

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Satoru Sakazume
U.S. Patent No.: 8,090,025 Attorney Docket No.: 50095-0247IP1
Issue Date: January 3, 2012
Appl. Serial No.: 11/787,623
Filing Date: April 17, 2007
Title: MOVING-PICTURE CODING APPARATUS, METHOD AND
PROGRAM, AND MOVING-PICTURE DECODING
APPARATUS, METHOD AND PROGRAM

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES

PATENT NO. 8,090,025 PURSUANT TO 35 U.S.C. §§ 311–319,

37 C.F.R. § 42

TABLE OF CONTENTS

I.	REQUIREMENTS FOR IPR	1
A.	Grounds for Standing.....	1
B.	Challenge and Relief Requested.....	1
C.	Claim Construction	2
D.	Level of Ordinary Skill in the Art.....	3
II.	THE '025 PATENT	3
A.	Brief Description.....	3
B.	Prosecution History.....	7
III.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	7
A.	[GROUND 1A/1B] – Claims 1-10 are rendered obvious by Mualla-Shirani-Saito and claims 1-10 are rendered obvious by Mualla-Shirani-Saito-Stockhammer.....	8
1.	Mualla (and Mualla-Stockhammer in Ground 1B)	8
2.	Shirani.....	18
3.	Combination of Mualla and Shirani (and the Combination of Mualla-Stockhammer and Shirani in Ground 1B)	22
4.	Saito	24
5.	Combination of Mualla-Shirani and Saito (and the Combination of Mualla-Stockhammer-Shirani and Saito in Ground 1B).....	26
6.	Analysis	33
B.	[GROUND 2] – Claims 1-10 are rendered obvious by Mualla-Shirani-Saito-Shimizu.....	85
1.	Shimizu.....	85
2.	Combination of Mualla-Shirani-Saito and Shimizu.....	88
3.	Analysis	98
IV.	PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION.....	103
V.	CONCLUSION AND FEES	103
VI.	MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1).....	103
A.	Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	103
B.	Related Matters Under 37 C.F.R. § 42.8(b)(2)	103
C.	Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	104
D.	Service Information	104

LIST OF EXHIBITS

APPLE-1001	U.S. Patent No. 8,090,025 to Sakazume (“the ’025 Patent”)
APPLE-1002	Excerpts from the Prosecution History of the ’025 Patent (“the Prosecution History”)
APPLE-1003	Declaration and Curriculum Vitae of Mr. Lett
APPLE-1004	Complaint, <i>Advanced Coding Technologies LLC v. Apple Inc.</i> , 2-24-cv-00572 (EDTX), filed July 22, 2024
APPLE-1005	Al-Mualla, et al., “Video Coding for Mobile Communications: Efficiency, Complexity and Resilience,” Elsevier, 2002 (Mualla)
APPLE-1006	S. Shirani, et al., “A Concealment Method for Video Communications in an Error-Prone Environment,” in <i>IEEE Journal on Selected Areas in Communications</i> , vol. 18, no. 6, pp. 1122-1128, June 2000 (Shirani)
APPLE-1007	N. Saito, et al., “The Polyharmonic Local Sine Transform: A New Tool for Local Image Analysis and Synthesis without Edge Effect,” <i>Applied and Computational Harmonic Analysis</i> , vol. 20, pp. 41-73, January 2006 (Saito)
APPLE-1008	B. Girod, et al., “Feedback-Based Error Control for Mobile Video Transmission,” in <i>Proceedings of the IEEE</i> , vol. 87, no. 10, pp. 1707-1723, Oct. 1999 (Girod)
APPLE-1009	T. Stockhammer, et al., “H.264/AVC in Wireless Environments,” in <i>IEEE Trans. on Circuits and Systems for Video Tech.</i> , vol. 13, no. 7, July 2003 (Stockhammer)
APPLE-1010	Perez, et al., “Poisson Image Editing,” <i>ACM, ACM Transactions on Graphics (TOG)</i> , vol. 22, Issue 3, Jul. 2003, pp. 313-318 (Perez)
APPLE-1011	Wexler, et al., “Space-Time Video Completion,” In <i>Proc. IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition</i> , 2004, pp. 120 –127 (Wexler)

APPLE-1012	Raskar, et al., “Image Fusion for Context Enhancement and Video Surrealism,” Proceedings of the 3rd Symposium on Non-Photorealistic Animation and Rendering, 2004, pp. 85-152 (Raskar)
APPLE-1013	Sun, et al., “Poisson Matting,” ACM Transactions on Graphics (Tog), Special Issue: Proceedings of the 2004 Siggraph Conference Session: Interacting With Images, vol. 23, Issue 3, 2004, pp. 315-321 (Sun)
APPLE-1014	U.S. Patent No. 7,636,128 B2 to Sun, et al. (Sun-128)
APPLE-1015	Fattal, et al., “Gradient Domain High Dynamic Range Compression,” In Proceedings of ACM SIGGRAPH, 2002, 249–256 (Fattal)
APPLE-1016	ITU-T Rec. H.263 (01/2005) (“the H.263 standard”)
APPLE-1017	M. Wada, “Selective Recovery of Video Packet Loss Using Error Concealment,” IEEE Journal on Selected Areas of Communication, vol. 7, pp. 807–814, June 1989 (Wada)
APPLE-1018	Declaration of June Munford
APPLE-1019	Memorandum, Interim Processes for PTAB Workload Management (USPTO March 26, 2025) (“Stewart Memorandum”)
APPLE-1020	RESERVED
APPLE-1021	ITU-T Rec. H.263 (02/1998)
APPLE-1022	List, et al., “Adaptive Deblocking Filter,” IEEE Journal on Circuits and Systems for Video Technology, vol. 13, pp. 614-619, July 2003 (List)
APPLE-1023	ISO/IEC 11172-2: 1993 (E) (MPEG1)
APPLE-1024	U.S. Patent Pub. No. 2003/0174776 A1 to Shimizu et al. (“Shimizu”)
APPLE-1025	P. Salama, et al., “Error Concealment for Shape Coding,” Proceedings, 2002 International Conference on Image

Processing, Rochester, NY, USA, Vol. 2, pp. 701-704, Sep. 2002 (Salama)

- APPLE-1026 L.D. Soares, et al., “Motion-based shape error concealment for object-based video,” Proceedings, 2004 International Conference on Image Processing, Singapore, Vol. 2, pp. 797-800, 2004 (Soares)
- APPLE-1027 Petition (Paper 2), IPR2024-00374, *Samsung Electronics. Co., Ltd. v. Advanced Coding Techs., LLC* (Jan. 8, 2024)
- APPLE-1028 Decision Instituting *Inter Partes* Review (Paper 11), IPR2024-00374, *Samsung Electronics. Co., Ltd. v. Advanced Coding Techs., LLC* (July 22, 2024)
- APPLE-1029 Dr. Cliff Reader declaration, Inter Partes Review (IPR) proceeding, IPR2024-00374
- APPLE-1030 ISO/IEC 13818-2: 2000 (E) (MPEG2/H.262 Video)
- APPLE-1031 RESERVED
- APPLE-1032 ISO/IEC 14496-10 (2004)

LISTING OF CHALLENGED CLAIM ELEMENTS¹

Claim 1	
[1.pre]	A moving-picture coding apparatus comprising:
[1.1.COD]	a predictive encoder to produce and encode a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided;
[1.2.COD]	a zone-border motion estimator to obtain a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones,
[1.3.COD]	find a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[1.4.COD]	and generate border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found; and
[1.5.COD]	a zone-border motion compensator to define a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data,
[1.6.COD]	and generate an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture,
[1.7.COD]	wherein the predictive encoder produces the residual picture with the first predictive picture as the predictive picture and encodes the residual picture;
[1.8.COD]	said predictive encoder including a first subtractor to produce a first residual picture from a difference between the picture to be coded and the first predictive picture;

¹ These claims reflect the certificate of correction filed Sep. 6, 2022.

[1.9.COD]	an orthogonal transformer to perform orthogonal transform to the first residual picture, thus generating orthogonal-transform coefficients data;
[1.10.COD]	a quantizer to perform quantization to the orthogonal transform coefficients data based on a specific quantization parameter, thus generating post-quantization data;
[1.11.DEC]	an inverse-quantizer to perform inverse-quantization to the post-quantization data based on a specific quantization parameter, thus generating post-inverse-quantization data; and
[1.12.DEC]	an inverse-orthogonal transformer to perform inverse-orthogonal transform to the post-inverse-quantization data, thus producing a decoded residual picture.

Claim 2

[2.1]	The moving-picture coding apparatus according to claim 1 wherein the predictive encoder further includes: a motion estimator to perform motion-vector search in the reference picture by block matching for each rectangular zone in the picture to be coded, thus generating motion vector data that indicates a matched rectangular zone in the reference picture;
[2.2]	a motion compensator to find a corresponding rectangular zone in the reference picture based on the motion-vector data, thus producing a second predictive picture;
[2.3]	a second subtractor to produce a second residual picture that is a difference between the second predictive picture and the picture to be coded; and
[2.4]	a residual determiner to compare the first predictive picture and the second predictive picture based on specific criteria, thus selecting the predictive picture having a smaller data amount.

Claim 3

[3.1]	The moving-picture coding apparatus according to claim 1 wherein the predictive encoder includes an entropy encoder to perform entropy encoding, at least, to the post-quantization data and the border motion-vector data, thus generating coded bitstreams,
[3.2]	wherein the moving-picture coding apparatus comprises a multiplexer to multiplex the coded bitstreams based on a specific syntax structure.

Claim 4	
[4.pre]	A non-transitory computer readable device having stored thereon a computer program comprising a set of instructions when executed by a computer to implement a method for moving-picture coding, the program comprising:
[4.1.COD]	a predictive encoding program code to produce and encode a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided;
[4.2.COD]	a zone-border motion estimation program code to obtain a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones,
[4.3.COD]	find a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[4.4.COD]	and generate border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found; and
[4.5.COD]	a zone-border motion compensation program code to define a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data,
[4.6.COD]	and generate an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture,
[4.7.COD]	wherein the predictive encoding program code produces the residual picture with the first predictive picture as the predictive picture and encodes the residual picture;
[4.8.COD]	said predictive encoding program code including a first subtraction program code to produce a first residual picture from a difference between the picture to be coded and the first predictive picture;

[4.9.COD]	an orthogonal transform program code to perform orthogonal transform to the first residual picture, thus generating orthogonal-transform coefficients data;
[4.10.COD]	a quantization program code to perform quantization to the orthogonal-transform coefficients data based on a specific quantization parameter, thus generating post-quantization data;
[4.11.DEC]	an inverse-quantization program code to perform inverse-quantization to the post-quantization data based on a specific quantization parameter, thus generating post-inverse-quantization data;
[4.12.DEC]	an inverse-orthogonal transform program code to perform inverse-orthogonal transform to the post-inverse-quantization data, thus producing a decoded residual picture; and
[4.13.COD]	an entropy coding program code to perform entropy coding, at least, to the post-quantization data and the border motion-vector data, thus generating coded bit strings,
[4.14.COD]	wherein the moving-picture coding program further comprises multiplex program code to multiplex the coded bit strings based on a specific syntax structure, thus generating a coded bitstream.
Claim 5	
[5.1]	The moving-picture coding program code according to claim 4 wherein the predictive encoding program code further includes: a motion estimation program code to perform motion-vector search in the reference picture by block matching for each rectangular zone in the picture to be coded, thus generating motion-vector data that indicates a matched rectangular zone in the reference picture;
[5.2]	a motion compensation program code to find a corresponding rectangular zone in the reference picture based on the motion-vector data, thus producing a second predictive picture;
[5.3]	a second subtraction program code to produce a second residual picture that is a difference between the second predictive picture and the picture to be coded; and
[5.4]	a selection program code to compare the first predictive picture and the second predictive picture based on specific criteria, thus selecting the predictive picture having a smaller data amount,

[5.5]	wherein the moving-picture coding program code further comprises a coding control program code to perform control to supply the selected residual picture and a predictive picture corresponding to the residual picture to the predictive encoding program code.
Claim 6	
[6.pre]	A moving-picture decoding apparatus comprising:
[6.1.DEC]	a demultiplexer to demultiplex coded data from an input signal based on a specific syntax structure,
[6.2.COD]	the input signal being obtained by multiplexing a coded bitstream obtained by predictive coding, border motion-vector data and post-quantization data obtained by quantization in the predictive coding,
[6.3.COD]	the coded bitstream obtained by producing and encoding a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided,
[6.4.COD]	obtaining a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones,
[6.5.COD]	finding a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[6.6.COD]	and generating the border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found, defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion vector data,
[6.7.COD]	and generating an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing the predictive picture;

[6.8.DEC]	an entropy decoder to perform entropy decoding to the data thus demultiplexed to generate, at least, motion-vector data, the post-quantization data, the border motion-vector data and parameter data required for constructing a specific syntax structure;
[6.9.DEC]	an inverse-quantizer to perform inverse-quantization to the post-quantization data to generate post-quantization orthogonal transform coefficients data;
[6.10.DEC]	an inverse-orthogonal transformer to perform inverse-orthogonal transform to the post-quantization orthogonal transform coefficients data to produce a decoded residual picture of one video area;
[6.11.DEC]	a zone-border motion compensator to define a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data, and generate an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture;
[6.12.DEC]	a combiner to combine the first predictive picture and the decoded residual picture to generate a decoded moving picture signal;
[6.13.DEC]	a memory to store the decoded moving-picture signal for at least one picture as a reference picture;
[6.14.DEC]	a motion compensator to specify a corresponding rectangular zone in the reference picture based on the motion vector data, thus generating a second predictive picture;
[6.15.DEC]	a selector to select either the first predictive picture or the second predictive picture and supply the predictive picture thus selected to the combiner; and
[6.16.DEC]	a decoding controller to receive decoding control data for decoding control from the parameter data and control the selector to switch the predictive picture to be supplied to the combiner between the first and second predictive pictures according to the decoding control data.
Claim 7	
[7.pre]	A non-transitory computer readable device having stored thereon a computer program comprising a set of instructions when executed by a computer to implement a method for moving-picture decoding, the program comprising:

[7.1.DEC]	a demultiplex program code to demultiplex coded data from an input signal based on a specific syntax structure,
[7.2.COD]	the input signal being obtained by multiplexing a coded bitstream obtained by predictive coding, border motion vector data and post-quantization data obtained by quantization in the predictive coding,
[7.3.COD]	the coded bitstream obtained by producing and encoding a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding, and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided,
[7.4.COD]	obtaining a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones,
[7.5.COD]	finding a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[7.6.COD]	and generating the border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found, defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion vector data,
[7.7.COD]	and generating an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing the predictive picture;
[7.8.DEC]	an entropy decoding program code to perform entropy decoding to the data thus demultiplexed to generate, at least, the post-quantization data, the border motion-vector data and parameter data required for constructing a specific syntax structure;
[7.9.DEC]	an inverse-quantization program code to perform inverse-quantization to the post-quantization data to generate post-quantization orthogonal transform coefficients data;
[7.10.DEC]	an inverse-orthogonal transform program code to perform inverse-orthogonal transform to the post-quantization orthogonal

	transform coefficients data to produce a decoded residual picture of one video area;
[7.11.DEC]	a zone-border motion compensation program code to define a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data, and generate an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture; and
[7.12.DEC]	a combine program code to combine the first predictive picture and the decoded residual picture to generate a decoded moving-picture signal, the decoded moving-picture signal for at least one picture being stored as a reference picture.
Claim 8	
[8.pre]	The moving-picture decoding program according to claim 7 wherein the entropy decoding program code further generates motion-vector data, the program further comprises:
[8.1]	a motion compensation program code to specify a corresponding rectangular zone in the reference picture based on the motion-vector data, thus generating a second predictive picture;
[8.2]	a selection program code to select either the first predictive picture or the second predictive picture and supply the predictive picture thus selected to the combiner; and
[8.3]	a decoding controlling program code to receive decoding control data for decoding control from the parameter data and control the selector to switch the predictive picture to be supplied to the combiner between the first and second predictive pictures according to the decoding control data.
Claim 9	
[9.pre]	A moving-picture coding method to be implemented in a moving-picture coding apparatus comprising the steps of:
[9.1.COD]	producing and encoding a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each

	composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided;
[9.2.COD]	obtaining a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones
[9.3.COD]	finding a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[9.4.COD]	and generating border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found; defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data,
[9.5.COD]	and generating an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture;
[9.6.COD]	producing a first residual picture from a difference between the picture to be coded and the first predictive picture;
[9.7.COD]	performing orthogonal transform to the first residual picture, thus generating orthogonal-transform coefficients data;
[9.8.COD]	performing quantization to the orthogonal-transform coefficients data based on a specific quantization parameter, thus generating post-quantization data;
[9.9.DEC]	performing inverse-quantization to the post-quantization data based on a specific quantization parameter, thus generating post-inverse-quantization data; and
[9.10.DEC]	performing inverse-orthogonal transform to the post-inverse quantization data, thus producing a decoded residual picture.
Claim 10	
[10.pre]	A moving-picture decoding method to be implemented in a moving-picture coding apparatus comprising the steps of:
[10.1.DEC]	demultiplexing coded data from an input signal based on a specific syntax structure,
[10.2.COD]	the input signal being obtained by multiplexing a coded bitstream obtained by predictive coding, border motion-vector data and post-quantization data obtained by quantization in the predictive coding,

[10.3.COD]	the coded bitstream obtained by producing and encoding a residual picture that is a residual signal between a picture to be coded that is an input moving-picture video signal to be subjected to coding, and a predictive picture produced from a reference picture that is a local decoded video signal for each of a plurality of rectangular zones, each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided,
[10.4.COD]	obtaining a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones,
[10.5.COD]	finding a border, of the reference picture, having a boundary condition that matches the boundary condition, by motion-vector search in the reference picture,
[10.6.COD]	and generating the border motion-vector data that is data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found, defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data,
[10.7.COD]	and generating an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing the predictive picture;
[10.8.DEC]	performing entropy decoding to the data thus demultiplexed to generate, at least, the post-quantization data, the border motion-vector data and parameter data required for constructing a specific syntax structure;
[10.9.DEC]	performing inverse-quantization to the post-quantization data to generate post-quantization orthogonal transform coefficients data;
[10.10.DEC]	performing inverse-orthogonal transform to the post-quantization orthogonal transform coefficients data to produce a decoded residual picture of one video area;
[10.11.DEC]	defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion-vector data, and generate an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture;

[10.12.DEC]	combining the first predictive picture and the decoded residual picture to generate a decoded moving-picture signal; and
[10.13.DEC]	storing the decoded moving-picture signal for at least one picture as a reference picture.

Apple Inc. (“Petitioner” or “Apple”) petitions for *Inter Partes* Review (“IPR”) of claims 1-10 (“the Challenged Claims”) of U.S. Patent No. 8,090,025 (“the ’025 Patent”). Compelling evidence presented in this Petition demonstrates that Apple will prevail.

I. REQUIREMENTS FOR IPR

A. Grounds for Standing

Petitioner certifies that the ’025 Patent is available for IPR. This petition is being filed within one year of service of a complaint against Apple. Apple is not barred or estopped from requesting review of the Challenged Claims on the below-identified grounds.

B. Challenge and Relief Requested

Apple requests institution of IPR and cancellation of the Challenged Claims based on the following grounds:

Ground	Claims	35 U.S.C. § 103
1A	1-10	Mualla, Shirani, and Saito
1B	1-10	Mualla, Shirani, Saito, and Stockhammer
2	1-10	Mualla, Shirani, Saito, and Shimizu

The grounds of this petition are further supported by the expert testimony of Mr. David Lett. *See* APPLE-1003, ¶¶[1]-[185].

The ’025 Patent claims priority to two foreign applications, the earliest of

which was filed April 17, 2006. APPLE-1001, 1:9-13. As shown below, each prior art reference that forms the basis of Ground 1 of this Petition pre-dates the '025 Patent's earliest proclaimed priority date (April 17, 2006; "Critical Date"), which Petitioner does not concede. APPLE-1018.

Reference	§102(a)/(b) Publication Date
Mualla	2002
Shirani	June, 2000
Saito	January, 2006
Stockhammer	July, 2003
Shimizu	September 18, 2003

C. Claim Construction

Petitioner submits that no formal claim constructions are necessary because "claim terms need only be construed to the extent necessary to resolve the controversy." *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); APPLE-1003, ¶¶[16]-[22]. Petitioner reserves the right to respond to any constructions offered by Patent Owner or adopted by the Board. Petitioner is not conceding that each challenged claim satisfies all statutory requirements, nor is Petitioner waiving any arguments concerning claim scope or grounds that can only be raised in district court. For this petition, Petitioner applies prior art in a manner consistent with Patent Owner's allegations of infringement before the district court.

D. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’025 Patent as of April 17, 2006 would have had (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience in digital video processing, or a related field. APPLE-1003, ¶¶[23]-[24]. Additional graduate education could substitute for professional experience, and *vice versa*. *Id.*

II. THE ’025 PATENT

A. Brief Description

The ’025 Patent is directed to “a moving-picture coding apparatus, method and program, and a moving-picture decoding apparatus, method and program that, without a smoothing procedure, can produce a predictive picture maintaining continuity of a video signal.” APPLE-1001, 2:30-38. The ’025 Patent discloses a “controller” (117a and 313a) and a “switch” (102/103 and 701) for selecting between “ordinary” motion estimation/compensation and “zone-border” motion estimation/compensation.” APPLE-1001, 25:21-42, 26:24-48, 28:21-39, FIGS. 5, 7. The ’025 Patent discloses that these two modes of “ordinary” and “zone-border” motion estimation/compensation are “adaptively switched based on the data amount, thus producing high-quality decoded videos with a fewer code amount overall.” APPLE-1001, 28:21-39. APPLE-1003, ¶[26].

The “ordinary” motion estimation/compensation is performed by a “motion estimator” and a “motion compensator,” which the ’025 Patent describes in its FIG. 13 description. APPLE-1001, 10:4-39, 21:63-22:14, 28:21-39, FIGS. 5, 7, 13. APPLE-1003, ¶[27].

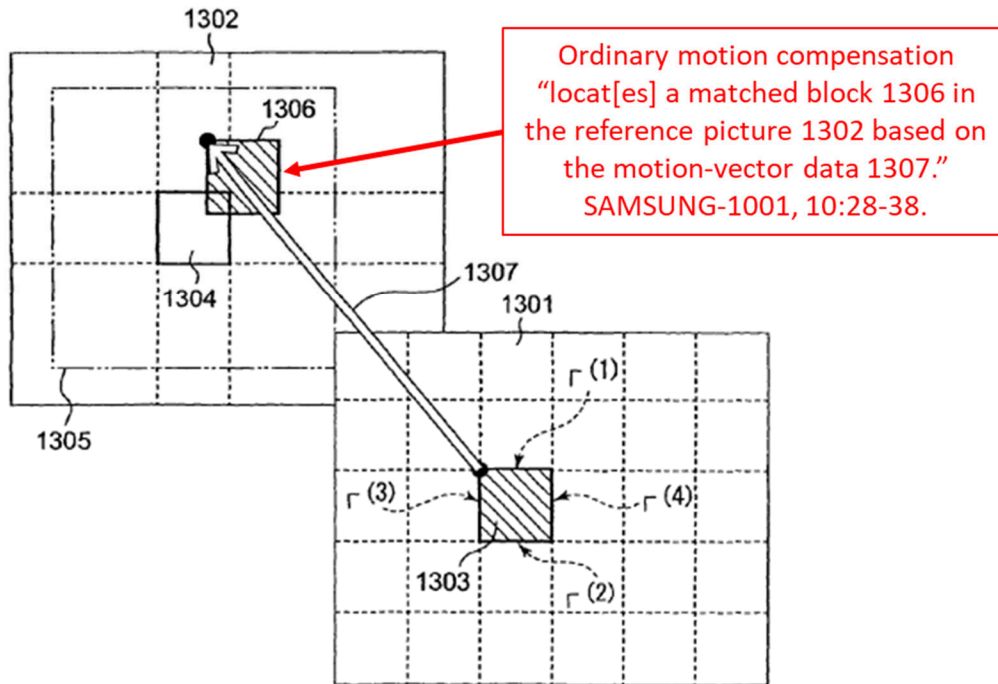


FIG. 13

APPLE-1001, FIG. 13 (annotated).

The “zone-border” motion estimation/compensation includes two steps.

APPLE-1001, 10:39-12:20, FIGS. 14-17. In the first step, “motion-vector search is performed on block borders.” APPLE-1001, 10:39-45, FIGS. 14-15. Block borders are associated with a “boundary condition” which is “a gradient of a video signal on a block border.” APPLE-1001, 10:46-56. The zone-border motion estimator “obtains border gradient data that is a boundary condition of a block border in a picture to be coded” and then selects the “closest gradient to the gradient of an original signal on the block border.” APPLE-1001, 10:57-65, FIGS. 14-15 (reproduced below). In the second step, “Poisson's Equation is then applied based on the gradient data of the video signal thus defined to generate a predictive signal in block which is then applied to a predictive block 1606a.” APPLE-1001, 11:20-34, FIGS. 16-17 (reproduced below). APPLE-1003, ¶[28].

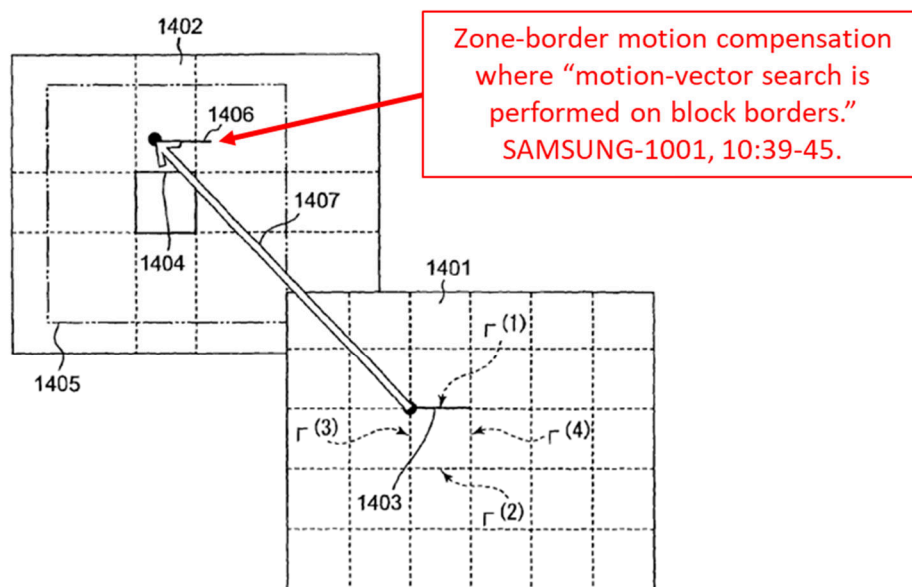


FIG. 14

APPLE-1001, FIG. 14 (annotated).

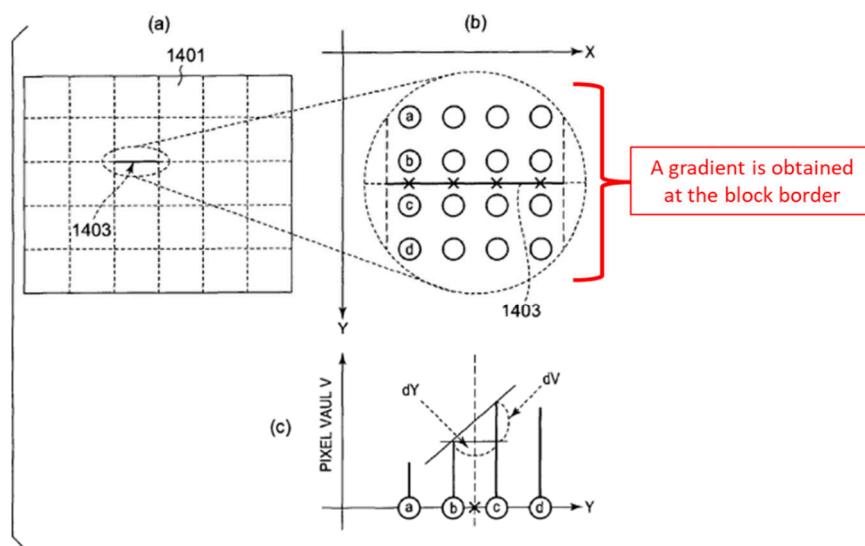
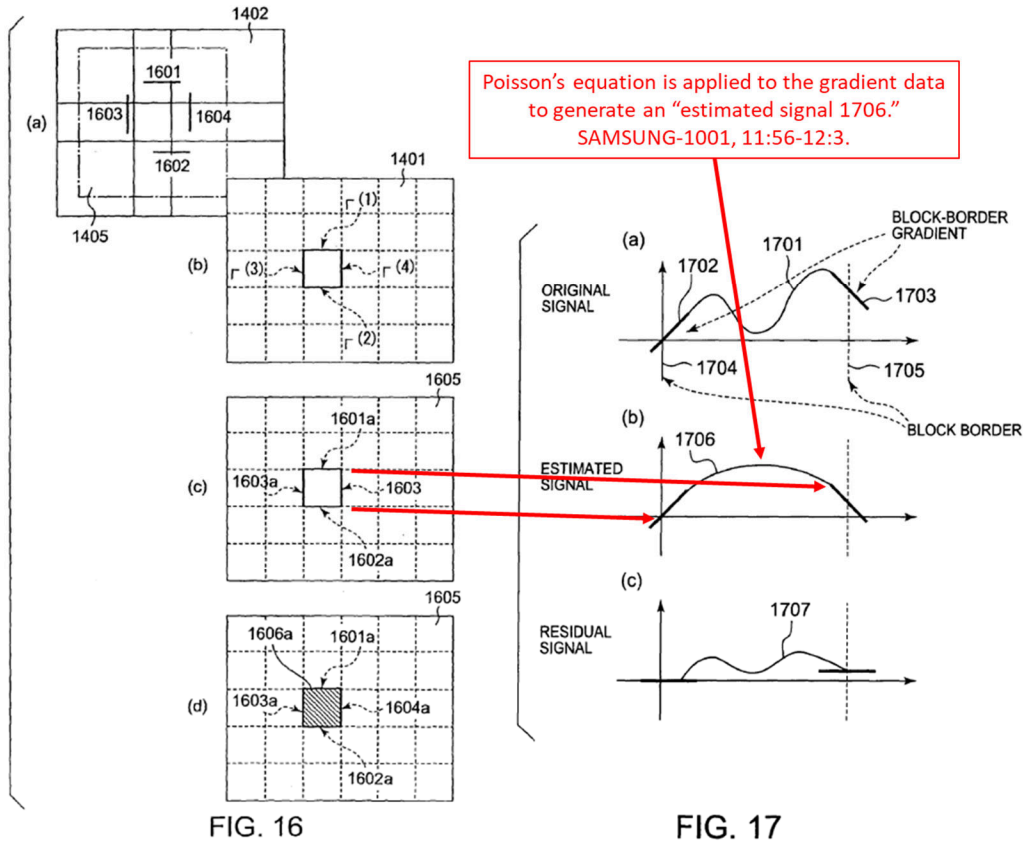


FIG. 15

APPLE-1001, FIG. 15 (annotated).



APPLE 1001, FIGS. 16-17 (annotated).

B. Prosecution History

The Examiner for the '025 Patent did not issue any prior art rejections and there is no indication that the Examiner considered Mualla, Shirani, Saito, Stockhammer, or Shimizu, which render the challenged claims obvious, as discussed below. *See generally*, APPLE-1002; APPLE-1003, ¶[29].

III. THE CHALLENGED CLAIMS ARE UNPATENTABLE

Grounds 1A/1B of the petition describe switching between conventional Motion Estimation/Motion Compensation (ME/MC) and error-concealment to correct coding and decoding errors, whereas Ground 2 describes both error-prone

and error-free scenarios, and how switching between conventional ME/MC and the hybrid 2-stage motion compensated prediction of Ground 1A/1B (in view of the additional disclosure of Shimizu) would have provided benefits in both scenarios. APPLE-1003, ¶¶[30]-[31].

A. [GROUND 1A/1B] – Claims 1-10 are rendered obvious by Mualla-Shirani-Saito and claims 1-10 are rendered obvious by Mualla-Shirani-Saito-Stockhammer

In ground [1A], the Petition relies on the disclosure of Mualla-Shirani-Saito and explains why this disclosure renders obvious claims 1-10. In ground [1B], the Petition relies on the same disclosure of Mualla-Shirani-Saito and explanation of obviousness as presented in ground [1A] but incorporates Stockhammer's disclosure to the extent its teachings would not have been within the general knowledge of a POSITA. APPLE-1003, ¶¶[30]-[31].

1. Mualla (and Mualla-Stockhammer in Ground 1B)

Mualla is a treatise on “fundamentals and standards of video source coding” and addresses “areas of coding efficiency, computational complexity, and error resilience in a single volume.” APPLE-1005, 9-10; APPLE-1003, ¶[32].

Mualla describes two types of statistical redundancy in video coding and decoding, namely “**spatial** (*intraframe*) and **temporal** (*interframe*) correlations between neighboring pels.” *Id.* **Spatial** (*intraframe*) redundancy is reduced by “intraframe coding” which can be implemented by “**transform coding**.” APPLE-

1005, 50-51. **Temporal** (interframe) redundancy can be reduced by “interframe coding” which can be implemented by “**motion estimation (ME)**” and “**motion compensation (MC)**.” APPLE-1005, 60-61. Mr. Lett provides a detailed explanation and summary of Mualla in his declaration, including a discussion of Mualla’s traditional ME and MC techniques. APPLE-1003, ¶¶[33]-[51].

Error Resilience Techniques

In addition to the conventional video coding and decoding operations discussed above, Mualla further describes techniques for “error resilience” in video coding, including “error detection and error control techniques” applicable to the H.263 standard. APPLE-1005, 11, 227-251. Mualla explains that “when transmitted over a mobile channel, compressed video can suffer severe degradation and the use of error-resilience techniques is vital.” APPLE-1005, 25; APPLE-1003, ¶[52].

To implement error resilience, Mualla describes a system in which “[t]he decoder was modified to perform error detection by detecting syntax and semantic violations.” APPLE-1005, 265. In particular, “[w]hen an error is detected, the decoding process is stopped” and “[a]ll macroblocks between the point where the error was detected and the synchronization point are marked as damaged macroblocks.” *Id.* The damaged macroblocks are recovered by using an error control technique called “**hybrid** error concealment” in which “**temporal**

concealment is first used to get an initial estimate of the damaged block” and “this initial estimate is then refined using **spatial** concealment.” APPLE-1005, 242-246, Figures 9.7, 9.8 (reproduced below); APPLE-1003, ¶[53].

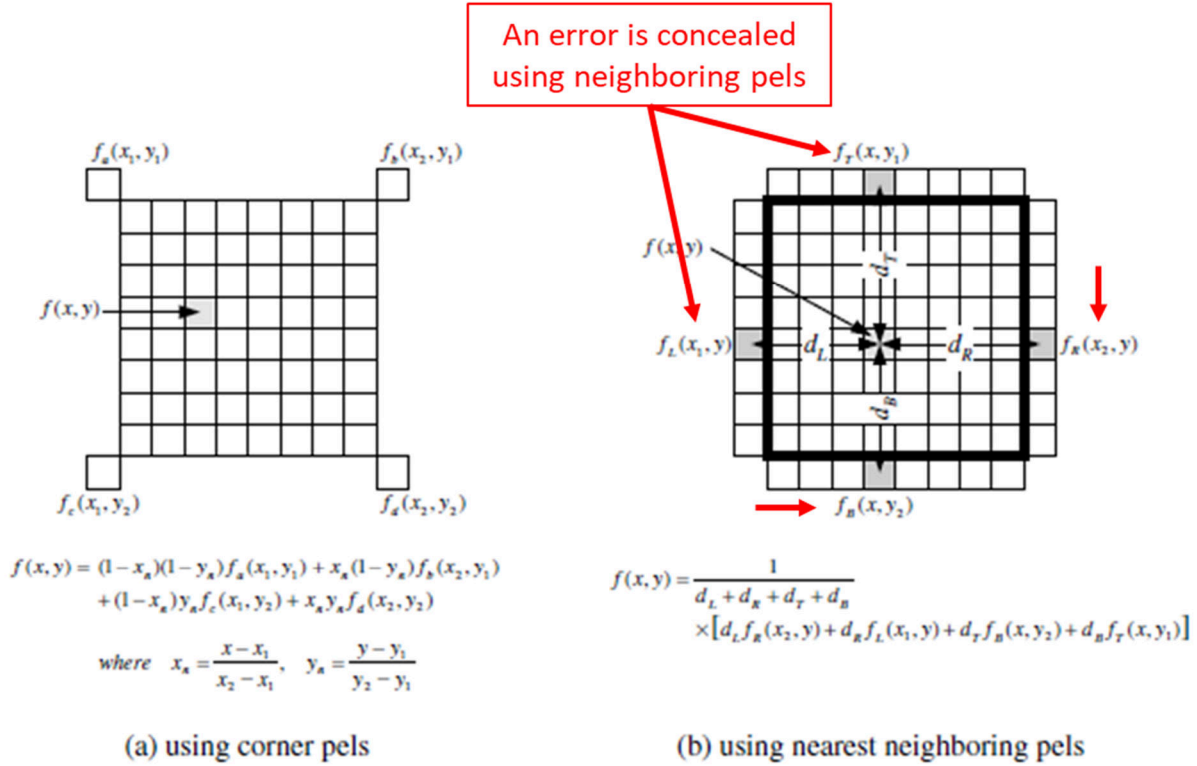


Figure 9.7: Error concealment using spatial interpolation

APPLE-1005, FIG. 9.7 (annotated).

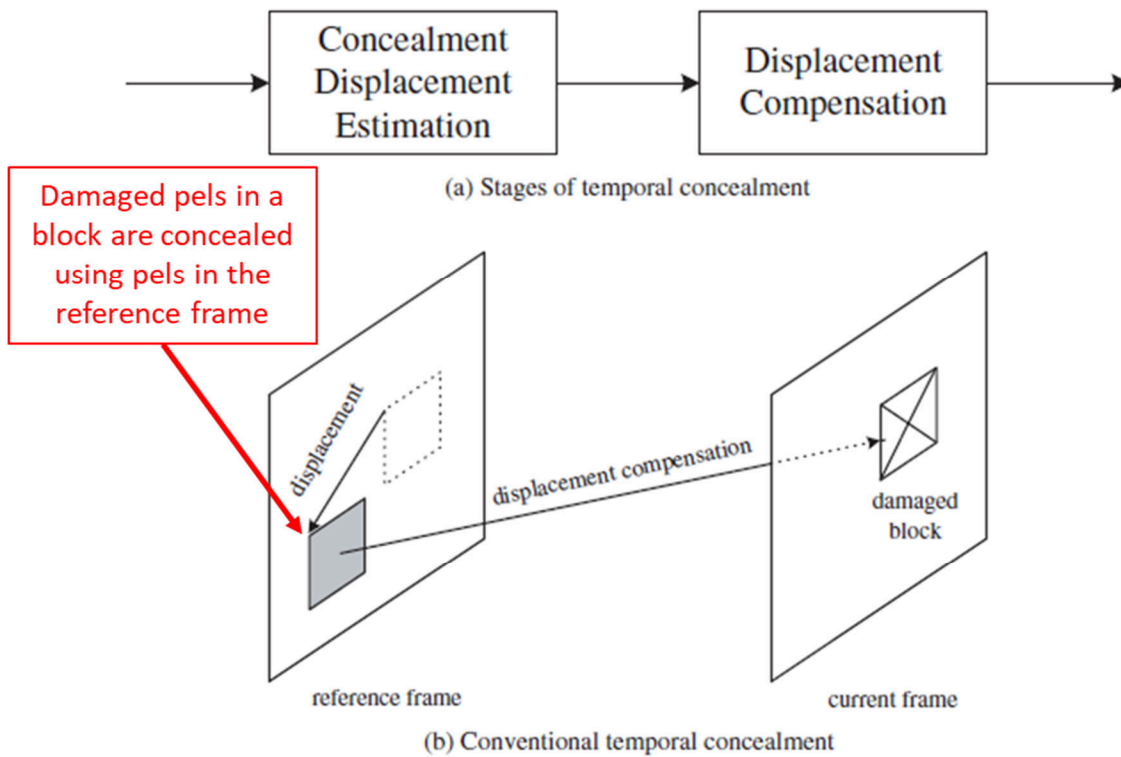


Figure 9.8: Temporal error concealment

APPLE-1005, FIG. 9.8 (annotated).

Furthermore, Mualla describes that error concealment can be performed at **both** the **encoder** and **decoder**. APPLE-1005, 246-250. For example, “the encoder and decoder cooperate to minimize the effects of transmission errors” and in particular “coding efficiency can also be improved by **performing error concealment** in the **encoder** so that both encoder and decoder use the same reference frames for prediction.” APPLE-1005, 249. To implement error concealment at the encoder, Mualla describes “**interactive techniques**” in which “the decoder uses a feedback channel to inform the encoder about which parts of the transmitted video have been received in error” and “[b]ased on this feedback

information, the encoder adjusts its operation to combat the effects of such errors.” APPLE-1005, 224. Mualla explains that “[w]hen feedback information is received, the encoder can reconstruct the error propagation process” so that “the encoder can track the error propagation.” APPLE-1005, 247; APPLE-1003, ¶[54]. Mualla also cites to numerous corroborating references, such as Girod (APPLE-1008) and Wada (APPLE-1017), for additional details of implementing “interactive” error concealment at both the encoder and decoder. APPLE-1005, 246-250. Mr. Lett provides a summary of these corroborating references. APPLE-1003, ¶¶[54]-[55].

Therefore, in view of the teachings of Mualla, a POSITA would have understood or found obvious that (i) an H.263-based **decoder** performs *error detection* to detect errors and *error concealment* to recover from those errors, and (ii) an H.263-based **encoder** receives feedback from the decoder regarding errors that were detected, and performs the same *error concealment* to replicate those same errors in its local decoder so as to use the same reference frames for error prediction. APPLE-1003, ¶[56].

Furthermore, Stockhammer provides illustrative diagrams for implementing feedback-based error detection and error correction in an H.264-based system. APPLE-1009, 8, Fig. 6 (reproduced below). Fig. 6 of Stockhammer illustrates an encoder that implements a local “decoder” to perform “video encoding” to

generate “packets.” *Id.* The packets are transmitted over a “packet loss channel” in which packets are randomly lost during transmission. *Id.* At the remote decoder, “[t]he decoder processes the received sequence of packets” such that “received packets are decoded as usual” whereas “for the lost packet, an error-concealment algorithm has to be invoked.” APPLE-1009, 8; APPLE-1003, ¶[57]. Furthermore, a “back-channel from the decoder to the encoder is available which allows reporting ... of the observed channel behavior at the decoder to the encoder.” *Id.*

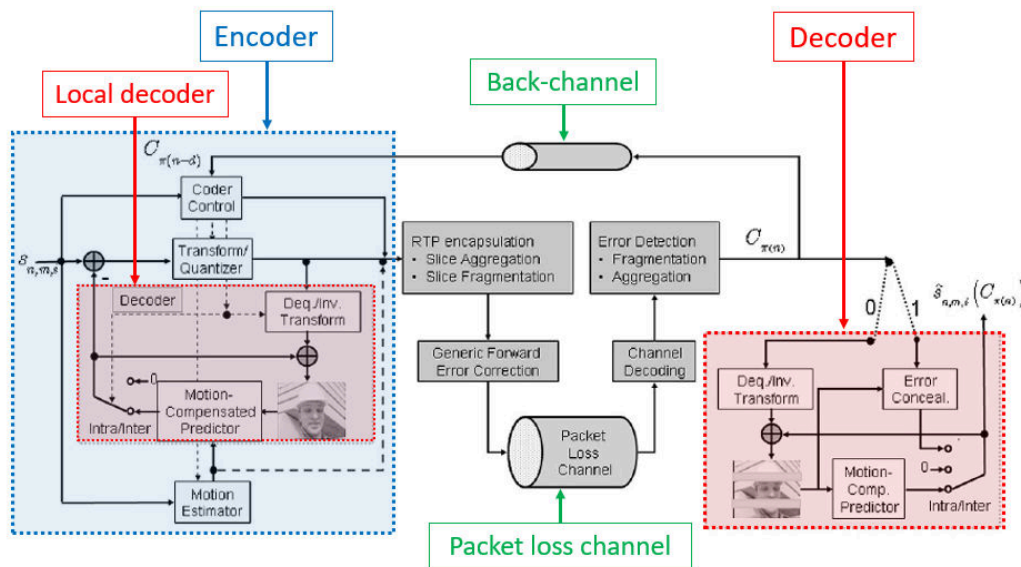


Fig. 6. H.264/AVC in IP-based packet-lossy environment with RTP encapsulation, generic forward error correction, delayed feedback information, and error concealment.

APPLE-1009, FIG. 6 (annotated and color added).

To the extent the implementation details of Stockhammer were not already understood to a POSITA with knowledge of error concealment techniques, a POSITA reviewing the teachings of Stockhammer would have also been motivated

to combine these same implementation details into the teachings of Mualla.

APPLE-1009, 8-11; APPLE-1003, ¶¶[58]-[59]. Indeed, as Mr. Lett explains, “although the techniques of Stockhammer were well understood, a POSITA would have naturally searched for specific implementation examples (e.g., Stockhammer’s FIG. 6 above) to refine the broader disclosure of Mualla.” *Id.*

To summarize, Mualla provides a self-contained teaching of video coding systems, including feedback-based error tracking in which (i) an H.263-based **decoder** performs *error detection* to detect errors and *error concealment* to recover from those errors, and (ii) an H.263-based **encoder** receives feedback from the decoder regarding errors that were detected, and performs the same *error concealment* to replicate those same errors in its local decoder so as to use the same reference frames for error prediction. Mualla refers to (i) as a “post-processing” technique and refers to (ii) as an “interactive” technique, and Mualla explains that a “combination” of these techniques can be employed in practical systems. APPLE-1005, 251; APPLE-1003, ¶[60]. Therefore, a POSITA would have understood or found obvious that feedback-based error resilience at both the encoder and decoder can be implemented by a system such as the one shown in Diagram C, below. APPLE-1009, Fig. 6; APPLE-1003, ¶¶[60]-[62]. In this diagram, Figure 6 of Stockhammer has been modified to show an example of how error concealment would be implemented at the encoder, based on the teachings of

Mualla. APPLE-1009, Fig. 6; APPLE-1003, ¶[60]. The modified portions of Stockhammer's Fig. 6 are annotated and shaded in red in Diagram C. APPLE-1009, Fig. 6; APPLE-1003, ¶[60]. As shown, the modification simply amounts to replacing the encoder's local "decoder" with an analogous error concealment-enabled structure as already implemented in the remote decoder. APPLE-1009, Fig. 6; APPLE-1003, ¶[60].

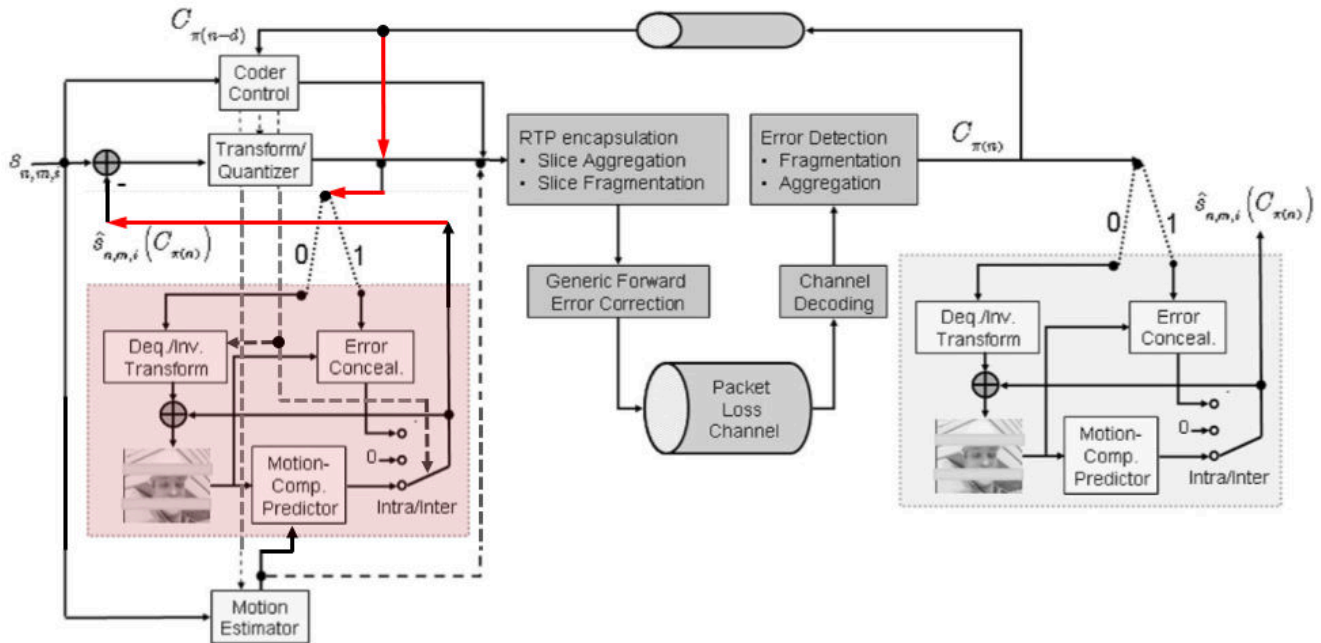


Diagram C (video coding system with interactive error concealment)

Diagram D, below, illustrates a detailed view of the modified encoder of Diagram C. APPLE-1009, Fig. 6; APPLE-1003, ¶[63]. As discussed previously, the modified encoder implements a local decoder (shaded in red) that performs error concealment. APPLE-1009, Fig. 6; APPLE-1003, ¶[63].

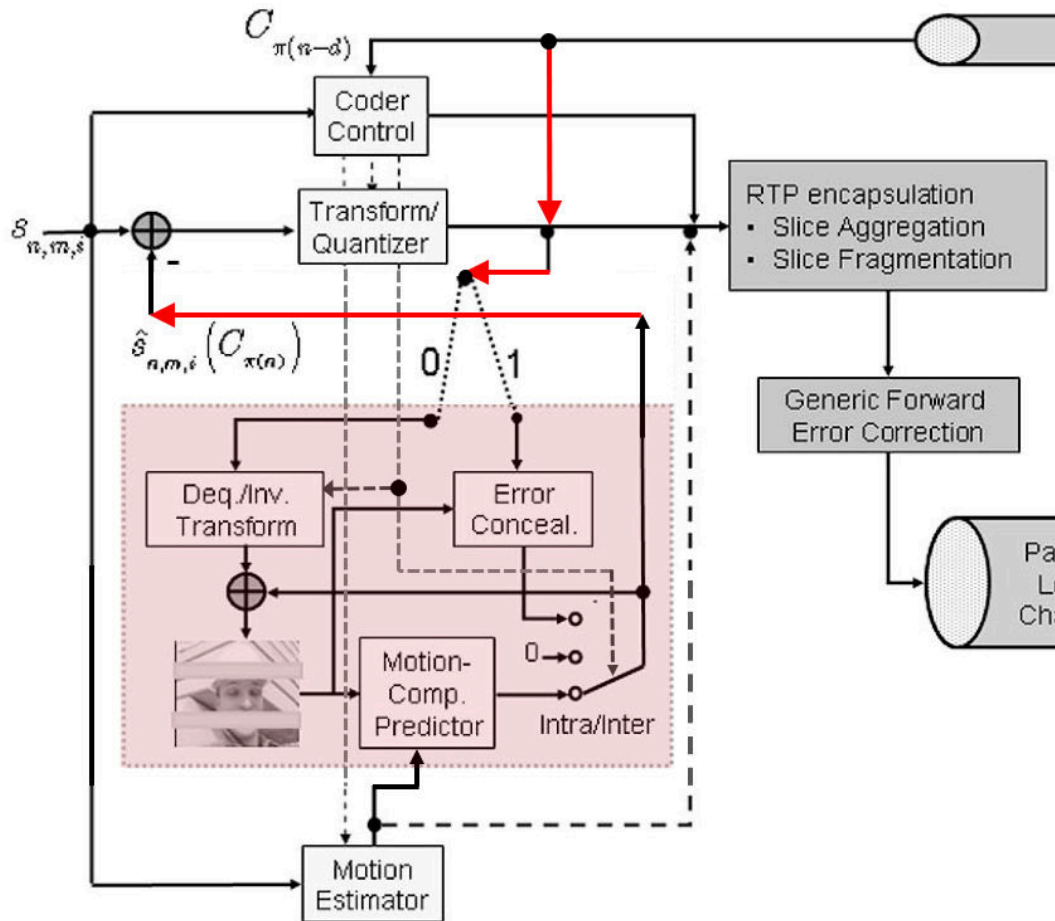


Diagram D (detailed view of the modified encoder of Diagram C)

The modified encoder receives feedback (annotated red line in the upper part of Diagram D) via the “back-channel” regarding blocks that were received in error at the remote decoder (not shown in Diagram D, refer to Diagram C). APPLE-1009, 8, Fig. 6; APPLE-1003, ¶[64]. Based on this feedback, a selector (the binary selector shown above the red shaded area in Diagram D) selects between “0” (if the feedback indicates no error in a macroblock) or “1” (if the feedback indicates error in a macroblock). APPLE-1009, 8, Fig. 6; APPLE-1003, ¶[64]. If the

feedback indicates no error (“0”), then the modified encoder performs the conventional steps of dequantization and inverse transform (upper-left-hand block in the red shaded area of Diagram D) with “motion compensated-predictor” (lower block in the red shaded area of Diagram D) and the second selector at the bottom-right-hand corner of the red shaded area is in the lower-most position. APPLE-1009, 8, Fig. 6; APPLE-1003, ¶[64]. Otherwise, if the feedback indicates an error (“1”), then the modified encoder performs “error concealment” (upper-right-hand block in the red shaded area of Diagram D) and the second selector at the bottom-right-hand corner of the red shaded area is in the upper-most position. APPLE-1009, 8, Fig. 6; APPLE-1003, ¶[64].

Specifically, in case of error (“1”), POSITA would have understood or found obvious, based on the technique of “error tracking” as described in Mualla (as corroborated by Wada) that “the encoder can track the error propagation from the original occurrence up to the current frame” (APPLE-1005, 247) by using a method in which “[t]he same error concealment as is performed at the receiver is carried out in the local-decoded frame stored” and then “[l]ocal decoding is reexecuted from the point of the concealed block(s) up to the currently encoded block” (APPLE-1017, 2); APPLE-1003, ¶[65].

In both scenarios of the feedback indicating error and no error, the encoder’s local “decoder” generates a “reconstructed image” (i.e., the aggregate signal

$\hat{s}_{n,m,i}(o, C_{\pi(n)})$ that labels the red annotated line on the left-hand side of Diagram D) which is provided to a subtractor (left-hand side of Diagram D) which subtracts the reconstructed image from the input image (i.e., the input signal $s_{n,m,i}$). APPLE-1009, Fig. 6; APPLE-1003, ¶[66]. This is consistent with the teachings of Mualla for conventional encoding (APPLE-1005, Figure 1.1, discussed above) where the encoder's local decoder generates a "prediction of the input frame" (the reconstructed image) is subtracted from the input frame (the input image).

Therefore, in view of the teachings of Mualla (as corroborated by Girod, Wada, and Stockhammer), a POSITA would have understood or found obvious that an error resilience technique (such as the "hybrid error concealment" technique described in Mualla) can be implemented at both the encoder and the decoder in a feedback-based video coding system. APPLE-1003, ¶[67].

2. Shirani

Shirani is cited in Mualla (as reference [199]) for teaching details of the "hybrid error concealment" technique that was described in Mualla. APPLE-1005, 246. Specifically, Shirani teaches a "two-stage error-concealment method for block-based compressed video which was transmitted in an error-prone environment" in an H.263 video coding framework. APPLE-1006, Abstract; 4. Shirani's "first stage" "obtain[s] an initial estimate of the missing block using information from the previous frame," which Shirani refers to as "temporal"

concealment (consistent with Mualla). APPLE-1005, 243-245; APPLE-1006, 1-2. In Shirani's "**second** stage," "the information around each missing block is used to refine the initial estimate," which Shirani refers to as "**spatial**" concealment (consistent with Mualla). APPLE-1005, 242-243; APPLE-1006, 1-3; APPLE-1003, ¶[68].

The **first stage** (temporal error concealment) utilizes motion vectors of "subblocks adjacent to the missing block" (subblocks "adjacent above and left" as well as "subblocks ... to the right and below of the missing block") to derive an "initial estimate" of the missing block that has the smallest sum of "border errors." APPLE-1006, 3. Shirani illustrates an example of "subblocks adjacent to the missing block, i.e., $u1$, $u2$, $l1$, and $l2$ in Fig. 1(a)." APPLE-1006, 3, FIG. 1(a); APPLE-1003, ¶[69].

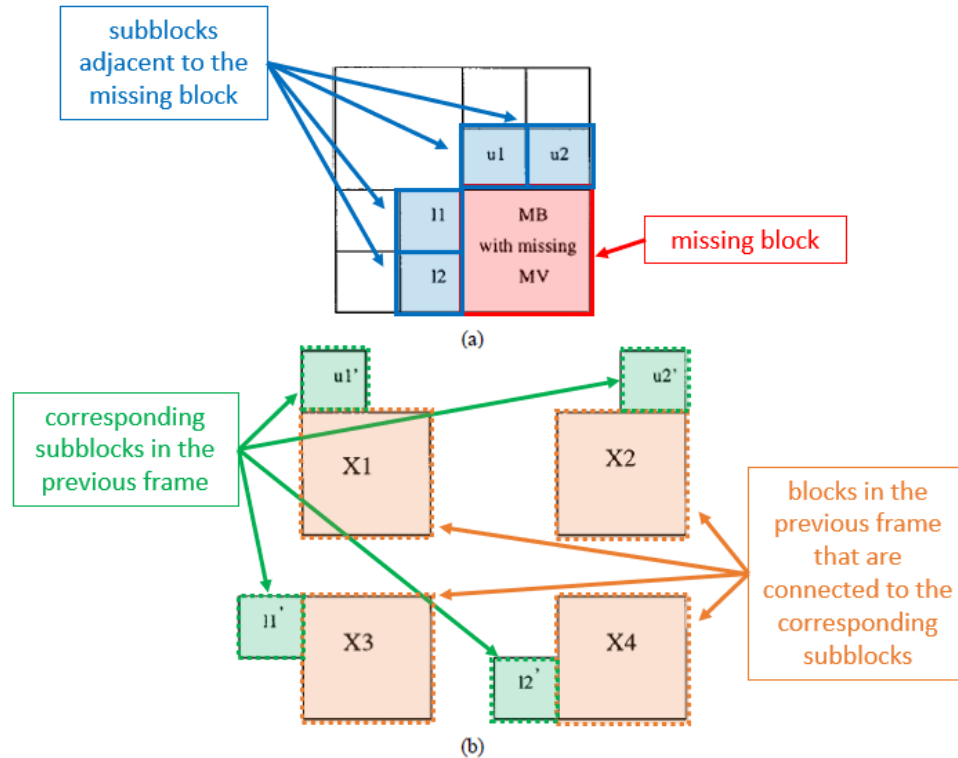


Fig. 1. (a) The four subblocks, (b) their corresponding subblocks in the previous frame, and the blocks connected to them.

APPLE-1006, FIG. 1(a) (annotated and color added).

Once these adjacent subblocks ($u1$, $u2$, $l1$, $l2$) are identified, “for each of these subblocks, a corresponding subblock in the previous frame is determined” (denoted $u1'$, $u2'$, $l1'$, and $l2'$ at the bottom of FIG. 1(a) of Shirani). APPLE-1006, 3. Then, “four blocks, namely X1, X2, X3, and X4, which are connected to $u1'$, $u2'$, $l1'$, and $l2'$, and (respectively), are determined” and “a block from the above four blocks that minimizes the squared sums of border errors, between the estimated block and its adjacent above and left blocks, is selected.” APPLE-1006, 3; APPLE-1003, ¶[70]. This is accomplished by defining “border error[s]” in “terms of pixels,” including an error for the “bottom line pixels of the block to the

top of the missing block” and an error for the “right column pixels of the block to the left of the missing block” and applying the same technique to define border errors for pixels adjacent to “the subblocks ... to the right and below of the missing block.” *Id.*

To estimate the motion of the adjacent subblocks, Shirani discloses that “the search for displacement of each of the subblocks is restricted to a set of candidate motion vectors” which can include “the motion vectors of available neighboring blocks, the median of the motion vectors of available neighboring blocks, the average of the motion vectors of available neighboring blocks.” APPLE-1006, 3; APPLE-1003, ¶[71]. Furthermore, although the above description focused on adjacent subblocks above and to the left of the missing block, Shirani also discloses that “the same method can be extended to include the subblocks of the blocks to the right and below of the missing block.” APPLE-1006, 3.

For the **second stage** (spatial error concealment), Shirani describes various techniques that can “refine the initial estimate” (obtained in the first stage, above) by utilizing “information around each missing block.” APPLE-1006, 1, 3. As examples of “spatial error concealment,” Shirani describes techniques in which neighboring pixels around the missing block are used to “refine the initial estimate” that was obtained in the first stage. APPLE-1006, 1, 3; APPLE-1003, ¶[72].

3. Combination of Mualla and Shirani (and the Combination of Mualla-Stockhammer and Shirani in Ground 1B²)

Given Mualla's explicit reference to Shirani, a POSITA would have naturally looked to Shirani and its "two-stage error concealment" technique for further details of how to implement "hybrid error concealment" as described in Mualla. APPLE-1005, 246; APPLE-1006, 1-6; APPLE-1003, ¶[78]. A POSITA implementing the supplemental teachings of Shirani into the existing disclosure of Mualla would have created the Mualla-Shirani combination (and the Mualla-Stockhammer-Shirani combination in Ground 1B). APPLE-1003, ¶[78].

In particular, the hybrid error concealment techniques of the Mualla-Shirani combination perform error detection to detect whether errors exist in a frame (also referred to as a picture). APPLE-1005, 241-246; APPLE-1006, 2-4. If errors are detected in a frame/picture, a two-stage motion-compensated error concealment algorithm is applied to perform both temporal and spatial concealment to conceal the errors and generate a predicted frame. APPLE-1005, 242-246; APPLE-1006, 2-4. Otherwise, if no errors are detected, conventional motion compensation is applied. *Id.* Specifically, as discussed above, the two-stage (hybrid) error

² The subsequent discussion in Sections III.A.3, III.A.5, and III.A.6 focuses on the combination of Mualla with Shirani and Saito, but is similarly applicable to the combination of Mualla-Stockhammer with Shirani and Saito in Ground 1B.

concealment algorithm of the Mualla-Shirani combination implements a “first stage” which applies “temporal” concealment and a “second stage” which applies “spatial” concealment. APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[79].

Furthermore, in Shirani’s “two-stage error concealment” technique, Shirani describes various alternatives for the “second stage” (in which “spatial” concealment is performed). Shirani describes a spatial concealment technique in which “each pixel in a damaged block is interpolated from the corresponding pixels in its four neighboring blocks such that the total squared border error is minimized.” APPLE-1006, 1; APPLE-1003, ¶¶[80]-[81].

As explained below in more detail, combining Mualla and Shirani would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results; and the (2) use of known techniques to improve similar devices in the same way. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421, 82 USPQ2d 1385, 1395-97 (2007); APPLE-1003, ¶[82]. Indeed, Mualla describes a similar hybrid error concealment technique as Shirani, and the combination would have simply involved adding Shirani’s teachings into the compatible teachings of Mualla, thus, for at least this reason, making the Mualla-Shirani combination predictable and foreseeable with a reasonable expectation of success. APPLE-

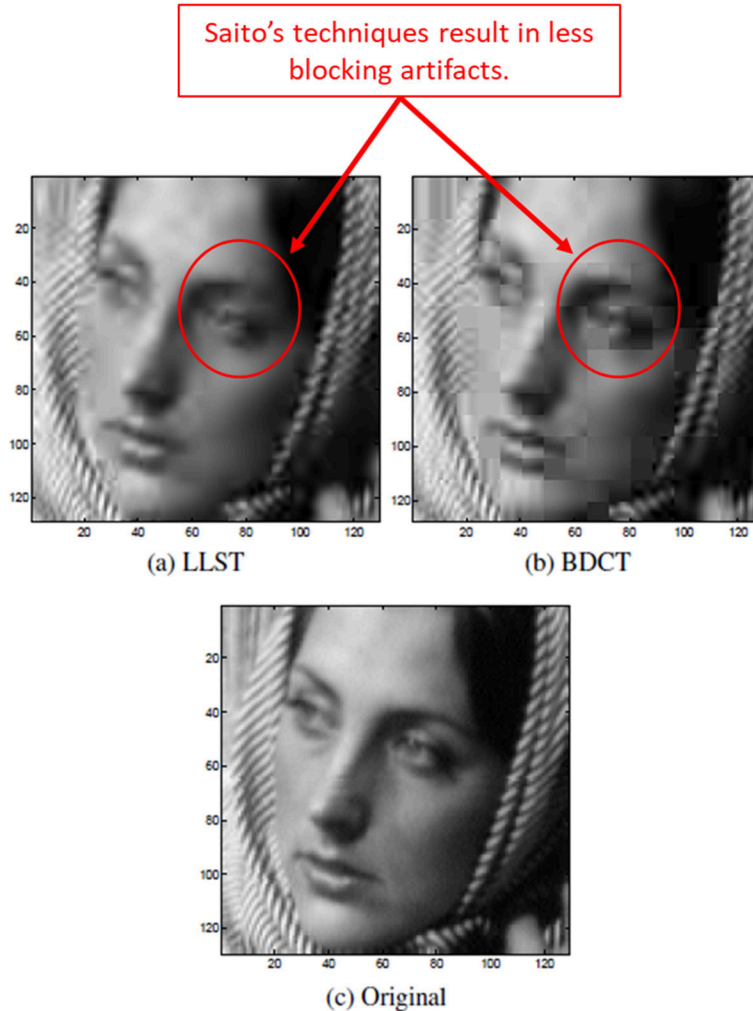
1005, 246; APPLE-1006, 1-6; APPLE-1003, ¶[82]. Moreover, because Mualla explicitly cites to Shirani's teachings when discussing its own techniques of hybrid error concealment, the use of Shirani's techniques in Mualla would have improved a similar device in the same way. APPLE-1005, 246. Additionally, Mualla and Shirani implement their teachings in similar video coding standards (e.g., H.263). APPLE-1005, 71-94, 231; APPLE-1006, 4-5; APPLE-1003, ¶[82].

4. Saito

Saito discloses "image approximation" using techniques based on solving Poisson's equation. APPLE-1007, 6, 27; APPLE-1003, ¶[73]. Saito expresses a goal of improving "the perceptual quality of these approximations" and a desire for a "perceptually correct metric" that can improve the "visual quality" of approximated images. APPLE-1007, 18; APPLE-1003, ¶[73].

To achieve these goals, Saito discloses a "polyharmonic local sine transform" (PHLST) and a "polyharmonic local cosine transform" (PHLCT) that both solve Poisson's equation and are "effective for image approximation." APPLE-1007, 6, 27; APPLE-1003, ¶[74]. Saito explains that "PHLST can represent smooth regions of an image very efficiently" and therefore "image approximation and compression are immediate and important applications of PHLST." APPLE-1007, 8-9. Saito further discloses "extensions and applications of PHLST" which include a version called "polyharmonic local cosine transform

(PHLCT).” APPLE-1005, 27. The PHLCT technique can “solve the ... Poisson equation with the Neumann boundary condition.” *Id.* The PHLCT utilizes “the first order normal derivative of data at the boundary.” *Id.* In particular, the solution to Poisson’s equation has “the same normal derivative at the boundary with the original function” by ensuring that “the residual function v_j ... clearly satisfies $\partial v_j / \partial v = 0$ on $\partial \Omega_j$.” APPLE-1007, 27. Thus, Saito’s PHLCT technique ensures that the derivative of the estimated signal at a boundary of a block is equal to the derivative of the signal to be estimated at the boundary, ensuring smoothness of the estimated image across block boundaries with less “blocking artifacts.” APPLE-1003, ¶[74]; APPLE-1007, 27; *see also* 5, 58, Abstract, FIG 11.



APPLE-1007, FIG. 11 (Cropped and annotated).

5. Combination of Mualla-Shirani and Saito (and the Combination of Mualla-Stockhammer-Shirani and Saito in Ground 1B)

As discussed above, the combination of Mualla and Shirani (and the combination of Mualla-Stockhammer and Shirani in Ground 1B) implements a two-stage (hybrid) error concealment technique. The second stage applies “spatial error-concealment” to “restore the missing blocks using the information in the current frame.” APPLE-1006, 1; *see also* APPLE-1005, 242. Both Mualla and

Shirani disclose that “**spatial**” concealment can be implemented using various techniques from the field of **conventional image-processing**, each with benefits and tradeoffs. APPLE-1005, 242-243; APPLE-1006, 1; APPLE-1003, ¶¶[83]-[84]. Indeed, as confirmed by Mr. Lett, it was well-known to use image-processing techniques to perform spatial concealment within an individual frame of a video. APPLE-1003, ¶¶[83]-[84] (citing APPLE-1011).

Furthermore, a POSITA would have understood or found obvious that image processing can be performed by techniques that **solve Poisson’s equation**. APPLE-1003, ¶[85]. As Mr. Lett explains, numerous research papers and patents had, several years before the Critical Date, described techniques for image editing and image reconstruction that involved solving Poisson’s equation. APPLE-1003, ¶[85] (citing APPLE-1010, APPLE-1012, APPLE-1013, APPLE-1014, and APPLE-1015).

As Mr. Lett explains, a POSITA would have understood or found obvious that image processing techniques involving Poisson’s equation are particularly effective in addressing the same challenges and achieving the same goals emphasized by Mualla and Shirani for error concealment applications. APPLE-1003, ¶[86]. For example, as Mr. Lett explains, Mualla repeatedly emphasizes a fundamental importance of considering the “human visual system (HVS)” in viewing the “subject quality” of video codec techniques, and in particular for error

concealment. APPLE-1005, 33, 36, 40, 52, 55, 59, 242, 251, 265, 268, Figures 10.8 and 10.10; APPLE-1003, ¶[86]. Like Mualla, Shirani also stresses the fundamental importance of considering the “limitations of the **human visual system**” when designing an error concealment technique.” APPLE-1006, 1, 4; APPLE-1003, ¶[86].

In view of Mualla and Shirani emphasizing a fundamental challenge and goal of improving the “subjective quality” of error concealment for the “human visual system,” a POSITA would have understood that an image approximation technique based on **solving Poisson’s equation** is an effective way to achieve this goal. APPLE-1003, ¶[87]. As Mr. Lett explains, “the Poisson equation has been used extensively in field of image processing and computer vision due to the specific benefits it provides for improving the perceptual quality of reconstructing images.” APPLE-1003, ¶[87] (citing APPLE-1010).

Therefore, as discussed above: (i) Mualla and Shirani both recognize that **image processing** techniques can be utilized to implement **second-stage spatial** error concealment in the Mualla-Shirani two-stage hybrid error concealment technique for video frames, (ii) image processing techniques that involve **solving Poisson’s equation** were well-known to provide specific benefits for improving the perceptual quality of reconstructed images for the human visual system, thus addressing a fundamental objective of Mualla and Shirani’s error concealment.

APPLE-1003, ¶[88]. As such, a POSITA would have understood or found obvious that image-processing techniques based on solving Poisson's equation would have been naturally applicable for implementing the second stage "spatial" concealment of the Mualla-Shirani two-stage hybrid error concealment algorithm for video frames. APPLE-1003, ¶[88]. As Mr. Lett further explains, literature in video coding spatial error concealment and literature in image processing by solving Poisson's equation both cited to similar techniques. APPLE-1003, ¶[88].

Saito provides details of implementing solutions to Poisson's equation for image approximation. APPLE-1003, ¶[89]. A POSITA would have also understood or found obvious that Saito, which was published in 2006 (over 4 years after Mualla and Shirani), was a natural advancement of applying Poisson's equation to improve the visual quality of image approximation and represented a state-of-the-art solution at the time of the Critical Date. APPLE-1003, ¶[89]. Like Mualla and Shirani, Saito expresses an overarching goal of improving "the perceptual quality of these approximations" and a desire for a "perceptually correct metric" that can improve the "visual quality" of approximated images. APPLE-1007, 18; APPLE-1003, ¶[89]. Saito also discusses the relevance of "psychophysical experiments" in selecting appropriate image approximation techniques that improve "visual fidelity" of reconstructed images. APPLE-1007, 24; APPLE-1003, ¶[89].

As discussed above, Saito discloses a “polyharmonic local sine transform” (PHLST) and a “polyharmonic local cosine transform” (PHLCT) that solve Poisson’s equation and are “effective for image approximation.” APPLE-1007, 6, 27; APPLE-1003, ¶[90]. In particular, Saito explains that its PHLCT technique “shall solve the [] Poisson equation with the Neumann boundary condition.” APPLE-1007, 27; APPLE-1003, ¶[90].

Furthermore, as discussed above, a POSITA would have recognized or found obvious that Mualla, Shirani, and Saito all strive to achieve similar goals in spatial image approximation, and accordingly a POSITA would have had a reasonable expectation of success in combining Saito’s techniques into Mualla-Shirani. As Mr. Lett explains, both Shirani and Saito strive to improve reconstruction of high-frequency edges and discontinuities in an image. APPLE-1003, ¶[91] (citing APPLE-1006, 5 and 7, and APPLE-1007, 1, 2, and 4).

As discussed above, Saito’s PHLCT technique ensures that the derivative of the estimated signal at a boundary of a block is equal to the derivative of the signal to be estimated at the boundary. APPLE-1007, 27; *see also* 5, Abstract; *supra* §III.4. A POSITA would have understood or found obvious that this is consistent with Shirani’s “second stage” technique generating a “refined initial estimate” that has border pixel values with a “border error” that is equal to the “border error” of the “initial estimate” that was generated in Shirani’s “first stage.” APPLE-1006, 1,

3; APPLE-1007, 27; *see also* 5, Abstract; APPLE-1003, ¶[92]. As Mr. Lett explains, Shirani's "border error" is equivalent to Saito's "derivative at the boundary" (which is equivalent to the '025 patent's "gradient" value at a boundary). APPLE-1003, ¶[92]. Furthermore, a POSITA would have understood or found obvious that because Saito's PHLCT technique can achieve a faster "decay rate" for the error signal at the boundaries of blocks, this is consistent with Shirani's goal of "improv[ing] the video reconstruction quality" by ensuring that "image continuity inside the block and across its boundaries is preserved." APPLE-1007, 27; APPLE-1006, Abstract, 3.

As explained below in more detail, combining Mualla-Shirani and Saito would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results; and the (2) use of known techniques to improve similar devices in the same way. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421, 82 USPQ2d 1385, 1395-97 (2007); APPLE-1003, ¶[93].

In the resulting combination of Mualla, Shirani, and Saito, if no errors are detected in a frame of video, then conventional motion compensation is applied to the frame "to provide a prediction of the current frame," as taught by Mualla and Shirani. APPLE-1005, 27, Figure 1.1, 115-116; APPLE-1006, Abstract 2-3. Otherwise, if an error is detected in one or more blocks of a frame, then a motion-

compensated two-stage hybrid error concealment technique is applied to the frame. The “first stage” applies Shirani’s “temporal” concealment to find a block X_i in the previous frame with border pixels that have the smallest sum of “border errors” relative to surrounding blocks, as taught by Shirani. APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[94]. The resulting block X_i is an “initial estimate” of the erroneous/missing block.

Then, the “second stage” refines the “initial estimate” X_i by applying “spatial” concealment in which “the information around each missing block” is utilized to estimate pixels values in the interior of the estimated block X_i . APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[95]. Here, the second-stage spatial concealment of Mualla-Shirani is modified to implement Saito’s PHLCT to interpolate the interior pixels of the Mualla-Shirani initial estimate X_i . APPLE-1003, ¶[95]. The second-stage spatial concealment obtains a solution to Poisson’s equation with Neumann boundary conditions equal to the normal derivatives at boundaries of the missing block (as taught by Saito). APPLE-1003, ¶[95]. These normal derivatives at the boundaries are based on Shirani’s “border errors” obtained in Shirani’s first stage. APPLE-1003, ¶[95]. As a result, the Mualla-Shirani-Saito combination uses a Poisson-equation-solving image approximation technique (e.g., PHLCT as taught in Saito) to implement spatial error concealment in the second stage of the Mualla-Shirani two-stage hybrid error concealment

technique, thereby generating a refinement of the “initial estimate” that was obtained in Shirani’s first stage. *Id.* The result is an estimated frame in which erroneous/missing blocks have been concealed. *Id.* As Mr. Lett explains, a POSITA would have recognized and found obvious that Saito’s image approximation techniques based on solving Poisson’s equation would have enhanced Shirani’s “second stage” techniques, “improv[ing] the video reconstruction quality” in order to achieve similar goals of image approximation. APPLE-1007, 27; APPLE-1006, Abstract, 3; APPLE-1003, ¶¶[95]-[96].

6. Analysis

“Moving picture decoding” claims 6-8, 10

[6.pre]

Mualla teaches extensive details of an H.263-based video codec system, which implements a *decoding apparatus* for decoding *moving pictures* (e.g., “video”). APPLE-1005, 23-26, 31, 36, 51-52, 227-228, 241-242, 265. For example, in describing “Transform Coding” which “forms the basis of almost all video coding standards,” Mualla describes a *decoder* implemented in a “transform coding system.” APPLE-1005, 51 (last paragraph), Figure 2.9 (reproduced below). As another example, in describing techniques for “error resilience” in video coding systems, Mualla discloses a *decoder* implemented in a “typical video communication system.” APPLE-1005, 227-228, Figure 9.1; APPLE-1003,

¶[112].

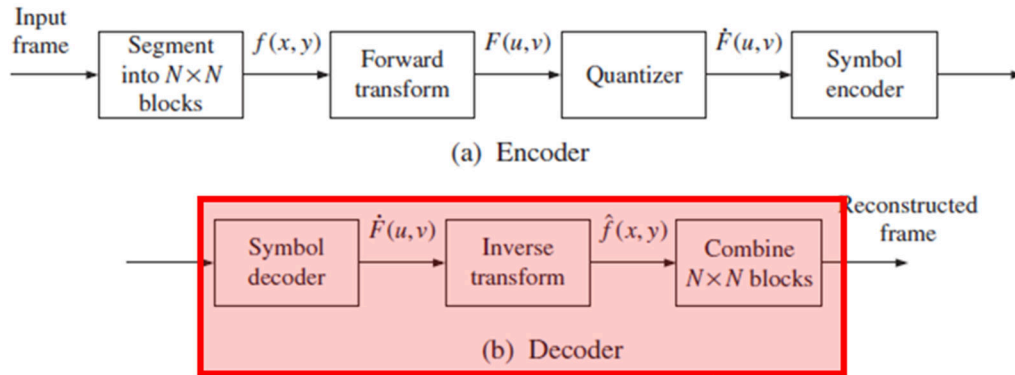


Figure 2.9: Block diagram of a transform coding system

A “moving-picture
decoding apparatus”

APPLE-1005, FIG. 2.9 (annotated and color added)

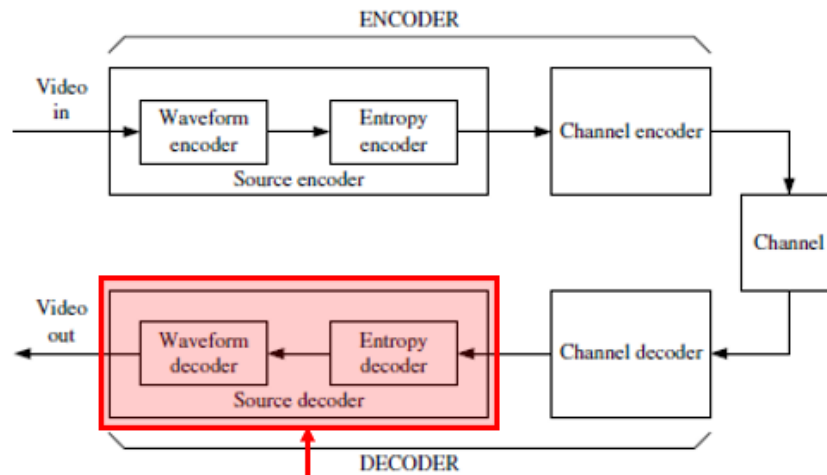


Figure 9.1: Typical video communication system

A “moving-picture
decoding apparatus”

APPLE-1005, FIG. 9.1 (annotated and color added)

Shirani similarly discloses using the H.263-based video codec system and its

associated *decoding apparatus*. APPLE-1006, 2-4, 4; APPLE-1003, ¶[113].

[6.1.DEC]

Shirani discloses that “[c]oded video information [(“*coded data from an input signal*”)] is first grouped into packets” and the “video packets are ... multiplexed with audio information” using an “H.223 multiplexing” protocol. APPLE-1006, 5. As Mr. Lett explains, multiplexing video packets at an encoder would have required a means for “*demultiplex[ing]*” the video packets at the decoder, for example, by using a *demultiplexer*. APPLE-1003, ¶[114]. Indeed, the H.263 standard (used by Mualla and Shirani) corroborates the use of demultiplexers in H.263 codecs for “*demultiplex[ing] coded data from an input signal.*” APPLE-1016, 2.

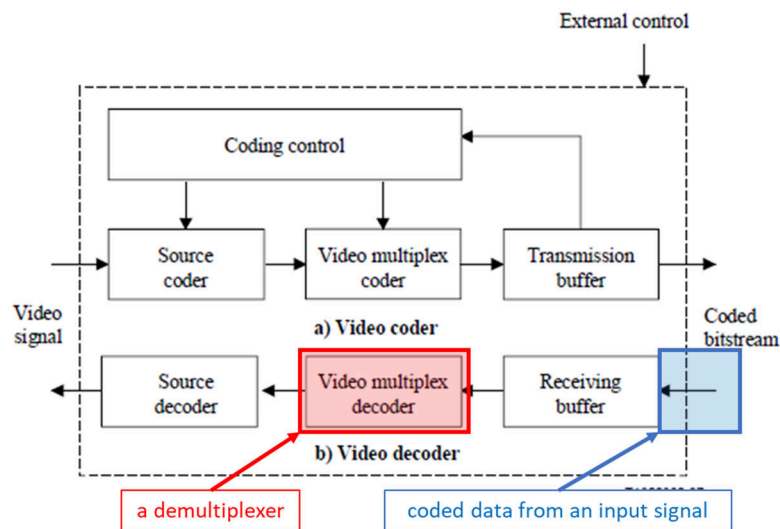
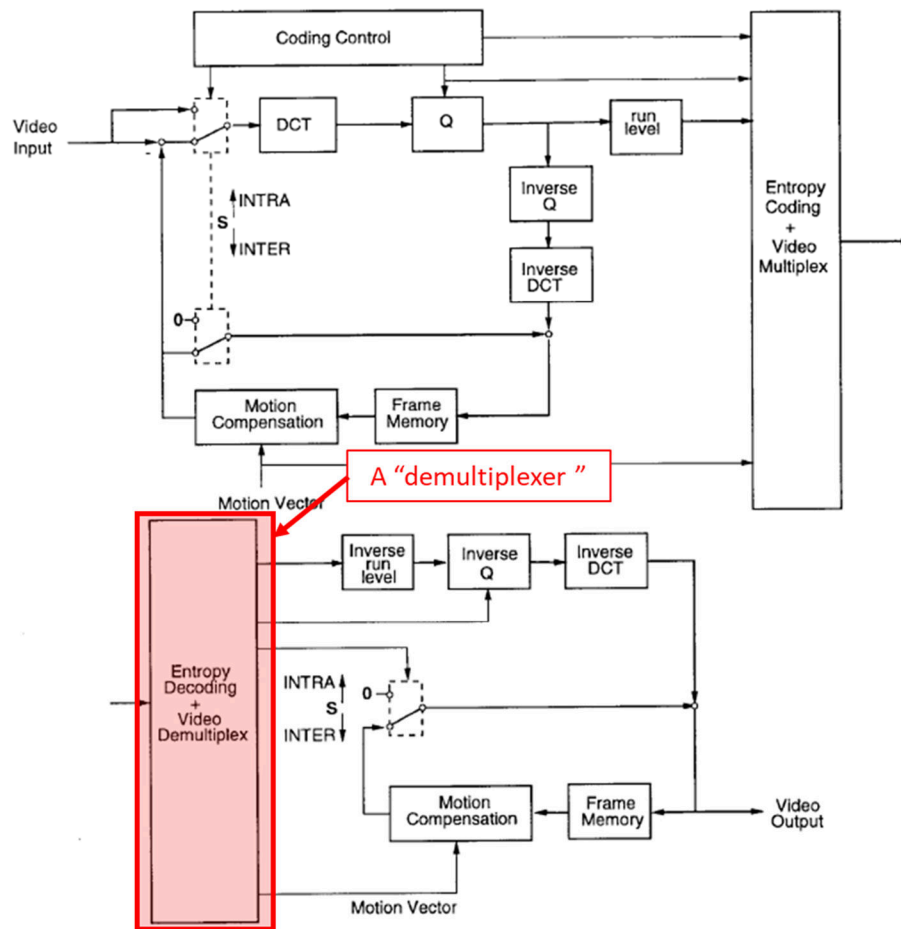


Figure 1/H.263 – Outline block diagram of the video codec

APPLE-1016, FIG. 1 (annotated and color added).

Additionally, Girod – a H.263 codec reference cited by Mualla as reference

201 – discloses that information is “multiplexed to the video bitstream.” APPLE-1008, 2, FIG. 1(b); APPLE-1003, ¶[115]. FIG. 1(b) illustrates an example decoder with a “*demultiplexer*.”



APPLE-1008, FIG. 1(a) (“encoder”) and 1(b) (“decoder”) (annotated and color added).

Furthermore, Mualla and Shirani both disclose the H.263 *specific syntax structure*. APPLE-1005, 93, 232; APPLE-1006, 2, 4-5. Indeed, Mualla refers to the H.263 codec standard as “the *syntax* of H.263.” APPLE-1005, 93. Mualla and Shirani also both use the specific syntax structure of the H.263 codec standard as a

gauge for detecting errors in a video bitstream. APPLE-1005, 231-232; APPLE-1006, 2. For example, Mualla's process of "error correction" uses "*syntax* ... violations" to detect errors in a video bitstream. APPLE-1005, 231-232. Shirani discloses a similar process where a decoder determines the "location of missing blocks" by "detecting ... semantic or *syntactic* violations." APPLE-1006, 2; APPLE-1003, ¶[116].

[6.2.COD]

As described above in 6.1.DEC, Shirani discloses *multiplexing a coded bitstream*, and a POSITA would have understood or found obvious that the H.263 codec standard, as used in Mualla and Shirani, included *multiplexing* operations. APPLE-1006, 5; APPLE-1003, ¶[117]; *supra*, [6.1.DEC]. Indeed, the H.263 standard discloses multiplexing and demultiplexing a "*coded bitstream*." APPLE-1016, 2, 9; APPLE-1003, ¶[117].

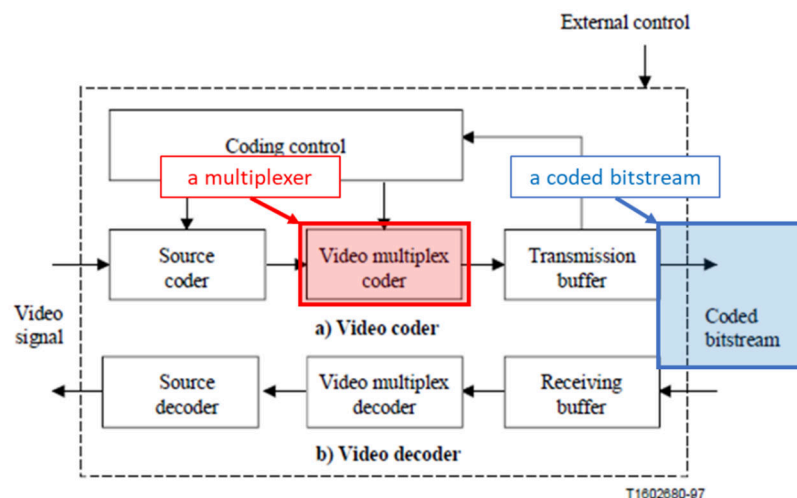


Figure 1/H.263 – Outline block diagram of the video codec

APPLE-1016, FIG. 1 (annotated and color added).

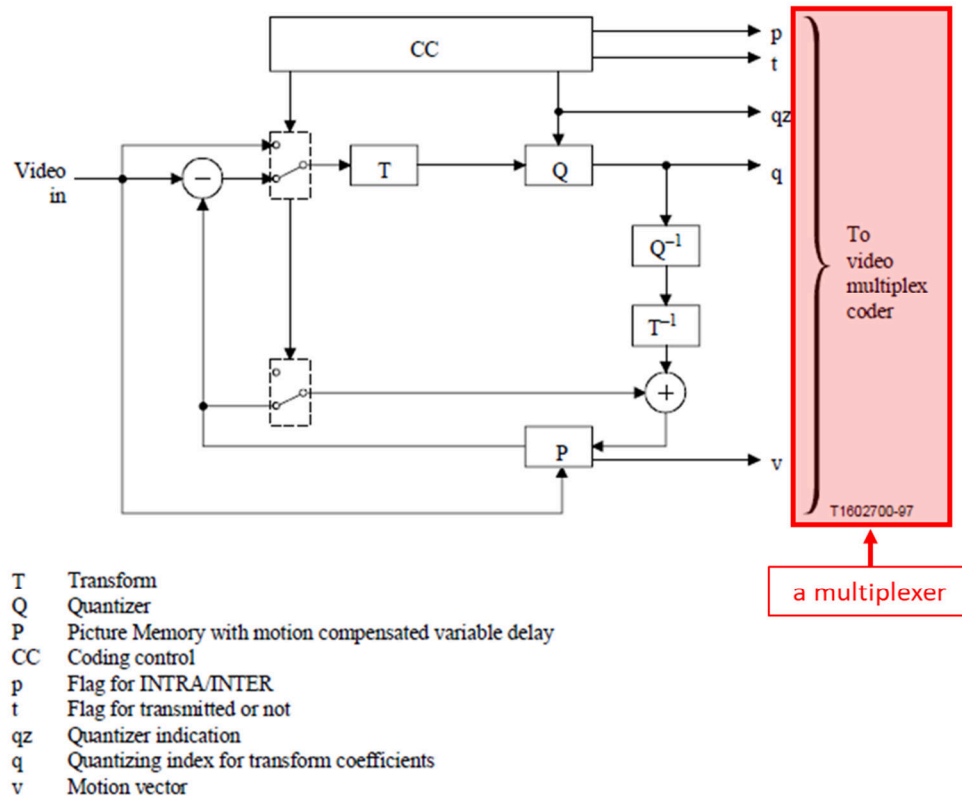


Figure 3/H.263 – Source coder

APPLE-1016, FIG. 3 (annotated and color added).

Furthermore, as shown in Mualla’s Figure 3.1 and explained in further detail below, a POSITA would have understood or found obvious, based on the teachings of Mualla, that the coded bitstream is obtained by encoding motion vectors (which include the claimed “*border motion-vector data*”) and a quantization index (the claimed “*post-quantization data*”), along with various coding control information. APPLE-1005, Figure 3.1, 71-75; APPLE-1003, ¶[118].

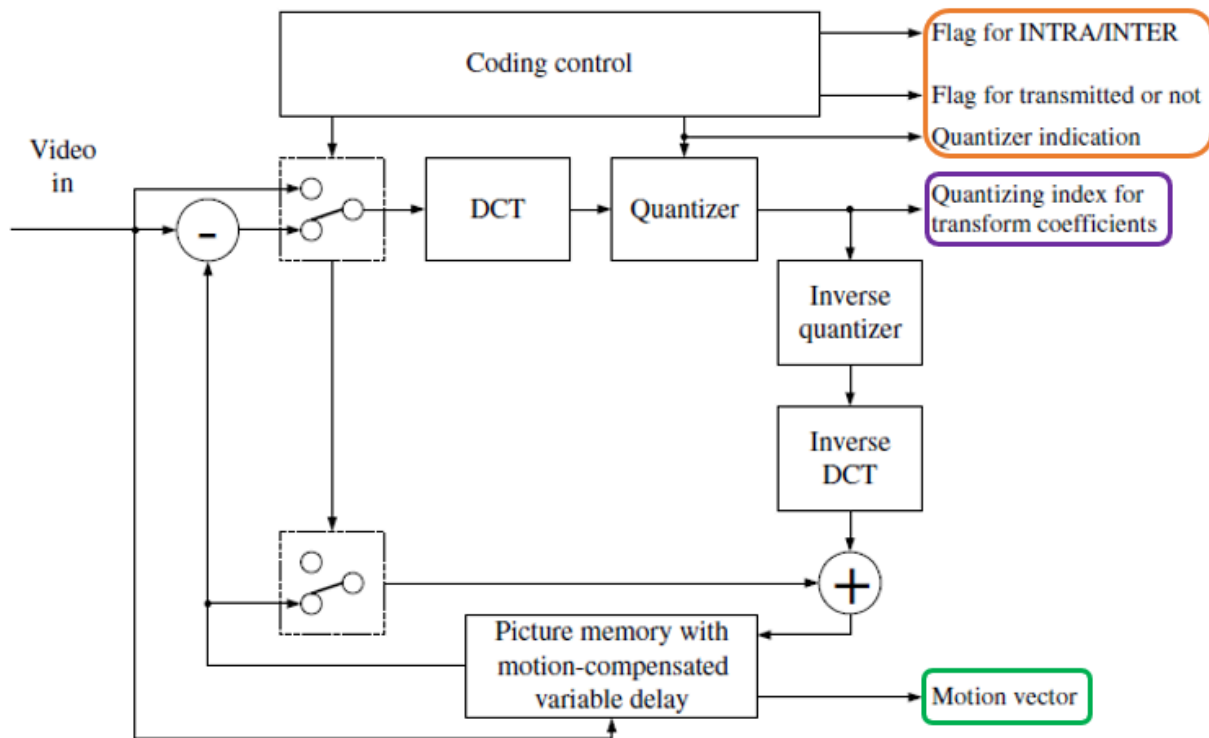


Figure 3.1: H.263 video encoder

...obtained by predictive coding

Mualla describes numerous operations at the encoder that perform predictive coding. APPLE-1003, ¶[119]. For example, Mualla discloses a “video codec” that includes a “motion estimation (ME) block ... to estimate the movement that occurred between the current frame and a reference frame.” APPLE-1005, 26-27, 115-117, FIG. 1.1. Mualla refers to this estimation as “motion information.” *Id.* Motion information is then “utilized by the motion compensation (MC) block to move the contents of the reference frame to provide a prediction of the current frame” (“*predictive coding*”) which Mualla also refers to as the “motion-compensated prediction (MCP).” *Id.* Mualla refers to “motion compensated

prediction” as a type of “predictive coding” in which regions of an input frame are predictively coded “with respect to the corresponding regions in the reference frame.” APPLE-1005, 61; APPLE-1003, ¶[119]. Mualla’s “typical video codec” illustrates these components.

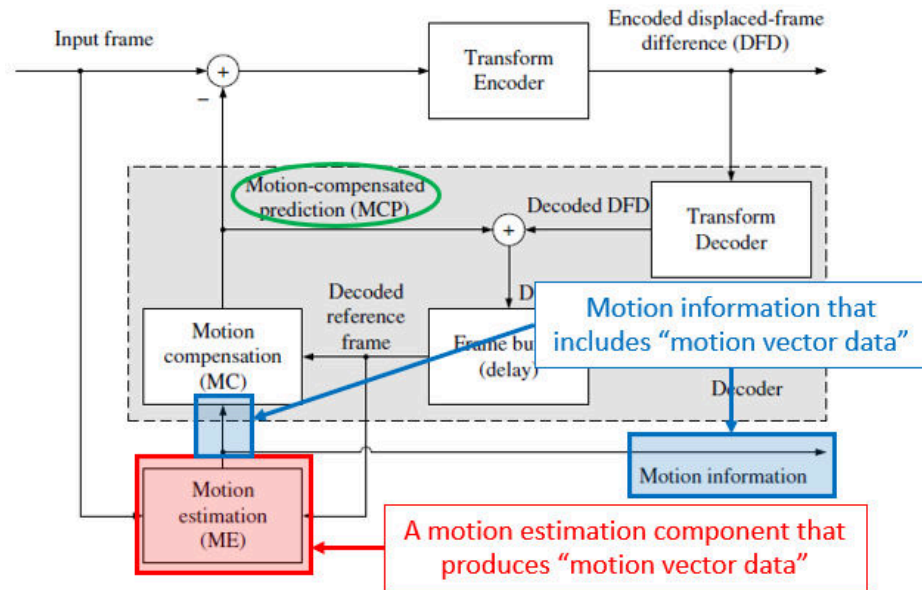


Figure 1.1: Typical video codec

APPLE-1005, FIG. 1.1 (annotated and color added).

As another example of predictive coding, Mualla discloses that the “motion information” (motion vector) itself is predictively coded. Mualla discloses that the “motion of the [a] current block” (e.g., the motion information) is “represented by a motion vector” (“*motion-vector data*”) which is the “displacement between the block and its best-match block in the reference frame.” APPLE-1005, 62, FIG. 2.16; APPLE-1003, ¶[120].

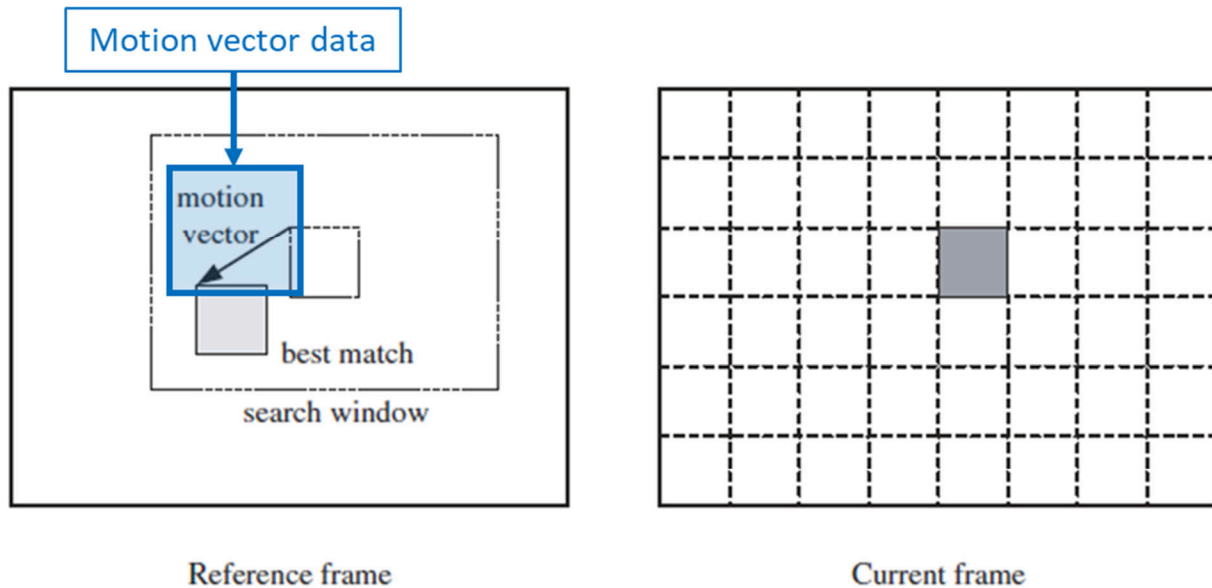


Figure 2.16: Block-matching motion estimation

APPLE-1005, FIG. 2.16 (annotated and color added).

Mualla discloses that the “estimated motion vector” is “predictively coded” (part of “*a coded bitstream obtained by predictive coding*”), meaning that the “motion vector difference $MVD = MV - MVP$ is encoded instead of the motion vector itself.” APPLE-1005, 74, *see also* 115-117; APPLE-1003, ¶[121].

border motion-vector data

As an initial matter, the '025 patent describes “**border motion-vector data**” as “data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found.” APPLE-1001, 2:59-61; *infra*, [6.6.COD]; APPLE-1003, ¶[122].

As discussed above, the Mualla-Shirani-Saito combination implements a

two-stage hybrid error concealment in the **encoder** (as well as the decoder), as taught by Mualla. APPLE-1005, 246-250. The encoder implements the first stage of Shirani's two-stage error concealment technique to the motion of a missing/erroneous macroblock by performing temporal concealment. *Supra*, §§III.A.3, A.5. In the combination, Shirani's first stage uses motion vectors of neighboring blocks to obtain estimates of pixels at the borders of a missing block. *Supra*, §§III.A.3, A.5; APPLE-1006, 2-3. In particular, the first stage of Shirani estimates motion vectors of subblocks adjacent to the missing block, and in doing so, utilizes “candidate motion vectors” (“data on a motion vector”) which can include motion vectors of “neighboring blocks.” *Supra*, §§III.A.2, A3, A.5; APPLE-1006, 2-3; APPLE-1003, ¶[123]. These motion vectors of neighboring blocks are used to obtain an estimate of pixels at each border of the missing block (“from a border of the rectangular zone in the picture to be coded”), by minimizing the “squared sums of border errors” (in total, “***border motion-vector data***” or, “data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found”). *Id.*

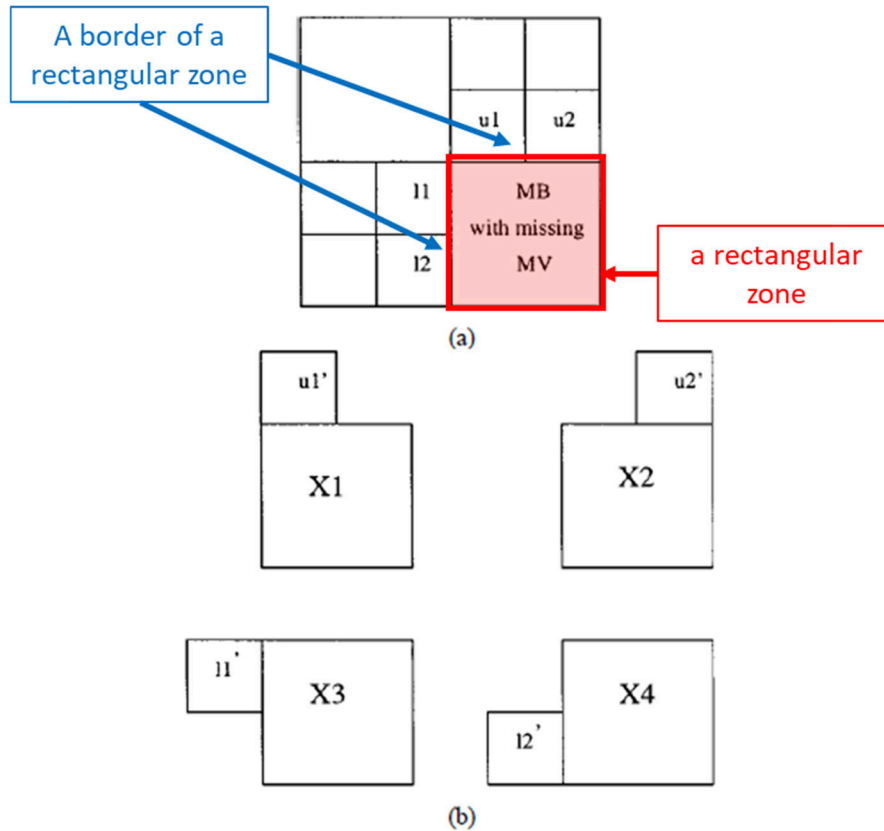


Fig. 1. (a) The four subblocks, (b) their corresponding subblocks in the previous frame, and the blocks connected to them.

APPLE-1005, FIG. 1(a) (annotated and color added).

After Shirani's **first stage** estimates pixels at the **borders** of the missing block, the **second stage** of the Mualla-Shirani-Saito combination estimates pixels in the **interior** of the missing block using Saito's technique for solving Poisson's equation with the border pixels (specifically, Shirani's "border errors" of the missing block) providing boundary conditions in Saito's techniques. *Supra*, §III.A.5; *infra* [6.7.COD]; APPLE-1003, ¶[124]. Therefore, because Shirani's first stage temporal concealment uses the motion vectors of neighboring blocks to estimate only the motion of pixels at the border of the missing block, Shirani

discloses the claimed “*border motion vector data.*”

post-quantization data obtained by quantization in the predictive coding

Mualla discloses that “quantization is a key element of a video coding system” and illustrates a “quantizer” in its example “video encoder” and “transform coding system.” APPLE-1005, 40-42, 52, FIGS. 2.3, 2.9, 3.1. Mualla discloses that, in “transform coding” (also known as “DCT coding”), “coefficients can be quantized according to their visual importance” (“*post-quantization data*”). APPLE-1005, 52, 75; APPLE-1003, ¶[125].

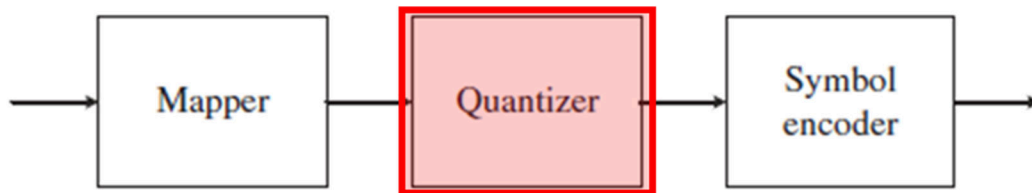


Figure 2.3: Elements of a video encoder

APPLE-1005, FIG. 2.3 (annotated and color added).

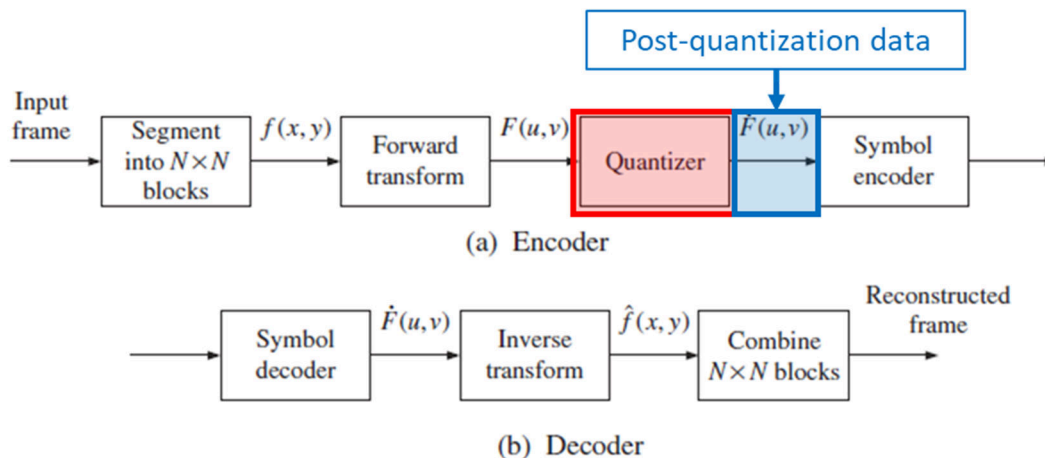


Figure 2.9: Block diagram of a transform coding system

APPLE-1005, FIG. 2.9 (annotated and color added).

Mualla discloses that these coefficients are derived by obtaining a “displaced frame difference (DFD)” that represents “the coded prediction error signal” (“*the predictive coding*”). APPLE-1005, 61, FIG. 2.15. The DFD signal can be coded “using any of the intraframe coding methods discussed in Section 2.6” and “the most commonly used method is transform coding, in particular block-based DCT transform coding.” APPLE-1005, 62. Mualla discloses that the DCT transform “is applied to the DFD values” and that this “results in ... transform coefficients for each [macro block].” APPLE-1005, 74. These coefficients are then “quantized using a uniform scalar quantizer” (“*post-quantization data obtained by quantization in the predictive coding*”). APPLE-1005, 75, FIG. 3.1; APPLE-1003, ¶[126]. This process is illustrated in Mualla’s example “H.263 video encoder.”

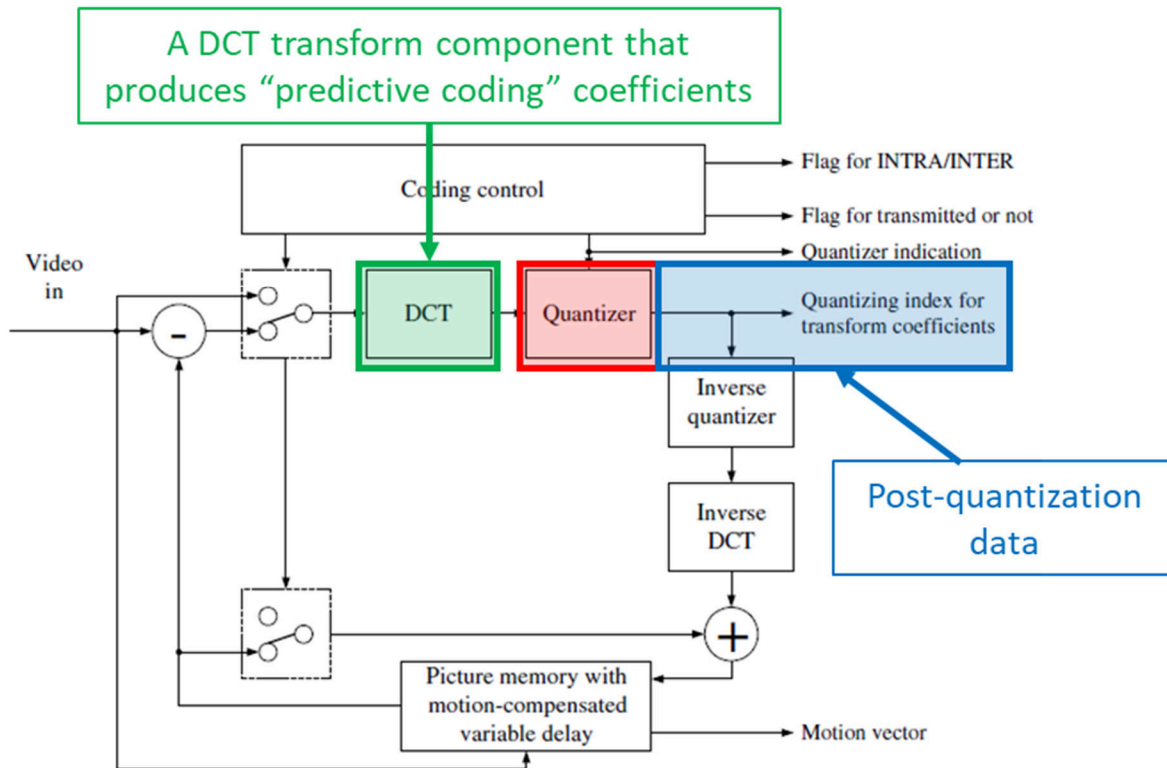


Figure 3.1: H.263 video encoder

APPLE-1005, FIG. 3.1 (annotated and color added).

[6.3.COD]

As discussed above in [6.2.COD], Mualla's DFD represents "the coded prediction error signal" ("*residual picture that is a residual signal*") which is created by subtracting a "motion compensation prediction (MCP)," produced from the "reference frame" ("*a predictive picture produced from a reference picture that is a local decoded video signal*"), from the "current frame" ("*a picture to be coded that is an input moving-picture video signal to be subjected to coding*"). APPLE-1005, 26-27, FIG. 1.1; APPLE-1003, ¶[127]. Mualla's "typical video codec" illustrates this process.

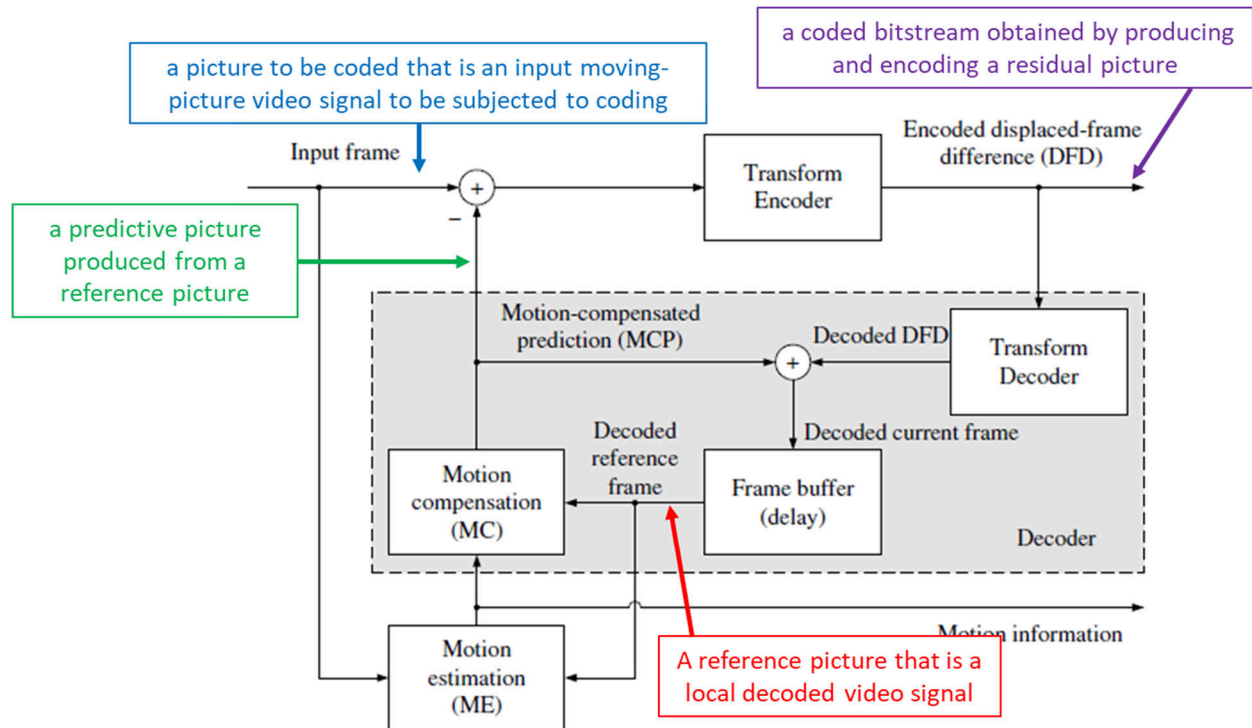


Figure 1.1: Typical video codec

APPLE-1005, FIG. 1.1 (annotated).

Mualla also discloses that an input video signal “consists of a sequence of pictures (or frames)” where “[e]ach picture is divided into groups of blocks (GOBs)” and each GOB “is divided into macroblocks (MBs)” (“*each of a plurality of rectangular zones*”). APPLE-1005, 71-72. These MBs are further divided into individual “pixels,” or “pels” (“*each composed of a specific number of pixels, into which a video area of the moving-picture video signal is divided*”). APPLE-1005, 36, 71-72, FIG. 3.2; APPLE-1003, ¶[128]. FIG. 3.2 of Mualla illustrates this breakdown.

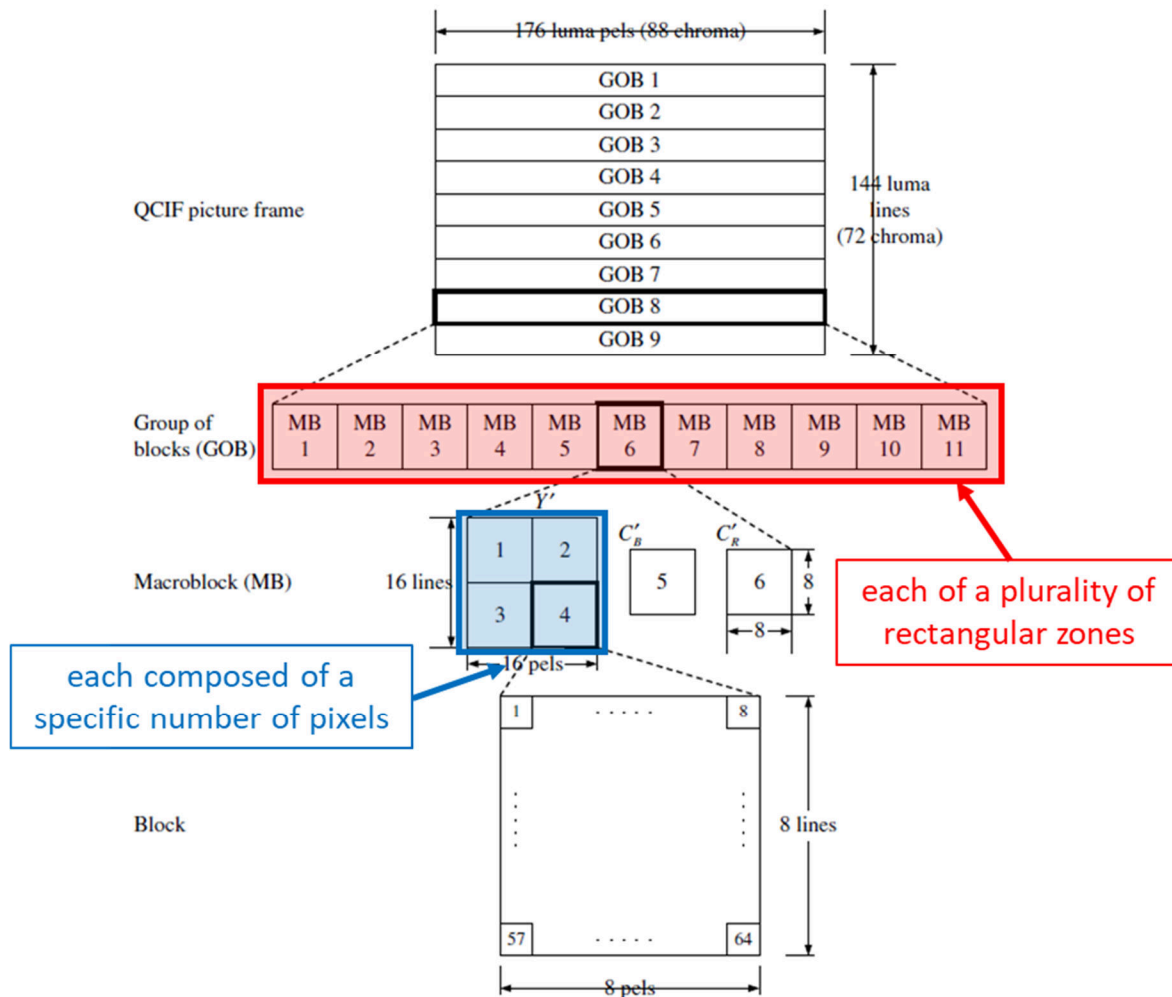


Figure 3.2: H.263 picture structure for a QCIF frame

APPLE-1005, FIG. 3.2 (annotated and color added).

[6.4.COD]

As discussed above, the Mualla-Shirani-Saito combination implements a two-stage hybrid error concealment in the **encoder** (as well as the decoder) using “interactive techniques,” as taught by Mualla. APPLE-1005, 246-250; APPLE-1003, ¶[129].

For the first stage, as discussed above in [6.2.COD], Mualla discloses

“boundary matching (BM)” to select a “suitable replacement [motion vector] from a set of candidate [neighboring] motion vectors. APPLE-1005, 245; *supra* [6.2.COD]. Shirani’s first stage applies a similar technique to estimate a missing motion vector by considering “subblocks adjacent to the missing block.” APPLE-1006, 2-3, FIG. 1(a); *supra* [6.2.COD]; APPLE-1003, ¶[130].

In more detail, Shirani explains that in order to obtain an “initial estimate” for each of the missing blocks (“*the rectangular zones*”), “subblocks adjacent to the missing block, i.e., $u1$, $u2$, $l1$, and $l2$ in Fig. 1(a), are considered” (“*another plurality of rectangular zones adjacent to the rectangular zones*”). APPLE-1006, 3, FIG. 1(a) (reproduced below); APPLE-1003, ¶[131].

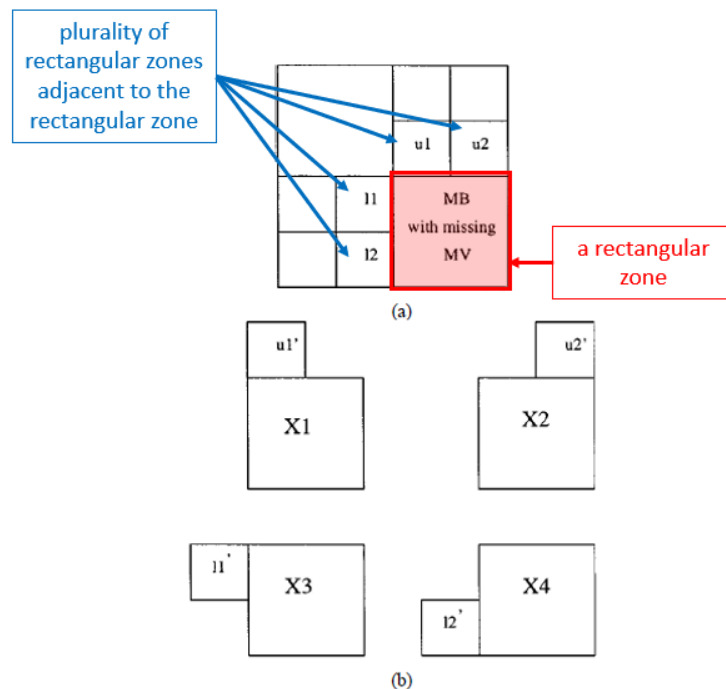


Fig. 1. (a) The four subblocks, (b) their corresponding subblocks in the previous frame, and the blocks connected to them.

APPLE-1006, FIG. 1(a) (annotated and color added).

As explained in Section III.A.2, above, Shirani discloses that “for each of these subblocks $[(u1, u2, l1, \text{ and } l2)]$, a corresponding subblock in the previous frame is determined” (denoted $u1', u2', l1', \text{ and } l2'$ at the bottom of FIG. 1(a), above). APPLE-1006, 3. Then, “four blocks, namely X1, X2, X3, and X4, which are connected to $u1', u2', l1', \text{ and } l2'$, and (respectively), are determined” and “a block from the above four blocks $[(X1, X2, X3, \text{ and } X4)]$ that minimizes the squared sums of border errors, between the estimated block and its adjacent above and left blocks, is selected.” APPLE-1006, 3; APPLE-1003, ¶[132]. Shirani achieves this by defining “border error[s]” in “terms of pixels,” including an error for the “bottom line pixels of the block to the top of the missing block” and an error for the “right column pixels of the block to the left of the missing block” and applying the same technique to define border errors for pixels adjacent to “the subblocks ... to the right and below of the missing block” (“*obtaining a boundary condition of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular zones*”). *Id.* Furthermore, although the above description focused on adjacent subblocks above and to the left of the missing block, Shirani also discloses that “the same method can be extended to include the subblocks of the blocks to the right and below of the missing block” (“... *of each of a plurality of borders between the rectangular zones and another plurality of rectangular zones adjacent to the rectangular*

zones”). *Id.*

FIG. 5 of Shirani illustrates an example of concealing errors for multiple damaged macroblocks (a “*plurality*” of “*rectangular zones*”) in a frame. APPLE-1006, FIG. 5; APPLE-1003, ¶[133].

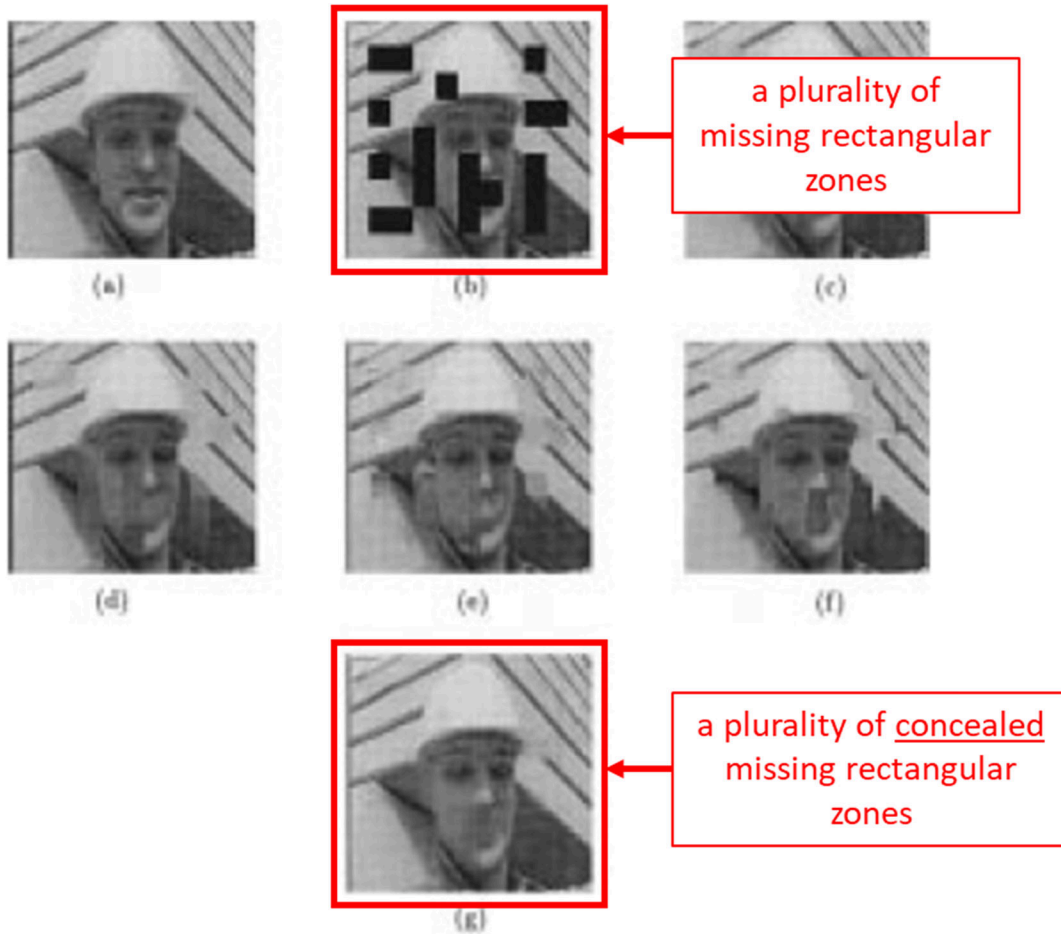


Fig. 5. A frame from the video sequence Foreman: (a) original; (b) missing blocks; reconstructed using (c) a nonadaptive GMRF model; (d) a suboptimal MRF model; (e) the method proposed in [4]; (f) the method proposed in [3]; and (g) our adaptive MRF model.

APPLE-1006, FIG. 5 (annotated).

[6.5.COD]

As discussed above, the Mualla-Shirani-Saito combination implements a two-stage hybrid error concealment in the **encoder** (as well as the decoder) using “interactive techniques,” as taught by Mualla. APPLE-1005, 246-250; APPLE-1003, ¶[134].

As described above in [6.4.COD], Shirani finds and evaluates the “border error” of a plurality of neighboring subblocks when considering the suitability of these subblocks for error concealment (“*finding a border, of the reference picture, having a boundary condition that matches the boundary condition*”). APPLE-1006, 2-3, FIG. 1; *supra* [6.2.COD], [6.4.COD]. Additionally, Shirani discloses that the “search for displacement of each of the subblocks is restricted to a set of candidate motion vectors,” specifically, “the motion vectors of available neighboring blocks” (“*finding a border ... by motion-vector search in the reference picture*”). APPLE-1006, 3, FIG. 1(a); *supra* [6.2.COD], [6.4.COD]; APPLE-1003, ¶[135].

[6.6.COD]

As discussed above, the Mualla-Shirani-Saito combination implements a two-stage hybrid error concealment in the **encoder** (as well as the decoder) using “interactive techniques,” as taught by Mualla. APPLE-1005, 246-250; APPLE-1003, ¶[136].

As described in [6.2.COD], [6.4.COD], and [6.5.COD] above, Shirani's first stage searches for candidate replacement motion vectors and generates border errors that define the similarity of the neighboring subblock's motion vector ("***border motion-vector data***"). *Supra*, [6.2.COD], [6.4.COD], [6.5.COD]; APPLE-1003, ¶[137]. Indeed, Shirani's "candidate motion vectors" are "***data on a motion vector from a border of the rectangular zone in the picture to be coded to the border of the reference picture thus found,***" at least because these candidate motion vectors conceal missing motion vectors between "***the reference picture***" and "***the picture to be coded.***" *Id.*

Moreover, as described above, Shirani generates border errors that quantify the similarity of the neighboring subblocks' motion vectors compared to the missing block to be concealed ("***defining a boundary condition of a border that corresponds to the border motion vector data, from the reference picture***"). *Supra*, [6.4.COD], [6.5.COD]; APPLE-1003, ¶[138]. Shirani's border errors evaluate the similarity of the neighboring subblocks' motion vectors, so they are therefore "***based on the border motion vector data***" of the neighboring subblocks. *Id.*

[6.7.COD]

As discussed above, the Mualla-Shirani-Saito combination implements a two-stage hybrid error concealment in the **encoder** (as well as the decoder) using

“interactive techniques,” as taught by Mualla. APPLE-1005, 246-250; APPLE-1003, ¶[139].

In particular, as described above in [6.6.COD], Shirani’s first stage generates *border motion-vector data* for each damaged or missing subblock. *Supra*, [6.2.COD], [6.4.COD]-[6.6.COD]. The second stage of the Mualla-Shirani-Saito combination uses “the information around each missing block ... to refine the initial estimate” (as taught by Shirani) by using Saito’s technique for solving Poisson’s equation with known boundary conditions. APPLE-1006, 3; APPLE-1007, 27; *supra* §§III.A.2, III.A.4, III.A.5; APPLE-1003, ¶[140].

As Mr. Lett explains, and as discussed above in the combination, a POSITA would have recognized and found obvious that Saito’s image approximation techniques based on solving Poisson’s equation would have enhanced Shirani’s “second stage” techniques, “improv[ing] the video reconstruction quality” in order to achieve similar goals of image approximation. APPLE-1007, 27; APPLE-1006, Abstract, 3; APPLE-1003, ¶[141]. Saito’s Polyharmonic Local Cosine Transform (PHLCT) is an enhancement of a Polyharmonic Local Sine Transform (PHLST) technique for image approximation. APPLE-1007, Abstract, 27. Saito discloses that the PHLST method “segments an image into local pieces using ... characteristic functions, then decomposes each piece into two components: the polyharmonic component and the residual.” APPLE-1007, Abstract, 4. Saito

explains that the polyharmonic component is “obtained by solving the elliptic boundary value problem associated with the so-called polyharmonic equation (e.g., Laplace’s equation, biharmonic equation, etc.) given the boundary values (the pixel values along the boundary created by the characteristic function).” *Id.* The PHLCT variation of PHLST approximates an image between two boundaries by “solv[ing] the ... Poisson equation with the Neumann boundary condition” (“*generating an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation*”). APPLE-1007, 27; APPLE-1003, ¶[141].

In implementing Saito’s teachings into the Mualla-Shirani combination, a POSITA would have recognized and found obvious that the “border errors” of Shirani are suitable to use as the “Neumann boundary condition” for Saito’s the PHLCT method, and that PHLCT would have been capable of approximating the image between these boundary conditions (“*thus producing the predictive picture*”). APPLE-1007, 27 (“this method of course requires to estimate the **first order normal derivative** of data at the **boundary**”); APPLE-1003, ¶[142].

As described in §III.A.5, above, Saito’s techniques represent a natural improvement of image approximation techniques, at least because they can achieve

³ Petitioner reserves the right to advance §112 arguments against this claim feature in litigation.

a faster “decay rate” for the error signal at the boundaries of blocks – an improvement that is consistent with the goals of Mualla-Shirani. APPLE-1007, 27; APPLE-1006, Abstract, 3; APPLE-1003, ¶[142].

Moreover, Saito’s process of PHLCT is described by the ’025 patent itself, and the ’025 patent cites a subsequent publication by Dr. Naoki Saito that explains the PHLCT method. APPLE-1001, Cover, 11:34-12:25; APPLE-1003, ¶[143].

[6.8.DEC]

As discussed above in [6.2.COD], Mualla-Shirani-Saito discloses “***border motion-vector data***” and “***post-quantization data.***” *Supra*, [6.2.COD]. In particular, as discussed above, the first stage of Shirani’s two-stage hybrid error concealment technique uses motion vectors of “neighboring blocks” (“***border motion-vector data***”) to estimate motion vectors of subblocks adjacent to the missing block. APPLE-1006, 3. Furthermore, as discussed in Section III.A.1, above, the motion vectors of these neighboring blocks (“***border motion-vector data***”) are encoded and transmitted to the decoder. APPLE-1005, 71, 74; APPLE-1003, ¶[144]. Because the encoding and transmission of motion vector information is performed for all macroblocks in a frame/picture, Mualla teaches that the motion vectors of neighboring blocks (“***border motion-vector data***”) around a missing block are encoded and transmitted to the decoder, and can be used by the decoder to perform temporal error concealment in Shirani’s first stage.

APPLE-1003, ¶[144].

Mualla-Shirani-Saito also discloses conventional “***motion vector data.***” *Supra*, III.A.1; *infra*, [6.14.DEC]. As discussed above, Mualla encodes and transmits motion information for each macroblock in a frame, which includes motion information for the macroblock being decoded (“***motion vector data***”) as well as motion information for neighboring blocks (“***border motion-vector data***”) which can be used for error concealment in case the macroblock being decoded is received in error or corrupted. APPLE-1005, 71, 74; APPLE-1003, ¶[145].

In particular, a POSITA would have understood or found obvious that, in the system of Mualla-Shirani-Saito, motion information for a macroblock being decoded (“***motion vector data***”) can be received and decoded, even if the macroblock is detected as having errors. APPLE-1003, ¶[146]. For example, Mualla explains that a macroblock is detected as having an “error” if the “**decoded motion vector** points outside the permissible range.” APPLE-1005, 232; 244 (explaining that “the motion vector of a damaged block is usually lost or **erroneously received**” and disclosing techniques for recovering a “damaged motion vector”); *see also* Girod, 4 (“Transmission errors can be detected in a variety of ways ... **detect** that the **decoded motion vectors**, DCT coefficients, or quantizer step sizes **exceed their permissible range** (semantic violations)”). In view of these teachings, even if a macroblock is detected as having errors, and

even if a particular example disclosed in Mualla or Shirani refers to motion vectors as “missing,” “lost,” or “not available,” a POSITA would have understood or found obvious that the techniques of Mualla and Shirani are applicable to motion vectors (“*motion vector data*”) that are **received** and detected as having errors. APPLE-1003, ¶[146].

As discussed in [6.1.DEC], above, Mualla-Shirani-Saito includes a multiplexer that “demultiplex[es]” received data. In addition, Mualla-Shirani-Saito implements an entropy decoder to perform entropy decoding on the demultiplexed data. *Supra*, [6.1.DEC]. In particular, Mualla also discloses an “*entropy decoder*” in a “[t]ypical video communication system” that “obtain[s] the output video” (“*generate ... motion-vector data, the post-quantization data, the border motion-vector data and parameter data required for constructing a specific syntax structure*”). APPLE-1005, 227-228, FIG. 9.1; APPLE-1003, ¶[147].

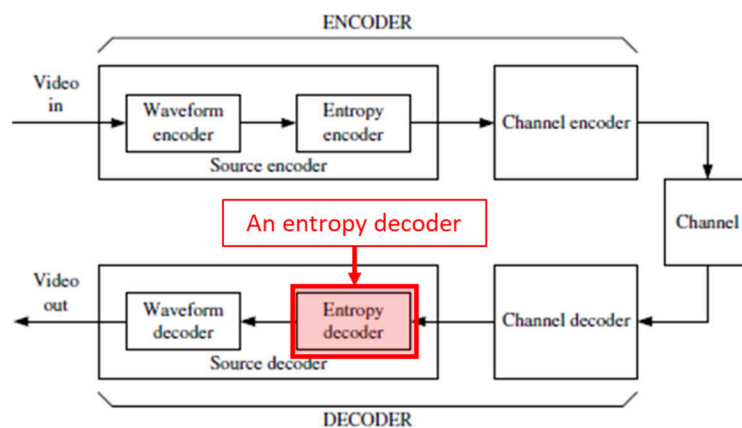


Figure 9.1: Typical video communication system

APPLE-1005, FIG. 9.1 (annotated and color added).

[6.9.DEC]

Mualla discloses that the H.263 standard includes the techniques of inverse-quantization. APPLE-1005, 51-52, 76. Specifically, Mualla discloses that “quantization is achieved by dividing the transform coefficient by a quantization step size and rounding the result (*“post-quantization data”*) ... [i]nverse quantization is the process of reconstructing an approximation of the original coefficient by multiplying the quantized coefficient by the same step size” (*“inverse-quantization to the post-quantization data to generate post-quantization orthogonal transform coefficients data”*). APPLE-1005, 76. Mualla illustrates an *“inverse quantizer”* in its example H.263 video encoder. APPLE-1005, FIG. 3.1. Mr. Lett explains that a POSITA would have recognized and found obvious that a similar component would have been included in the decoder of a codec. APPLE-1003, ¶[148].

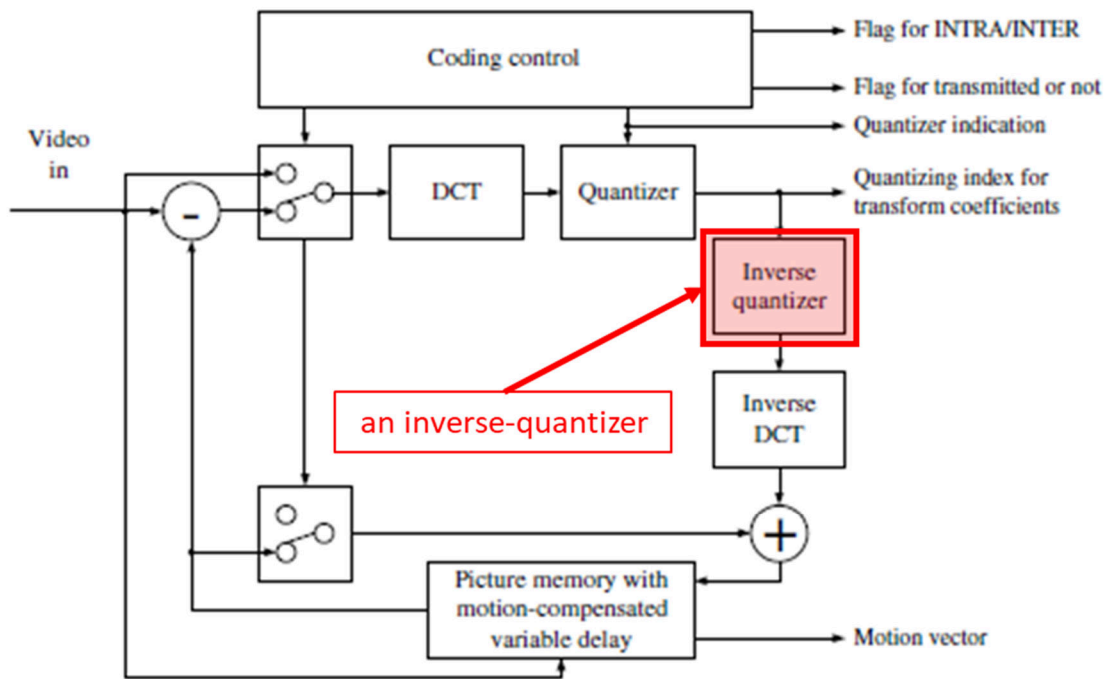
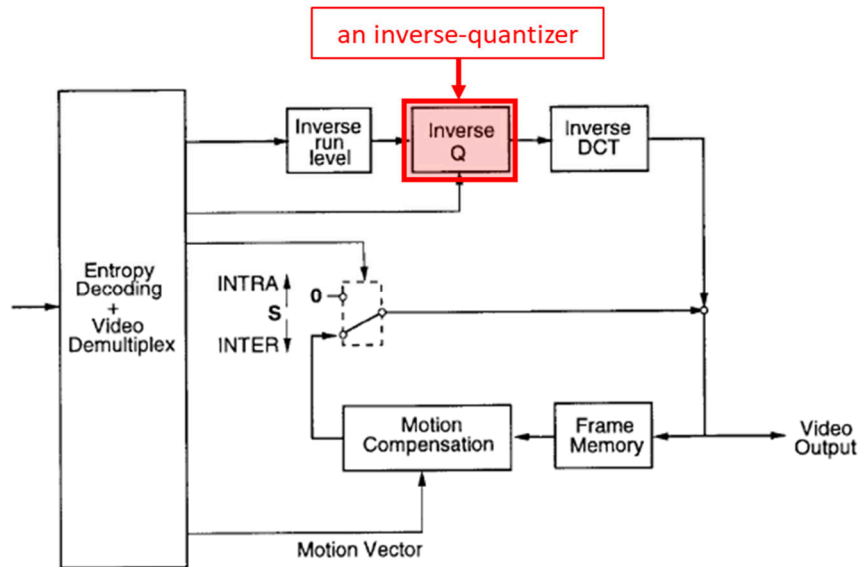


Figure 3.1: H.263 video encoder

APPLE-1005, FIG. 3.1 (annotated and color added).

In particular, a POSITA would have recognized or found obvious that the decoder would “*perform inverse quantization*,” for example, by using an “*inverse-quantizer*,” as this was well-known before the Critical Date. APPLE-1003, ¶[149]. Indeed, Girod corroborates the use of “inverse-quantizer[s]” in H.263 decoders and illustrates this component in its example H.263 decoder. APPLE-1008, 2, FIG.

1(b)



APPLE-1008, FIG. 1(b) (annotated and color added).

[6.10.DEC]

As discussed in §III.A.1, above, Mualla discloses an H.263-based system that implements inverse-orthogonal transforms. APPLE-1005, 51-52, 76. Specifically, Mualla discloses the encoder performs “transform coding” where a “unitary space-frequency transform is applied to each block [in an input frame] to produce an $N \times N$ block of transform (spectral) coefficients that are then suitably quantized” (“*post-quantization orthogonal transform coefficients data*”). APPLE-1005, 52, FIG. 2.9; APPLE-1003, ¶[150]. At the decoder, “an inverse transform is applied to reconstruct the frame” (“*an inverse-orthogonal transformer to perform inverse-orthogonal transform to the post-quantization orthogonal transform coefficients data*”). *Id.* FIG. 2.9 of Mualla illustrates an “[i]nverse transform” component in an example “transform coding system.” *Id.*

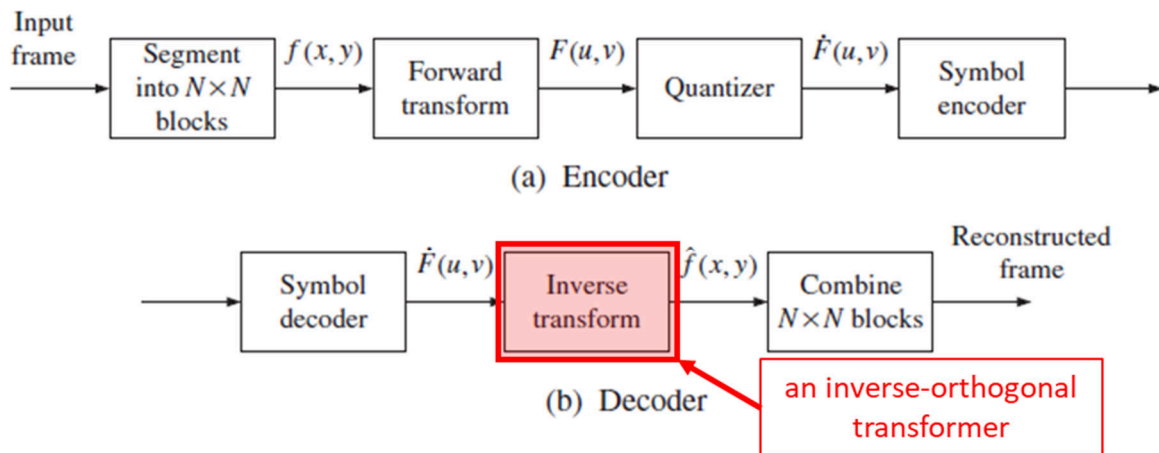
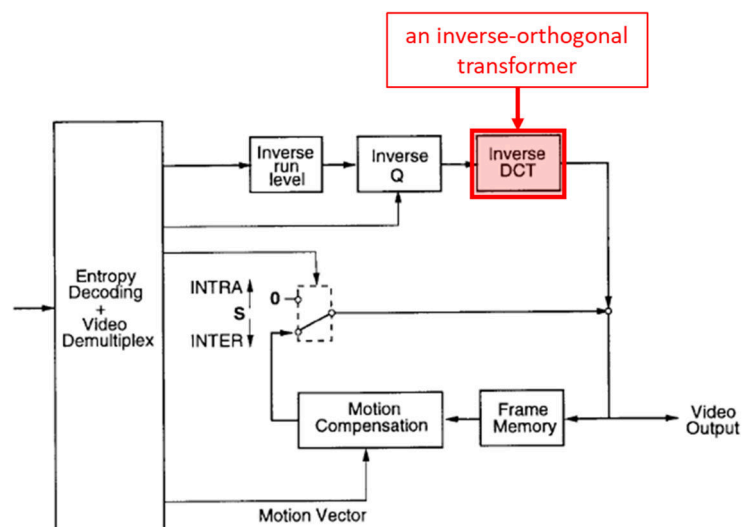


Figure 2.9: Block diagram of a transform coding system

APPLE-1005, FIG. 2.9 (annotated and color added).

Moreover, a POSITA would have recognized or found obvious that the decoder would include an “*inverse-orthogonal transformer*” as this was well-known in the art before the Critical Date. APPLE-1003, ¶[151]. Indeed, Girod corroborates the use of inverse transforms in H.263 decoders. APPLE-1008, 2, FIG. 1(b); APPLE-1003, ¶[151].



APPLE-1005, FIG. 1(b) (annotated and color added).

[6.11.DEC]

As discussed above, the Mualla-Shirani-Saito combination implements a two-stage hybrid error concealment in **both** the **encoder** and the **decoder** by using “interactive techniques,” as taught by Mualla. APPLE-1005, 246-250; APPLE-1003, ¶[152]. As described above in [6.2.COD] and [6.6.COD]-[6.7.COD], the **encoder** of the Mualla-Shirani-Saito combination defines “*a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion vector data*” and “*generate[s] an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture.*” *Supra*, [6.2.COD], [6.6.COD]-[6.7.COD]. Thus, a POSITA would have understood or found obvious that the **decoder** of the Mualla-Shirani-Saito combination would perform analogous operations by using the same two-stage hybrid error concealment technique. APPLE-1003, ¶[152].

Furthermore, Mualla refers to the first-stage temporal concealment as “**motion-compensated** error concealment.” APPLE-1005, 27. Mualla explains that “[a]s in **motion-compensated prediction**, the process of temporal concealment involves two stages: *concealment displacement estimation* and *displacement compensation*, as shown in Figure 9.8(a). For this reason, temporal concealment is sometimes referred to as **motion-compensated concealment.**”

APPLE-1005, 243, Figure 9.8(a) (reproduced below, annotated); APPLE-1003, ¶[153].

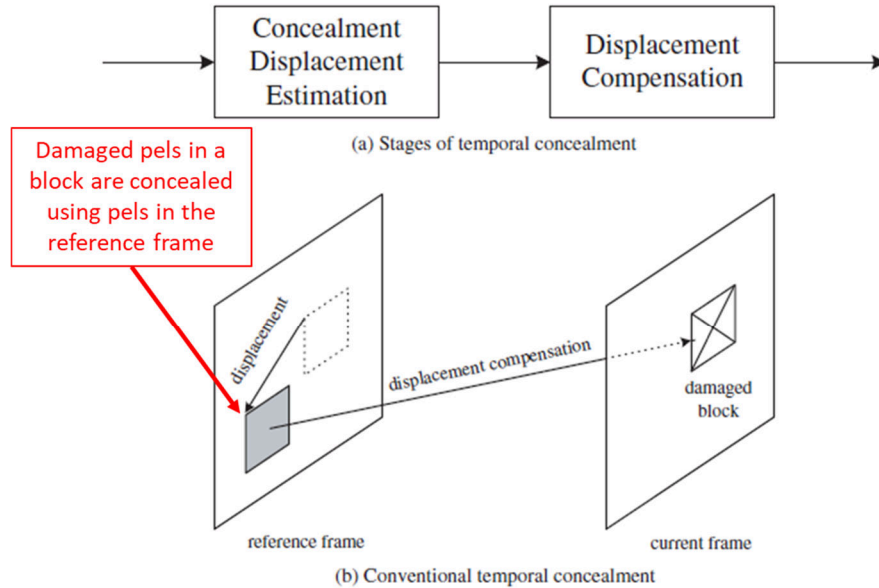


Figure 9.8: Temporal error concealment

APPLE-1005, FIG. 9.8 (annotated)

Moreover, as Mr. Lett explains, “the additional techniques of Shirani and Saito are simply ‘algorithms’ that can be executed by computer program modules” (“*a zone-border motion compensator*”). APPLE-1006, 1-2; APPLE-1007, 6; APPLE-1003, ¶[154]. Accordingly, the encoding features discussed in encoding limitations [6.2.COD], [6.6.COD]-[6.7.COD] would be similarly implemented within the decoder of Mualla-Shirani-Saito. APPLE-1003, ¶[154].

Thus, a POSITA would have understood or found obvious that the **decoder** of the Mualla-Shirani-Saito combination implements “motion compensated error concealment techniques” as part of a two-stage hybrid “error concealment”

algorithm (the claimed “*zone-border motion compensator*”), which includes a first-stage temporal concealment stage (which defines “*a boundary condition of a border that corresponds to the border motion vector data, from the reference picture based on the border motion vector data*”) and a second-stage spatial concealment stage (which “*generate[s] an estimated video signal in each rectangular zone in the picture to be coded, that satisfies Poisson's Equation, thus producing a first predictive picture*”). APPLE-1003, ¶[155].

[6.12.DEC]

Mualla describes a well-known decoding process in which a “prediction” (“*the first predictive picture*”) generated by motion-estimation and motion-compensation (the “*zone border motion compensation*” of Mualla-Shirani-Saito) is “added to” (or summed with) (“*combine the first predictive picture and the decoded residual picture*”) an “error signal” (“*the decoded residual picture*”), which is the result of inverse DCT, to “reconstruct the current frame” (“*generate a decoded moving picture signal*”). APPLE-1005, 27, 76, FIG. 10.8; *supra* [6.6.11.DEC]; APPLE-1003, ¶[156]. Indeed, Mualla’s error concealment technique “*generate[s] a decoded moving picture signal*” by concealing missing/erroneous/damaged blocks, as shown in Figure 10.8. *Id.*

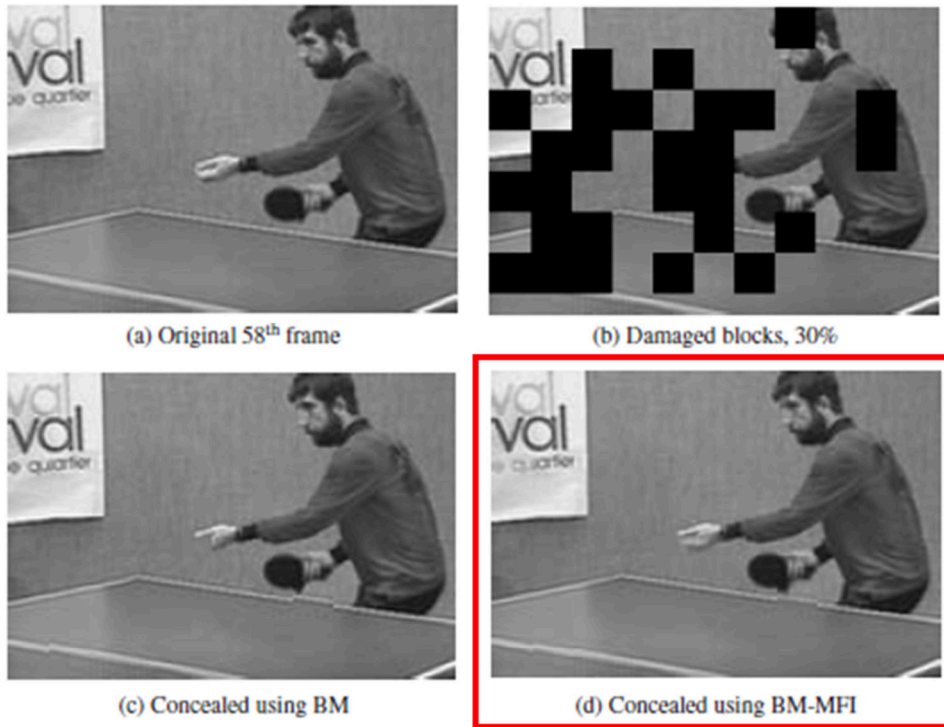


Figure 10.8: Subjective quality of concealed 58th frame of QSIF TABLE TENNIS at 30 frames/s with a block loss rate of 30%

combin[ing] the first predictive picture
and the decoded residual picture to
generate a decoded moving picture signal

APPLE-1005, FIG. 10.8 (annotated).

Thus, for a given frame (picture) that is being decoded, (i) coded data for some macroblocks of the frame are received without error and are decoded using Mualla's conventional "motion estimation" and "motion compensation," whereas (ii) coded data for other macroblocks of the frame are erroneously received and concealed using the Mualla-Shirani-Saito two-stage hybrid error concealment technique. The result is an aggregate predicted frame ("*the first predictive picture*") that is combined with Mualla's "decoded [displaced-frame difference]

DFD” (“*the decoded residual picture*”). The combining technique is analogous to the combining technique implemented in Mualla’s conventional motion-estimation-compensation decoding. APPLE-1005, FIG. 1.1; APPLE-1003, ¶[157].

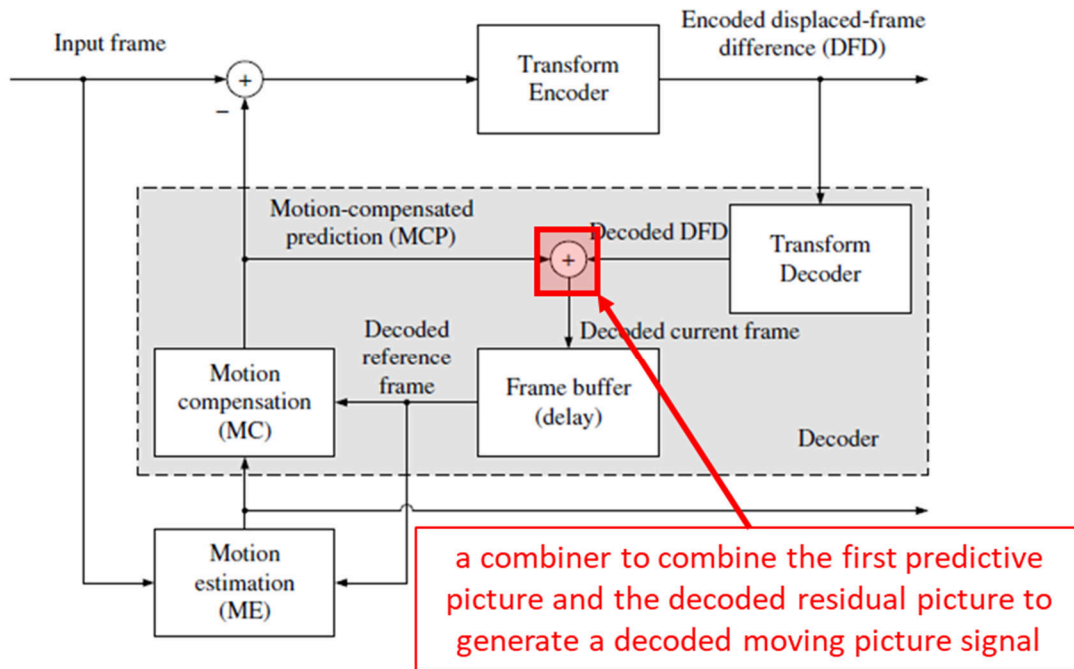
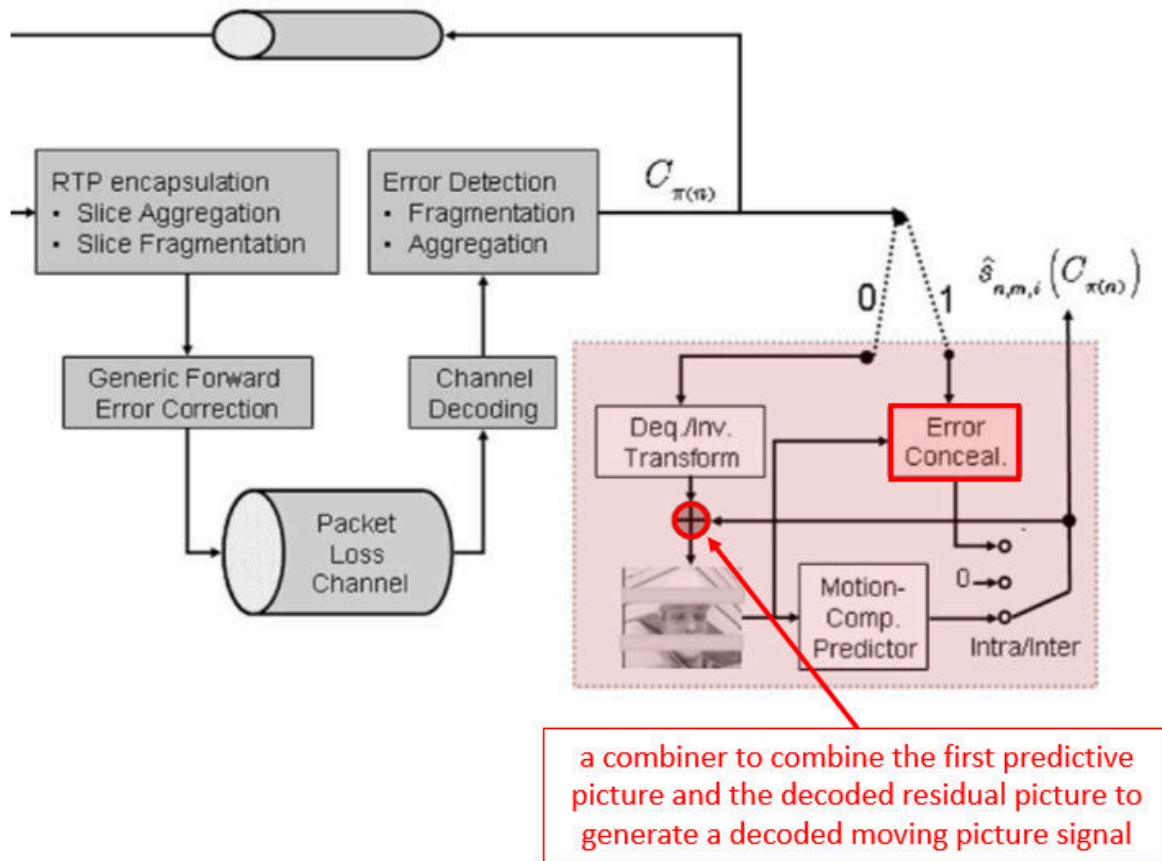


Figure 1.1: Typical video codec

APPLE-1005, FIG. 1.1 (annotated and color added).

A detailed example of such a combining step is provided in Stockhammer (APPLE-1009). As discussed in Section III.A.1, above, decoding is performed on a per-macroblock basis. For a given macroblock of the frame, if an error is detected in the coded data of the macroblock, then the decoder performs “error concealment” and the second selector at the bottom-right-hand corner of the red shaded area is in the upper-most position and provides the error concealment output (the “*first predictive picture*”) to a combiner (annotated). APPLE-1009, 8,

Fig. 6 (reproduced below, annotated); APPLE-1003, ¶[158]. As shown in Fig. 6, the other input to the combiner is the output of the “dequantizer/inverse transform.” *Id.* As explained in [6.9.DEC] and [6.10.DEC], the output of dequantizer/inverse transform is a “*decoded residual picture.*”



APPLE-1009, FIG. 6 (annotated and color added).

Conversely, if no errors are detected in the coded data of the macroblock, then the decoder applies the conventional “motion compensated predictor” and the second selector at the bottom-right-hand corner of the red shaded area is in the lower-most position and provides the output of the motion compensated predictor

to the combiner (annotated). APPLE-1009, 8, Fig. 6; APPLE-1003, ¶[159].

For all macroblocks in the frame (both those received in error and without error), the decoder generates reconstructed samples $\hat{s}_{n,m,i}(o, C_{\pi(n)})$ that form a reconstructed image (the “*decoded moving picture signal*”). APPLE-1003, ¶[160].

[6.13.DEC]

Mualla describes the well-known feature of a decoder including a “*memory*” to store a “*reference picture*” which is the “*decoded moving-picture signal*” that was generated by combining the residual picture and the predictive picture. APPLE-1005, 26-27, FIG. 1.1; *supra*, [6.12.DEC]; APPLE-1003, ¶[161]. For example, Mualla discloses that “reference frame[s]” (“*reference picture[s]*”) are “stored in a frame buffer” (“*a memory*”). *Id.*

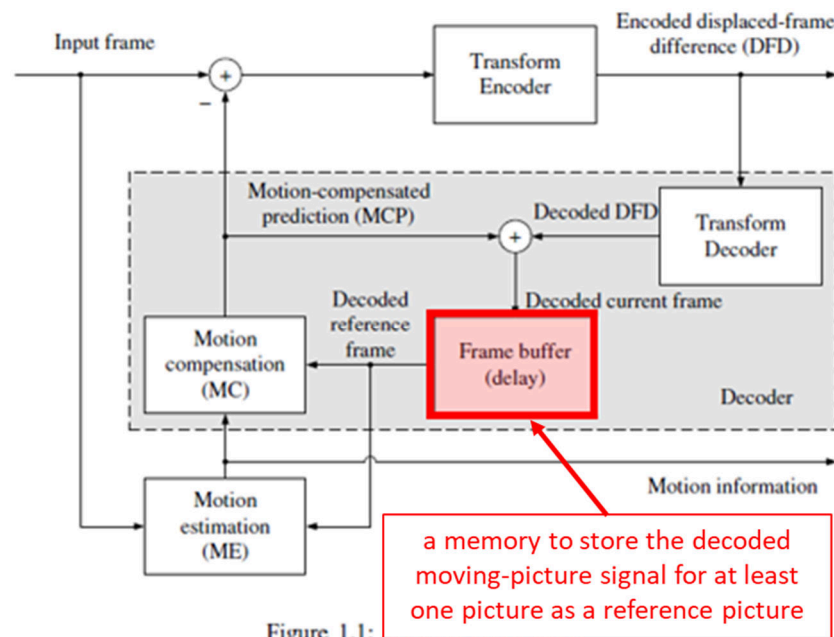
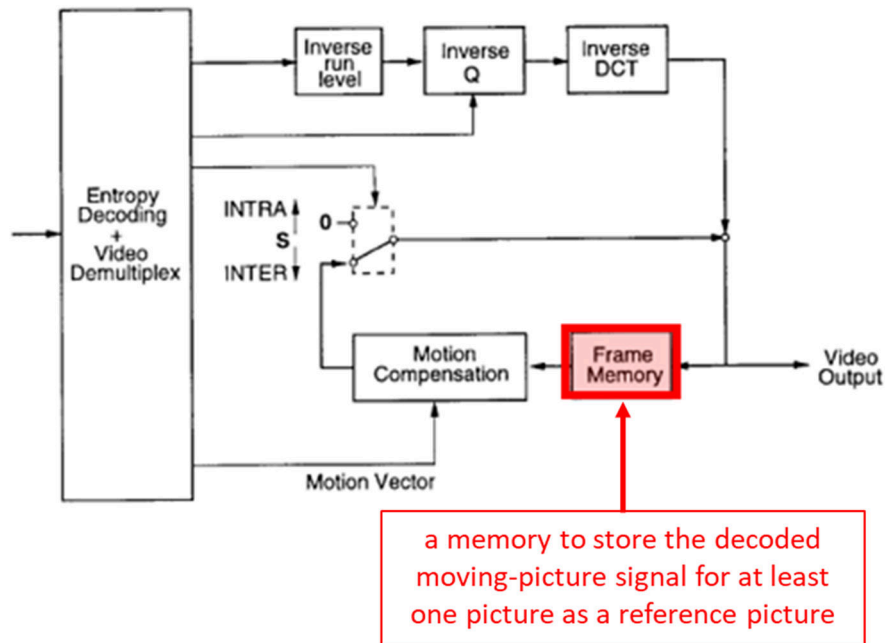


Figure 1.1:

APPLE-1005, FIG. 1.1 (annotated and color added).

Additionally, the use of a “*memory*” to “*store ... a reference picture*” was well-known in the art before the Critical Date. APPLE-1003, ¶[162]. For example, Girod discloses that a H.263 decoder includes a “frame memory” which stores “decoded frames.” APPLE-1008, 2.



APPLE-1008, FIG. 1(b) (annotated and color added).

[6.14.DEC]

Mualla’s “typical video codec” includes the well-known feature of “motion compensation.” APPLE-1005, 26-27, 76, 115-116, FIG. 1.1, 2.16; APPLE-1003, ¶[163]. Indeed, Mualla discloses that a “motion estimation (ME) block uses a motion model to estimate the movement that occurred between [a] current frame and a reference frame” (“*specify a corresponding rectangular zone in the reference picture based on the motion vector data*”). APPLE-1005, 26-27, 116-

117; APPLE-1003, ¶[163]. Then, “[a]t the decoder, the same reference frame is used along with the received motion information to produce the same prediction” (*“thus generating a second predictive picture”*). *Id.* FIG. 1.1 below illustrates motion compensation in Mualla’s typical video codec.

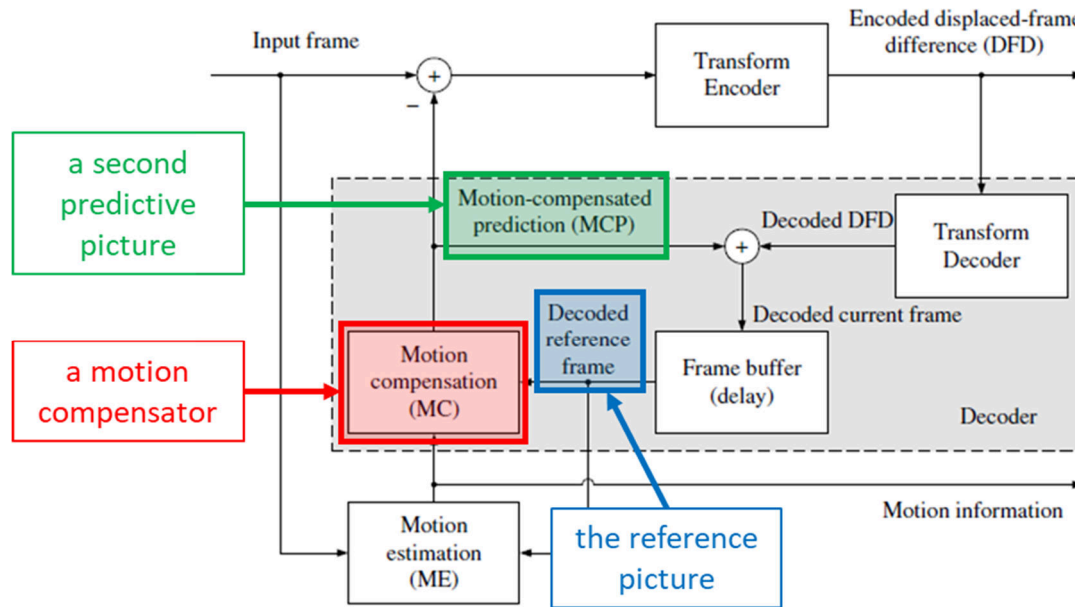
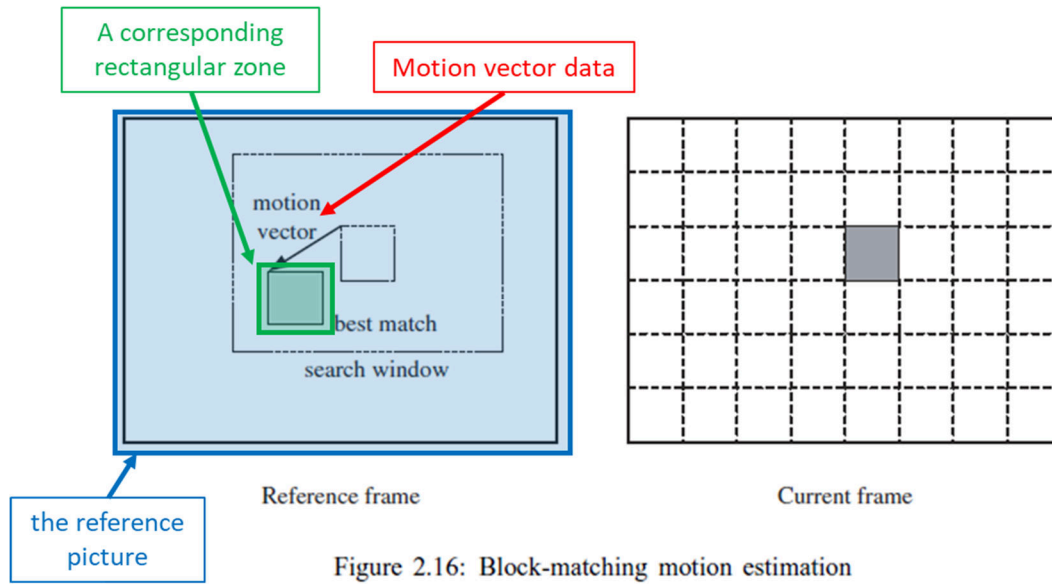


Figure 1.1: Typical video codec

APPLE-1005, FIG. 1.1 (annotated and color added).

Additionally, Mualla discloses a process of using a “motion vector” (*“based on the motion vector data”*) to determine a “block” in the reference frame (*“a corresponding rectangular zone in the reference picture”*) that matches the current frame. APPLE-1005, 76; APPLE-1003, ¶[164]. This process is illustrated in FIG. 2.16 below.



APPLE-1005, FIG. 2.16 (annotated and color added).

Moreover, the techniques of motion compensation were well-known before the Critical Date. APPLE-1003, ¶[165]. Indeed, Shirani describes the use of “motion compensation” (“*a motion compensator*”) and “motion vectors” (“*motion vector data*”) to identify “motion compensated blocks” (“*a corresponding rectangular zone in the reference picture*”). APPLE-1006, 1-2.

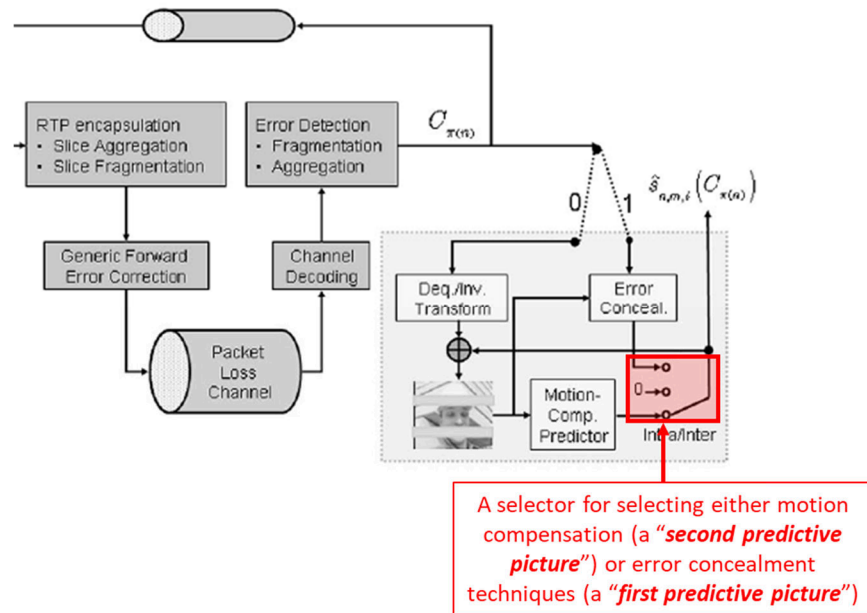
[6.15.DEC]

Mualla discloses that its error concealment techniques can be deployed “[w]hen an error is detected,” and that during this period, the “decoding process is stopped.” APPLE-1005, 265. From at least this disclosure, a POSITA would have recognized and found obvious that tradition motion compensation (producing a “*second predictive picture*”) would execute until an error was detected by the decoder in the bitstream, at which point the error concealment techniques of the

Mualla-Shirani-Saito combination would be used to generate replacement motion vector and video data (a “*first predictive picture*”). APPLE-1005, 265; APPLE-1003, ¶[166].

As Mr. Lett explains, a POSITA would have recognized or found obvious that switching between these two methods of decoding (“*select[ing] either the first predictive picture or the second predictive picture*”) would be effectuated by a switching component, such as a “*selector*.” APPLE-1003, ¶[167]. Indeed, Girod (cited by Mualla) corroborates selecting between two different decoding modes in response to detected errors, as “an erroneous bit stream cannot be gracefully decoded by an ‘off-the-shelf’ video decoder build [sic] for error-free transmission.” APPLE-1008, 4. Girod explains that, in the case of errors, “[s]pecial provisions for error detection, resynchronization, and concealment are required.” *Id.* Stockhammer⁴ also corroborates the use of a “selector” to switch between traditional motion compensation and error concealment. APPLE-1009, FIG. 6; APPLE-1003, ¶[167].

⁴ Although Stockhammer’s teachings are within the general knowledge of a POSITA, a POSITA would have additionally combined Stockhammer into the Mualla-Shirani-Saito combination for the reasons discussed above. *Supra*, §III.A.1-3.



APPLE-1009, FIG. 6 (annotated and color added).

Moreover, Mualla uses a “*selector*” to control inter or intra coding. APPLE-1005, 72, FIG. 3.1. As Mr. Lett explains, using a selector to switch between normal decoding and error concealment “would have been a natural extension of techniques disclosed within Mualla.” APPLE-1003, ¶[168].

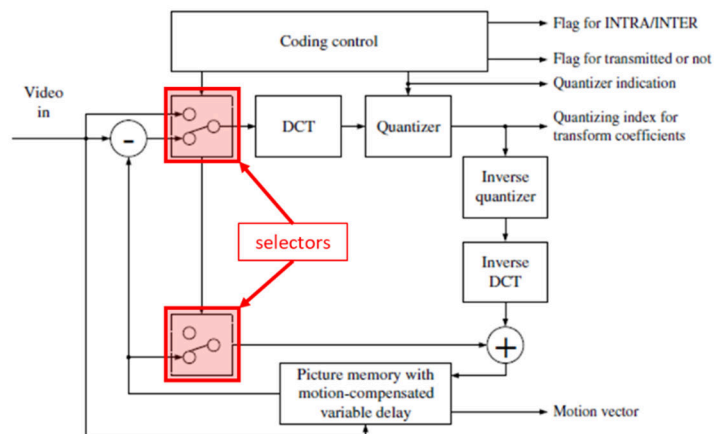


Figure 3.1: H.263 video encoder

APPLE-1005, FIG. 3.1 (annotated and color added).

[6.16.DEC]

As explained above in [6.15.DEC], the Mualla-Shirani-Saito combination would have switched, via a “selector” or equivalent component, between conventional motion compensation and error concealment, depending on the presence of errors in the bitstream. *Supra*, [6.15.DEC]. Additionally, Mualla also discloses that these errors can be detected from “syntax and semantic violations,” the “header information” being one source of this data (“*decoding control data for decoding control from the parameter data*”). APPLE-1005, 231-232.

Additionally, Mualla’s H.263 video encoder includes a “coding control” component (“*a decoding controller*”) that is “responsible for varying several parameters to control the rate or the quality of the coded video.” APPLE-1005, 72, 75, FIG. 3.1; APPLE-1003, ¶[169]. Examples of functions that are controlled by the coding control component are “INTER/INTRA mode decision at the picture or MB level, the update pattern of the forced INTRA refresh, the TRANSMIT=SKIP decision at the MB level, and the QP and its change at the picture, GOB, or MB level.” APPLE-1005, 75; APPLE-1003, ¶[169]. Illustrated below is Mualla’s example H.263 encoder with a “coding control” component that controls operations of the local decoder (the bottom switch).

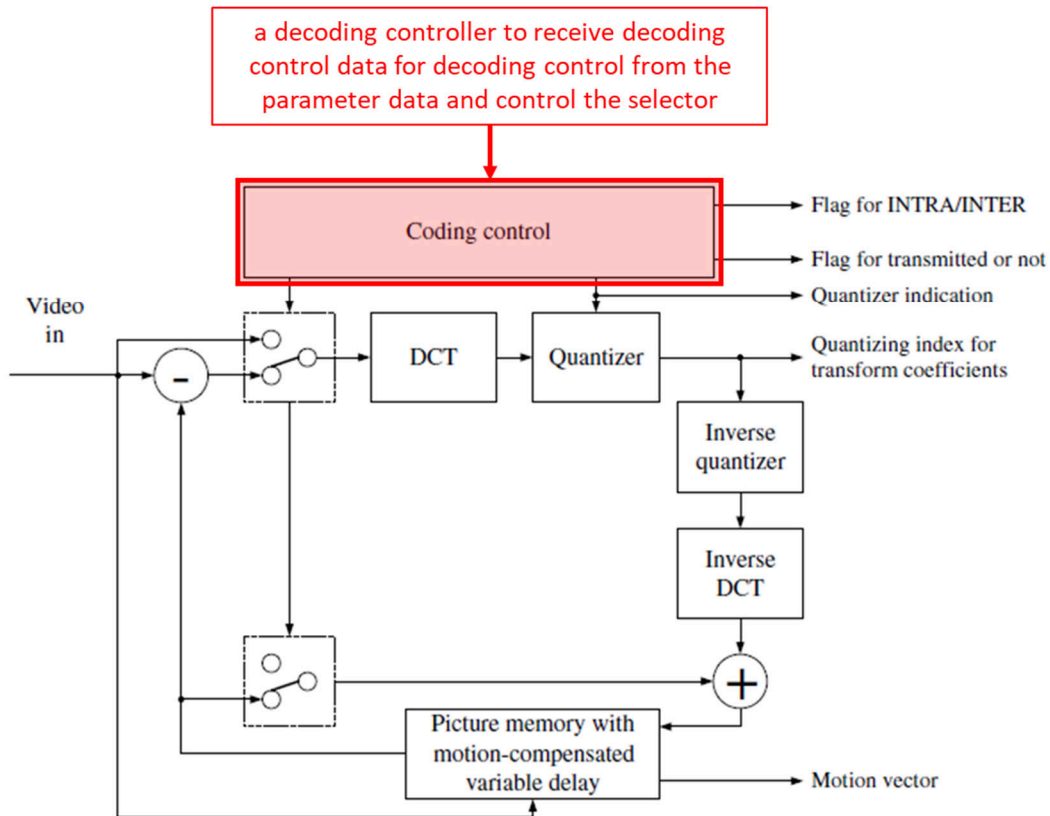


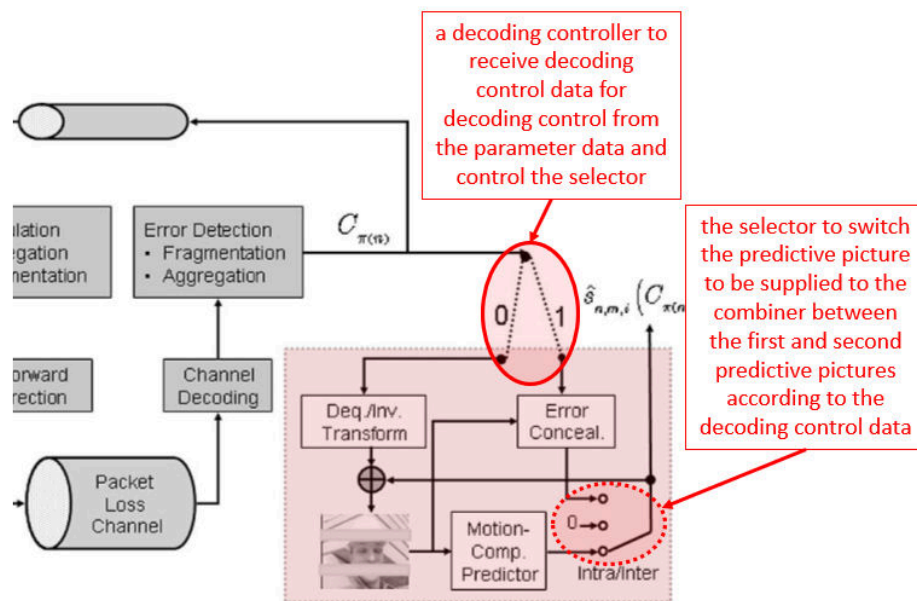
Figure 3.1: H.263 video encoder

APPLE-1005, FIG. 3.1 (annotated and color added).

As Mr. Lett explains, a POSITA would have recognized or found obvious that Mualla's decoder would have implemented an analogous coding control component ("*a decoding controller*") would have "*receive[d] decoding control data for decoding control from the parameter data*" and subsequently, in response to a detected error, "*control[ed] the selector to switch the predictive picture to be supplied to the combiner between the first and second predictive pictures according to the decoding control data.*" APPLE-1003, ¶[170]. Indeed, this function is consistent with the operations of the coding controller described by

Mualla, and as Mr. Lett explains, “Mualla’s coding controller must necessarily process syntax data (*‘decoding control data for decoding control from the parameter data’*) at least because it must make decisions regarding ‘inter’ or ‘intra’ prediction.” APPLE-1003, ¶[170].

Additionally, as illustrated in Stockhammer’s Figure 6 (reproduced below) and as discussed in [6.15.DEC], above, it was known that an H.264-based decoder can implement a *decoding controller* that determines whether an error has been detected (either binary 0 or 1) and controls a *selector* to switch between the outputs of the “error concealer” (the “*first predictive picture*”) or the output of the conventional “motion compensated predictor” (the “*second predictive picture*”) APPLE-1009, FIG. 6; APPLE-1003, ¶[171].



APPLE-1009, FIG. 6 (annotated and color added).

[7.pre]

Mualla discloses “software-based video codecs” (“*non-transitory computer readable device*”) that utilize various “algorithms” (“*a computer program comprising a set of instructions*”). APPLE-1005, 71, 182, 265, 299. For example, Mualla provides an example of a specific software implementation of H.263 called “Telenor” (citing to a “H.263 TMN software codec” as reference [144]) (“*a method for moving-picture decoding*”). APPLE-1005, 265, 299; APPLE-1003, ¶[172].

Moreover, the use of algorithms in implementing codecs was well-known before the Critical Date. APPLE-1003, ¶[173]. Indeed, both Shirani and Stockhammer corroborate the use of “algorithms” and “software” in implementing H.263 and H.264 codecs. APPLE-1006, 1 (“many of the video compression algorithms (e.g., H.263)"); APPLE-1009, 1, 5 (“in the standardization of H.264/AVC for the development of the reference software”).

[7.1.DEC]-[8.3], [10.pre]-[10.13.DEC]

These limitations are obvious as explained above for the corresponding limitations listed in the following table. APPLE-1003, ¶[174].

Claim	Corresponding Claim
[10.pre]	[6.pre]
[7.1.DEC], [10.1.DEC]	[6.1.DEC]
[7.2.COD], [10.2.COD]	[6.2.COD]
[7.3.COD], [10.3.COD]	[6.3.COD]
[7.4.COD], [10.4.COD]	[6.4.COD]
[7.5.COD], [10.5.COD]	[6.5.COD]
[7.6.COD], [10.6.COD]	[6.6.COD]
[7.7.COD], [10.7.COD]	[6.7.COD]
[7.8.DEC], [8.pre], [10.8.DEC]	[6.8.DEC]
[7.9.DEC], [10.9.DEC]	[6.9.DEC]
[7.10.DEC], [10.10.DEC]	[6.10.DEC]
[7.11.DEC], [10.11.DEC]	[6.11.DEC]
[7.12.DEC], [10.12.DEC]	[6.12.DEC]
[8.1]	[6.14.DEC]
[8.2]	[6.15.DEC]
[8.3]	[6.16.DEC]
[10.13.DEC]	[6.13.DEC]

“Moving picture coding” claims 1-5, 9

[1.pre], [4.pre], [9.pre]

Mualla teaches extensive details of an H.263-based video codec system, which implements ***coding apparatus, method, and media*** for decoding ***moving pictures*** (e.g., “video”). APPLE-1005, 23-26, 31, 36, 51-52, 61-63, 227-228, 241-242, 265. For example, as discussed in §III.A.1, above, Mualla’s Figure 9.1 and accompanying description disclose a “typical video communication system” which includes a “source encoder” (“***coding apparatus, method, and media***”). APPLE-1005, Figure 9.1 (reproduced below), 227-228. Further details of Mualla’s “source encoder” are illustrated and described with reference to Figures 2.3, 1.1, and 2.9 of Mualla, as explained in §III.A.1, above. To facilitate understanding, aggregated views of the relation between these different figures of Mualla are shown in Diagrams A and B discussed in §III.A.1, above, which are reproduced again below. APPLE-1003, ¶[175].

Additionally, as discussed above in [7.pre], “***non-transitory computer readable device***” is both disclosed by Mualla and further would have been an obvious implementation choice to a POSITA. *Supra* [7.pre]; APPLE-1003, ¶[176].

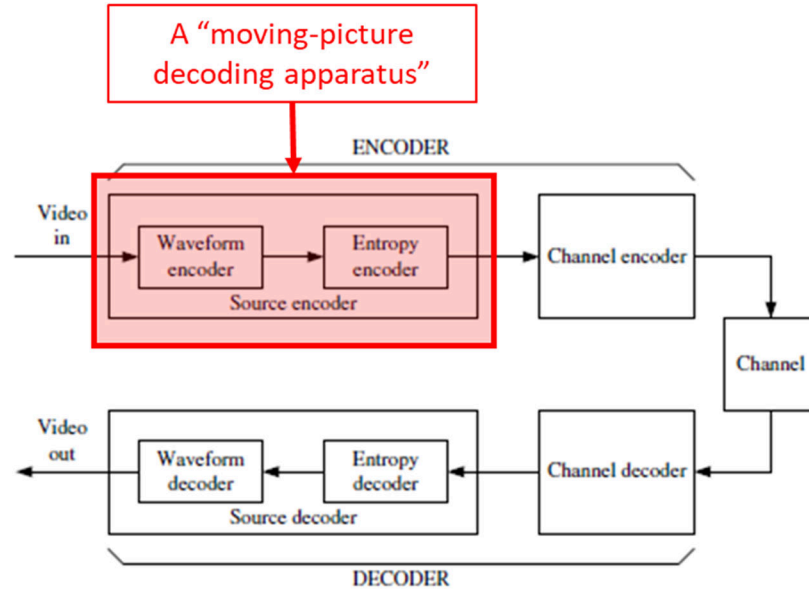


Figure 9.1: Typical video communication system

APPLE-1005, FIG. 9.1 (annotated and color added).

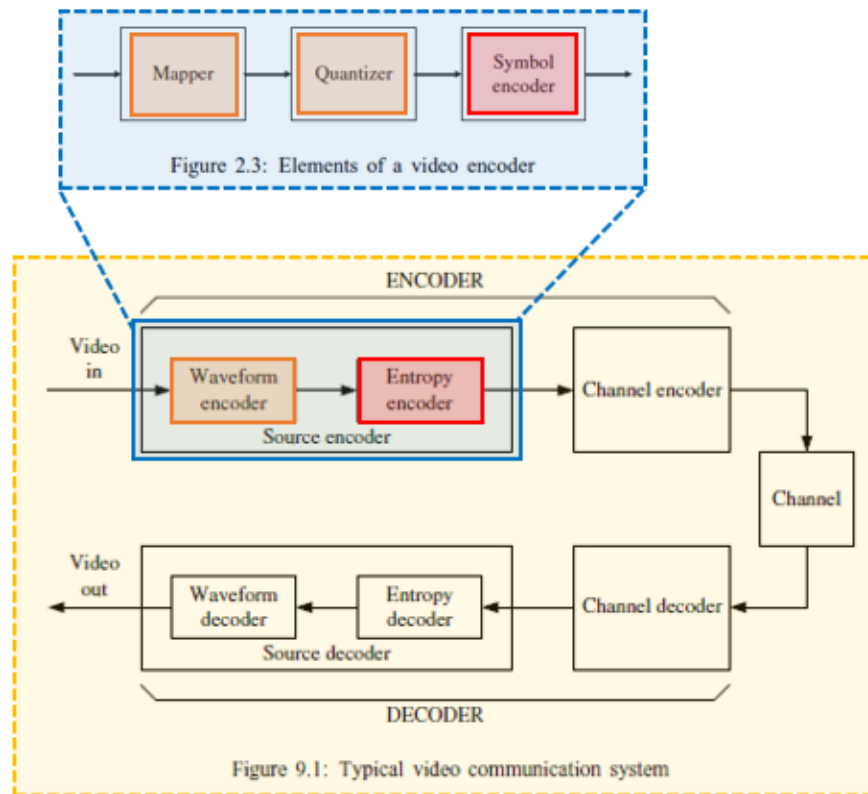


Diagram A (showing the relation between Mualla's Figures 9.1 and 2.3)

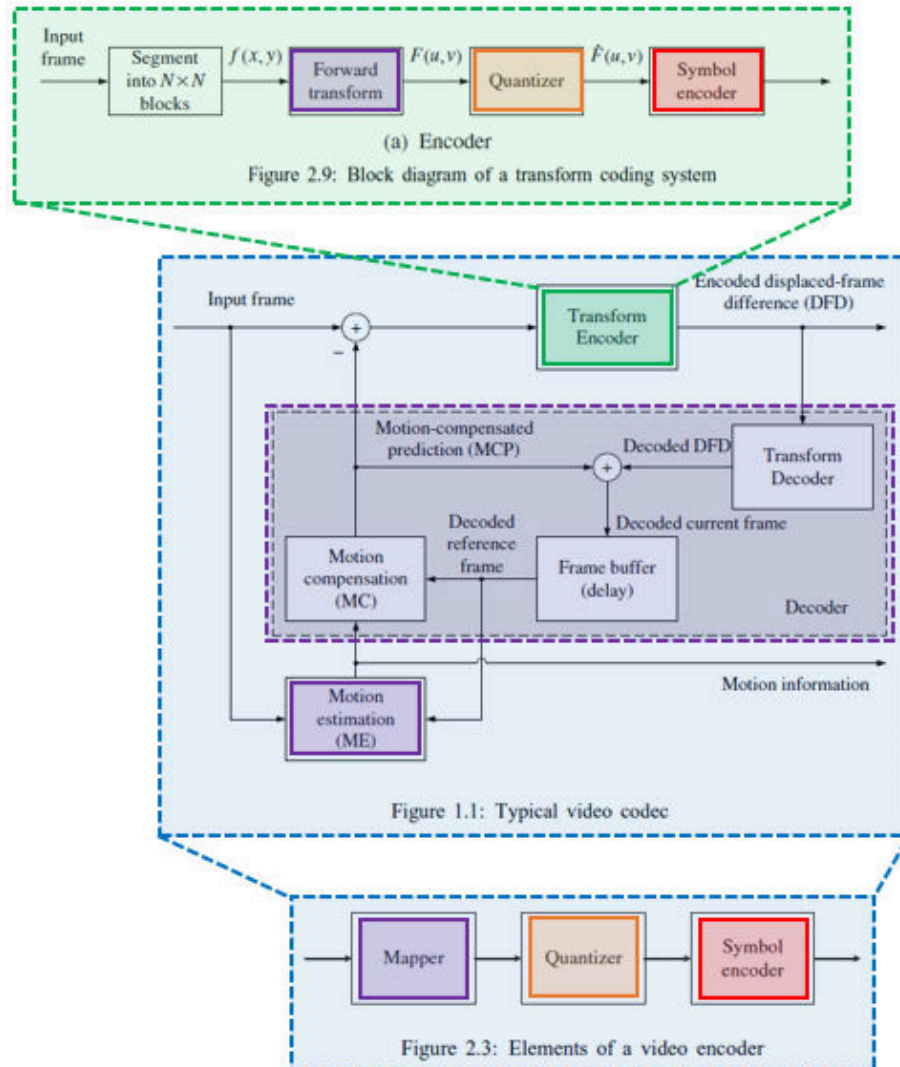


Diagram B (showing the relation between Mualla's Figures 2.3, 1.1, and 2.9)

[1.8.COD], [2.3], [4.8.COD], [5.3], [9.6.COD]

Mualla's "typical video codec" illustrates a component ("*a first subtractor*") that subtracts the "[m]otion-compensated prediction (MCP)" ("*the first predictive picture*") from an "[i]nput frame" ("*the picture to be coded*") to produce, after transform encoding, an "[e]ncoded displaced-frame difference (DFD)" ("*a first residual picture*"). APPLE-1005, 23-26, FIG. 1.1; APPLE-1003, ¶[177].

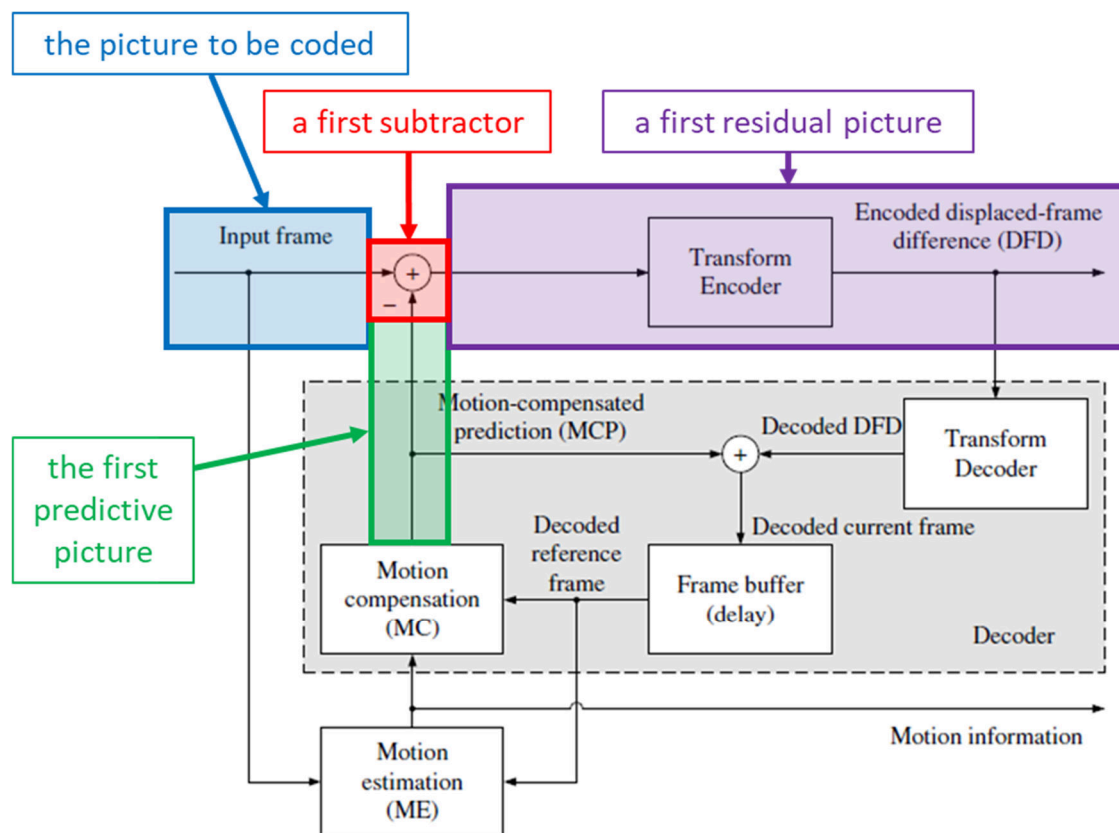


Figure 1.1: Typical video codec

APPLE-1005, FIG. 1.1 (annotated and color added).

Remaining limitations of Claims 1, 3-4, and 9

These limitations are obvious as explained above for the corresponding limitations listed in the following table. APPLE-1003, ¶[214].

Claim	Corresponding Claim
[1.1.COD], [4.1.COD], [9.1.COD]	[6.3.COD]
[1.2.COD], [4.2.COD], [9.2.COD]	[6.4.COD]
[1.3.COD], [4.3.COD], [9.3.COD]	[6.5.COD]
[1.4.COD], [1.5.COD], [4.4.COD], [4.5.COD], [9.4.COD]	[6.6.COD]
[1.6.COD], [4.6.COD], [9.5.COD]	[6.7.COD]
[1.7.COD], [4.7.COD]	[6.3.COD]
[1.9.COD], [1.10.COD], [3.2], [4.9.COD], [4.10.COD], [4.14.COD], [9.7.COD], [9.8.COD]	[6.2.COD]
[1.11.DEC], [4.11.DEC], [9.9.DEC]	[6.9.DEC]
[1.12.DEC], [4.12.DEC], [9.10.DEC]	[6.10.DEC]
[3.1], [4.13.COD]	[6.8.DEC]
[2.1], [5.1]	[6.2.COD]
[2.2], [5.2]	[6.14.DEC]
[2.4], [5.4]	[6.15.DEC]
[5.5]	[6.16.DEC]

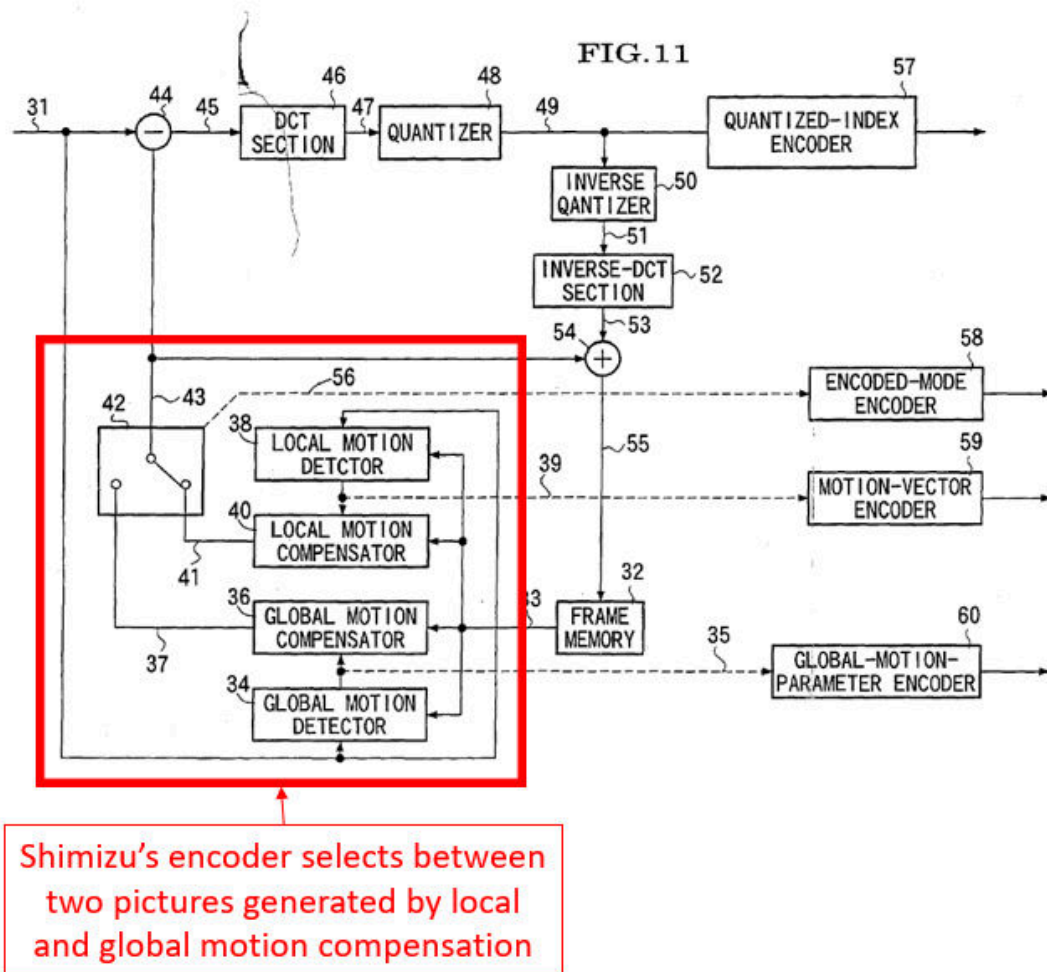
**B. [GROUND 2] – Claims 1-10 are rendered obvious by
Mualla-Shirani-Saito-Shimizu**

1. Shimizu

Shimizu discloses techniques for improving the efficiency of interframe predictive coding by switching between two different modes of motion compensation. APPLE-1024, ¶¶[0047]-[0059], FIG. 11. Specifically, Shimizu discloses “local motion compensation” which “detects, for each macro block ... a motion vector between input picture [] and reference picture” and “generates a local motion-compensating predicted picture [] based on the motion vector.” *Id.*, ¶[0031]. Shimizu notes that “[t]his method equals the conventional motion-compensating method used in the conventional MPEG or the like.” *Id.* As an alternative to the conventional “local” motion compensation, Shimizu also discloses “global motion compensation” which “generates a global motion-compensating predicted picture [] by making the motion vector of each pixel ... act on the reference picture.” *Id.*, [0030]. As such, Shimizu explains that global motion compensation uses a “motion vector representing a global motion over a plurality of areas” while local motion compensation uses a “motion vector for each area.” *Id.*, ¶¶[0049]-[0050]-[0052]; APPLE-1003, ¶[75].

Using a reference picture, Shimizu’s encoder generates both a “global motion-compensating predicted picture” and a conventional “local motion-compensating predicted picture,” and then selects between these two predicted

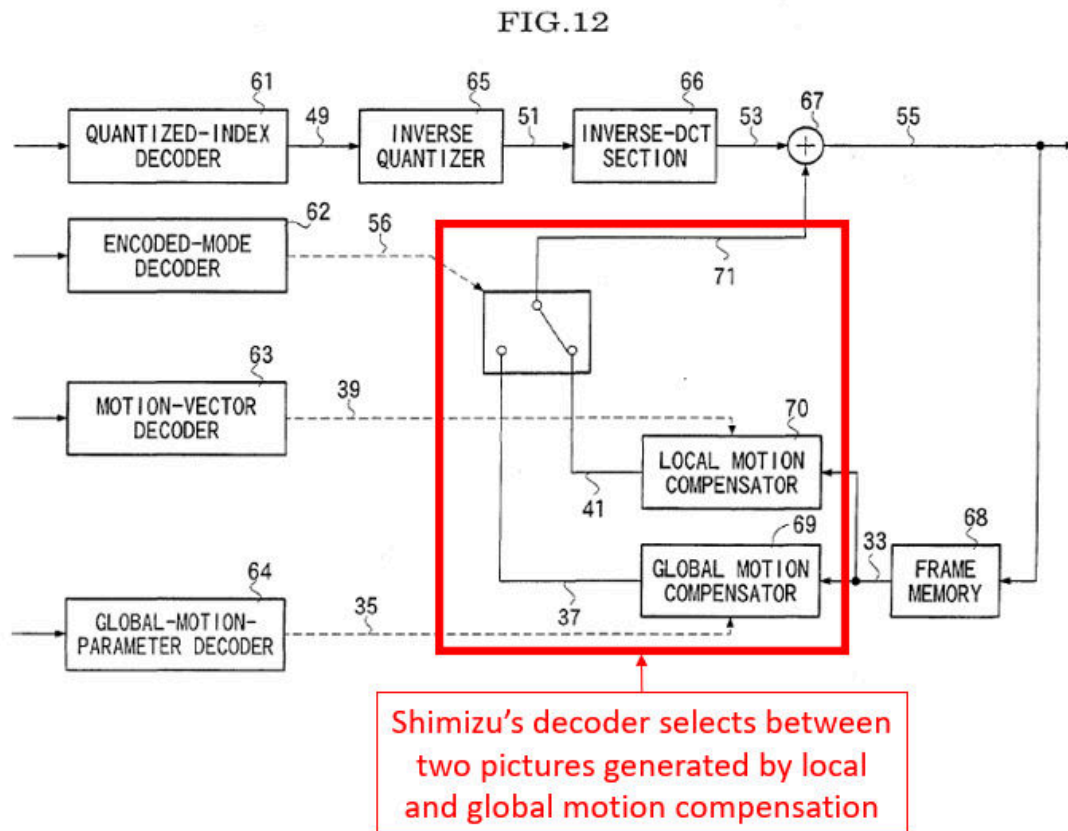
pictures based on “whichever has the smaller error with respect to the input picture.” *Id.*; ¶¶[0030]-[0036], FIG. 11 (reproduced below, showing the encoder implementing encoding mode selector 42 that selects between the two predicted pictures provided from lines 37 and 41); APPLE-1003, ¶[76].



APPLE-1024, FIG. 11.

Analogously, Shimizu's decoder also generates and selects between two motion-compensating predicted pictures generated by global motion compensation and conventional local motion compensation. APPLE-1024, ¶¶[0034], [0035],

FIG. 12 (reproduced below, showing the decoder implementing a switch 71 that selects between the two predicted pictures provided from lines 37 and 41); APPLE-1003, ¶[77]. Specifically, Shimizu discloses that the decoder's switch 71 is controlled by encoded-mode choice information 56 received from the encoder. *Id.*, ¶[0035]; APPLE-1003, ¶[77]. As such, Shimizu's encoder and decoder both generate global and local motion compensating predicted pictures and select between the two predicted pictures, such that the same predicted picture is selected at the encoder and decoder. *Id.*



APPLE-1024, FIG. 12.

2. Combination of Mualla-Shirani-Saito and Shimizu

As discussed above, the combination of Mualla, Shirani, and Saito in Ground 1A (and the combination of Mualla-Stockhammer-Shirani and Saito in Ground 1B) implements a two-stage hybrid error concealment technique where in the first stage, an “initial estimate” X_i of the erroneous/missing block is generated using “temporal” error concealment and then, the second stage refines the initial estimate X_i by applying “spatial” concealment in which “the information around each missing block” is utilized to estimate pixels values in the interior of the estimated block X_i (as taught by Mualla-Shirani). APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[97]. The Mualla-Shirani-Saito combination uses a Poisson-equation-solving image approximation technique (e.g., PHLCT, as taught by Saito) to implement the spatial error concealment in the second stage of the two-stage hybrid error concealment technique. APPLE-1003, ¶[97].

Furthermore, a POSITA would have understood that error resilience techniques, such as the Mualla-Shirani-Saito hybrid two-stage motion-compensated technique discussed above, are not limited to error-prone environments, but can also be used to enhance coding performance in error-free environments. APPLE-1003, ¶[98] (citing Stockhammer, APPLE-1009, 10, as explaining “the tools for increased **error resilience**, in particular those to limit error propagation, do not significantly differ from those used for compression

efficiency. Features like multiframe prediction or intra MB coding are **not exclusively error-resilience** tools. They are **also** used to **increase coding efficiency** in **error-free** environments providing a **trade-off** that is left to the encoder”). In particular, Mualla notes the similarities between motion compensated prediction (for error-free environments) and temporal concealment (for error resilience), explaining that “[a]s in motion-compensated prediction, the process of temporal concealment involves two stages: concealment displacement estimation and displacement compensation, as shown in Figure 9.8(a). For this reason, temporal concealment is sometimes referred to as motion-compensated concealment.” APPLE-1005, 243; APPLE-1003, ¶[98].

Indeed, a POSITA would have understood that the Mualla-Shirani-Saito hybrid 2-stage motion compensated predictor using a Poisson-equation-solving image approximation technique could naturally operate as a motion compensated predictor in error-free environments (as suggested by Mualla and corroborated by Stockhammer, above) because it requires similar inputs (motion vectors and a reference picture stored in frame memory) as a motion compensated predictor to generate similar outputs (a predicted picture). APPLE-1003, ¶[99]. Further, a POSITA would have understood that Poisson approximation techniques, as disclosed in Saito, are not limited to error concealment but instead is more generally applicable to “image approximation” with a goal of improving “the

perceptual quality of these approximations” and a desire for a “perceptually correct metric” that can improve the “visual quality” of approximated images. APPLE-1007, 6, 18, 27; APPLE-1003, ¶[99]. As Mr. Lett explained in section III.A, numerous research papers and patents had, several years before the Critical Date, described techniques for image editing and image reconstruction that involved solving Poisson’s equation. APPLE-1003, ¶[99] (citing Perez, APPLE-1010, Raskar, APPLE-1012, Sun/Sun-128, APPLE-1013 and APPLE-1014, and Fattal, APPLE-1015). Therefore, a POSITA would have understood and found obvious that Saito’s Poisson approximation technique, as implemented in the second stage of the Mualla-Shirani hybrid 2-stage prediction, is also applicable to general motion compensated prediction, even in an error-free environment. APPLE-1003, ¶[99].

Thus, a POSITA would have understood that error resilience techniques, such as the Mualla-Shirani-Saito hybrid two-stage motion-compensated technique described above, could also be used in error-free environments to improve coding efficiency. In particular, a POSITA would have understood and found obvious that a coding system could utilize an error resilience technique, such as the Mualla-Shirani-Saito hybrid two-stage motion-compensated technique, as an alternative to conventional motion compensated prediction to achieve a desired trade-off of performance metrics such as coding efficiency. APPLE-1003, ¶[100].

Shimizu describes details of how to implement a system to adaptively achieve such a trade-off. As discussed in section III.B.1, above, Shimizu discloses a coding system that dynamically selects, on a per-block basis, between two different motion compensation techniques and Shimizu further explains that this technique can “reduce the amount of generated code with respect to the motion vector” and thus “improve the efficiency of the motion-vector prediction.”

APPLE-1024, ¶[0046]. In particular, Shimizu’s system selects between (i) a conventional motion compensation technique that obtains motion vectors for each macroblock and generates predicted pictures for the motion vector and (ii) an alternative motion compensation technique that operates on individual pixels to provide a global motion compensation over a larger area. APPLE-1003, ¶[101]. Furthermore, a POSITA would have understood that Shimizu’s alternative motion compensation technique (“global motion compensation”) was also a well-known error resilience technique. APPLE-1003, ¶[101] (citing Salama, APPLE-1025, 1, as disclosing “[e]rror concealment of shape using global motion compensation” and Soares, APPLE-1026, 2, as disclosing “[g]lobal motion compensation concealment”).

Therefore, in view of the knowledge in the art that error resilience techniques (such as the Mualla-Shirani-Saito hybrid two-stage motion-compensated technique) could also be used in error-free environments as an

alternative to conventional motion compensation, and in view of Shimizu's teaching of a system that adaptively selects between conventional motion compensation and an alternative motion compensation to "reduce the amount of generated code with respect to the motion vector" and thus "improve the efficiency of the motion-vector prediction," and a POSITA would have found it obvious to implement a hybrid 2-stage motion compensated predictor using a Poisson-equation-solving image approximation technique (as taught by Mualla-Shirani-Saito) in Shimizu's system in error-free environments to switch/select between (i) the hybrid 2-stage motion compensated prediction and (ii) the conventional motion compensated prediction. APPLE-1003, ¶[102].

The modified portions of Mualla's encoder (in Fig. 1.1) are shown in Diagram E, below, where the components added to Mualla's encoder are annotated in red (based on annotated Fig. 11 of Shimizu that includes the Mualla-Shirani-Saito hybrid 2-stage motion compensated predictor to facilitate explanation). APPLE-1005, Fig. 1.1; APPLE-1003, ¶[103]. As shown, the modification simply amounts to adding two components to Mualla's encoder: (i) a hybrid 2-stage motion compensated predictor using a Poisson-equation-solving image approximation technique (as taught by Mualla-Shirani-Saito, annotated in red) and (ii) an encoding mode selector 42 (as taught by Shimizu, annotated in red) that selects between Mualla's conventional motion compensation/estimation (annotated

in blue and green) and the hybrid 2-stage motion compensated predictor of Mualla-Shirani-Saito. APPLE-1003, ¶[103].

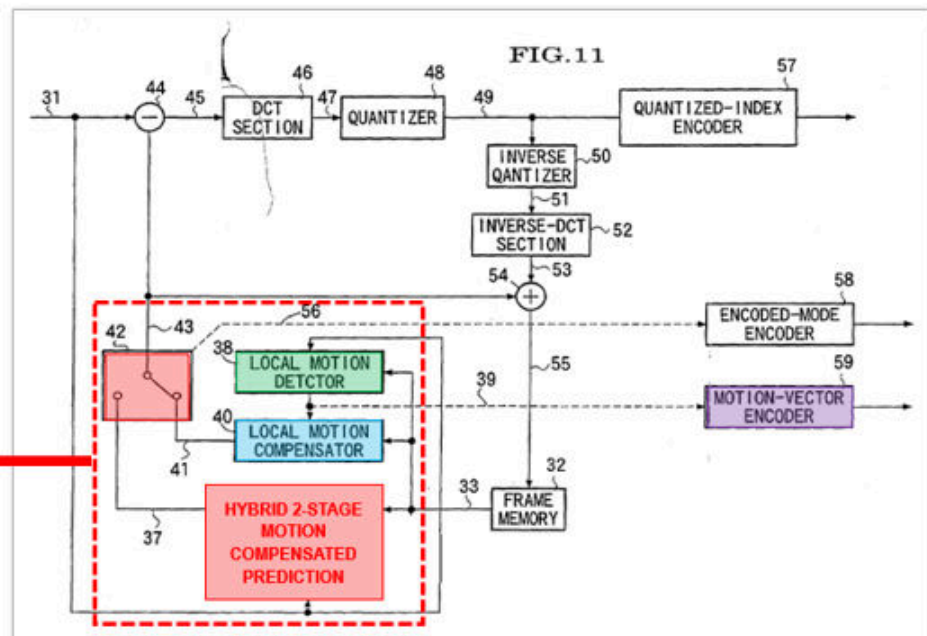
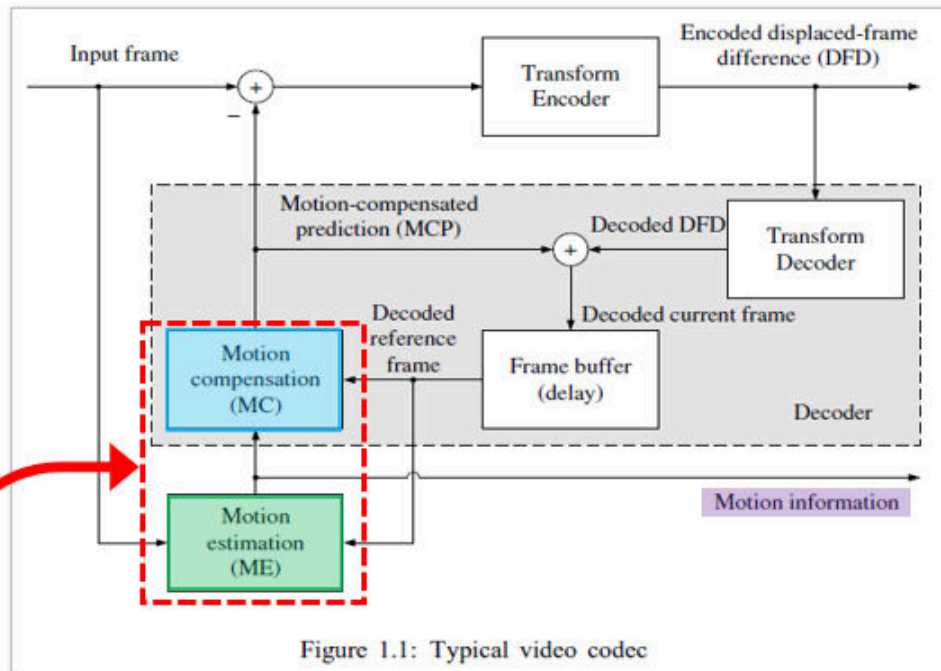


Diagram E (encoder with selective motion compensation/estimation)

In Mualla's modified encoder, the hybrid 2-stage motion compensation predictor operates using the basic principles described in section III.A.3 and III.A.5, namely by implementing a "first stage" that applies Shirani's "temporal" concealment to find a block X_i in the previous frame with border pixels that have the smallest sum of "border errors" relative to surrounding blocks, as taught by Shirani. APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[104]. The resulting block X_i is an "initial estimate" of the block to be decoded. Then, the "second stage" refines the "initial estimate" X_i by applying "spatial" concealment in which "the information around" the block to be decoded is utilized to estimate pixels values in the interior of the initially estimated block X_i . APPLE-1005, 242-245; APPLE-1006, 1-3; APPLE-1003, ¶[104]. Here, the second-stage spatial concealment of Mualla-Shirani is modified to implement Saito's PHLCT to interpolate the interior pixels of the Mualla-Shirani initial estimate X_i . APPLE-1003, ¶[104].

An analogous modification would be made to Mualla's decoder, as shown in Diagram F, below (based on annotated Fig. 12 of Shimizu that includes the Mualla-Shirani-Saito hybrid 2-stage motion compensated predictor to facilitate explanation), by adding: (i) a hybrid 2-stage motion compensated predictor using a Poisson-equation-solving image approximation technique (as taught by Mualla-Shirani-Saito) and (ii) a switch 71 (as taught by Shimizu) that receives encoded-

mode choice information 56 from the encoder and selects between Mualla's conventional motion compensation/estimation and the hybrid 2-stage motion compensated predictor of Mualla-Shirani-Saito. APPLE-1024, [0035], FIG. 12; APPLE-1003, ¶[105].

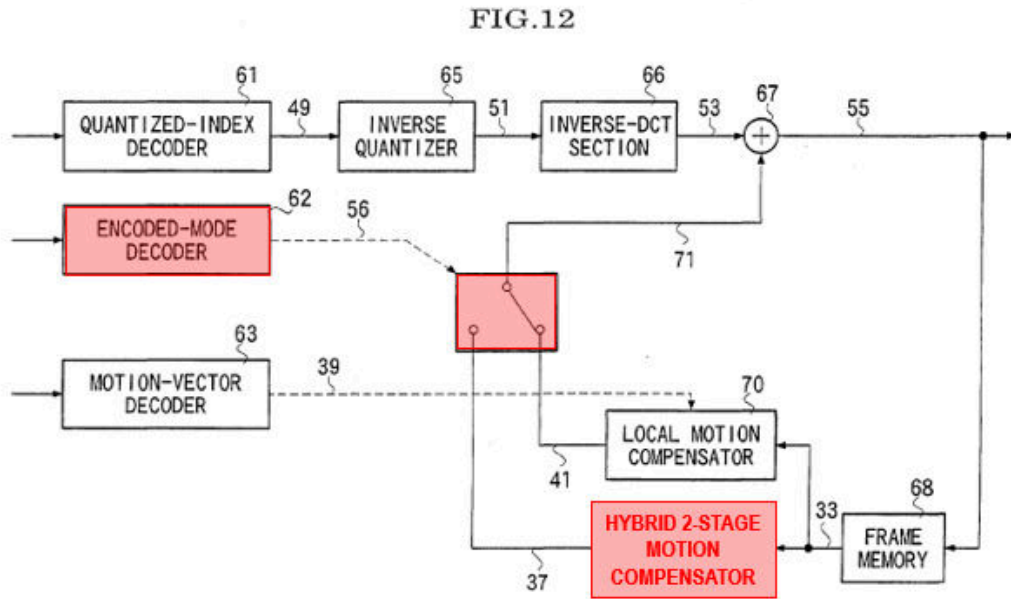


Diagram F (decoder with selective motion compensation/estimation)

As explained below in more detail, combining Mualla-Shirani-Saito-Shimizu (with or without the error-resilience details of Stockhammer) would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results; and the (2) use of known techniques to improve similar devices in the same way. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421, 82 USPQ2d 1385, 1395-97 (2007); APPLE-1003, ¶[106].

In view of the applicability of the Shirani-Mualla-Saito hybrid 2-stage prediction technique to error-free motion compensated prediction, a POSITA would have recognized the advantages of further applying Shimizu's technique to select between the Shirani-Mualla-Saito hybrid 2-stage prediction technique and conventional motion compensated prediction, particularly because this is aligned with Mualla's goal of minimizing information transmitted between encoder and decoder would have led to greater system efficiency. APPLE-1005, 40 ("The aim of video coding is to reduce, or compress, the number of bits used to represent video"), 90 ("The radio spectrum is a limited and scarce resource. ... Given the enormous amount of data generated by video, the use of efficient coding techniques is vital. One of the most important factors that decide the coding efficiency of a video codec is the motion estimation and compensation technique"). Using this knowledge, a POSITA would have been motivated to apply Shimizu's techniques of selecting between two different forms of motion compensation to the Mualla-Shirani-Saito system to select between (i) conventional motion compensation and (ii) the Mualla-Shirani-Saito hybrid 2-stage motion compensation. APPLE-1003, ¶[107].

In the resulting combination of Mualla, Shirani, Saito, and Shimizu, the encoder would have produced two predictive pictures—one using conventional motion compensation and one using hybrid 2-stage motion compensated

prediction—and compared the results to determine which picture produced the most efficient result (as disclosed in Shimizu, e.g., “reduce the amount of generated code with respect to the motion vector” and thus “improve the efficiency of the motion-vector prediction”). APPLE-1024, ¶¶[0032], [0037], [0046], [0058]-[0059]. The encoder would notify the decoder of this selection, through the encoded-mode choice information 56 which controls the decoder’s switch 71. APPLE-1024, ¶¶[0032], [0035]. In the scenario of error-free operation, the hybrid 2-stage motion compensated prediction of Mualla-Shirani-Saito would generate predictions of blocks to be encoded/decoded, rather than blocks that are missing/erroneous. APPLE-1003, ¶[108].

As explained above, A POSITA would have understood and found obvious that Shimizu’s techniques of reducing the amount of generated code would aligned with Mualla’s goal of reducing the number of bits used to represent video by using efficient coding techniques. APPLE-1003, ¶[109]. For example, Shimizu’s techniques of comparing the results of two different motion compensation techniques would have reduced the amount of information transitted by the encoder, allowing the combined system to more efficiently encode information and make the most of precious bandwidth—motivations that are echoed by Mualla. APPLE-1024, ¶¶[0032], [0037], [0046], [0058]-[0059]; APPLE-1003, ¶[109].

3. Analysis

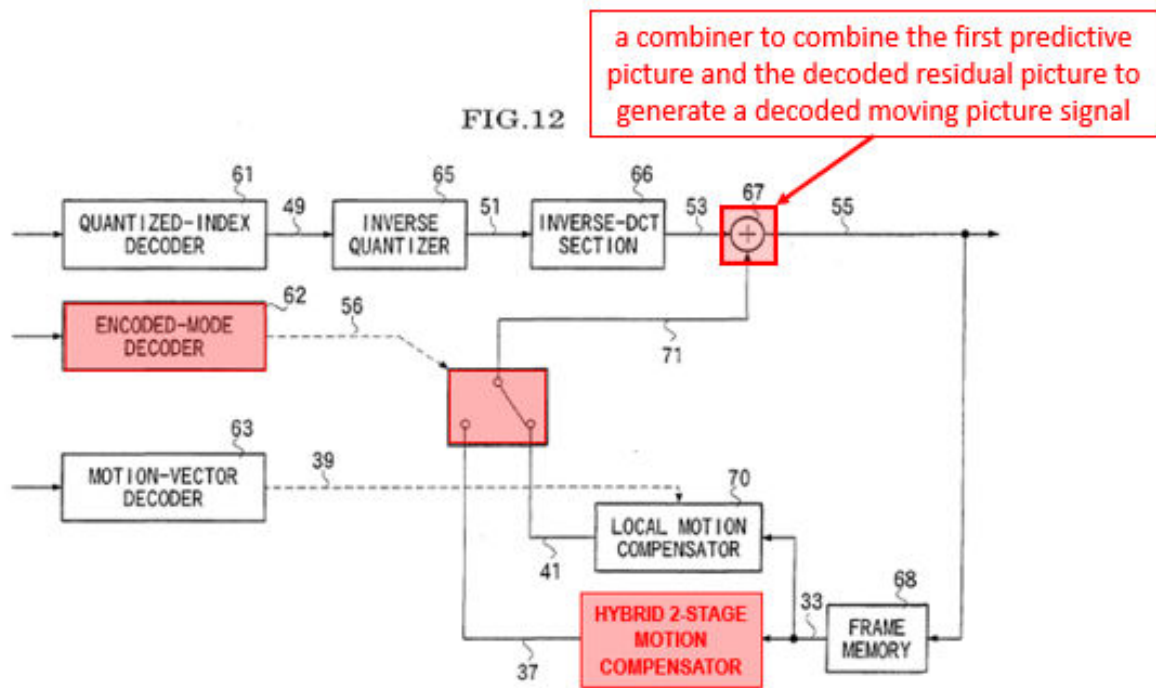
[6.pre]-[6.14.DEC]

Each of these elements are rendered obvious for the same reasons as discussed in their respective sections in Grounds 1A/1B above, supplemented with the additional remarks below. *Supra*, §III.A.6.[6.pre]-[6.14.DEC]; APPLE-1003, ¶[179]. Specifically, as explained in section III.B.2, above, in the combination of Mualla, Shirani, Saito, and Shimizu, the only change to Mualla's encoder and decoder is to add (i) a 2-stage hybrid motion compensated predictor using Poisson approximation techniques (as taught by Mualla-Shirani-Saito) and (ii) a selector/switch that selects between this 2-stage hybrid motion compensated predictor and Mualla's conventional motion estimator/compensator. As such, the basic operations of Mualla's encoder and decoder with respect to claim elements [6.pre]-[6.14.DEC] would remain unchanged in the Mualla-Shirani-Saito-Shimizu system operating in an error-free environment, with supplemental details of certain claim elements described below. APPLE-1003, ¶[179].

For [6.8.DEC], as explained in section III.A.6, Mualla discloses both "border motion vector data" (motion vectors of neighboring blocks around the block to be encoded/decoded) and as explained in section III.A.1, Mualla also discloses conventional "motion vector data" (motion vector of the block to be encoded/decoded). *Supra*, III.A.1; [6.8.DEC] in section III.A.6; APPLE-1003,

¶[180].

For [6.12.DEC], as explained in III.A.6 and III.A.1, Mualla discloses a combiner that is implemented both in the encoder (APPLE-1005, Fig. 1.1) and in the decoder. APPLE-1005, 27, 76, FIG. 10.8; *supra* [6.6.11.DEC] in III.A.6; APPLE-1003, ¶[181]. Further, Shimizu provides an example of a combiner implemented in the decoder. APPLE-1024, Fig. 12, [0036] (“difference picture 53 and predicted picture 43 are added to each other by adder 67 so that local decoded picture 55 is generated”); APPLE-1003, ¶[181].

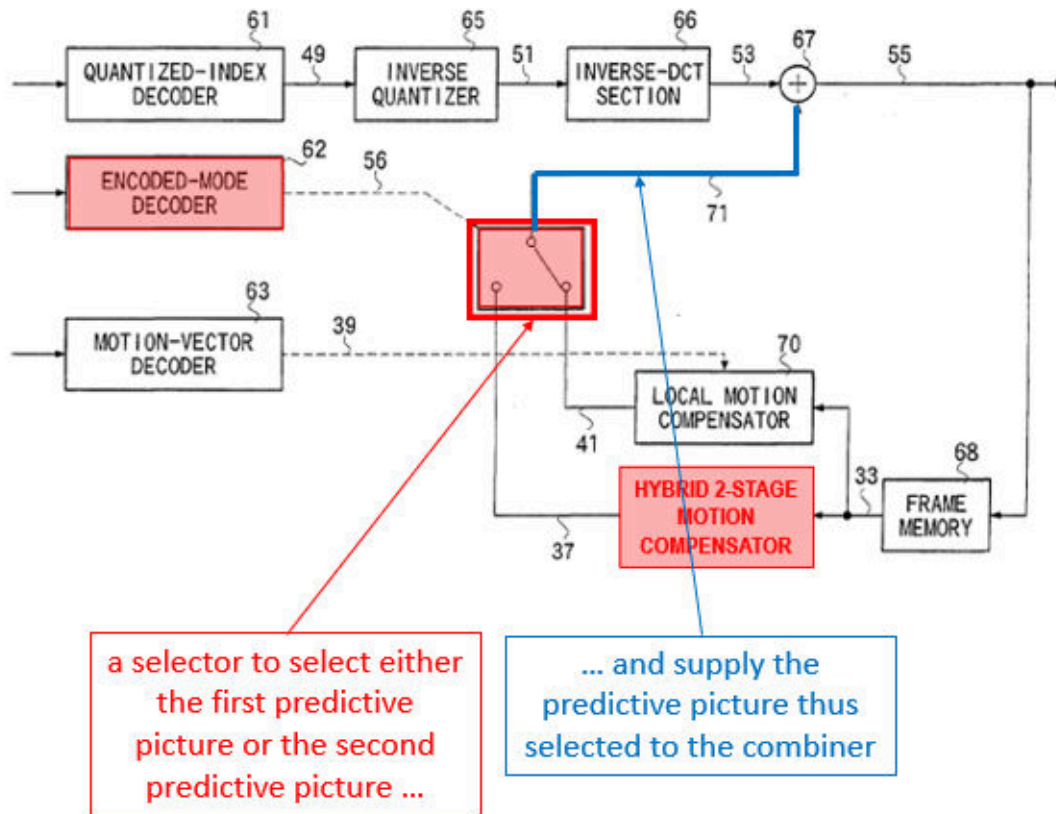


APPLE-1024, FIG. 12 (annotated and color added, modified to show the Mualla-Shirani-Saito hybrid 2-stage motion compensator implemented as the alternative motion compensator).

[6.15.DEC]

Shimizu discloses that “encoded mode information 56 activates switch 71” (*“a selector to select either the first predictive picture or the second predictive picture ...”*) so that one of two motion compensation pictures is selected and provided to adder 67 (*“... and supply the predictive picture thus selected to the combiner”*). APPLE-1024, ¶¶[0034]-[0036], FIG. 12. In particular, the decoder’s switch 71 is controlled by encoded mode information 56 which is received from the encoded mode encoder 58 of the encoder (which communicates with the decoder’s encoded mode decoder 62 and vice-versa). *Id.*, ¶¶[0030]-[0033], FIG. 11; APPLE-1003, ¶[182].

FIG.12

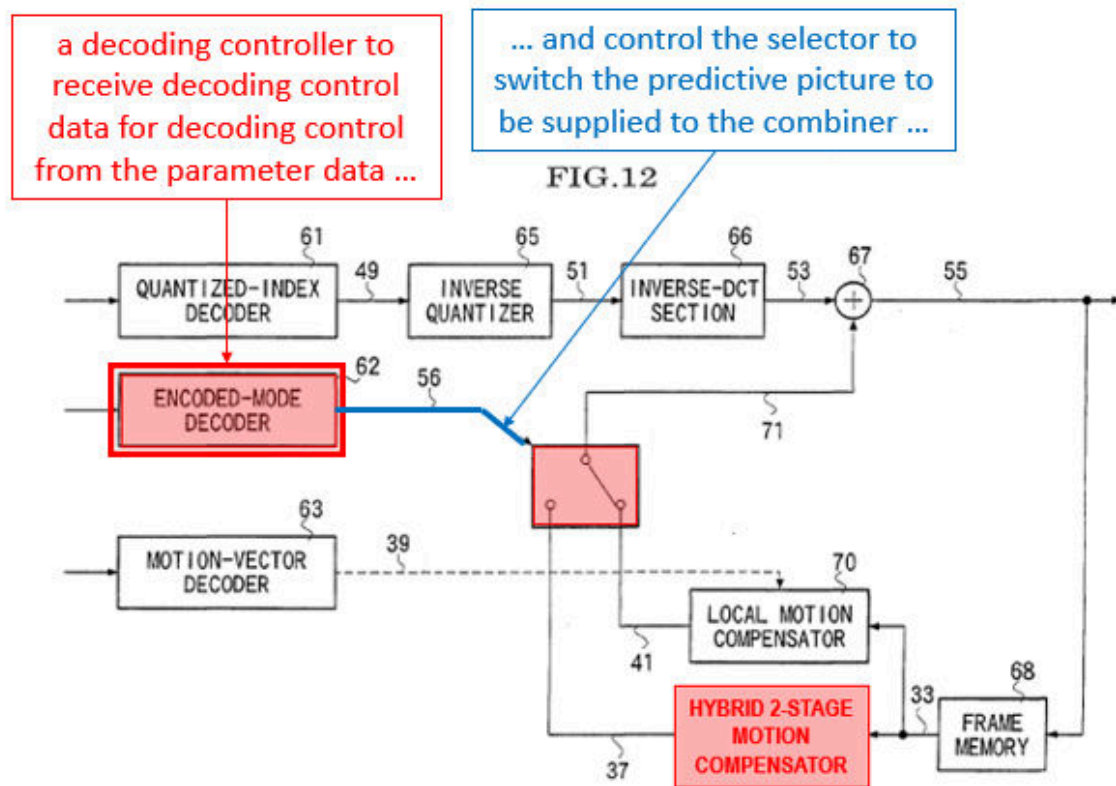


APPLE-1024, FIG. 12 (annotated and color added, modified to show the Mualla-Shirani-Saito hybrid 2-stage motion compensator implemented as the alternative motion compensator).

[6.16.DEC]

As discussed above in 6.15.DEC, Shimizu discloses an “encoded-mode decoder 62” (“*a decoding controller*”) within the decoder that “decodes encoded-mode choice information 56” (“... *to receive decoding control data for decoding control from the parameter data ...*”) that activates switch 71. APPLE-1024, ¶¶[0034]-[0036], FIG. 12. As discussed above in 6.15.DEC, switch 71 causes the

selection of one of two motion compensation pictures, and in the combination of Mualla-Shirani-Saito-Shimizu, would have switched the mode of decoding between conventional motion compensation and the Mualla-Shirani-Saito two-stage hybrid motion compensated prediction (“... *and control the selector to switch the predictive picture to be supplied to the combiner between the first and second predictive pictures according to the de-coding control data*”). APPLE-1005, 247; APPLE-1024, ¶¶[0034]-[0036], FIG. 12; APPLE-1003, ¶[183].



APPLE-1024, FIG. 12 (annotated and color added, modified to show the Mualla-Shirani-Saito hybrid 2-stage motion compensator implemented as the alternative motion compensator).

Remaining claims 1-5, 7-10

Each of these claims are rendered obvious for the same reasons as discussed in Grounds 1A/1B above, in view of the additional disclosure of Shimizu discussed above with respect to elements [6.15.DEC]-[6.16.DEC]. *Supra*, §III.A.6; APPLE-1003, ¶[184].

IV. PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION

Petitioner believes that discretionary denial is unwarranted, and yet, Petitioner intends to utilize the bifurcated briefing process contemplated by the March 26, 2025, Stewart Memorandum to rebut contentions if offered by Patent Owner to the contrary. APPLE-1019.

V. CONCLUSION AND FEES

The Challenged Claims are unpatentable. Petitioner authorizes charge of fees to Deposit Account 06-1050.

VI. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Apple Inc. is the petitioner and real party-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

Petitioner is not aware of any disclaimers, reexamination certificates for the '025 Patent. The '025 Patent was previously challenged in IPR2024-00374, which was instituted and subsequently terminated before a final written decision.

The '025 Patent is the subject of a number of civil actions including: *Advanced Coding Technologies LLC v. Apple Inc.*, 2-24-cv-00572 (EDTX), filed July 22, 2024; *Advanced Coding Technologies LLC v. Google LLC*, 2-24-cv-00353 (EDTX), filed May 10, 2024; *Samsung Electronics Co., Ltd. v. Advanced Coding Technologies LLC*, IPR2024-00372 (PTAB), filed January 08, 2024; *Advanced Coding Technologies LLC v. Samsung Electronics Co., Ltd. et al.*, 2-22-cv-00499 (EDTX), filed December 30, 2022; and *Advanced Coding Technologies LLC v. LG Electronics, Inc. et al.*, 2-22-cv-00501 (EDTX), filed December 30, 2022.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR50095-0247IP1@fr.com	Jeremy J. Monaldo, Reg. No. 58,680 Won S. Yoon, Reg. No. 71,058 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 IPR50095-0247IP1@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR50095-0247IP1@fr.com.

Respectfully submitted,

Dated: May 12, 2025

/Jeremy J. Monaldo/

W. Karl Renner, Reg. No. 41,265

Jeremy J. Monaldo, Reg. No. 58,680

Won S. Yoon, Reg. No. 71,058

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

(Control No. IPR2025-00991)

Attorneys for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,959 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: May 12, 2025

/Jeremy J. Monaldo/

W. Karl Renner, Reg. No. 41,265

Jeremy J. Monaldo, Reg. No. 58,680

Won S. Yoon, Reg. No. 71,058

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

Attorneys for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6I(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on May 12, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided via Federal Express, to the Patent Owner, by serving the correspondence address of record as follows:

FABRICANT LLP
411 Theodore Fremd Road
Suite 206 South
Rye, NY 10580
(212) 257-5795

/Crena Pacheco/

Crena Pacheco
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
pacheco@fr.com