101	US
Current	H04N 19/46		20141101	US
Current	H04N 19/593	-	20141101	KR; US
Original	H04N 19/136	-	20141101	US
Original	H04N 19/11		20141101	US
Original	H04N 19/147		20141101	US
Original	H04N 19/157		20141101	US
Original	H04N 19/159		20141101	US
Original	H04N 19/176		20141101	US
Original	H04N 19/196		20141101	US
Original	H04N 19/46		20141101	US
Original	H04N 19/463		20141101	US
Original	H04N 19/593		20141101	US
Original	H04N 19/70		20141101	US
Original	H04N 19/96		20141101	US

ECLA:

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.

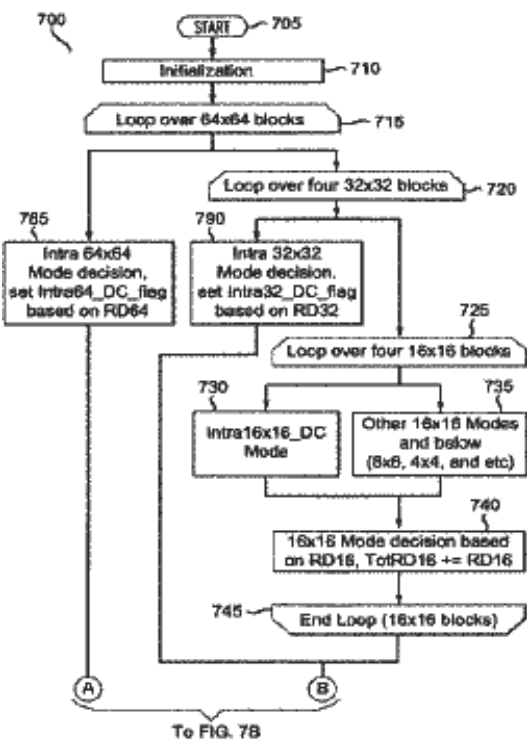
Language of Publication: EN

INPADOC Legal Status Table:

Gazette Date	Code	INPADOC Legal Status Impact
2024-02-28	STCF	-
Description: INFORMATION ON STATUS: PATENT GRANT PATENTED CASE		
2024-02-12	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL PUBLICATIONS -- ISSUE FEE PAYMENT VERIFIED		

2024-02-09	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL PUBLICATIONS -- ISSUE FEE PAYMENT RECEIVED		
2023-12-18	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL NOTICE OF ALLOWANCE MAILED -- APPLICATION RECEIVED IN OFFICE OF PUBLICATIONS		
2023-12-06	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL DOCKETED NEW CASE - READY FOR EXAMINATION		
2023-09-14	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL NOTICE OF ALLOWANCE MAILED -- APPLICATION RECEIVED IN OFFICE OF PUBLICATIONS		
2023-09-06	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL AWAITING TC RESP., ISSUE FEE NOT PAID		
2023-09-01	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL NOTICE OF ALLOWANCE MAILED -- APPLICATION RECEIVED IN OFFICE OF PUBLICATIONS		
2023-06-29	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL DOCKETED NEW CASE - READY FOR EXAMINATION		
2022-08-18	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL NON FINAL ACTION MAILED		
2021-10-15	STPP	-
Description: INFORMATION ON STATUS: PATENT APPLICATION AND GRANTING PROCEDURE IN GENERAL DOCKETED NEW CASE - READY FOR EXAMINATION		
2021-06-18	FEPP	-
Description: FEE PAYMENT PROCEDURE ENTITY STATUS SET TO UNDISCOUNTED (ORIGINAL EVENT CODE: BIG.); ENTITY STATUS OF PATENT OWNER: LARGE ENTITY		

Post-Issuance (US):
Reassignment (US) Table:
Maintenance Status (US):
Litigation (US):
Opposition (EP):
License (EP):
EPO Procedural Status:
Front Page Drawing:



Priority Number - DWPI: US2009222177P | WO2010US1862A | CN201510242289A | CN201510242361A | US13381538A | US13381538A | US2017806893A | US2021351649A | US2023227124A | US18830023A
Priority Date - DWPI: 2009-07-01 | 2010-06-29 | 2010-06-29 | 2010-06-29 | 2011-12-29 | 2011-12-29 | 2017-11-08 | 2021-06-18 | 2023-07-27 | 2024-09-10
Assignee - Current US: InterDigital VC Holdings Inc.



United States Patent and Trademark Office

Office of the Commissioner for Patents

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

PATENT #11936876

APPLICATION #17351649

FILING DATE06/18/2021

ISSUE DATE03/19/2024

Payment Window Status

WINDOW		STATUS		FEES	
3.5 Year		Not Open		Not Due	
Window	First Day to Pay	Surcharge Starts	Last Day to Pay	Status	Fees
3.5 Year	03/19/2027	09/21/2027	03/20/2028	Not Open	Not Due
7.5 Year	03/19/2031	09/20/2031	03/19/2032	Not Open	Not Due
11.5 Year	03/19/2035	09/20/2035	03/19/2036	Not Open	Not Due

No maintenance fees are due at this time. 3.5 year window opens on 03/19/2027.

Patent Holder Information

Customer #15800

Entity StatusUNDISCOUNTED

Phone Number2155585740

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