

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LENOVO (UNITED STATES), INC.,
Petitioner,

v.

TELEFONAKTIEBOLAGET LM ERICSSON,
Patent Owner.

U.S. Patent No. 10,708,618

Issued: July 7, 2020

Application No. 16/440,274

Filed: June 13, 2019

Title: REFERENCE PICTURE SIGNALING

DECLARATION OF DR. CHARLES D. CREUSERE

LENOVO
EXHIBIT 1003

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND ENGAGEMENT	5
II. BACKGROUND AND QUALIFICATIONS.....	5
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '618 PATENT.....	11
IV. UNDERSTANDING OF PATENT LAW	13
V. PERSON OF ORDINARY SKILL IN THE ART	17
VI. DEVELOPMENT OF VIDEO CODING TECHNOLOGIES.....	18
A. Video Coding Standards.....	27
1. Advanced Video Coding (H.264/AVC).....	27
2. High Efficiency Video Coding (H.265/HEVC).....	45
VII. THE '618 PATENT.....	46
A. The '618 Patent's Specification	46
B. The Challenged Claims	50
C. The Relevant Prosecution History Of The '618 Patent And Its Family Members.....	51
VIII. CLAIM CONSTRUCTION	54
IX. GROUND 1: CLAIMS 1-7 ARE OBVIOUS OVER MULROY IN VIEW OF H.264/AVC.....	55
A. Mulroy (U.S. 2009/0052531; EX1004)	55
B. H.264/AVC (EX1005)	60
C. Combining Mulroy And H.264/AVC	67
D. The Mulroy-H.264/AVC Combination.....	71

1.	The Mulroy-H.264/AVC Combination Can Maintain The Same Set Of Frames In The DPB From One Slice To the Next.....	72
2.	The Mulroy-H.264/AVC Combination Can Remove Frames From The DPB To Recover From The Loss Of MMCO Commands	74
3.	The Mulroy-H.264/AVC Combination Can Add Frames To The DPB To Recover From The Loss Of Encoded Data For A Previous Frame	79
E.	The Mulroy-H.264/AVC Combination Renders Claims 1-7 Of The '618 Patent Obvious.....	84
1.	Claim 1	84
2.	Claim 2	94
3.	Claim 3	96
4.	Claim 4	98
5.	Claim 5	102
6.	Claim 6	103
7.	Claim 7	104
X.	GROUND 2: CLAIMS 8-19 ARE OBVIOUS OVER MULROY IN VIEW OF H.264/AVC AND RAVEENDRAN	105
A.	Raveendran (U.S. 2006/0165176; EX1006)	105
B.	Combining Mulroy, H.264/AVC, And Raveendran	106
C.	The Mulroy-H.264/AVC-Raveendran Combination	108
1.	Claim 8	108
2.	Claim 9	110
3.	Claim 10	111

4.	Claim 11	112
5.	Claim 12	113
6.	Claim 13	116
7.	Claim 14	118
8.	Claim 15	121
9.	Claim 16	122
10.	Claim 17	123
11.	Claim 18	125
12.	Claim 19	125
XI.	OBJECTIVE INDICIA OF NON-OBVIOUSNESS	126

I, Charles D. Creusere, do hereby declare:

I. INTRODUCTION AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Lenovo in connection with the above-captioned Petition for Inter Partes Review (“IPR”) to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 10,708,618 (hereinafter “the ’618 patent”).

2. I am being compensated at my usual and customary rate for the time I spend in connection with this IPR. My compensation is not affected by the outcome of this IPR.

3. Specifically, I have been asked to provide my opinions regarding whether claims 1-19 (each a “challenged claim” and collectively the “challenged claims”) of the ’618 Patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of June 30, 2011. It is my opinion that each challenged claim would have been obvious to a POSITA after reviewing the prior art discussed herein.

II. BACKGROUND AND QUALIFICATIONS

4. This section contains a summary of my educational background, career history, publications, and other relevant qualifications. My full curriculum vitae is attached as Appendix A to this declaration.

5. I received a Bachelor of Science degree in Electrical and Computer Engineering from the University of California at Davis in 1985. I received a Master's of Science degree in Electrical and Computer Engineering from the University of California at Santa Barbara in 1990, and I received my Ph.D. in Electrical and Computer Engineering, also from the University of California at Santa Barbara in 1993.

6. I have more than 30 years of experience with data compression, decompression, and data storage.

7. I am currently a Full Professor in the Klipsch School of Electrical & Computer Engineering at New Mexico State University, and I hold the Frank Carden Endowed Chair in Telemetry and Telecommunications. I was an Assistant Professor at New Mexico State from January 2000 until I became an Associate Professor in 2004. I have been a Full Professor since August 2010. My research and coursework at New Mexico State have focused on digital signal and image and video processing. Much of the research I have done over the course of my career is in the area of image and video compression.

8. My first exposure to the field of signal compression came in the fall of 1989, when I took a course entitled Vector Quantization and Signal Compression at UCSB from Prof. Allen Gersho—an internationally renowned researcher in the area of speech compression. As my Ph.D. research progressed, I began to focus on

transform-based compression as my main application area. My first paper dealing with image compression was published in 1991. I have since written 24 other journal and conference papers directly related to compression, and I am the named inventor on two issued United States patents related to compression.

9. Since joining the faculty of New Mexico State University in 2000, I have taught numerous classes at both the graduate and undergraduate levels. At the graduate level, I have taught the following courses: Signal Compression (EE573), Image Processing (EE596), Digital Signal Processing (EE545), Pattern Recognition (EE565), Advanced Linear Systems (EE555), Telemetry Systems (EE585), Information Theory (EE586), Adaptive Signal Processing (EE594), Multirate Signal Processing and Wavelets (EE595), and Neural Signal Processing (EE597). At the undergraduate level, I have taught the following courses: Linear Algebra & Probability (EE200), Signals and Systems I (EE312), Image Processing (EE446), Introduction to Digital Signal Processing (EE395), and Digital Communications (EE497).

10. From 1993 through 1999, I was a Researcher and Team Leader, at the Naval Air Warfare Center, China Lake. At China Lake, my research efforts focused on high speed image and video compression technologies including embedded wavelet video compression. In particular, I developed and demonstrated a real-time video streaming system that combined scalable video coding with

motion compensated prediction to communicating video over TCP/IP networks from a remote server to a centralized client.

11. From 1990 through 1993, I worked as a Research Assistant in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. In this position, I worked on subband coding (compression) and multirate filter bank theory. I also implemented real-time filter banks on a digital signal processor. In the summer of 1992, I worked at AT&T Bell labs where I developed and simulated new methods of extremely low bit rate video coding for video telephone applications.

12. From 1985 through 1989, I worked as a Design Engineer at the Naval Weapons Center, China Lake. In this role, I built and tested the guidance electronics for various laser guided munitions. This project included mixed analog and digital circuit design as well as the programming of an embedded digital signal processor. I also developed software for an advanced video processor and studied ground target tracking.

13. A full listing of my publications is found in my curriculum vitae. See Appendix A.

14. I have published 17 peer-reviewed journal articles and 94 conference papers, including 9 journal articles and 30 conference papers directly related to data compression; the following are representative:

- C.D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," IEEE Trans. on Image Processing, Vol. 6, No. 10, pp. 1436-42, Oct. 1997.
- C.D. Creusere, "Fast embedded compression for video," IEEE Trans. on Image Processing, Vol. 8, No. 12, pp. 1811-16, Dec. 1999.
- C.D. Creusere, "Motion compensated video compression with reduced complexity encoding for remote transmission," Signal Processing: Image Communications, Vol. 16, No. 7, pp. 627-42, April 2000.
- C.D. Creusere, "Fast embedded compression for video," IEEE Trans. On Image Processing, Vol. 8, No. 12, pp. 1811-16, Dec. 1999.
- Castorena, J.; Creusere, C.D., "The Restricted Isometry Property for Banded Random Matrices," Signal Processing, IEEE Trans. on Image Processing, Vol. 62, No. 19, pp. 5073-84, Oct. 1, 2014.
- Castorena, J.; Creusere, C.D., "Sampling of Time-Resolved Full-Waveform LIDAR Signals at Sub- Nyquist Rates," IEEE Trans. on Geoscience and Remote Sensing, Vol.53, No.7, pp. 3791-802, July 2015.
- S. Kandadai and C.D. Creusere, "Scalable Audio Compression at Low Bitrates," IEEE Trans. on Audio, Speech, and Language Processing, Vol. 16, No. 5, pp. 969-79, July 2008.

15. I am a named co-inventor on two issued patents, both relating specifically to image and video compression. I am the listed inventor on U.S. Patent No. 6,148,111 entitled “Parallel digital image compression system for exploiting zerotree redundancies in wavelet coefficients” and U.S. Patent No. 6,466,698 entitled “Efficient embedded image and video compression system using lifted wavelets.”

16. In addition to the experience and publications listed above, I have also received the following awards and distinctions that are relevant to the subject matter of this declaration. I am currently deputy Editor-in-Chief for IEEE Transactions on Image Processing as well as an Area Editor for IEEE Open Journal on Signal Processing. I have previously served as an Associate Editor for IEEE Transactions on Image Processing from 2010 through 2014. I have also served in this capacity from 2002 through 2005. From 2008-2013, I served as an Associate Editor for IEEE Transactions on Multimedia. I also served as a Senior Area Editor for IEEE Transactions on Image Processing from 2016-2022.

17. In 2004, I served as the co-general chair for the IEEE Digital Signal Processing Workshop in Taos, New Mexico. In 2012 and 2014, I served as the co-technical chair for the Southwest Symposium on Image Analysis and Interpretation held in Santa Fe, New Mexico and San Diego, CA, respectively. In addition, I served as the technical chair for the 2015 and 2021 International Telemetering

Conference held in Las Vegas, NV in October. I am also a member of the technical program committees for the IEEE Data Compression Conference, IEEE International Conference on Image Processing and the IEEE Acoustics, Speech, and Signal Processing Conference.

18. The above outline of my experience with communications systems is not comprehensive of all of my experience over my years of technical experience. Additional details of my background are set forth in my curriculum vitae, attached as **Appendix A** to this Declaration, which provides a more complete description of my educational background and work experience.

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '618 PATENT

19. In preparing this declaration, I have reviewed the following materials bearing Exhibit Nos. that I understand are being referenced in the IPR to which my declaration accompanies:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 10,708,618 (“ the '618 patent ”)
1002	File History of U.S. Patent No. 10,708,618
1003	N/A
1004	U.S. Patent Pub. No. 2009/0052531 (“ Mulroy ”)
1005	ITU-T Recommendation H.264 (Nov. 2007) <i>SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of</i>

No.	Description
	<i>audiovisual services – Coding of moving video - Advanced video coding for generic audiovisual services (“H.264/AVC”)</i>
1006	U.S. Patent Application Pub. No. 2006/0165176 (“ Raveendran ”)
1007	N/A
1008	ITU-T Recommendation H.263 (Nov. 2000), <i>Annex U: Enhanced reference picture selection mode</i> , International Telecommunication Union (prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on Nov. 17, 2000)
1009	File History of U.S. Patent Application No. 15/299,997, which issued as U.S. Patent No. 9,807,418
1010	Comparison of Challenged Claims
1011	ITU-T Recommendation H.263 (Feb. 1998) <i>SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video – Video coding for low bit rate communication (“H.263 (1998)”)</i>
1012	N/A
1013	ITU-T Recommendation H.263 (Jan. 2005) <i>SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video – Video coding for low bit rate communication (“H.263”)</i>
1014	Recommendation ITU-T H.265 (Apr. 2013) <i>SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video – High efficiency video coding (“H.265”)</i>
1015	Stephan Wenger, <i>H.264/AVC Over IP</i> , IEEE Transactions on Circuits and Systems for Video Tech., Vol. 13, No. 7 (July 2003) (“ Wenger ”)
1016	Thomas Wiegand et al., <i>Overview of the H.264/AVC Video Coding Standard</i> , IEEE Transactions on Circuits and Systems for Video Tech., Vol. 13, No. 7 (July 2003) (“ Wiegand ”)

IV. UNDERSTANDING OF PATENT LAW

20. I am not an attorney. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. My understanding of the law was provided to me by the Petitioner's attorneys.

21. I understand that when considering the scope of the claims of a patent that the patent claim terms should generally be given the ordinary meaning that the terms would have to a POSITA after reading the patent as of the earliest claimed priority date.

22. I understand that the POSITA is deemed to read the claim term not only in the context of the particular claim in which the term appears, but in the context of the entire patent, including the specification. I further understand that the principal considerations regarding the scope and meaning of a claim are the plain language of the claim (including the surrounding claim language and context), the patent specification, and the prosecution history. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification. I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope

narrower than it would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the '618 patent.

23. I understand that a claim is unpatentable if it would have been obvious. Obviousness of a claim requires that the claim would have been obvious from the perspective of a POSITA at the time the alleged invention was made. I understand that a claim could have been obvious from a single prior art reference or from a combination of two or more prior art references.

24. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

25. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I understand that there is no strict requirement of finding an express teaching, suggestion, or motivation to combine within the references. When a product is available, design incentives and other market forces can prompt variations of it, either in the same field or different one. If a POSITA can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a

technique has been used to improve one device and a POSITA would have recognized that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if common sense directs one to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

26. I am informed by counsel for the Petitioner and understand that when a patented invention is a combination of known elements, a court must determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue by considering the teachings of prior art references, the effects of demands known to people working in the field or present in the marketplace, and the background knowledge possessed by a POSITA.

27. I am informed by counsel for the Petitioner and understand that a patent claim composed of several limitations is not proved obvious merely by demonstrating that each of its limitations was independently known in the prior art. I am informed by counsel for the Petitioner and understand that identifying a reason those elements would be combined can be important because inventions in many instances rely upon building blocks long since uncovered and claimed discoveries almost of necessity tend to be combinations of what, in some sense, is already known. I am informed by counsel for the Petitioner and understand that it

is improper to use hindsight in an obviousness analysis, and that a patent's claims should not be used as a "roadmap" to combine prior art references.

28. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field.

29. I understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

30. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the descriptions found in the patent being considered.

31. I understand that other challenges to the patentability of a patent, including patent ineligibility, non-enablement, lack of written description support,

and lack of definiteness or clarity of claim language, cannot be raised in IPR proceedings before the Board.

32. I understand that Petitioner has the burden of proving unpatentability by a preponderance of evidence, which means that the claims are more likely than not unpatentable.

33. The analysis in this declaration is in accordance with the above-stated legal principles.

V. PERSON OF ORDINARY SKILL IN THE ART

34. It is my understanding that when interpreting the claims of the '618 patent I must do so based on the perspective of one of ordinary skill in the art at the relevant priority date. My understanding is that the earliest claimed priority date of the '618 patent is June 30, 2011.

35. It is my opinion that a person of ordinary skill in the art in June 2011 ("POSITA") would have had at least a bachelor's degree in electrical engineering, computer engineering, or a related field, with about two years of experience in video decoding or encoding technologies. More work or practical experience may qualify one not having the requisite education as a person with ordinary skill in the art while a higher level of education could offset less experience.

36. A person of skill in the art would have also had knowledge of the relevant video coding standards at the time, which would have included the H.263 and H.264/MPEG 4 AVC standards and the nascent H.265/HEVC standard.

37. Well-known industry publications, such as the recommendations and standards published by the international bodies for audiovisual standards, reflect the knowledge of a person of ordinary skill in the art. Additional knowledge and skills possessed by a person of ordinary skill in the art are discussed below in Section VI.

38. I am a person of at least ordinary skill in the art and was so at the time of the alleged invention of the '618 patent.

VI. DEVELOPMENT OF VIDEO CODING TECHNOLOGIES

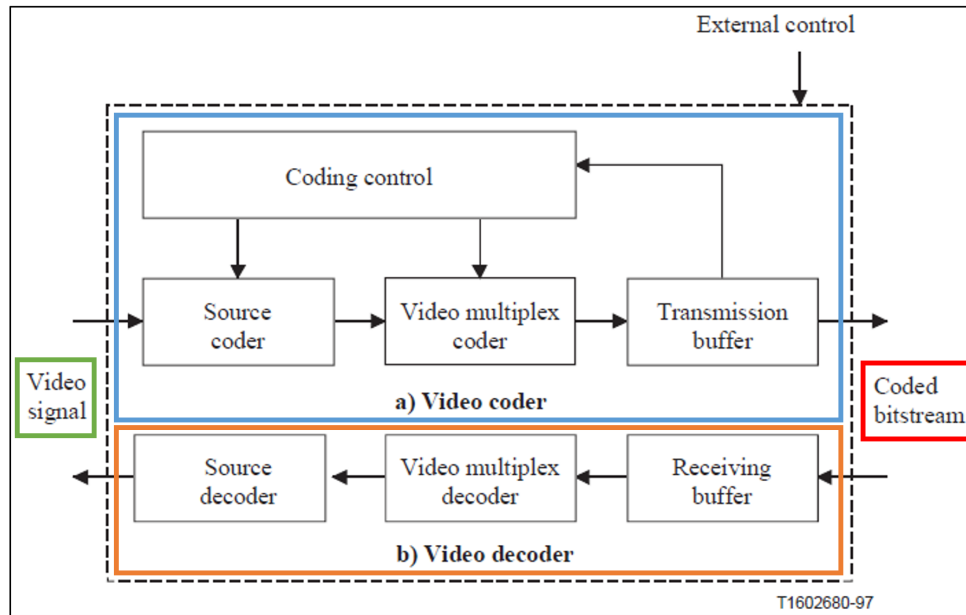
39. Like film, video comprises a rapid sequence of still images that the human vision system integrates into a perception of motion. The “moving images” are often called video frames or video pictures.

40. Digital video in its raw format is highly redundant. Video coding also known as video compression, is the art of removing the spatial and temporal redundancy to enable storage or transmission of digital video data with vastly fewer bits, while preserving the quality of the raw video data to the maximum extent possible. Each video frame comprises a regular, rectangular array of pixels.

The density of the pixel array determines the highest spatial resolution enabled by a particular video system.

41. Within a given scene being imaged by a video system, the content is very likely not to contain the highest level of detail except in a few, small areas. While a prescribed density of pixels may accurately describe such high detail areas, that density of pixels is overkill for the rest of the scene to a greater or lesser degree. Thus, the digital representation of the typical digital video frame contains spatial redundancy. For example, if the video scene contains the sky, or a uniformly painted wall, adjacent pixels will have almost identical intensity and color.

42. Video compression and decompression are typically performed by a video codec (coder-decoder). A codec is a hardware or software tool that compresses and decompresses digital video according to a video coding standard. A **coder** (or encoder) compresses an input **video** signal by applying techniques to reduce the data size, resulting in an encoded **bitstream** of the compressed video. A **decoder** performs the inverse operations on the bitstream to reconstruct a representation of the original video signal.



EX1013, 2 (depicting a block diagram of the H.263 video codec) (annotated).

43. One of the earliest examples of video compression used change detection to effect the removal of temporal redundancy. For example, U.S. Patent No. 1,796,030 to Ray D. Kell, filed in 1929, describes the efficient transmission of video by only transmitting changed elemental areas of the image from one picture to the next. U.S. Patent No. 1,796,030, 1:1-7. Kell's work was further developed at Bell Labs in the 1960s by applying a process called conditional update that distinguishes stationary from non-stationary pixels and transmits only the pixels that have changed. *E.g.*, U.S. Patent No. 3,553,362, 1:8-27 (describing "conditional replenishment video systems" where a "transmitted sample is used to update or replenish its corresponding sample in a receiving frame memory" upon a "conditional determination"). Related work by F.W. Mounts, in 1968, described

using a threshold to determine significant changes between pixels, where the threshold identifies the number of pixels having significant change required in order to trigger transmission of the change. *E.g.*, U.S. Patent No. 3,571,505, 1:42-61. Mounts also acknowledges the threshold that applies to distinguishing noise in the video from a true change. *Id.*, 16:49-53.

44. Bell Labs also developed spatial compression by predicting the value of a pixel from neighboring pixels. This recognizes that in many areas of a picture, the pixels are highly correlated, and the values of adjacent pixels are very similar in luminance and chrominance. *E.g.*, U.S. Patent No. 2,701,274, 4:1-27. Spatial prediction evolved from simply predicting a current pixel from its neighbor (to the left) in the raster scan, to adaptively predicting from the neighbor to the left or the neighbor on the raster scan line above (requiring the buffering of a line of video data). This was further extended to include adaptively predicting from the co-located pixel in the previous picture (requiring the buffering of a picture of video data). *E.g.*, U.S. Patent No. 3,051,778, 3:46-4:2.

45. The image and video compression work at Bell Labs was almost exclusively pixel-based in the 1960s and 1970s. At the time, memory sizes were very small by today's standards while its cost was very high, meaning that buffering many rows of pixels to perform processing on two-dimensional blocks of pixels was prohibitive. Furthermore, the cost of the digital logic required to process

blocks of pixels at real-time video rates was also prohibitive. Some researchers, however, pursued an approach called transform coding, in which blocks of pixels such as 16×16 pixels were converted into a set of transform coefficients to facilitate the removal of spatial redundancy. *E.g.*, U.S. Patent No. 3,679,821 (entitled “Transform coding of image difference signals”); U.S. Patent No. 4,245,248, 1:14-31 (explaining that “it has been found advantageous to partition individual frames of video data into blocks or groups of picture elements, and to perform a two-dimensional, generally linear transformation of each block so as to generate a set of transform coefficients,” which are “then used for prediction purposes”). Instead of performing pixel-based processing, the transform approach transforms blocks of pixels into spatial frequency coefficients, which can be more easily and effectively encoded than the raw pixels themselves because as discussed above, most regions of a picture contain only low spatial frequency information (i.e., low-detail information), thus only a few low spatial frequency coefficients will need to be encoded.

46. Block-based motion compensation was developed in 1974. Motion is modeled as a translation of objects within the scene, in the plane of the imaging system. While this model is clearly an approximation (for example, it ignores rotation, zoom and occlusion), it works remarkably well, especially when applied to small blocks of pixels into which each video picture may be divided.

47. This technique identifies blocks in the scene in which the content has not changed in the current picture from the previous picture, and does not update them, while blocks that have changed are predicted from blocks in the previous picture that may be displaced. The displacement is represented in the encoder by a motion vector.

48. Because the prediction may not be perfect, the pixel values of the predicted block are subtracted from the pixels of the block to be coded, yielding a residual block of prediction errors. If the simple translation model for motion is valid for the current block to be coded, the residual pixel values are zero or very small, and can be coded using just a few bits. It should be noted however that because the model can fail, a block of pixels may be coded more efficiently without reference to another picture, and a decision is made on a block-by-block basis as to the use of temporal prediction – which is often called inter-prediction or inter-picture prediction– or whether to predict the values of the pixels in the block from neighboring blocks of pixels in the same picture. Such spatial prediction within a picture is often called intra-prediction or intra-picture prediction, and it also may produce a block of residual pixel values. Modern video coding standards all use these techniques.

49. It should be noted that the use of prediction from one picture to the next means the encoder/transmitter and decoder/receiver must maintain identical

copies of the pictures used for prediction, which are reconstructed versions of previously encoded/decoded pictures. This implies the decoder must receive a perfect copy of the transmitted differences between a current picture and a prediction (reference) picture, or else it will not correctly reconstruct the current picture. Worse yet, the errors introduced will then continue to future encoded pictures, resulting in degraded video quality long after the original error occurred.

50. An encoder and decoder each typically include a buffer referred to as the “Decoded Picture Buffer” or simply “DPB.” EX1005, 297 (§C.2); EX1013, 149 (describing a “memory buffer of reference pictures” in H.263). A DPB can be managed by simple rules (e.g., only store the most recently decoded picture). But to fully harness the advantages of inter prediction, however, the reference pictures sometimes must be actively managed.

51. Some ways that reference pictures were managed was to allow storage of multiple reference pictures in the DPB, to allow storage of long-term reference pictures in the DPB and the reordering of reference pictures. *E.g.*, EX1013, 173-174 (§U.4.5). The storage of multiple reference pictures allows the encoder to choose a reference picture from amongst multiple candidates when encoding the current picture in the video sequence. Reordering reference pictures allows the encoder to use the lowest number for a reference picture index value to identify the most-used reference frame in a section of a picture. Since lower-numbered

reference picture index values require fewer bits of data to encode than higher-numbered values, this reduces the total number of bits required to encode a picture.

52. After a picture is decoded, the picture can be stored in the DPB for use as a reference picture in predicting and decoding¹ subsequent pictures in the video sequence. At some point during decoding, a decoder will “run out” of memory locations at which to store reference pictures. At this point, the decoder can discard one or more reference pictures, which are no longer available for use in subsequent motion-compensated prediction. According to one simple approach to buffer management, the decoder uses a “sliding window” to keep up to M reference pictures. For example, the H.263 standard describes “a ‘Sliding Window’ over time” approach. *See* EX1008, 2-3. “In such a buffering scheme using M picture memories $PM_0 \dots PM_{M-1}$, the most recent preceding (up to M) decoded and reconstructed pictures are stored in the picture memories and can be used as references for decoding.” *Id.*, 3. When the current (just decoded) picture is added to the buffer, the oldest reference picture is discarded according to the sliding window.

53. According to another approach to buffer management, an encoder signals information in encoded data, which the decoder receives and uses to update

¹ As discussed above, both the encoder and decoder have a DPB, and the details laid out here apply to both DPBs.

which reference pictures are stored for use in subsequent motion-compensated prediction. For example, the H.263 standard also describes an “‘Adaptive Memory Control’ type of operation” that “can be used for a more flexible and specific control of the picture memories.” EX1008, 3. For the Adaptive Memory Control approach, the encoder signals memory management control operation (“MMCO”) values in the encoded data for a current picture. EX1008, 11-14 (§§U.3.1.5.6 – U.3.1.5.9), 22 (§U.4.1). The MMCO values, and rules for the decoder to react to the MMCO values, indicate which reference pictures the decoder should retain after decoding the current picture and whether the respective reference pictures are “short-term” or “long-term” reference pictures. *Id.* Storing a long-term reference picture allows retention of a reference picture that is especially useful for prediction, such as in resynchronizing the encoder and decoder’s respective DPBs when transmitted data packets are lost in transmission.

54. MMCO values in the Adaptive Memory Control approach of the H.263 standard use what the ’618 patent describes as “relative” references – references that are interpreted relative to correctly decoded previous pictures. *See, e.g.,* EX1001, 1:48-52, 2:5-9, 9:15-18, 20:29-35. Alternatively, a reference picture can be identified with what the ’618 patent describes as an “absolute reference” – independent of correct decoding of previous pictures. *See, e.g.,* EX1001, 4:39-48, 6:41-48, 7:42-46. Such “absolute” references were used in some prior approaches.

For example, in the February 1998 version of the H.263 standard, Annex N describes identifying a reference picture using an absolute reference with a “Temporal Reference for Prediction (TRP)” syntax element. EX1011, 99 (§N.4.1.4). That version of H.263 describes that “[w]hen present (as indicated in TRPI), TRP indicates the Temporal Reference which is used for prediction of the encoding,” where the Temporal Reference is an 8-bit identifier of each picture (whether used for reference or not). EX1011, 99 (§N.4.1.4); *see also id.*, 23 (§5.1.2), 99 (§N.4.1.2).

55. Features of Annex U of the H.263 standard were adopted in the H.264/AVC standard, as described below, subject to further refinements and modifications in the H.264/AVC standard.

A. Video Coding Standards

56. Two international bodies are responsible for audiovisual standards – the ITU-T and ISO/IEC. For almost three decades, these bodies have collaborated to develop generations of standards, most frequently publishing identical texts under their own standardization programs.

1. Advanced Video Coding (H.264/AVC)

57. The ’618 patent describes the H.264/AVC standard as the state of the art at the time of the ’618 patent’s filing. EX1001, 1:31-35. H.264/AVC was first published in May 2003, and it remains in common use today. From 2003 to the

present, subsequent versions of the H.264/AVC standard have made minor corrections and also added features for extensions. EX1005, 2. The H.264/AVC standard defines syntax and semantics of encoded data in a bitstream for video (EX1005, 34-104 (§7)) as well as certain decoding processes (EX1005, 104-212 (§8)). Although it does not specify implementation details, the H.264/AVC standard sets “interoperability points” for decoding in varied operational environments. EX1005, 285; *see also id.*, 285-300 (Annex A). In an implementation of encoding or decoding for a specific operational environment, an encoder or decoder can use specific tools defined in the H.264/AVC standard in various ways, while remaining conformant with the H.264/AVC standard. *See* EX1006, xvi (§0.5).

58. In general, in H.264/AVC each picture has a frame number (`frame_num`) indicating its position in decoding order and a picture order count (`pic_order_cnt`) indicating its position in display order. EX1005, 79-85 (§7.4.3). Each picture is further subdivided into macroblocks and slices, to efficiently encode the various pieces of the picture. EX1005, 23-24 (§6.3).

59. As with its predecessor standards, the H.264/AVC standard uses inter-picture prediction to encode video, among other prediction techniques. To do this,

the encoded data for a current picture² in the H.264/AVC standard includes references to previously reconstructed pictures (reference pictures) that are used for the inter-picture prediction. The encoded data for the current picture can include information that specifies how to reorder the reference pictures for decoding of the current picture (or a slice of the current picture), so that the most commonly used reference picture can be addressed most efficiently, and so on, and to mark references pictures according to how they will be used in subsequent decoding. The syntax structure `ref_pic_list_reordering()`, which includes syntax elements for control of reference picture reordering, is present in a slice header for a picture, and the syntax structure `dec_ref_pic_marking()`, which includes syntax elements for control of reference picture marking, may also be present in the slice header for the picture. The pertinent syntax and semantics for control of reference picture reordering are shown here:

7.3.3 Slice header syntax		
<code>slice_header() {</code>	C	Descriptor
<code>ref_pic_list_reordering()</code>	2	
<code>if(nal_ref_idc != 0)</code>		
<code>dec_ref_pic_marking()</code>	2	

² Because the early generations of standards included coding of interlaced fields, the term “picture” was used to collectively refer to frames and fields.

EX1005, 46 (excerpts).

7.3.3.1 Reference picture list reordering syntax

	C	Descriptor
ref_pic_list_reordering() {		
if(slice_type % 5 != 2 && slice_type % 5 != 4) {		
ref_pic_list_reordering_flag_l0	2	u(1)
if(ref_pic_list_reordering_flag_l0)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0		
reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
if(slice_type % 5 == 1) {		
ref_pic_list_reordering_flag_l1	2	u(1)
if(ref_pic_list_reordering_flag_l1)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0		
reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
}		

EX1005, 47.

7.4.3.1 Reference picture list reordering semantics

The syntax elements `reordering_of_pic_nums_idc`, `abs_diff_pic_num_minus1`, and `long_term_pic_num` specify the change from the initial reference picture lists to the reference picture lists to be used for decoding the slice.

EX1005, 85.

reordering_of_pic_nums_idc together with **abs_diff_pic_num_minus1** or **long_term_pic_num** specifies which of the reference pictures are re-mapped. The values of **reordering_of_pic_nums_idc** are specified in Table 7-4. The value of the first **reordering_of_pic_nums_idc** that follows immediately after **ref_pic_list_reordering_flag_l0** or **ref_pic_list_reordering_flag_l1** shall not be equal to 3.

EX1005, 85.

Table 7-7 – reordering_of_pic_nums_idc operations for reordering of reference picture lists	
reordering_of_pic_nums_idc	Reordering specified
0	abs_diff_pic_num_minus1 is present and corresponds to a difference to subtract from a picture number prediction value
1	abs_diff_pic_num_minus1 is present and corresponds to a difference to add to a picture number prediction value
2	long_term_pic_num is present and specifies the long-term picture number for a reference picture
3	End loop for reordering of the initial reference picture list

EX1005, 85.

abs_diff_pic_num_minus1 plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value. **abs_diff_pic_num_minus1** shall be in the range of 0 to **MaxPicNum** – 1. The allowed values of **abs_diff_pic_num_minus1** are further restricted as specified in subclause 8.2.4.3.1.

long_term_pic_num specifies the long-term picture number of the picture being moved to the current index in the list. When decoding a coded frame, **long_term_pic_num** shall be equal to a **LongTermPicNum** assigned to one of the reference frames or complementary reference field pairs marked as "used for long-term reference". When decoding a coded field, **long_term_pic_num** shall be equal to a **LongTermPicNum** assigned to one of the reference fields marked as "used for long-term reference".

EX1005, 86.

60. At the beginning of decoding a slice that has been encoded using inter-picture prediction (such as a P slice or a B slice), a decoder performs operations to derive a reference picture list, which is a list of reference pictures that may be used for inter-picture prediction when decoding the slice. EX1005, 11 (§3.124), 112 (§8.2.4). The decoder derives an initial reference picture list based

on marking status and identifying information of reference pictures in a decoded picture buffer (“DPB”). EX1005, 112 (§8.2.4), 113-116 (§8.2.4.2). The decoder can then modify the initial reference picture list according to syntax elements in a `ref_pic_list_reordering()` syntax structure. EX1005, 112 (§8.2.4), 116-118 (§8.2.4.3).

61. A reference picture list in H.264/AVC has indices, which are organized from a lowest index (0) to a highest index, whose value depends on a syntax element (such as `num_ref_idx_l0_active_minus1` or `num_ref_idx_l1_active_minus1`) in a picture parameter set. EX1005, 75 (§7.4.2.2; `num_ref_idx_l0_active_minus1`, `num_ref_idx_l1_active_minus1`), 113 (§8.2.4.2). When the decoder derives an initial reference picture list, reference pictures in the DPB are ordered in the initial reference picture list according to rules. EX1005, 113-116 (§8.2.4.2). If the DPB has more reference pictures than can fit in the reference picture list, “the extra entries past position `num_ref_idx_l0_active_minus1` or `num_ref_idx_l1_active_minus1` are discarded from the initial reference picture list.” EX1005, 113. On the other hand, if the reference picture list has more indices than the number of reference pictures in the DPB, “the remaining entries in the initial reference picture list are set equal to ‘no reference picture’.” EX1005, 114.

62. The operations performed to derive the initial reference picture list vary depending on the type of slice being decoded (e.g., P slice or B slice) and picture organization (e.g., frame or field). EX1005, 113-116 (§8.2.4.2). For example, when decoding a P slice of a frame, the reference pictures in an initial reference picture list are ordered so that short-term reference frames have lower indices than long-term reference frames. EX1005, 114 (§8.2.4.2.1). More specifically, short-term reference frames are ordered starting with the reference frame having the highest PicNum value (in general, the most recent reference frame) at index 0, and putting reference frames with lower PicNum values, in descending order, in successive indices of the initial reference picture list. *Id.* Then, long-term reference frames are ordered starting with the lowest LongTermPicNum value and proceeding through reference frames having higher values of LongTermPicNum, in ascending order, in successive indices of the initial reference picture list. *Id.* In the following example, a reference picture list RefPicList0 for a P slice includes 5 positions, which are organized by indices 0 to 4.

For example, when three reference frames are marked as "used for short-term reference" with PicNum equal to 300, 302, and 303 and two reference frames are marked as "used for long-term reference" with LongTermPicNum equal to 0 and 3, the initial index order is:

- RefPicList0[0] is set equal to the short-term reference picture with PicNum = 303,
- RefPicList0[1] is set equal to the short-term reference picture with PicNum = 302,
- RefPicList0[2] is set equal to the short-term reference picture with PicNum = 300,
- RefPicList0[3] is set equal to the long-term reference picture with LongTermPicNum = 0, and
- RefPicList0[4] is set equal to the long-term reference picture with LongTermPicNum = 3.

Id.

63. When the decoder modifies the initial reference picture list according to syntax elements in a `ref_pic_list_reordering()` syntax structure, the decoder can reorder reference pictures in the reference picture list, add one or more reference pictures to the reference picture list, and/or remove one or more reference pictures from the reference picture list. EX1005, 112 (§8.2.4), 116-118 (§8.2.4.3). During decoding, a reference picture used for inter-picture prediction of the content of a macroblock or sub-macroblock portion is identified with a syntax element that specifies an index in a reference picture list. EX1005, 97 (§7.4.5.1; `ref_idx_l0` and `ref_idx_l1`), 100 (§7.4.5.3; `ref_idx_l0` and `ref_idx_l1`). During encoding, the value of a reference picture index is entropy coded as a truncated Exp-Golomb-coded syntax element ($te(v)$) or as a context-adaptive arithmetic entropy-coded syntax element ($ae(v)$). EX1005, 38-39 (§7.2; $ae(v)$ and $te(v)$), 52 (§7.3.5.2). By adding a reference picture that is not in an initial reference picture list, the added reference picture can be used during decoding of the slice. Conversely, by removing (from the initial reference picture list) a reference picture that is not to be used during decoding, the reference picture index values of remaining reference pictures can be lowered in the reference picture list (to save bits on reference picture index values), or space can be freed up for an additional reference picture to be used during decoding of the slice.

64. In the `ref_pic_list_reordering()` syntax structure, the syntax provides a “do-while” loop in which the syntax element `reordering_of_pic_nums_idc` controls the number of times the syntax elements within the do-while loop are present for a reference picture list. EX1005, 47 (§7.3.3.1), 85-86 (§7.4.3.1), 116-118 (§8.2.4.3). Within the loop, the syntax structure can provide a syntax element for adjusting a short-term reference picture at an index of a reference picture list or a syntax element specifying a long-term reference picture at an index of a reference picture list. *Id.* If the value of the `reordering_of_pic_nums_idc` syntax element is 0 or 1, the syntax structure includes an `abs_diff_pic_num_minus1` syntax element, which specifies a short-term reference picture at an index of the reference picture list. *Id.* If the value of the `reordering_of_pic_nums_idc` syntax element is 2, the syntax structure includes a `long_term_pic_num` syntax element, which specifies a long-term reference picture at an index of the reference picture list. *Id.* Finally, if the value of the `reordering_of_pic_nums_idc` syntax element is 3 (“End loop”), the syntax structure includes no additional syntax elements for modifying the reference picture list. *Id.*

65. During decoding, a variable (such as `refIdxL0`) tracks a current index in a reference picture list, starting at index 0. EX1005, 116-118 (§8.2.4.3). Successive occurrences of `reordering_of_pic_nums_idc` are “processed in the order they occur in the bitstream.” EX1005, 116.

66. If the value of the `reordering_of_pic_nums_idc` syntax element is 0 or 1, (a) the value of $(\text{abs_diff_pic_num_minus1} + 1)$ is subtracted from or added to a prediction value, (b) the short-term reference picture having the resulting picture number (`picNumLX`, after some additional condition checks and adjustments) is placed at the current index, (c) positions are shifted for any remaining reference pictures later in the reference picture list, and (d) the variable that tracks the current index in the reference picture list is incremented. EX1005, 117-118 (§8.2.4.3.1). For the first occurrence of `reordering_of_pic_nums_idc` equal to 0 or 1 in the syntax structure, the prediction value is the frame number of the current frame. EX1005, 82 (§7.4.3; `CurrPicNum`), 117-118 (§8.2.4.3.1). For a subsequent occurrence of `reordering_of_pic_nums_idc` equal to 0 or 1 in the syntax structure, the prediction value is the previous picture number in the reference picture list. *Id.*

67. If the value of the `reordering_of_pic_nums_idc` syntax element is 2, (a) the long-term reference picture having the picture number `long_term_pic_num` is placed at the current index, (b) positions are shifted for any remaining reference pictures later in the reference picture list, and (c) the variable that tracks the current index in the reference picture list is incremented. EX1005, 118 (§8.2.4.3.2).

68. Thus, the variable that tracks the current index in the reference picture list is incremented as the decoder proceeds through operations for successive indices of the reference picture list. EX1005, 116-118 (§8.2.4.3). If the value of the

reordering_of_pic_nums_idc syntax element is 3, the decoder finishes the modification process for the reference picture list.

69. When a reference picture list is modified according to syntax elements in a ref_pic_list_reordering() syntax structure, the reference picture at an index of the reference picture list can change or remain the same. *See* EX1005, 47 (§7.3.3.1), 85-86 (§7.4.3.1), 116-118 (§8.2.4.3). Examples of how the picture list can change or remain the same using the appropriate combination of syntax elements reordering_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_num syntax are given below in Section IX.D.

70. During decoding, reference pictures in a DPB can be marked according to a “sliding window” marking process or adaptive marking process. EX1005, 118-119 (§8.2.5). For the adaptive marking process, the encoded data for the current slice can include information that specifies how to mark reference pictures to indicate the reference pictures may be used in subsequent decoding, including MMCO commands that indicate which pictures should be saved in the DPB for later use as reference pictures when decoding one or more subsequent pictures or that indicate which pictures may be removed from the DPB, after output, by marking them as “unused for reference.” EX1005, 86-89 (§7.4.3.3), 120-122 (§8.2.5.4). The MMCO commands affect which reference pictures are retained in the DPB and how the reference pictures may be used as “short-term”

reference pictures or “long-term” reference pictures. *Id.* Because of the timing of processing these MMCO commands, encoded data for a current picture can include MMCO commands that indicate which pictures should be saved in the DPB for use in decoding subsequent pictures. *See* EX1005, 120-122 (§8.2.5.4). From the perspective of decoding the current picture, encoded data for previous pictures can include MMCO commands that indicate which pictures should be saved in the DPB for use in decoding the current picture. *Id.* The syntax structure `dec_ref_pic_marking()`, which includes syntax elements to control reference picture marking, may be present in a slice header for a picture. *Id.* The pertinent syntax and semantics for control of reference picture marking are shown here:

7.3.3.3 Decoded reference picture marking syntax

dec_ref_pic_marking() {	C	Descriptor
if(IdrPicFlag) {		
no_output_of_prior_pics_flag	2 5	u(1)
long_term_reference_flag	2 5	u(1)
} else {		
adaptive_ref_pic_marking_mode_flag	2 5	u(1)
if(adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	2 5	ue(v)
if(memory_management_control_operation == 1 memory_management_control_operation == 3)		
difference_of_pic_nums_minus1	2 5	ue(v)
if(memory_management_control_operation == 2)		
long_term_pic_num	2 5	ue(v)
if(memory_management_control_operation == 3 memory_management_control_operation == 6)		
long_term_frame_idx	2 5	ue(v)
if(memory_management_control_operation == 4)		
max_long_term_frame_idx_plus1	2 5	ue(v)
} while(memory_management_control_operation != 0)		
}		
}		

EX1005, 49.

7.4.3.3 Decoded reference picture marking semantics

The syntax elements `no_output_of_prior_pics_flag`, `long_term_reference_flag`, `adaptive_ref_pic_marking_mode_flag`, `memory_management_control_operation`, `difference_of_pic_nums_minus1`, `long_term_frame_idx`, `long_term_pic_num`, and `max_long_term_frame_idx_plus1` specify marking of the reference pictures.

The marking of a reference picture can be "unused for reference", "used for short-term reference", or "used for long-term reference", but only one among these three. When a reference picture is referred to as being marked as "used for reference", this collectively refers to the picture being marked as "used for short-term reference" or "used for long-term reference" (but not both). A reference picture that is marked as "used for short-term reference" is referred to as a short-term reference picture. A reference picture that is marked as "used for long-term reference" is referred to as a long-term reference picture.

The syntax element `adaptive_ref_pic_marking_mode_flag` and the content of the decoded reference picture marking syntax structure shall be identical for all coded slices of a coded picture.

EX1005, 86.

adaptive_ref_pic_marking_mode_flag selects the reference picture marking mode of the currently decoded picture as specified in Table 7-8. **adaptive_ref_pic_marking_mode_flag** shall be equal to 1 when the number of frames, complementary field pairs, and non-paired fields that are currently marked as "used for long-term reference" is equal to $\text{Max}(\text{num_ref_frames}, 1)$.

Table 7-8 – Interpretation of **adaptive_ref_pic_marking_mode_flag**

adaptive_ref_pic_marking_mode_flag	Reference picture marking mode specified
0	Sliding window reference picture marking mode: A marking mode providing a first-in first-out mechanism for short-term reference pictures.
1	Adaptive reference picture marking mode: A reference picture marking mode providing syntax elements to specify marking of reference pictures as "unused for reference" and to assign long-term frame indices.

EX1005, 87.

memory_management_control_operation specifies a control operation to be applied to affect the reference picture marking. The **memory_management_control_operation** syntax element is followed by data necessary for the operation specified by the value of **memory_management_control_operation**. The values and control operations associated with **memory_management_control_operation** are specified in Table 7-9. The **memory_management_control_operation** syntax elements are processed by the decoding process in the order in which they appear in the slice header, and the semantics constraints expressed for each **memory_management_control_operation** apply at the specific position in that order at which that individual **memory_management_control_operation** is processed.

EX1005, 87.

Table 7-9 – Memory management control operation (memory_management_control_operation) values

memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation syntax element loop
1	Mark a short-term reference picture as “unused for reference”
2	Mark a long-term reference picture as “unused for reference”
3	Mark a short-term reference picture as “used for long-term reference” and assign a long-term frame index to it
4	Specify the maximum long-term frame index and mark all long-term reference pictures having long-term frame indices greater than the maximum value as “unused for reference”
5	Mark all reference pictures as “unused for reference” and set the MaxLongTermFrameIdx variable to “no long-term frame indices”
6	Mark the current picture as “used for long-term reference” and assign a long-term frame index to it

EX1005, 88.

71. In the dec_ref_pic_marking() syntax structure, the syntax provides a do-while loop in which values of the memory_management_control_operation syntax element control the number of times the syntax elements within the do-while loop are present for the syntax structure. EX1005, 49 (§7.3.3.3), 86-88 (§7.4.3.3). If the value of memory_management_control_operation is 1, the syntax structure includes a syntax element that specifies a short-term reference picture to mark as “unused for reference.” *Id.* If the value of memory_management_control_operation is 2, the syntax structure includes a syntax element that specifies a long-term reference picture to mark as “unused for reference.” *Id.* If the value of memory_management_control_operation is 3, the

syntax structure includes syntax elements that specify a short-term reference picture to mark as a long-term reference picture having an indicated long-term frame index. *Id.* If the value of `memory_management_control_operation` is 4, the syntax structure includes a syntax element that specifies a long-term frame index above which any long-term reference picture is marked as “unused for reference.” *Id.* If the value of `memory_management_control_operation` is 5, all reference pictures are marked as “unused for reference.” *Id.* If the value of `memory_management_control_operation` is 6, the syntax structure includes a syntax element that indicates a long-term frame index for the current picture, which is marked as a long-term reference picture. *Id.* Finally, if the value of `memory_management_control_operation` is 0, the syntax structure includes no additional syntax elements for marking. *Id.*

72. For the adaptive marking process, MMCO commands in a `dec_ref_pic_marking()` syntax structure are “processed in the order they occur in the bitstream after the current picture has been decoded.” EX1005, 120. An MMCO command can provide for marking a short-term reference picture or long-term reference picture as “unused for reference” (MMCO command 1 or 2, respectively), changing a short-term reference picture to a long-term reference pictures (MMCO command 3), marking certain long-term reference pictures or all reference pictures as “unused for reference” (MMCO command 4 or 5,

respectively), or marking the current picture as a long-term reference picture (MMCO command 6). EX1005, 49 (§7.3.3.3), 87-89 (§7.4.3.3), 120-122 (§8.2.5.4). If the value of the MMCO command is 0, the decoder finishes the adaptive marking process performed after decoding the current picture.

73. The DPB stores decoded picture for use as reference pictures, for output reordering, or for output delay. EX1005, 7 (§3.40). In general, a picture is removed from the DPB at the latter of the time of its output and the time the reference picture is marked as “unused for reference.” EX1005, 298 (§C.2.3). Before the decoded current picture is (possibly) stored in the DPB, the decoder removes certain reference pictures from the DPB. *Id.* A reference picture that has already been output will be removed from the DPB when the reference picture is marked as “unused for reference.” *Id.* On the other hand, a reference picture that remains to be output will not be removed from the DPB when the reference picture is marked as “unused for reference” but instead will remain in the DPB until after the reference picture has been output. *Id.*

All pictures m in the DPB, for which all of the following conditions are true, are removed from the DPB.

- picture m is marked as “unused for reference” or picture m is a non-reference picture. When a picture is a reference frame, it is considered to be marked as “unused for reference” only when both of its fields have been marked as “unused for reference”.
- picture m is marked as “non-existing” or it has OutputFlag equal to 0 or its DPB output time is less than or equal to the CPB removal time of the current picture n ; i.e., $t_{o,dpb}(m) \leq t_r(n)$

EX1005, 298.

74. When H.264/AVC was first drafted, as a general rule, the standard did not specify operations that a decoder must perform for error correction or recovery. However, occasionally encoded data for a given picture is lost, including reference picture information (such as MMCO commands) that indicates how to store reference pictures in the DPB for decoding of subsequent pictures. *E.g.*, EX1001, 2:56-3:6. If the reference picture information indicating how to store reference pictures for a certain subsequent picture is lost, the state of the DPB may be incorrect when decoding the subsequent picture and inter prediction may result in incorrect decoding of the subsequent picture. *Id.*

75. This shortcoming in H.264/AVC had been recognized for years prior to the filing of the '618 patent, when later versions of the standard were being developed to include applications involving error resilience and scalable video coding. For example, the Mulroy reference, which was filed in 2007, recognized that “[i]f the decoder loses a set of MMCO commands, there will be mismatch between the contents of the encoder and decoder buffers” and proposes a number of strategies to mitigate that weakness. EX1004, ¶27. In particular, Mulroy describes using a feature of H.264/AVC—reference picture list reordering information—to reiterate which reference pictures should be present in a decoder buffer (not just to indicate how to reorder reference pictures). *Id.* Thus, in order to mitigate this recognized weakness in H.264/AVC, Mulroy describes using

reference picture list reordering information from H.264/AVC for a new purpose, one not described in the H.264/AVC standard.

2. High Efficiency Video Coding (H.265/HEVC)

76. The H.265/HEVC standard was developed to improve compression efficiency over the H.264/AVC codec and, especially, to improve its ability to handle high-resolution images. EX1014, 1. H.265/HEVC was first published in 2013. EX1014, (a), i.

77. H.265/HEVC expands on many of the same approaches that were used in the H.264/AVC standard. For example, it uses intra prediction, but allows for more modes of prediction in different directions. *E.g.*, EX1014, 2, 102. It also uses inter prediction from reference pictures but uses a different approach than H.264/AVC to manage those references pictures. *E.g.*, *id.*, 2, 9.

78. In H.265/HEVC the encoded data does not include MMCO commands. Instead, H.265/HEVC specifies a reference picture set and reference picture lists that are determined based on that reference picture set. EX1014, 95-99 (§8.3.2) (describing process for reference picture set and picture marking, and five reference picture set lists). This makes it more robust to loss of the encoded data for any given picture.

VII. THE '618 PATENT

79. The '618 patent issued on July 7, 2020, from U.S. Patent Application No. 16/440,274 (the “'274 application”), filed on June 13, 2019, which is a continuation of U.S. Patent Application No. 16/048,556, filed on July 30, 2018, now U.S. Patent No. 10,368,088, which is a continuation of U.S. Patent Application No. 15/718,374, filed on September 28, 2017, now U.S. Patent No. 10,063,882, which is a continuation of U.S. Patent Application No. 15/299,997, filed on October 21, 2016, now U.S. Patent No. 9,807,418, which is a continuation of U.S. Patent Application No. 13/508,442, now Patent No. 9,706,223, which is a national stage application of P.C.T. International Application No. PCT/SE2012/050439, filed on April 26, 2012, which claims the benefit of U.S. Provisional Application No. 61/503,019, filed on June 30, 2011. EX1001, Cover, 1:6-20.

A. The '618 Patent's Specification

80. The '618 patent describes a method of decoding encoded videos using an “absolute” approach to signaling reference picture information for a current picture. In the approach described, the encoded data for a current picture includes information that indicates reference pictures available in the DPB for decoding of the current picture, as opposed to a “relative” approach of managing the DPB

based on information in encoded data for one or more previous pictures. EX1001, 4:39-54.

81. The '618 patent describes that its “general objective” is to “provide an efficient reference picture signaling and buffer management in connection with video encoding and decoding.” EX1001, 3:20-22.

82. According to the '618 patent, it is necessary to modify the buffer management approach from the H.264/AVC standard because the MMCO structures in encoded data for a previous picture (according to H.264/AVC) risked the loss of information indicating which reference pictures to retain for decoding of a current picture and how those reference pictures should be identified, even when encoded data for the current picture is received. *See, e.g.*, EX1001, 6:1-4 (“The prior art techniques may suffer from problems with regard to using incorrect reference pictures when a picture carrying MMCO information is unintentionally lost...”). Some specific issues identified in the '618 patent's specification are that “if a picture containing MMCO of type 2, 3, 4, 5 or 6 is lost there is a risk that the number of long term pictures in the DPB is different from what it would have been if the picture was received, resulting in an ‘incorrect’ sliding window process for all the following pictures.” EX1001, 2:56-60. This creates differences in the “state” of the DPB at the encoder and decoder, which is problematic, for example, because “the encoder and decoder will contain a different number of short-term pictures

resulting in out-of-sync behavior of the sliding window process.” EX1001, 2:60-63. “Thus, the prior art solution has a problem that correct reference picture management is dependant [sic] on that previously decoded pictures have been correctly received and decoded.” EX1001, 6:34-37.

83. As discussed in Section VI.A.1 above and further discussed in Section IX.A below, these criticisms of the H.264/AVC MMCO and sliding window processes for reference picture marking were not new in the art but had been the subject of discussion between POSITAs since at least 2007.

84. To address these alleged deficiencies, the '618 patent describes using absolute signaling of information *in the encoded data for a current picture* about which reference pictures to use for decoding of the current picture, i.e., numbers that indicate the exact identifiers of the reference pictures used to decode the current picture. The specification attempts to draw a “clear contrast to the prior art solutions in which correct reference picture management is dependent on that previously encoded pictures have been correctly received and decoded” (EX1001, 4:39-42), where the “previously encoded pictures” include the buffer description information indicating reference pictures used to decode the current picture. Instead, the approaches in the '618 patent use signaling of the “buffer description information that is used for reference pictures in an absolute and explicit way instead of a relative or implicit way.” EX1001, 4:42-44. By doing that “the

encoded representation of a picture contains the information about which reference pictures to use for reference during decoding independent of the encoded representations of previous pictures in the video stream.” EX1001, 4:45-48.

85. The benefit of this approach is alleged to be that by using “absolute” signaling of information about reference pictures to be used for the current picture (in the encoded representation of the current picture), the decoder only need rely on the information included in the current picture itself, instead of requiring that the information in the previous pictures also be received and decoded. *See* EX1001, 6:38-56 (“the encoded representation, i.e. the bitstream, for a current picture contains the information about what pictures to use for reference, i.e. reference pictures, independent of the encoded representations of previous pictures. It can therefore be said that the logical responsibility for maintaining correct decoded picture buffer is moved from the decoder to the bitstream.”).

86. The '618 patent also suggests that to reduce the number of bits needed to encode the absolute identifiers of reference pictures, “signal[ing] the reference picture properties, i.e. picture identifiers, relative to the value of these properties as signaled for the current picture.” EX1001, 9:34-36. “For instance, if the current picture has a picture identifier 7 the list of reference pictures with identifiers 3, 5 and 6 could be defined as -1 , -2 , and -4 , which typically can be represented by fewer bits as compared to 3, 5 and 6, in particular if variable length coding is

employed for the picture identifiers.” EX1001, 9:32-42. (The value -4 indicates the reference picture with identifier 3 since $7 - 4 = 3$, and so on.) In the terminology of the ’618 patent, using a differential value to identify a reference picture relative to the picture identifier of the current picture is still “absolute” signaling, since the differential value and picture identifier of the current picture are part of the encoded data for the current picture. According to the ’618 patent, a picture identifier can be a Picture Order Count (“POC”) value, frame number, or POC value and additional identifier. EX1001, 7:54-56 (“Examples of such picture identifiers include Picture Order Count (POC), frame number (frame_num) or POC and an additional identifier (additional_picture_id).”).

87. As discussed below, however, solutions to the issues identified in the ’618 patent were addressed by the technology disclosed in the Mulroy reference that predates the ’618 patent’s earliest priority date by more than two years.

B. The Challenged Claims

88. Challenged claim 1 is representative and listed here, with element numbering added in the lefthand column:

[1.1]	A method of decoding an encoded representation of a picture in a video stream of multiple pictures, the method comprising:
[1.2]	retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures;
[1.3]	determining, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the

	plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture, wherein determining the reference picture identifier comprises:
[1.4]	(i) retrieving, based on the buffer description information, a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture; and
[1.5]	(ii) calculating the reference picture identifier based on the picture identifier and the delta identifier; and
[1.6]	updating a decoded picture buffer based on the reference picture identifier.

The other independent claims recite similar limitations, and the dependent claims add additional details about handling the reference pictures, such how to handle missing reference pictures.

C. The Relevant Prosecution History Of The '618 Patent And Its Family Members

89. The '618 patent claims priority to U.S. Patent Application No. 16/048,556, filed on Jul. 30, 2018, now U.S. Pat. No. 10,368,088; which is a continuation of U.S. Pat. Application No. 15/718,374, filed on Sep. 28, 2017, now U.S. Pat. No. 10,063,882; which is a continuation of U.S. Pat. Application No. 15/299,997, filed on Oct. 21, 2016, now U.S. Pat. No. 9,807,418; which is a continuation of U.S. Pat. Application No. 13/508,442, filed on Apr. 25, 2013, now U.S. Pat. No. 9,706,223; which is a national stage application of P.C.T. International Application No. PCT/SE2012/050439, filed on Apr. 26, 2012, which claims the benefit of U.S. Provisional Application No. 61/503,019, filed Jun. 30, 2011. EX1001, Cover, 1:6-20.

90. The claims in the '618 patent's ancestor applications are similar to the claims of the '618 patent. In particular, U.S. Pat. App. No. 15/299,997 (the "'997 application") contained a claim that was amended in an April 24, 2017, response to a Non-final Rejection to claim elements such as a retrieving buffer description information and determining a reference picture identifier based on the buffer description information and a delta identifier. The Examiner rejected the original form of that claim as obvious over Borgwardt et al. ("Multi-Picture Buffer Semantics for Interlace," JVT-C049, May 10, 2002) in view of two other patent publications (U.S. Publication No. 2010/0208828 to Lu et al. and U.S. Publication No. 2010/0238822 to Koyabu et al.). EX1009, 309.

91. In response to that rejection, the Applicant amended the claim as indicated below and argued that the amended claim was distinct from Borgwardt, because "'[e]mbodiments represented by Claim 1 avoid this problem by including, within the encoded picture itself, buffer description information that identifies the reference picture(s) needed to decode the picture." EX1009, 345. The Applicant further argued that "[i]n this way, each encoded picture may itself include sufficient information to identify the reference pictures needed for decoding the encoded picture." *Id.* (emphasis omitted.) Other aspects that Applicant contended were not disclosed in the combination of references include buffer description information that defines a plurality of reference pictures and either a delta

identifier, a picture identifier, or a calculation of a reference picture identifier. *Id.*, 346-349.

92. That claim as amended is reproduced below:

1. A method of decoding an encoded representation of a picture in a video stream of multiple pictures, said method comprising:

retrieving, using a decoder configured to decode said encoded representation of said picture, buffer description information defining at least one reference picture, wherein said buffer description information is retrieved from said encoded representation of said picture;

determining, based on said buffer description information, at least one reference picture identifier, said at least one reference picture identifier identifying a respective reference picture among said at least one reference picture, said respective reference picture comprising as a decoding reference for said picture, wherein determining said at least one reference picture identifier comprises:

(i) retrieving, based on said buffer description information, a respective delta identifier; and

(ii) calculating said at least one reference picture identifier based on a picture identifier identifying said picture and said respective delta identifier; and

updating a decoded picture buffer based on said at least one reference picture identifier.

EX1009, 335-336.

93. During prosecution of the '618 patent, the same Examiner rejected claims for non-statutory double patenting over claims of the ancestor patents,

including U.S. Patent No. 9,807,418 which issued from the '997 application, noting that the claims are not patentably distinct from each other because the claims “perform the same function” despite the use of “different terminology” in the claims. EX1002, 100. In the same Office action, the Examiner rejected claims 1-3 as obvious over Borgwardt and Lu, similar to the rejection of the claims in the '997 application. EX1002, 104-105.

94. In response, the Applicant repeated the arguments against Borgwardt and Lu used during prosecution of the '997 application, namely that Borgwardt and Lu do not disclose or suggest values that “define buffer description information that define a plurality of reference pictures,” “a delta identifier ... identifying the picture, a picture identifier identifying a picture, or calculation of a reference picture identifier.” EX1002, 134-137. Notably, while the Applicant argued that MMCO commands in Borgwardt do not disclose those elements, Applicant did not argue that the claimed method must not use any MMCO commands. *Id.* The Examiner then allowed the claims. *Id.*, 147-148.

VIII. CLAIM CONSTRUCTION

95. I understand that, in an IPR, claim terms should be given the ordinary meaning that the terms would have to a POSITA on the earliest effective filing date, in view of the specification and file history.

96. The terms “frame” and “picture” are used interchangeably in the ’618 patent’s specification. EX1001, 5:45-47 (“The present embodiments generally relate to encoding and decoding of pictures, also referred to as frames in the art, of a video stream.” No claim uses the term “frame,” but I note that, since the ’618 patent’s specification uses the terms “frame” and “picture” interchangeably (EX1001, 5:45-47), a POSITA would have read the specification with that usage in mind.

**IX. GROUND 1: CLAIMS 1-7 ARE
OBVIOUS OVER MULROY IN VIEW OF H.264/AVC**

97. As explained below, claims 1-7 are obvious over Mulroy in view of H.264/AVC.

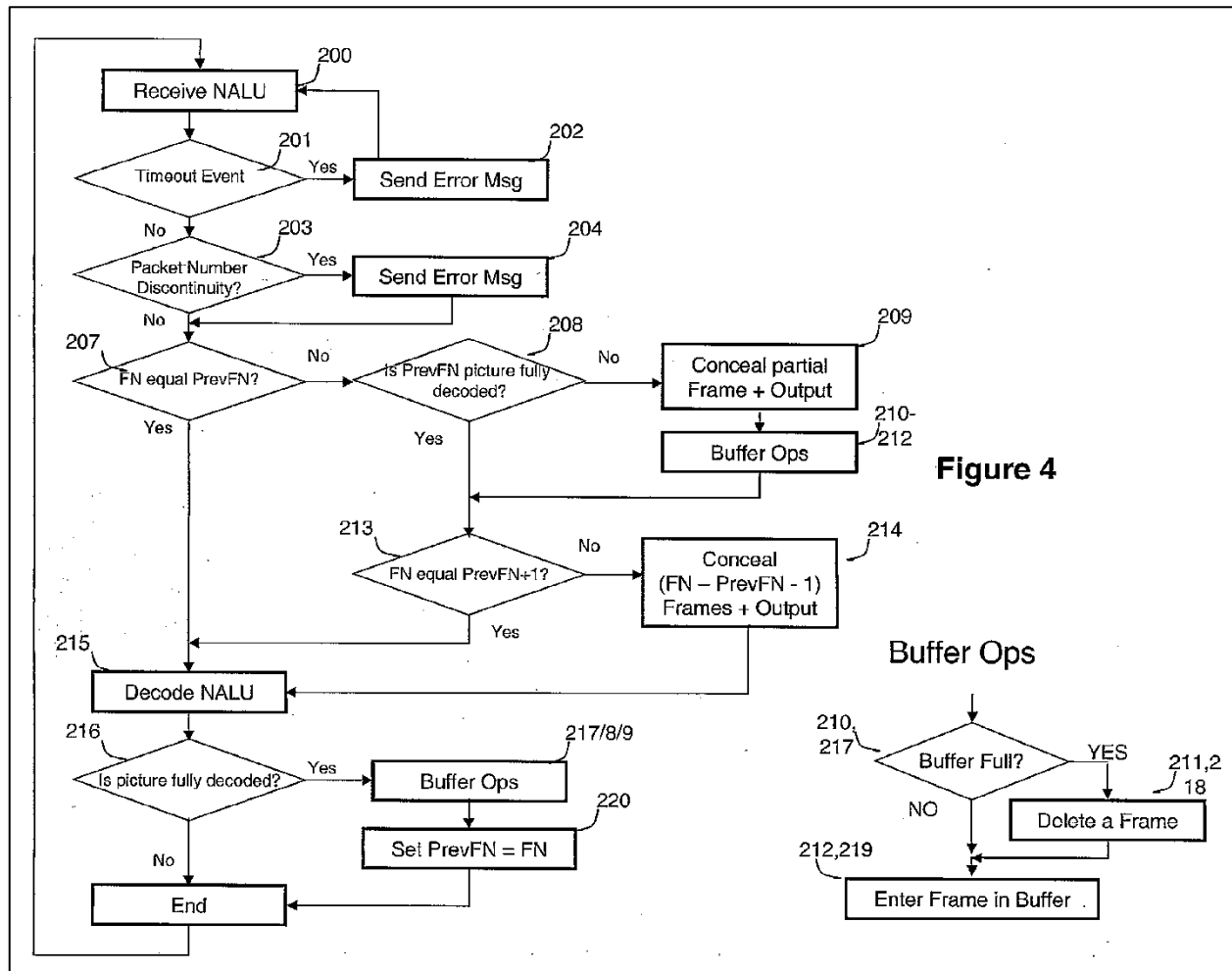
A. Mulroy (U.S. 2009/0052531; EX1004)

98. Mulroy is a U.S. patent application publication that was published February 26, 2009, and is thus prior art under at least Sections 102(a-b). EX1004, Cover. Mulroy is analogous art to the ’618 patent because they are both within the same field of endeavor of video coding. EX1001, 1:24-27; EX1004, ¶1.

99. Mulroy discloses methods to improve encoding and decoding video streams in an H.264 codec system. The H.264 codec system in Mulroy implements features of the H.264/AVC standard (EX1004, ¶¶8-10), but Mulroy does not expressly include all details of the H.264/AVC standard. Also, Mulroy explicitly adds to the H.264 codec system elements that the alleged inventors of the ’618

patent claim to have invented. In particular, Mulroy, in view of H.264/AVC, discloses each of the elements that the Applicant argued distinguished its claims from the prior art, including buffer description information that defines a plurality of reference pictures and a delta identifier, a picture identifier, and calculation of a reference picture identifier.

100. Figure 4 of Mulroy, reproduced below, shows operations of a decoder system at a remote receiver. The receiver receives (at 200) a network abstraction layer unit (“NALU”) that includes encoded data for a slice of a current frame. EX1004, ¶30. (According to the H.264/AVC standard, a picture is organized as multiple slices, and encoded data for a slice is received as a NALU. EX1005, 23-24 (§6.3), 58-61 (§7.4.1).)



EX1004, FIG. 4. The receiver checks that certain conditions have been satisfied, such as the NALU being timely received (at 201) in a packet with an expected packet number (at 203). EX1004, ¶30. If so, “[a]t 207, the decoder checks whether the current slice is for the same frame as the preceding slice.” *Id.* If the current slice and preceding slice are for the same frame (that is, the current slice is not the first slice to be decoded for the current frame), the current slice is decoded (at 215) as a non-first slice of the current frame. EX1004, ¶32.

101. Otherwise, the current slice is the first slice to be decoded for the current frame. After checking various conditions and performing operations (at 208-214), the current slice is decoded (at 215) as the first slice of the current frame. EX1004, ¶32. With respect to the conditions and operations (at 208-214), “it is also checked (208) whether the preceding frame has been fully decoded.” EX1004, ¶30. If the preceding frame was not fully decoded, the decoder performs operations to generate substitute content for the preceding frame (at 209) and insert the concealed frame in the DPB (at 210-212). *Id.* The decoder also checks (at 213) whether the current frame immediately follows the preceding frame. EX1004, ¶31. If not (“i.e., there are one or more whole frames missing”), to conceal the loss of the missing frame(s), the decoder generates substitute frame(s) for display. *Id.*

102. When decoding the encoded data for the current slice of the current frame (at 215), the decoder can use one or more reference pictures in the DPB of the decoder, which are managed as explained below. If the decoder has encoded data for the current slice but a reference picture used to decode the current slice is “absent from the buffer,” the decoder can use an “alternative frame that is present in the decoder.” EX1004, ¶33. Also, if errors are present in the current slice, the decoder may use “concealment techniques” to alleviate the errors. EX1004, ¶32. After the current frame is decoded, the decoder inserts the current frame in the DPB (at 217-219).

103. Mulroy teaches managing the reference pictures in the DPB of the decoder before, during, and after decoding a current picture to maintain the proper reference pictures in the buffer. EX1004, ¶¶30-32, FIG. 4. According to Mulroy, the reference pictures of the DPB of the decoder can be managed using MMCO commands (*see* EX1004, ¶26) as well as “‘reference picture list reordering’ commands defined by H.264” (*see* EX1004, ¶¶27-29). Mulroy teaches using existing H.264 coding tools, such as the encoded data for the current picture combined with the reference picture identifiers used for decoding the current picture, to define a plurality of reference pictures. EX1004, ¶¶21-22, 27. This method avoids sole reliance on potentially lost MMCO commands from previous pictures. *Id.* Mulroy makes use of H.264/AVC tools, such as the use of MMCO commands to delete pictures from the decoder’s reference buffer that the encoder has deleted (EX1004, ¶25), and Mulroy makes use of “sets” of MMCO messages, i.e., buffer description information (EX1004, ¶26). But Mulroy does not stop there. For its additions to the H.264 codec system, Mulroy indicates “every slice contains a message informing the decoder as to exactly which frames should be in the decoder buffer at that time.” EX1004, ¶27. Mulroy also states that “[o]ne way of doing this is to make use of the ‘reference picture list reordering’ commands defined by H.264.” *Id.* Mulroy reiterates:

The way we do this is for the encoder to insert remapping instructions in every slice (NAL unit) that reference all

frames in its buffer (there is not necessarily any changing of the default order), and for the decoder to use these to determine which frame to remove from its buffer when it otherwise does not have the required knowledge.

EX1004, ¶28.

104. Thus, during decoding of a current picture, the decoder in Mulroy uses the “remapping instructions” in the encoded data for the current picture to determine which frames should be in its buffer. EX1004, ¶27. The decoder can then delete a reference picture in its buffer that is not supposed to be in its buffer – “determine which frame to remove from its buffer.” EX1004, ¶28. As noted above, the decoder can also determine that a reference picture is missing from its buffer and instead use an alternative reference picture:

It can occur that a frame is received without errors but cannot be decoded correctly because its reference frame is absent from the buffer. This can be dealt with by decoding the NAL unit in box 215, using an alternative frame that is present in the decoder.

EX1004, ¶33.

B. H.264/AVC (EX1005)

105. The H.264/AVC standard was the leading video codec standard at the time the patent application for Mulroy was filed, and the November 2007 version of the H.264/AVC standard, published as H.264/AVC (EX1005), was widely known at the time as a state-of-the-art video compression standard. EX1001, 1:31-15. Earlier versions of the H.264/AVC standard were published in May 2003 and

March 2005. EX1005, 2. H.264/AVC was disseminated and made available on the ITU website (<https://www.itu.int/rec/T-REC-H.264/en>) before June 30, 2010. The posted editions of the H.264/AVC standard are accessible to the public, and can easily be found on the ITU-T website. Although I am not a member of the ITU, as an interested participant in the field, I was aware of the ITU standardization process (*see* ¶56 above) and where documents are available to the public on the ITU's website. I used the published source code of H.264 prior to 2004 to do the research that led to a published paper. V. Thilak and C.D. Creusere, *Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter*, Proceedings of the 12th European Signal Processing Conference, 281-84 (September 2004; Vienna, Austria). From my recollection and personal knowledge, the final, approved version of the third edition was posted and made publicly available on the ITU-T website on or around May 29, 2008, as is indicated on the ITU-T website. Other persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence could also locate and access it, without restriction, at that time. H.264/AVC is thus prior art under at least Sections 102(a-b). H.264/AVC is analogous art to the '618 patent because they are both within the same field of endeavor of video coding. EX1001, 1:24-27; EX1005, i (Summary).

106. An H.264 codec system, as referenced in Mulroy and described in H.264/AVC, uses syntax elements in reference picture list reordering information to identify reference pictures. EX1005, 47, 85-86; EX1004, ¶27. As discussed earlier, the reference picture list reordering information is signaled as part of a `ref_pic_list_reordering()` syntax structure, which is signaled as part of a `slice_header()` syntax structure for a slice:

7.3.3 Slice header syntax		
<code>slice_header() {</code>	C	Descriptor
<code> first_mb_in_slice</code>	2	ue(v)
<code> slice_type</code>	2	ue(v)
...		
<code> ref_pic_list_reordering()</code>	2	
<code> if((weighted_pred_flag && (slice_type == P slice_type == SP)) (weighted_bipred_idc == 1 && slice_type == B))</code>		
<code> pred_weight_table()</code>	2	

EX1005, 46 (§7.3.3) (excerpt) (highlighting added). The reference picture list reordering syntax (`ref_pic_list_reordering()`) signals an `abs_diff_pic_num_minus1` syntax element:

7.3.3.1 Reference picture list reordering syntax

ref_pic_list_reordering() {	C	Descriptor
if(slice_type % 5 != 2 && slice_type % 5 != 4) {		
ref_pic_list_reordering_flag_l0	2	u(1)
if(ref_pic_list_reordering_flag_l0)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
if(slice_type % 5 == 1) {		
ref_pic_list_reordering_flag_l1	2	u(1)
if(ref_pic_list_reordering_flag_l1)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
}		

EX1005, 47 (§7.3.3.1) (highlighting added). In particular, the `abs_diff_pic_num_minus1` syntax element (plus 1) indicates the difference between the picture number of a reference picture and a prediction value, which in some cases is the picture number (`frame_num`) of the current picture (`CurrPicNum`). EX1005, 79-85 (§7.4.3), 85-86 (§7.4.3.1), 117-118 (§8.2.4.3.1)).

The variable CurrPicNum is derived as follows.

- If field_pic_flag is equal to 0, CurrPicNum is set equal to frame_num.
- Otherwise (field_pic_flag is equal to 1), CurrPicNum is set equal to 2 * frame_num + 1.

EX1005, 82 (§7.4.3).

abs_diff_pic_num_minus1 plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value. **abs_diff_pic_num_minus1** shall be in the range of 0 to MaxPicNum – 1. The allowed values of **abs_diff_pic_num_minus1** are further restricted as specified in subclause 8.2.4.3.1.

EX1005, 86 (§7.4.3.1).

The variable picNumLXNoWrap is derived as follows.

- If reordering_of_pic_nums_idc is equal to 0
 - if(picNumLXPred – (abs_diff_pic_num_minus1 + 1) < 0)
 - picNumLXNoWrap = picNumLXPred – (abs_diff_pic_num_minus1 + 1) + MaxPicNum (8-35)
 - else
 - picNumLXNoWrap = picNumLXPred – (abs_diff_pic_num_minus1 + 1)

- Otherwise (reordering_of_pic_nums_idc is equal to 1),
 - if(picNumLXPred + (abs_diff_pic_num_minus1 + 1) >= MaxPicNum)
 - picNumLXNoWrap = picNumLXPred + (abs_diff_pic_num_minus1 + 1) – MaxPicNum (8-36)
 - else
 - picNumLXNoWrap = picNumLXPred + (abs_diff_pic_num_minus1 + 1)

picNumLXPred is the prediction value for the variable picNumLXNoWrap. When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of reordering_of_pic_nums_idc equal to 0 or 1 in the ref_pic_list_reordering() syntax), picNumL0Pred and picNumL1Pred are initially set equal to CurrPicNum. After each assignment of picNumLXNoWrap, the value of picNumLXNoWrap is assigned to picNumLXPred.

The variable picNumLX is derived as follows

- if(picNumLXNoWrap > CurrPicNum)
- picNumLX = picNumLXNoWrap – MaxPicNum (8-37)
- else
- picNumLX = picNumLXNoWrap

picNumLX shall be equal to the PicNum of a reference picture that is marked as “used for short-term reference” and shall not be equal to the PicNum of a short-term reference picture that is marked as “non-existing”.

EX1005, 117 (§8.2.4.3.1). In decoding, the value (abs_diff_pic_num_minus1 + 1)

is combined with (added to or subtracted from) the prediction value

picNumLXPred, which in some cases is the frame number of the current picture

(CurrPicNum). EX1005, 79-85 (§7.4.3), 117-118 (§8.2.4.3.1). “When the process specified in this subclause is invoked the first time for a slice (that is, for the first occurrence of `reordering_of_pic_nums_idc` equal to 0 or 1 in the `ref_pic_list_reordering()` syntax), `picNumL0Pred` and `picNumL1Pred` are initially set equal to `CurrPicNum`.” EX1005, 117 (§8.2.4.3.1). By signaling the picture number of a reference picture as a difference value compared to `CurrPicNum`, the number of bits used to signal the reference picture is typically reduced. *Id.*

107. According to the H.264/AVC standard, a decoder should infer loss of a picture when an entry in reference picture list reordering information refers (by frame number) to a short-term reference picture that does not exist. EX1005, 120 (§8.2.5.2).

NOTE 3 – The decoding process should infer an unintentional picture loss when any of these values of `frame_num` pertaining to “non-existing” pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or is referred to in the assignment process of a `LongTermFrameIdx` to a short-term reference picture (subclause 8.2.5.4.3). The decoding process should not infer an unintentional picture loss when a memory management control operation not equal to 3 is applied to a frame marked as “non-existing”.

Id. For a non-existing picture, the decoder generates a picture having samples that “may be set to any value,” and marks the generated picture “as ‘non-existing’ and ‘used for short-term reference.’” *Id.*, 119 (§8.2.5.2).

The decoding process generates and marks a frame for each of the values of `frame_num` pertaining to “non-existing” pictures, in the order in which the values of `UnusedShortTermFrameNum` are generated by Equation 7-22, using the “sliding window” picture marking process as specified in subclause 8.2.5.3. The generated frames are also marked as “non-existing” and “used for short-term reference”. The sample values of the generated frames may be set to any value. The bitstream shall not contain data that results in a reference to these generated frames which are marked as “non-existing” in the inter prediction process, a reference to these frames in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or a reference to these frames in the assignment process of a `LongTermFrameIdx` to a short-term reference picture (subclause 8.2.5.4.3).

Id. In some cases (when `pic_order_cnt_type` is not equal to 0), the decoder also generates picture order count values (`TopFieldOrderCnt` and `BottomFieldOrderCnt`) for the non-existing picture. *Id.*, 120 (§8.2.5.2).

When `pic_order_cnt_type` is not equal to 0, `TopFieldOrderCnt` and `BottomFieldOrderCnt` are derived for each of the "non-existing" frames by invoking the decoding process for picture order count in subclause 8.2.1. When invoking the process in subclause 8.2.1 for a particular "non-existing" frame, the current picture is considered to be a picture considered having `frame_num` inferred to be equal to `UnusedShortTermFrameNum`, `nal_ref_idc` inferred to be not equal to 0, `nal_unit_type` inferred to be not equal to 5, `IdrPicFlag` inferred to be equal to 0, `field_pic_flag` inferred to be equal to 0, `adaptive_ref_pic_marking_mode_flag` inferred to be equal to 0, `delta_pic_order_cnt[ 0 ]` (if needed) inferred to be equal to 0, and `delta_pic_order_cnt[ 1 ]` (if needed) inferred to be equal to 0.

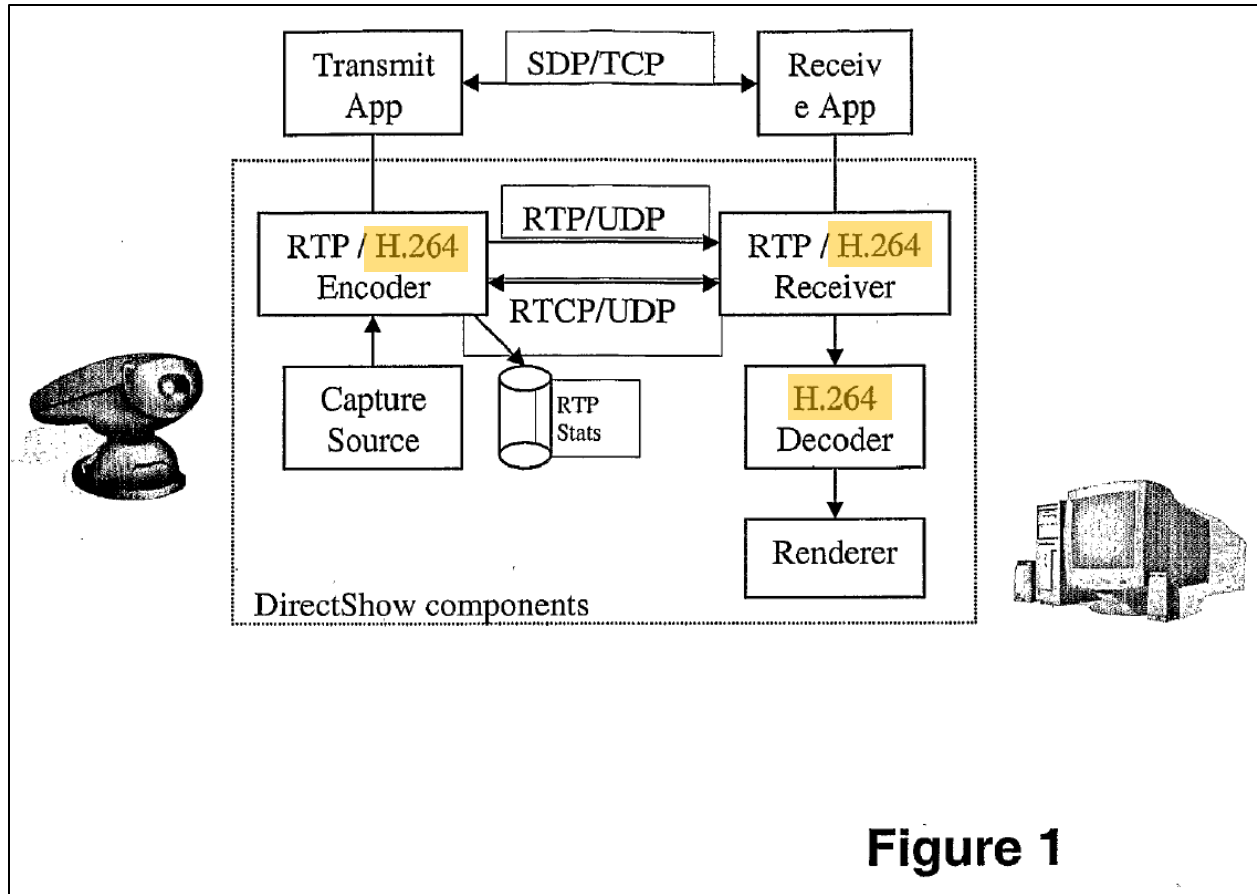
Id., 120 (§8.2.5.2).

108. Every time this syntax is used, the prediction value is the picture number (`frame_num`) of the current picture (`CurrPicNum`) for the first picture in the buffer. EX1005, 86 (“**abs_diff_pic_num_minus1** plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value.”), 80 (“**frame_num** is used as an identifier for pictures”), 82 (“`CurrPicNum` is set equal to `frame_num`”), 79-85, 85-86 (describing reference picture list reordering semantics), 117-118 (describing “[r]eordering process of reference picture lists for short-term reference pictures”). In decoding, the value (`abs_diff_pic_num_minus1 + 1`) is combined with (added to or subtracted from) the prediction value `picNumLXPred`. EX1005, 80, 117-118. The first time the reordering syntax is invoked for a given slice, “`picNumL0Pred` and `picNumL1Pred`,” these prediction values, “are initially set equal to `CurrPicNum`.” EX1005, 117-118 (§8.2.4.3.1). After each step, the

prediction values for the next frame in the reference picture list is identified using the frame number of the previous frame in the list, plus or minus $\text{abs_diff_pic_num_minus1} + 1$. *Id.* Thus, the frame numbers of each picture in the reference picture list are specified using only information in the current picture, not any prior information.

C. Combining Mulroy And H.264/AVC

109. A POSITA would have been motivated to combine the video coding methods of Mulroy with the teachings of H.264/AVC, and would have had a reasonable expectation of success in doing so, because Mulroy's teachings are expressly designed to be used with the "architecture of a real-time multi-reference H.264 encoder and decoder." EX1004, ¶8. FIG. 1 of Mulroy shows "a video transmission system incorporating apparatus according to embodiments of the invention," and expressly depicts H.264 elements such as an H.264 encoder and H.264 decoder:



EX1004, FIG. 1 (annotated), ¶4; *id.*, ¶¶9-13. Mulroy further describes the “[en]coder of FIG. 1” with regard to FIG. 2A and “the decoder of FIG. 1” with regard to FIG. 2B. EX1004, ¶¶5-6, 17-18.

110. Mulroy describes its teachings as a “modified version” of the H.264 encoder and decoder that “alleviate[s]” the scenario where “[i]f the decoder loses a set of MMCO commands, there will be a mismatch between the contents of the encoder and decoder buffers.” EX1004, ¶27. That is, Mulroy discloses the same problem and the same solution claimed in the ’618 patent, and relies on the H.264/AVC standard for certain aspects of the implementation. In some scenarios,

such as high packet-loss environments encountered in wireless transmissions, Mulroy's approach could be preferable to the standard H.264.

111. Given that H.264/AVC was the state-of-the-art version of the H.264/AVC standard at the time Mulroy was filed, a POSITA implementing Mulroy would have naturally turned to H.264/AVC for the relevant implementation details. H.264/AVC, in turn, discloses such implementation details as: the retrieval and use of a delta identifier to calculate a reference picture identifier, identifying reference pictures for subsequent pictures in the video stream, setting a picture order count of a non-existing picture, and marking non-existing pictures as non-existing.

112. Mulroy's reliance on the structures and implementation of H.264/AVC is explicit. For example, Mulroy states that a way of compensating for the loss of MMCO commands in encoded data for a previous picture, which is a stated goal of the '618 patent (EX1001, 6:34-37), would be "to make use of the 'reference picture list reordering' commands defined by H.264" in "every slice...informing the decoder as to exactly which frames should be in the decoder buffer at that time." EX1004, ¶27. By doing this, Mulroy-H.264/AVC can determine that a reference picture is missing from the decoder buffer and use an alternative reference picture instead of proceeding with a "mismatch between the

contents of the encoder and decoder buffers” (EX1004, ¶¶27, 33), which in some scenarios can be “severe to the decoding process” (EX1001, 2:43-45).

113. Elsewhere, Mulroy describes that the decoder in its codec system should receive encoded data for a slice of a current picture in a NALU syntax structure of H.264/AVC, and some types of NALUs contain slice data for a current picture. EX1004, ¶¶28, 30. A POSITA would have particularly recognized that Mulroy’s descriptions of how to employ specific syntax structures from H.264/AVC (e.g., EX1004, ¶¶27-30) and to use specific types of syntax structures and syntax elements (e.g., EX1004, ¶¶26-28) to provide a roadmap for an implementation of Mulroy with H.264/AVC, “in which every slice contains a message informing the decoder as to exactly which frames should be in the decoder buffer at that time.” EX1004, ¶27. Furthermore, Mulroy explicitly notes that “[f]urther information can be found” in two IEEE articles about H.264/AVC and implementations thereof for network delivery of encoded data. EX1004, ¶¶80-81 (citing EX1015, EX1016). A POSITA thus would have understood Mulroy as providing an explicit teaching, suggestion, and/or motivation to combine Mulroy with H.264/AVC.

114. A POSITA thus would have been motivated to use the elements of H.264/AVC described above when implementing Mulroy and would have had a reasonable expectation of success in doing so, because of the express directions in

Mulroy to use the syntax structures and syntax elements of H.264/AVC. Mulroy specifically gives directions “to make use of the ‘reference picture list reordering’ commands defined by H.264,” which includes the `ref_pic_list_reordering()` syntax structure and `abs_diff_pic_num_minus1` syntax element discussed in Section IX.B above. EX1004, ¶27. Examples of how a POSITA would implement the method of Mulroy using these syntax structures are laid out in Section IX.D below.

115. So implemented, the “modified version” of H.264/AVC proposed by Mulroy, which encodes every slice with “a message informing the decoder as to exactly which frames should be in the decoder buffer at that time” (EX1004, ¶27), provided improved error resilience due to redundancy in buffer management.

D. The Mulroy-H.264/AVC Combination

116. The decoder system of Mulroy, in combination with features of the H.264/AVC standard referenced in Mulroy, implements each of the elements of claims 1-7 of the '618 patent. The Mulroy-H.264/AVC combination describes that “every slice contains a message informing the decoder as to exactly which frames should be in the decoder buffer at that time.” EX1004, ¶27. A POSITA would have understood Mulroy as providing directions on how to use the H.264/AVC reordering commands in this way. EX1004, ¶27. Exemplary implementations in common use scenarios that follow Mulroy’s disclosure are described below.

1. The Mulroy-H.264/AVC Combination Can Maintain The Same Set Of Frames In The DPB From One Slice To the Next

117. One necessary function of the DPB is to maintain, for use in decoding a current slice, the reference frames that were in the DPB during decoding of a previous slice, and the Mulroy-H.264/AVC combination provides a method for doing this. For example, a POSITA would have understood that this could be implemented using ordinary skill and conventional techniques as illustrated by the example below with a DPB that stores up to five reference frames. The following set of commands could be used in a slice header of a current slice of a current frame with frame_num 150 to indicate the DPB is supposed to contain frames 149, 138, and 144 as short-term reference pictures, and frames 12 and 17 as long-term reference pictures, for decoding of the current slice.

118. **Initial state of DPB at decoder:** for decoding of the current frame with frame_num 150, suppose the DPB at the decoder contains frames 149, 138, 144, 12, and 17, where the underlined frames are marked as long-term reference frames.

119. **Step 1 (current DPB index 0):** in the slice header for the current slice, a reordering command indicates short-term reference frame 149 in DPB position 0, relative to the frame number of the current frame ($150 - (0 + 1) = 149$): the reordering command is given with the parameters

$\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 0$. The DPB contents after this operation are understood at the decoder to be *149*.

120. **Step 2 (current DPB index 1):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *138* in DPB position *1*, relative to the frame number of the reference frame in DPB position *0* ($149 - (10 + 1) = 138$), with $\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 10$. The DPB contents after this operation are understood at the decoder to be *149* and *138*.

121. **Step 3 (current DPB index 2):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *144* at DPB position 2. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 1$ and $\text{abs_diff_pic_num_minus1} = 5$. $138 + (5 + 1) = 144$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, and *144*.

122. **Step 4 (current DPB index 3):** In the slice header for the current slice, the next reordering command indicates long-term reference frame *12* at DPB position 3. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 2$ and $\text{long_term_pic_num} = 12$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, *144*, and *12*.

123. **Step 5 (current DPB index 4):** In the slice header for the current slice, the next reordering command indicates long-term reference frame *17* at DPB position *4*. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 2$ and $\text{long_term_pic_num} = 17$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, *144*, *12*, and *17*.

124. **Step 6:** In the slice header for the current slice, the next reordering command is given with the parameter $\text{reordering_of_pic_nums_idc} = 3$, which indicates the end of the reordering commands. The final DPB contents after processing this reordering command are understood at the decoder to be *149*, *138*, *144*, *12*, and *17*.

2. **The Mulroy-H.264/AVC Combination Can Remove Frames From The DPB To Recover From The Loss Of MMCO Commands**

125. Another exemplary operation provided by the Mulroy-H.264/AVC combination is to continue decoding a current frame even when an MMCO command in the encoded data for a previous frame has been lost. The Mulroy-H.264/AVC combination addresses the loss of the MMCO command (in the encoded data for the previous frame) when decoding the next slice for which the remapping instructions in the slice header are properly received. A POSITA would have understood that this would be implemented using ordinary skill and conventional techniques as illustrated by the following example.

126. As Mulroy describes, “[i]f the decoder loses a set of MMCO commands, there will be mismatch between the contents of the encoder and decoder buffers.” EX1004, ¶27. Suppose the decoder does not receive an MMCO command in the encoded data for a slice of a previous frame, where the MMCO command would cause a decoder to mark one or more reference frames as “unused for reference” (an MMCO command of type 1, 2, 4, or 5). The decoder incorrectly understands its DPB to still include the reference frame(s) that should have been marked as “unused for reference” according to the lost MMCO command. Even if the encoded data for a current slice of the current frame includes MMCO commands, those MMCO commands do not repeat the previous (lost) MMCO command to mark the reference frame(s) as “unused for reference” since the encoder assumes the reference frame(s) were previously marked as “unused for reference.” At the time of decoding the current slice of the current frame, the state of the DPB is different at the encoder and decoder. Mulroy addresses this scenario by inserting remapping instructions (as reference picture list reordering commands in H.264/AVC) in a slice header for the slice of the current frame. “The way we do this is for the encoder to insert remapping instructions in every slice (NAL unit) that reference all frames in its buffer (there is not necessarily any changing of the default order), and for the decoder to use these to determine which frame to

remove from its buffer when it otherwise does not have the required knowledge.”

EX1004, ¶28.

127. The following example illustrates how a POSITA would have understood Mulroy to recover from the loss of MMCO commands. In this example, the DPB at the decoder stores up to five reference frames. For a slice of a previous frame with *frame_num 149*, encoded data includes a single MMCO command that, if received and processed by the decoder, would cause the decoder to mark long-term reference frame *17* as “unused for reference” (MMCO command of type 2) and remove it from the DPB (since it was output long before). The decoder does not receive the MMCO command, however, even though Mulroy describes that its decoder can determine that the reference picture is missing from the decoder buffer and instead use an alternative reference picture. EX1004, ¶33. Nevertheless, after decoding the previous frame with *frame_num 149*, the decoder updates its DPB to contain the just-decoded frame with *frame_num 149*. Because space is available in the DPB at the decoder before frame *149* is inserted, no reference frame is “bumped” from the DPB (that is, marked as “unused for reference” and removed). When decoding a current slice in a current frame with *frame_num 150*, the decoder processes the following reordering commands in the slice header of the current slice, which reiterate the correct state of the DPB at the decoder.

128. **Initial state of DPB at decoder:** for decoding of the current frame with frame_num *150*, the DPB at the decoder contains frames *149*, *138*, *144*, *12*, and *17*, where the italicized frames are marked as long-term reference frames. Due to loss of the MMCO command in the encoded data for the previous frame, the decoder has not marked long-term reference frame *17* as “unused for reference” and removed it from the DPB at the decoder. For comparison, the encoder expects the DPB at the decoder to contain frames *149*, *138*, *144*, and *12*.

129. **Step 1 (current DPB index 0):** in the slice header for the current slice, a reordering command indicates short-term reference frame *149* in DPB position *0*, relative to the frame number of the current frame ($150 - (0 + 1) = 149$): the reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 0$. The DPB contents after this operation are understood at the decoder to be *149*.

130. **Step 2 (current DPB index 1):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *138* in DPB position *1*, relative to the frame number of the reference frame in DPB position *0* ($149 - (10 + 1) = 138$), with $\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 10$. The DPB contents after this operation are understood at the decoder to be *149* and *138*.

131. **Step 3 (current DPB index 2):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *144* at DPB position 2. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 1$ and $\text{abs_diff_pic_num_minus1} = 5$: $138 + (5 + 1) = 144$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, and *144*.

132. **Step 4 (current DPB index 3):** In the slice header for the current slice, the next reordering command indicates long-term reference frame *12* at DPB position 3. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 2$ and $\text{long_term_pic_num} = 12$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, *144*, and *12*.

133. **Step 5:** In the slice header for the current slice, the next reordering command is given with the parameter $\text{reordering_of_pic_nums_idc} = 3$, which indicates the end of the reordering commands. The final DPB contents after processing this reordering command are understood at the decoder to be *149*, *138*, *144*, and *12*. At this point, the decoder understands long-term reference frame *17* to no longer be in its DPB, and the decoder can remove long-term reference frame *17* from its DPB before decoding the current slice of the current frame with $\text{frame_num } 150$.

134. Although retaining an additional reference frame in the DPB (here, long-term reference frame *17*) might not interfere with the decoding of the current frame with frame_num *150*, the incorrectly retained reference frame continues to occupy a position in the DPB. This can cause problems in subsequent decoding. For example, if the current frame with frame_num *150* is added to the DPB (containing frames *149*, *138*, *144*, *12*, and *17*) using the sliding window process, a reference frame (such as short-term reference frame *138*) is incorrectly “bumped” from the DPB, which stores at most five reference frames. Specifically, if the DPB at the decoder incorrectly contains frames *149*, *138*, *144*, *12*, and *17* when frame *150* is inserted, frame *138* would be marked as “unused for reference” and removed. Going forward, the DPB at the decoder would incorrectly contain frames *150*, *149*, *144*, *12*, and *17*, instead of correctly containing frames *150*, *149*, *138*, *144*, and *12*.

3. The Mulroy-H.264/AVC Combination Can Add Frames To The DPB To Recover From The Loss Of Encoded Data For A Previous Frame

135. Another exemplary operation provided by the Mulroy-H.264/AVC combination is to continue decoding a current frame even when encoded data for a previous frame has been lost. The Mulroy-H.264/AVC combination addresses the loss of the encoded data for the previous frame when decoding the next slice for which encoded data, including remapping instructions in the slice header, is

properly received. A POSITA would have understood that this would be implemented using ordinary skill and conventional techniques as illustrated by the following example.

136. Mulroy describes that “[i]t can occur that a frame is received without errors, but cannot be decoded correctly because its reference frame is absent from the buffer. This can be dealt with by decoding the NAL unit in box 215, using an alternative frame that is present in the decoder.” EX1004, ¶33. Although error concealment operations using a generated, non-existing reference picture are outside the scope of the H.264/AVC standard, Mulroy describes options for setting the values of the samples of the generated picture. EX1004, ¶¶30 (“generate substitute slices for display (e.g. by copying pixels from a previous frame). Then, in Steps 210, 211 and 212 (which are the same as Steps 105, 106 and 107 at the encoder), the concealed frame is inserted into the buffer, deleting an existing frame if necessary.”), 36 (“error concealment techniques used may be any of those conventionally employed”). Suppose the decoder does not receive the encoded data for slices of a previous frame. At the time of decoding the current slice of the current frame, the state of the DPB is different at the encoder and decoder.

137. The following example illustrates how a POSITA would have understood Mulroy to recover from the loss of encoded data for a previous frame. In this example, the DPB at the decoder stores up to five reference frames. The

decoder receives and decodes encoded data for the previous frame with frame_num *148*. The decoder does not, however, receive the encoded data for the previous frame with frame_num *149*. Instead, the decoder next receives encoded data for the current frame with frame_num *150*. When decoding a current slice in the current frame with frame_num *150*, the decoder processes the following reordering commands in the slice header of the current slice, which reiterate the correct state of the DPB at the decoder.

138. **Initial state of DPB at decoder:** for decoding of the current frame with frame_num *150*, the DPB at the decoder contains frames *148*, *138*, *144*, *12*, and *17*, where the italicized frames are marked as long-term reference frames. Due to loss of the encoded data for the previous frame *149*, the decoder has not added short-term reference frame *149* to its DPB, nor has the decoder marked short-term reference frame *148* as “unused for reference” and removed it from the DPB at the decoder. For comparison, the encoder expects the DPB at the decoder to contain frames *149*, *138*, *144*, *12*, and *17*. (In this example, an MMCO command in the encoded data for previous frame *149* that was lost should have caused the decoder to mark reference frame *148* as “unused for reference” and remove frame *149* from its DPB. Alternatively, a different frame, such as short-term reference frame *138* or long-term reference frame *17* could be removed from the DPB, but that does not significantly change this example.)

139. **Step 1 (current DPB index 0):** in the slice header for the current slice, a reordering command indicates short-term reference frame *149* in DPB position *0*, relative to the frame number of the current frame ($150 - (0 + 1) = 149$): the reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 0$. The DPB contents after this operation are understood at the decoder to be *149*.

140. **Step 2 (current DPB index 1):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *138* in DPB position *1*, relative to the frame number of the reference frame in DPB position *0* ($149 - (10 + 1) = 138$), with $\text{reordering_of_pic_nums_idc} = 0$ and $\text{abs_diff_pic_num_minus1} = 10$. The DPB contents after this operation are understood at the decoder to be *149* and *138*.

141. **Step 3 (current DPB index 2):** in the slice header for the current slice, the next reordering command indicates short-term reference frame *144* at DPB position *2*. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 1$ and $\text{abs_diff_pic_num_minus1} = 5$: $138 + (5 + 1) = 144$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, and *144*.

142. **Step 4 (current DPB index 3):** In the slice header for the current slice, the next reordering command indicates long-term reference frame *12* at DPB

position 3. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 2$ and $\text{long_term_pic_num} = 12$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, *144*, and 12.

143. **Step 5 (current DPB index 4):** In the slice header for the current slice, the next reordering command indicates long-term reference frame *17* at DPB position 4. The reordering command is given with the parameters $\text{reordering_of_pic_nums_idc} = 2$ and $\text{long_term_pic_num} = 17$. The DPB contents after this operation are understood at the decoder to be *149*, *138*, *144*, 12, and 17.

144. **Step 6:** In the slice header for the current slice, the next reordering command is given with the parameter $\text{reordering_of_pic_nums_idc} = 3$, which indicates the end of the reordering commands. The final DPB contents after processing this reordering command are understood at the decoder to be *149*, *138*, *144*, 12, and 17. At this point, the decoder understands short-term reference frame *149* should be in its DPB. The decoder can generate a substitute frame *149* using one of the concealment approaches described in Mulroy (e.g., using “alternative frame that is present in the decoder” (EX1004, ¶33) or generating “substitute slices for display (e.g. by copying pixels from a previous frame)” (EX1004, ¶30)). The substitute frame can be inserted in the DPB for use in decoding the current slice of the current frame with $\text{frame_num } 150$. EX1004, ¶30 (“the concealed frame is inserted into the buffer, deleting an existing frame if necessary.”). At this point, the

decoder also understands short-term reference frame 148 should no longer be in its DPB, and the decoder can remove reference frame 148 from its DPB.

E. The Mulroy-H.264/AVC Combination Renders Claims 1-7 Of The '618 Patent Obvious

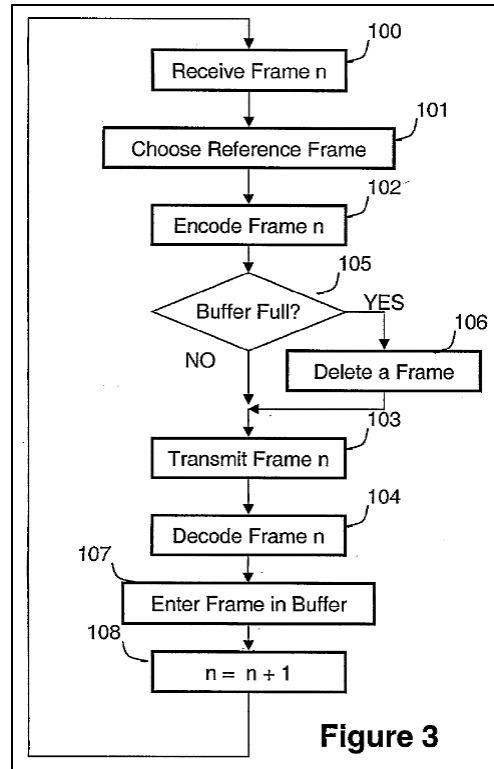
1. Claim 1

[1.1] A method of decoding an encoded representation of a picture in a video stream of multiple pictures, the method comprising:

145. To the extent the preamble is limiting, Mulroy-H.264/AVC discloses it. Mulroy is titled “Video Coding” and describes “[v]ideo frame capture and H.264 real-time encode” (an encoded representation of a picture in a video stream of multiple pictures) and “H.264 real-time decode and display” (decoding an encoded representation of a picture in a video stream of multiple pictures).

EX1004, ¶¶8-10; *id.*, ¶1 (“present invention concerns video coding”); EX1005, 18 (“The video source that is represented by the bitstream is a sequence of either or both frames or fields (called collectively pictures) in decoding order.”). Further, a POSITA understood that a video stream comprises multiple pictures, and that video coding involved the decoding of an encoded representation of a picture in a video stream of multiple pictures. *Supra* Section VI.A.

146. For example, with respect to encoding, Mulroy describes encoding a number of pictures. EX1004, ¶22. The method of encoding described in Mulroy is summarized in the flowchart of FIG. 3, shown below.



EX1004, FIG. 3; *id.*, ¶22 (“FIG. 3 shows the operation of the encoder.”). With respect to decoding, Mulroy also discloses a method of decoding an encoded representation of a picture in a video stream of multiple pictures. *E.g.*, EX1004, Abstract (describing that encoded pictures are decoded by a decoder at a remote receiver). The decoding method of Mulroy is illustrated in Figure 4, shown below.

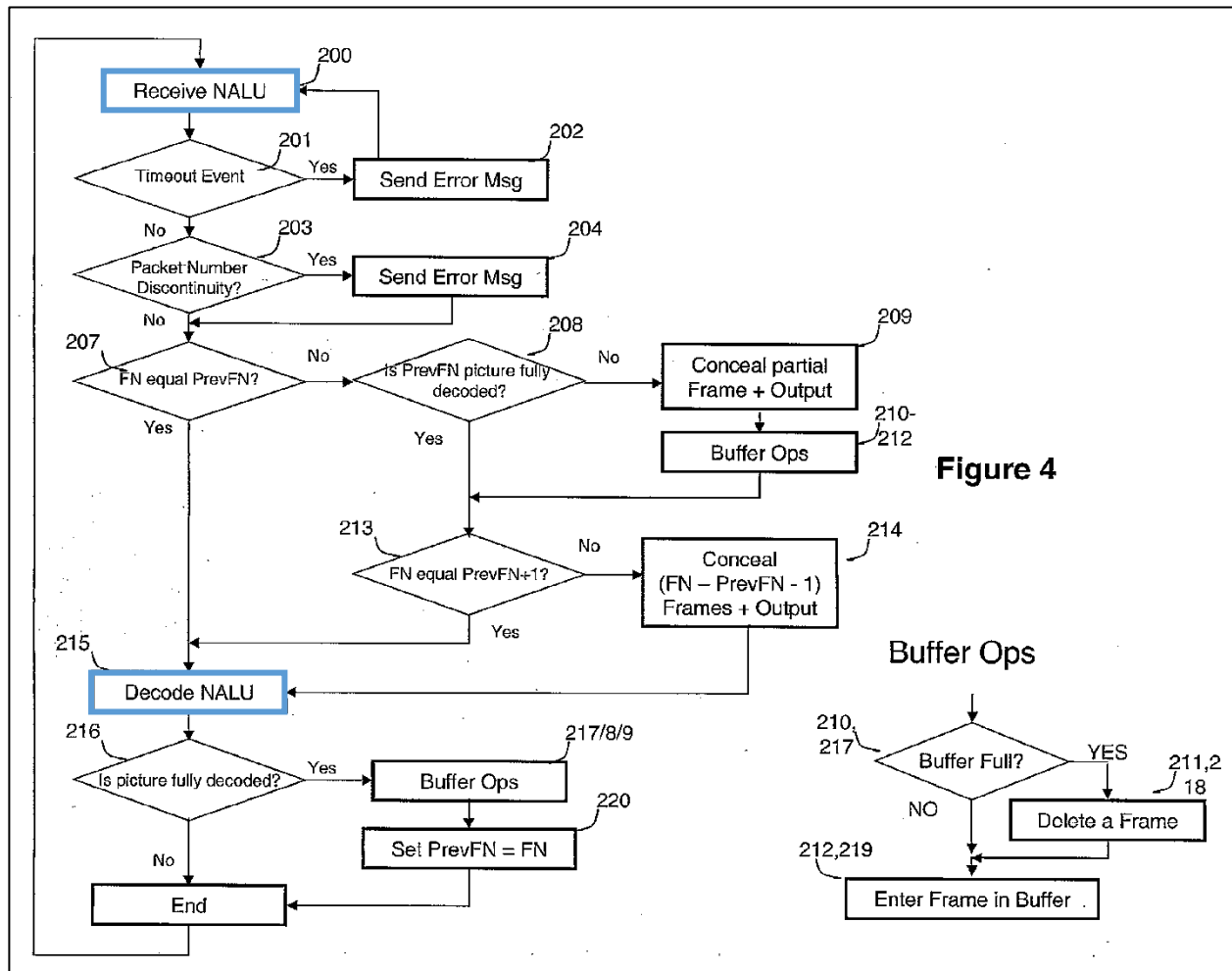


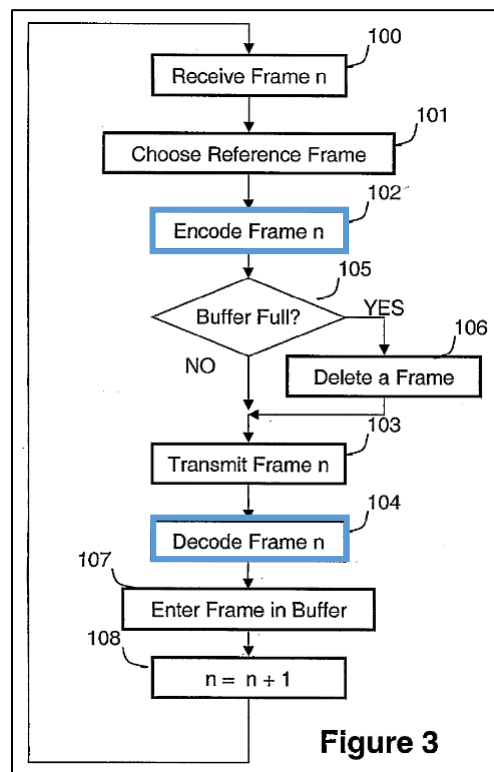
Figure 4

EX1004, FIG. 4 (annotated). A decoder receives encoded data for a slice in a NALU at step 200 (EX1004, ¶30) and decodes the encoded data for the slice at step 215 (EX1004, ¶32). The decoder repeats the operations of receiving and decoding encoded data in NALUs for slices of a current picture. EX1004, ¶32.

[1.2] retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures;

147. Mulroy-H.264/AVC renders this element obvious because Mulroy discloses it.

148. Mulroy discloses “retrieving, from the encoded representation of the picture,...” the information recited in this element. Mulroy shows operations of an encoder in FIG. 3 (below) that “insert[s] remapping instructions in every slice (NAL unit) that reference all frames in its buffer.” EX1004, ¶28. These remapping instructions are added to the encoded representation of the picture based on reference pictures in a buffer. EX1004, ¶¶21-22.



EX1004, FIG. 3 (annotated).

149. The buffer description information would then be decoded, and thus retrieved, from the encoded frame. When the decoder receives encoded data in NALUs for slices of a picture and decodes the encoded data, the decoder retrieves remapping instructions from the encoded data and “use[s] these to determine which

frame to remove from its buffer when it otherwise does not have the required knowledge.” EX1004, ¶28. Mulroy thus discloses “retrieving, from the encoded representation of the picture.”

150. Mulroy further discloses that what is retrieved is “...*buffer description information....*” Mulroy describes a “modified version [of H.264] in which every slice contains a message informing the decoder as to *exactly which frames should be in the decoder buffer at that time.*” EX1004, ¶27 (emphasis added).

151. According to Mulroy, the encoder can change which reference pictures are stored in the buffer, for example, by adding a newly decoded picture or deleting an existing reference picture. EX1004, ¶¶23-26. Mulroy describes sending a “message” in every slice “informing the decoder as to exactly which frames should be in the decoder buffer at that time.” EX1004, ¶27. To do this, the message can use “‘reference picture list reordering’ commands defined by H.264” as “remapping instructions,” where the “remapping instructions can be made use of in the present context for indicating to the decoder which frames (by their frame numbers) should be in its buffer.” *Id.* These commands can be used to maintain the order of the pictures in the reference picture list, remove pictures from the list, or rearrange pictures identified by it, and specific examples of each of these operations are laid out in Section IX.D above.

152. Because this “message” in Mulroy states the frames that should be in the decoder buffer, i.e., describes the contents of the buffer, a POSITA would have understood it to be “buffer description information.”

153. Mulroy further discloses that its buffer description information “...defin[es] a plurality of reference pictures.” Mulroy describes that “the transmitted information includes the reference indices of the predictor, or all the predictors, used so that the decoder can use the correct one(s) for decoding,” which indicates identification of reference pictures in the encoder buffer (and decoder buffer) by the reference indices. EX1004, ¶22. Mulroy states this process for decoding was “known.” *Id.*

154. A POSITA would have understood that Mulroy discloses an unambiguous identification in the message that informs “the decoder as to exactly which frames should be in the decoder buffer at that time.” EX1004, ¶27. According to Mulroy, the message can use reference picture list reordering commands as in H.264/AVC to indicate the frames in a decoder buffer. *Id.* Mulroy does not require using reference picture list reordering commands to indicate the frames in the decoder buffer, but instead describes “‘reference picture list reordering’ commands defined by H.264” as “[o]ne way of doing this.” *Id.* Reference picture list reordering commands, used even when reference pictures are not reordered, provide a way to indicate the frames in the decoder buffer.

155. A POSITA would have understood that a way to implement this buffer description information would be to use a series of H.264/AVC reference picture list reordering commands. *Supra* §IX.D (discussing detailed examples of such implementation). By including the set of commands in the slice header, the buffer description information indicates the contents of the DPB.

[1.3] determining, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture, wherein determining the reference picture identifier comprises:

156. Mulroy-H.264/AVC renders this element obvious because Mulroy discloses a decoder determining, based on reordering commands in encoded data for a slice of a current picture, a reference picture identifier (frame number of a reference picture) that identifies a reference picture among the reference pictures usable to decode the current picture. EX1004, ¶27.

157. Mulroy describes that “every slice contains a message informing the decoder as to exactly which frames should be in the decoder buffer at that time. One way of doing this is to make use of the ‘reference picture list reordering’ commands defined by H.264.” EX1004, ¶27.

158. Importantly, Mulroy also describes that the encoder should “insert remapping instructions in every slice (NAL unit) that reference all frames in its buffer (there is not necessarily any changing of the default order)” and that the

decoder then should “use these to determine which frame to remove from its buffer when it otherwise does not have the required knowledge.” EX1004, ¶28.

[1.4] (i) retrieving, based on the buffer description information, a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture; and

159. Mulroy-H.264/AVC renders this element obvious because H.264/AVC discloses it.

160. H.264/AVC teaches retrieving an `abs_diff_pic_num_minus1` syntax element from a reference picture list reordering syntax structure (`ref_pic_list_reordering()`) in a slice header. *See* EX1005, 47 (§7.3.3.1, laying out the “[r]eference picture list reordering syntax” structure), 85-86 (§7.4.3.1, explaining the “[r]eference picture list reordering semantics”). H.264/AVC defines the `abs_diff_pic_num_minus1` syntax element:

`abs_diff_pic_num_minus1` plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value. `abs_diff_pic_num_minus1` shall be in the range of 0 to `MaxPicNum - 1`. The allowed values of `abs_diff_pic_num_minus1` are further restricted as specified in subclause 8.2.4.3.1.

EX1005, 86 (§7.4.3.1). As defined, the `abs_diff_pic_num_minus1` syntax element indicates a difference between the picture number (`frame_num`) of a reference picture and a picture number prediction value. *Id.*; *see also id.*, 79-85 (§7.4.3), 85-86 (§7.4.3.1), 117-118 (§8.2.4.3.1) For the first reordering command in each slice header for which the `reordering_of_pic_nums_idc` syntax elements is equal to 0 or 1, the picture number prediction value is “a picture identifier identifying the

picture,” i.e., the picture number (frame_num) of the current picture (CurrPicNum). *Id.* In other words, the abs_diff_pic_num_minus1 syntax element associated with the first reordering command is “a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture.”

161. Sections 7.4.3, 7.4.3.1, and 8.2.4.3.1 of H.264/AVC describe a structure for reordering a “plurality of reference pictures” for efficient coding. EX1005, 79-85 (§7.4.3), 85-86 (§7.4.3.1), 117-118 (§8.2.4.3.1). Mulroy describes using the same reference picture list reordering information to reiterate which reference pictures should be in a decoder buffer. EX1004, ¶¶27-28. Thus, the abs_diff_pic_num_minus1 syntax element of Mulroy-H.264/AVC is an example of a “delta identifier” in claim 1, which is “a difference or delta identifier or value” as described in the ’618 patent. EX1001, 9:48.

[1.5] (ii) calculating the reference picture identifier based on the picture identifier and the delta identifier; and

162. Mulroy-H.264/AVC renders this element obvious because H.264/AVC teaches calculating the reference picture identifier based on the picture identifier and the delta identifier. For example, H.264/AVC describes using a delta identifier, its abs_diff_pic_num_minus1 syntax element, from a reference picture list reordering syntax structure (ref_pic_list_reordering()) in a slice header, to

determine the picture number of a reference picture. EX1005, 47 (§7.3.3.1). The decoder combines the value $(\text{abs_diff_pic_num_minus1} + 1)$ with the prediction value picNumLXPred , which when the reordering process is “invoked the first time for a slice,” it is “initially set equal to CurrPicNum ” to determine a variable picNumLXNoWrap indicating the picture number of the reference picture. EX1005, 117-118 (§8.2.4.3.1). Thus, Mulroy-H.264/AVC discloses a delta identifier or at least renders it obvious.

163. The delta identifier is used to calculate the reference picture identifier as follows. If the value of the $\text{reordering_of_pic_nums_idc}$ syntax element is 0, the decoder subtracts the value $(\text{abs_diff_pic_num_minus1} + 1)$ from the variable picNumLXPred . EX1005, 117 (§8.2.4.3.1 (equation 8-35)). If the value of the $\text{reordering_of_pic_nums_idc}$ syntax element is 1, the decoder adds the value $(\text{abs_diff_pic_num_minus1} + 1)$ to the variable picNumLXPred . *Id.* (equation 8-36). When $\text{reordering_of_pic_nums_idc}$ is 0 or 1 for the first time in a reference picture list reordering syntax structure, the variable picNumLXPred is equal to CurrPicNum , which is the frame number (frame_num) of the current picture, i.e., the picture identifier. EX1005, 79-85 (§7.4.3), 117-118 (§8.2.4.3.1). Thus, Mulroy-H.264/AVC discloses “calculating the reference picture identifier based on the picture identifier and the delta identifier” once for each slice that has a $\text{reordering_of_pic_nums_idc}$ syntax element of 0 or 1. *See supra* §IX.D (laying out

examples of how the `abs_diff_pic_num_minus1` syntax element would be used in the Mulroy-H.264/AVC method).

[1.6] updating a decoded picture buffer based on the reference picture identifier.

164. Mulroy-H.264/AVC renders this element obvious because Mulroy discloses the decoder using the picture number of the reference picture (determined using “remapping instructions” in the encoded data for the current picture) to update its buffer. EX1004, ¶¶28, 33. Mulroy discloses “using the same algorithm as used at the coder for selecting pictures to be deleted” (EX1004, ¶29) as an alternative, instead of using, for example, the “rules for the reference buffer management” that Mulroy lays out at ¶¶47-51 to manage the DPB. Mulroy further discloses that when the decoder “does not have the required knowledge,” the decoder can use the “remapping instructions in every slice (NAL unit) that reference all frames in its buffer” to “determine which frame to remove from its buffer,” thus providing redundancy if, for example, the MMCO commands are lost. EX1004, ¶28.

2. Claim 2

[2.1] The method according to claim 1, wherein:

165. Mulroy-H.264/AVC renders claim 1 obvious. *Supra* Section IX.E (claim 1).

[2.2] determining the reference picture identifier comprises determining, based on the buffer description information,

166. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E (elements 1.3, 1.4, and 1.5).

[2.3] i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and

167. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E (element 1.3).

[2.4] ii) a picture identifier identifying a respective reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video stream.

168. Mulroy-H.264/AVC renders this element obvious because H.264/AVC teaches a picture identifier identifying a respective reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video stream. It would have also been natural and obvious to implement Mulroy with H.264/AVC because Mulroy was explicitly designed to be implemented with H.264/AVC. *Supra* Section IX.C.

169. H.264/AVC describes signaling reference picture list reordering information in the ref_pic_list_reordering() syntax structure, which is part of the slice_header() syntax structure for a slice. EX1005, 47 (§7.3.3.1). This reordering information can include one or more abs_diff_pic_num_minus1 syntax elements, each identifying a picture number of a short-term reference picture. EX1005, 47

(§7.3.3.1), 85-86 (§7.4.3.1). This reordering information can include long_term_pic_num syntax elements, identifying the picture number of a long-term reference picture. *Id.*

170. A POSITA would have understood that some reference pictures identified as long-term reference pictures are used during decoding of a subsequent picture in the decoding order. In some cases, a given long-term reference picture is used for decoding both the current picture and a subsequent picture in Mulroy-H.264/AVC, while in other cases, a given long-term reference picture is used for decoding a subsequent picture but not the current picture in Mulroy-H.264/AVC. *Compare* EX1001, 8:32-38 (“the encoded picture therefore carries the buffer description information that can be used at the decoder to define and identify the reference pictures that are needed to decode the current picture and/or any subsequent picture of the video stream.”); *see also id.*, 15:35-43, 15:44-53, 22:11-30.

3. Claim 3

[3.1] The method according to claim 1, further comprising:
--

171. As discussed above, Mulroy-H.264/AVC satisfies the elements of claim 1. *Supra* Section IX.E (claim 1).

[3.2] determining whether the reference picture identified by the reference picture identifier is one of missing or non-existing; and

172. Mulroy-H.264/AVC renders this element obvious because Mulroy teaches that its decoder determines that a reference picture is missing from its buffer and instead uses an alternative reference picture. EX1004, ¶33.

[3.3] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generating a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier.

173. Mulroy-H.264/AVC renders this element obvious because Mulroy discloses generating a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier, responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, in view of the teachings of H.264/AVC as discussed below.

174. According to the H.264/AVC standard, a decoder should infer loss of a picture when an entry in reference picture list reordering information refers to a reference picture that does not exist in the DPB. EX1005, 119-120 (§8.2.5.2).

NOTE 3 – The decoding process should infer an unintentional picture loss when any of these values of frame_num pertaining to “non-existing” pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or is referred to in the assignment process of a LongTermFrameIdx to a short-term reference picture (subclause 8.2.5.4.3). The decoding process should not infer an unintentional picture loss when a memory management control operation not equal to 3 is applied to a frame marked as “non-existing”.

Id., 120. For a non-existing picture, according to the H.264/AVC standard, the decoder generates a picture with samples that “may be set to any value,” and marks the generated picture “as ‘non-existing’ and ‘used for short-term reference.’” *Id.*, 119.

175. According to Mulroy, its decoder can determine that a reference picture is missing from the decoder buffer and instead use an alternative reference picture. EX1004, ¶33. A POSITA, reading this description in Mulroy, would have understood that one way to provide an “alternative frame” would be to generate a non-existing reference picture as specified in the H.264/AVC standard—with samples that “may be set to any value” and marking the generated picture “as ‘non-existing’ and ‘used for short-term reference.’” *Id.* Although error concealment operations using a generated, non-existing reference picture are outside the scope of the H.264/AVC standard, Mulroy describes options for setting the values of the samples of the generated picture. EX1004, ¶¶30, 36.

4. Claim 4

[4.1] The method according to claim 3, wherein generating the non-existing picture with parameters comprises at least one of:

176. Mulroy-H.264/AVC renders claim 3 obvious. *Supra* Section IX.E (claim 3).

177. For Mulroy-H.264/AVC to render this claim obvious, it need only disclose or render obvious one of the four options for generating the non-existing picture. The language “at least one of ... and ...” in this context could not indicate “one or more” because it is describing method steps, not objects. Mulroy-H.264/AVC discloses at least limitations [4.2] and [4.5], and thus renders claim 4 obvious.

[4.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

178. Mulroy-H.264/AVC renders this element obvious because H.264/AVC teaches setting a picture order count (POC) of the non-existing picture to a POC of the reference picture. According to H.264/AVC, a decoder should infer loss of a picture when an entry in reference picture list reordering information refers to a reference picture that does not exist in the DPB. EX1005, 119-120 (§8.2.5.2).

NOTE 3 – The decoding process should infer an unintentional picture loss when any of these values of frame_num pertaining to “non-existing” pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or is referred to in the assignment process of a LongTermFrameIdx to a short-term reference picture (subclause 8.2.5.4.3). The decoding process should not infer an unintentional picture loss when a memory management control operation not equal to 3 is applied to a frame marked as “non-existing”.

Id., 120. In some cases (when pic_order_cnt_type is not equal to 0), the decoder generates picture order count values (TopFieldOrderCnt and BottomFieldOrderCnt) for the non-existing picture. *Id.*

179. According to Mulroy, its decoder can determine that a reference picture is missing from the decoder buffer and instead use an alternative reference picture. EX1004, ¶33. A POSITA, reading this description in paragraph 33 of Mulroy, would have understood that one way to provide an “alternative frame” would be to generate a non-existing reference picture as specified in H.264/AVC – with picture order count values generated for the non-existing reference picture as specified in H.264/AVC.

[4.3] setting an `additional_picture_id` of the non-existing picture to an `additional_picture_id` of the reference picture;

180. Mulroy-H.264/AVC discloses at least elements [4.2] and [4.5], and thus renders claim 4 obvious. Element [4.3] is not addressed at this time.

[4.4] setting a `temporal_id` of the non-existing picture to a `temporal_id` of the reference picture; and

181. Mulroy-H.264/AVC discloses at least elements [4.2] and [4.5], and thus renders claim 4 obvious. Element [4.4] is not addressed at this time.

[4.5] marking the non-existing picture with a non-existing mark and used for reference mark.

182. Mulroy-H.264/AVC renders this element obvious because H.264/AVC teaches marking the non-existing picture with a non-existing mark and used for reference mark. According to H.264/AVC, a decoder should infer loss of a

picture when an entry in reference picture list reordering information refers to a reference picture that does not exist in the DPB. EX1005, 119-120 (§8.2.5.2).

NOTE 3 – The decoding process should infer an unintentional picture loss when any of these values of frame_num pertaining to “non-existing” pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term reference pictures (subclause 8.2.4.3.1), or is referred to in the assignment process of a LongTermFrameIdx to a short-term reference picture (subclause 8.2.5.4.3). The decoding process should not infer an unintentional picture loss when a memory management control operation not equal to 3 is applied to a frame marked as “non-existing”.

Id., 120. For a non-existing picture, according to H.264/AVC, the decoder generates a picture and marks the generated picture “as ‘non-existing’ and ‘used for short-term reference.’” *Id.*, 119.

183. According to Mulroy, its decoder can determine that a reference picture is missing from the decoder buffer and instead use an alternative reference picture. “It can occur that a frame is received without errors, but cannot be decoded correctly because its reference frame is absent from the buffer. This can be dealt with by decoding the NAL unit in box 215, using an alternative frame that is present in the decoder.” EX1004, ¶33. A POSITA, reading this description in Mulroy, would have understood that one way to provide an “alternative frame” would be to generate a non-existing reference picture as specified in H.264/AVC – including marking the generated picture “as ‘non-existing’ and ‘used for short-term reference.’” EX1005, 119. In FIG. 4 of Mulroy, this operation takes place in box 215. EX1004, ¶33. After the operations of box 215 are completed, the process continues through boxes 216-220 until the new frame is added to the buffer and reference picture list. EX1004, ¶32.

5. Claim 5

[5.1] A method of decoding an encoded representation of a picture in a video stream of multiple pictures, the method comprising:

184. Mulroy-H.264/AVC renders this obvious. *Supra* Section IX.E
(element 1.1).

[5.2] retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures;

185. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E
(element 1.2).

[5.3] determining, based on the buffer description information, a reference picture identifier that identifies a reference picture among the plurality of reference pictures that comprises a decoding reference for the picture;

186. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E
(element 1.3).

[5.4] determining whether the reference picture identified by the reference picture identifier is one of missing or non-existing;

187. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E
(element 3.2).

[5.5] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generating a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier; and

188. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E (element 3.3).

[5.6] updating a decoded picture buffer based on the reference picture identifier.

189. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E (element 1.6).

6. Claim 6

[6.1] The method according to claim 5, wherein generating the non-existing picture with parameters comprises at least one of:

190. Mulroy-H.264/AVC renders claim 5 obvious. *Supra* Section IX.E (claim 5). For Mulroy-H.264/AVC to render this claim obvious, it need only disclose or render obvious one of the four options for generating the non-existing picture. *Supra* Section IX.E (element 4.1).

[6.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

191. Mulroy-H.264/AVC renders this element obvious. *Supra* Section IX.E (element 4.2).

[6.3] setting an additional_picture_id of the non-existing picture to an additional_picture_id of the reference picture;

192. Mulroy-H.264/AVC discloses at least elements 6.2 and 6.5. Element 6.3 is not addressed at this time.

[6.4] setting a temporal_id of the non-existing picture to a temporal_id of the reference picture; and

193. Mulroy-H.264/AVC discloses at least elements 6.2 and 6.5. Element 6.4 is not addressed at this time.

[6.5] marking the non-existing picture with a non-existing mark and used for reference mark.

194. Mulroy-H.264/AVC renders this element obvious, as discussed above.
Supra Section IX.E (element 4.5).

7. Claim 7

[7.1] The method according to claim 5, wherein determining the reference picture identifier comprises:

195. As discussed above, Mulroy-H.264/AVC satisfies the elements of claim 5. *Supra* Section IX.E (claim 5).

[7.2] (i) retrieving, based on the buffer description information, a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture; and

196. Mulroy-H.264/AVC renders this element obvious, as discussed above.
Supra Section IX.E (element 1.4).

[7.3] (ii) calculating the reference picture identifier based on the picture identifier and the delta identifier.

197. Mulroy-H.264/AVC renders this element obvious, as discussed above.
Supra Section IX.E (element 1.5).

**X. GROUND 2: CLAIMS 8-19 ARE OBVIOUS OVER
MULROY IN VIEW OF H.264/AVC AND RAVEENDRAN**

A. Raveendran (U.S. 2006/0165176; EX1006)

198. Raveendran is a U.S. patent application publication that was published on July 27, 2006, and thus is prior art under at least Sections 102(a-b). EX1006, Cover. Raveendran is analogous art to the '618 patent because they are both in the same field of endeavor of video coding. EX1006, ¶¶4, 14.

199. Raveendran discloses a method and apparatus to perform frame rate up conversion for video compression. Specifically, Raveendran discusses performing operations on bitstreams that are defined by video coding standards and discusses aspects of H.264/AVC. EX1006, ¶¶65-66, 70-71, 73, 75-76. Raveendran also discloses the use of the logic it describes in “a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein.” EX1006, ¶84. It further discusses implementation of video compression algorithms in an access point in a network. EX1006, ¶¶77-81.

200. The disclosures that Raveendran made in 2006 about operating video coding systems in software and hardware implementations provide at least as much

detail as the specification of the '618 patent. *Compare* EX1001, 25:22-42, 28:66-29:7, *with* EX1006, ¶¶77-85.

B. Combining Mulroy, H.264/AVC, And Raveendran

201. A POSITA would have been motivated to combine the video coding methods taught by Mulroy and H.264/AVC with the hardware and/or software implementations taught by Raveendran, and would have had a reasonable expectation of success in doing so, because Raveendran provides express teachings, suggestions, and/or motivations to implement such video coding methods in specific hardware and/or software environments, and its disclosure reflects that an ordinary level of skill would easily do so. Although Mulroy and H.264/AVC do not expressly disclose hardware and/or software implementations, a POSITA would have recognized that such implementations would have been necessary to perform the video coding methods they teach. Raveendran, in turn, expressly teaches the implementation in hardware and/or software of video coding methods such as H.264/AVC in specific hardware and software environments. EX1006, ¶¶77-86. Raveendran teaches methods that are “[s]tandards [c]ompliant” with standards such as H.264/AVC (EX1006, ¶¶70-72) and further teaches that “the methods described herein may be implemented on a variety of communication hardware, processors and systems known by one of ordinary skill in the art”

(EX1006, ¶83). Raveendran also gives express examples of such communication hardware, processors and systems. EX1006, ¶¶77-86.

202. Thus, when implementing Mulroy and H.264/AVC, a POSITA would have been motivated, with a reasonable expectation of success, to do so in hardware and/or software implementations, as described in Raveendran. The POSITA would only need to combine known prior art elements, such as the methods of Mulroy-H.264/AVC and the specific known “communication hardware, processors and systems known by one of ordinary skill in the art” (EX1006, ¶83) taught by Raveendran, to achieve predictable results, namely encoding and decoding video as taught by Mulroy-H.264/AVC. The POSITA knew how to do so, as Raveendran explains that its disclosure “enable[s] any person skilled in the art to make or use the various embodiments.” EX1006, ¶86.

203. The obviousness to combine these references is further supported by the '618 patent's limited disclosures on implementation, admitting that one of skill in the art would not have needed further disclosure to expect successful implementation. For example, the '618 patent's discussion of software (EX1001, 25:1-2, 28:46-47) and hardware (EX1001, 25:26-30, 29:3-7) implementations is given at a high level, and does not include any discussion of the network being used. In contrast, Raveendran's discussions of software and hardware

implementations gives at least as much detail and includes a discussion of the network architecture used. EX1006, ¶¶77-85.

C. The Mulroy-H.264/AVC-Raveendran Combination

204. The resulting Mulroy-H.264/AVC-Raveendran combination, which includes Mulroy-H.264/AVC described above implemented in hardware/software environments as taught by Raveendran, renders claims 8-19 obvious as discussed in the following sections.

1. Claim 8

[8.1] A decoder configured to decode an encoded representation of a picture in a video stream of multiple pictures, the decoder comprising:

205. Mulroy-H.264/AVC-Raveendran renders this obvious because Mulroy discloses a decoder configured to decode encoded representations. EX1004, ¶¶18-21. Mulroy further describes a remote receiver that functions as a decoder. EX1004, ¶¶30-36.

[8.2] at least one processor; and

206. Mulroy-H.264/AVC-Raveendran renders this element obvious because Raveendran teaches implementing a decoder with a processor. EX1006, ¶84.

[8.3] a memory storing instructions which, when executed by the at least one processor causes the decoder to:

207. Mulroy-H.264/AVC-Raveendran renders this element obvious because Raveendran teaches using memory to store instructions that are executed by a processor to cause an encoder/decoder to carry out its various operations. EX1006, ¶85.

[8.4] retrieve buffer description information defining a plurality of reference pictures, wherein the buffer description information is retrieved from the encoded representation of the picture;

208. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.2).

[8.5] determine, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture, wherein the decoder is configured to

209. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.3).

[8.6] i) retrieve, based on the buffer description information, a respective delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture, and

210. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.4).

[8.7] ii) calculate the reference picture identifier based on the picture identifier and the respective delta identifier; and

211. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.5).

[8.8] update a decoded picture buffer based on the reference picture identifier.

212. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.6).

2. Claim 9

[9.1] The decoder according to claim 8, wherein the memory stores further instructions which, when executed by the at least one processor causes the decoder to:

213. Mulroy-H.264/AVC-Raveendran renders this element obvious because Mulroy-H.264/AVC-Raveendran renders claim 8 obvious and because Raveendran discloses a “software module” or “instructions” stored in memory that are “executed by a processor” to perform a method. *Supra* Section X.C (claim 8); EX1006, ¶¶84-85, 17.

[9.2] retrieve the buffer description information defining multiple reference pictures from the encoded representation of the picture; and

214. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.2).

[9.3] determine, based on the buffer description information,

215. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 2.2).

[9.4] i) a picture identifier identifying a respective reference picture among the multiple reference pictures as the decoding reference for the picture and

216. Mulroy-H.264/AVC-Raveendran renders this element obvious because Mulroy discloses it. *Supra* Section IX.E (element 2.3).

[9.5] ii) a picture identifier identifying a respective reference picture among the multiple reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video stream.

217. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 2.4).

3. Claim 10

[10.1] The decoder according to claim 8, wherein the memory stores further instructions which, when executed by the at least one processor causes the decoder to further configured to:

218. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Sections X.C (claim 8), X.C (element 9.1).

[10.2] determine whether the reference picture identified by the reference picture identifier is one of missing or non-existing; and

219. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.2).

[10.3] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generate a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier.

220. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.3).

4. Claim 11

[11.1] The decoder according to claim 10, wherein generating the non-existing picture with parameters comprises at least one of:

221. Mulroy-H.264/AVC-Raveendran combination renders claim 10 obvious, as discussed above. *Supra* Section X.C (claim 10). For Mulroy-H.264/AVC-Raveendran to render this claim obvious, it need only disclose or render obvious one of the four options for generating the non-existing picture. *Supra* Section IX.E (element 4.1). Mulroy-H.264/AVC-Raveendran renders claim 11 obvious because Mulroy-H.264/AVC-Raveendran renders obvious at least elements 11.2 and 11.5, as noted below.

[11.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

222. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.2).

[11.3] setting an `additional_picture_id` of the non-existing picture to an `additional_picture_id` of the reference picture;

223. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 11.2 and 11.5. Element 11.3 is not addressed at this time.

[11.4] setting a `temporal_id` of the non-existing picture to a `temporal_id` of the reference picture; and

224. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 11.2 and 11.5. Element 11.4 is not addressed at this time.

[11.5] marking the non-existing picture with a non-existing mark and a used for reference mark.

225. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.5).

5. Claim 12

[12.1] A decoder configured to decode an encoded representation of a picture in a video stream of multiple pictures, the decoder comprising:

226. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (element 8.1).

[12.2] a data retriever circuit configured to retrieve buffer description information defining a plurality of reference pictures, wherein the buffer description information is retrieved from the encoded representation of the picture;

227. Mulroy-H.264/AVC-Raveendran renders this element obvious.
Mulroy-H.264/AVC renders obvious retrieving buffer description information

defining a plurality of reference pictures, wherein the buffer description information is retrieved from the encoded representation of the picture. *Supra* Section IX.E (element 1.2). Raveendran teaches implementing video coding methods, such as the teachings of Mulroy-H.264/AVC, in “circuits” such as an ASIC, FPGA, or other programmable logic device. EX1006, ¶84. A POSITA would have been motivated with a reasonable expectation of success to apply the teachings of Raveendran when implementing the obvious method of Mulroy-H.264/AVC, because Raveendran’s disclosure “is provided to enable any person skilled in the art to make or use the various embodiments.” EX1006, ¶86; *supra* Sections IX.C, X.B. A POSITA thus would have found it obvious to implement the known Mulroy-H.264/AVC method with an application specific integrated circuit (data retriever circuit) in order to “perform the function[]” as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

[12.3] a picture identifier determiner circuit configured to:

228. Mulroy-H.264/AVC-Raveendran renders this element obvious. Raveendran describes implementing H.264/AVC methods in hardware circuits, such as an application specific integrated circuit or another hardware component. EX1006, ¶84. Further, Mulroy-H.264/AVC renders obvious the steps of determining a picture identifier. A POSITA thus would have found it obvious to

implement the known Mulroy-H.264/AVC method of determining a picture identifier with an application specific integrated circuit (picture indicator determiner circuit) to “perform the function[]” as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

[12.4] determine, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture;

229. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.3).

[12.5] determine whether the reference picture identified by the reference picture identifier is one of missing or non-existing;

230. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.2).

[12.6] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generate a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier; and

231. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.3).

[12.7] a buffer manager circuit configured to update a decoded picture buffer based on the reference picture identifier.

232. Mulroy-H.264/AVC-Raveendran renders this element obvious.

Mulroy-H.264/AVC renders obvious updating a DPB based on the reference picture identifier. *Supra* Section IX.E (element 1.6).

233. Raveendran further teaches implementing video coding methods, such as the teachings of Mulroy-H.264/AVC, in a circuit. EX1006, ¶84. A POSITA thus would have found it obvious to implement the known Mulroy-H.264/AVC method of updating a decoded picture buffer with an application specific integrated circuit (buffer manager circuit) to “perform the function[]” as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

6. Claim 13

[13.1] The decoder according to claim 12, wherein generating the non-existing picture with parameters comprises at least one of:

234. Mulroy-H.264/AVC-Raveendran renders claim 12 obvious. *Supra* Section X.C (claim 12). To render this claim obvious, Mulroy-H.264/AVC-Raveendran need only disclose or render obvious one of the four options for generating the non-existing picture. *Supra* Section IX.E (element 4.1). As

discussed below, Mulroy-H.264/AVC-Raveendran renders obvious at least elements 13.2 and 13.5, and thus renders claim 13 obvious.

[13.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

235. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.2).

[13.3] setting an additional_picture_id of the non-existing picture to an additional_picture_id of the reference picture;

236. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 13.2 and 13.5, so element 13.3 is not addressed at this time.

[13.4] setting a temporal_id of the non-existing picture to a temporal_id of the reference picture; and

237. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 13.2 and 13.5, so element 13.4 is not addressed at this time.

[13.5] marking the non-existing picture with a non-existing mark and used for reference mark.

238. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.5).

7. **Claim 14**

[14.1] A receiver comprising:

239. To the extent the preamble is limiting, Mulroy-H.264/AVC-Raveendran renders it obvious because Mulroy discloses the operation of a receiver. EX1004, ¶30.

[14.2] an input section circuit configured to receive encoded representations of multiple pictures in a video stream;

240. Mulroy-H.264/AVC-Raveendran renders this element obvious. Mulroy describes a remote receiver that receives pictures in a video stream that are encoded as NALUs containing H.264/AVC slice data. EX1004, ¶30, FIG. 1, FIG. 4. Raveendran further teaches implementing video coding methods, such as the teachings of Mulroy-H.264/AVC, in a circuit. EX1006, ¶84. A POSITA thus would have found it obvious to implement the known Mulroy-H.264/AVC method of receiving encoded pictures in a video stream with an application specific integrated circuit (input section circuit) in order to “perform the function[]” as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

[14.3] a decoder circuit configured to decode the encoded representations of the multiple pictures; and

241. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (element 8.1). A POSITA would have found it obvious to implement

the known Mulroy-H.264/AVC decoder with an application specific integrated circuit (decoder circuit) to “perform the functions” of the decoder as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

[14.4] an output section circuit configured to output decoded pictures of the video stream, wherein the decoder circuit comprises:

242. Mulroy-H.264/AVC-Raveendran renders this element obvious. The ’618 patent describes that the “output section...is configured to output decoded pictures.” EX1001, 3:63-64. Mulroy describes this output section in its description that when a frame is decoded “the newly decoded frame is inserted into the buffer” for use. EX1004, ¶32. A POSITA would have understood that the newly decoded frame inserted into the buffer, in addition to being available as a reference picture, would also be output for eventual display, because the typical purpose of decoding the video stream is to display it. *E.g.*, EX1005, 1, 3. Mulroy also discloses output of the H.264 Decoder going to a “Renderer” (EX1004, FIG.1), and that the “video transmission system” depicted in FIG. 1 “performs the following functions” such as “H.264 real-time decode and display” (EX1004, ¶¶4, 8, 10).

243. A POSITA thus would have found it obvious to implement the known Mulroy-H.264/AVC method of outputting decoded pictures of the video stream with an application specific integrated circuit (output section circuit) to “perform

the function[]” as taught by Raveendran, because Raveendran provides an express teaching, suggestion, and/or motivation to do so. EX1006, ¶84.

[14.5] a data retriever circuit configured to retrieve buffer description information defining a plurality of reference pictures, wherein the buffer description information is retrieved from the encoded representation of a picture;

244. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (element 12.2).

[14.6] a picture identifier determiner circuit configured to

245. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (element 12.3).

[14.7] determine, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture, wherein the picture identifier determiner is configured to

246. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.3).

[14.8] i) retrieve, based on the buffer description information, a respective delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture, and

247. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.4).

[14.9] ii) calculate the reference picture identifier based on the picture identifier and the respective delta identifier; and

248. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 1.5).

[14.10] a buffer manager circuit configured to update a decoded picture buffer based on the reference picture identifier.

249. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (element 12.7).

8. Claim 15

[15.1] The receiver according to claim 14, wherein the decoder circuit is further configured to perform operations comprising:

250. Mulroy-H.264/AVC-Raveendran renders this element obvious, as discussed above. *Supra* Section X.C (claim 14).

[15.2] determining whether the reference picture identified by the reference picture identifier is one of missing or non-existing; and

251. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.2).

[15.3] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generating a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier.

252. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.3).

9. Claim 16

[16.1] The receiver according to claim 15, wherein generating the non-existing picture with parameters comprises at least one of:

253. Mulroy-H.264/AVC-Raveendran renders claim 15 obvious. *Supra* Section X.C (claim 15). Additionally, for Mulroy-H.264/AVC-Raveendran to render this claim obvious, it need only disclose or render obvious one of the four options for generating the non-existing picture. *Supra* Section IX.E (element 4.1). Mulroy-H.264/AVC-Raveendran renders this claim obvious because it renders obvious at least elements 16.2 and 16.5, as discussed in the following paragraphs.

[16.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

254. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.2).

[16.3] setting an `additional_picture_id` of the non-existing picture to an `additional_picture_id` of the reference picture;

255. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 16.2 and 16.5, so element 16.3 is not addressed at this time.

[16.4] setting a `temporal_id` of the non-existing picture to a `temporal_id` of the reference picture; and

256. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 16.2 and 16.5, so element [16.3] is not addressed at this time.

[16.5] marking the non-existing picture with a non-existing mark and used for reference mark.

257. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.5).

10. Claim 17

[17.1] A computer program product comprising a non-transitory computer-readable medium storing program instructions for decoding an encoded representation of a picture in a video stream of multiple pictures, the program instructions when executed by a processor of a node in an access network cause the node to perform operations comprising:

258. Mulroy-H.264/AVC-Raveendran renders this element obvious because Raveendran discloses “a computer readable medium” with instructions for executing a method for construction of a video sequence. EX1006, ¶17.

Raveendran also describes “access point 504x” (node) as “a device in an access network” (EX1006, ¶¶78, 81) a “receive [] data processor 558” within access point 504x that processes the video stream and provides decoded data (EX1006, ¶78), and non-transitory computer readable media, such as EEPROM, a hard disk, or a CD-ROM (EX1006, ¶85). A POSITA would have found it obvious to implement Mulroy-H.264/AVC in hardware and/or software described by Raveendran, because Raveendran provides express teachings, suggestions, and/or motivations to do so. *Supra* Section X.B.

[17.2] retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures;

259. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra*
Section IX.E (element 1.2).

[17.3] determining, based on the buffer description information, a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture, wherein determining the reference picture identifier comprises:

260. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra*
Section IX.E (element 1.3).

[17.4] (i) retrieving, based on the buffer description information, a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture; and

261. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra*
Section IX.E (element 1.4).

[17.5] (ii) calculating the reference picture identifier based on the picture identifier and the delta identifier; and

262. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra*
Section IX.E (element 1.5).

[17.6] updating a decoded picture buffer based on the reference picture identifier.

263. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra*
Section IX.E (element 1.6).

11. Claim 18

[18.1] The computer program product according to claim 17, wherein the instructions comprise further instructions that are executable by the processor of the node in the access network to perform operations comprising:

264. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section X.C (claim 17).

[18.2] determining whether the reference picture identified by the reference picture identifier is one of missing or non-existing; and

265. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.2).

[18.3] responsive to determining the reference picture identified by the reference picture identifier is one of missing or non-existing, generating a non-existing picture with parameters set to parameters of the reference picture identified by the reference picture identifier.

266. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 3.3).

12. Claim 19

[19.1] The computer program product according to claim 18, wherein generating the non-existing picture with parameters comprises at least one of:

267. Mulroy-H.264/AVC-Raveendran renders claim 18 obvious. *Supra* Section X.C (claim 18). To render this claim obvious, the combination need only disclose or render obvious one of the four options for generating the non-existing picture. *Supra* Section IX.E (element 4.1). As discussed in the following

paragraphs, Mulroy-H.264/AVC-Raveendran render obvious at least elements 19.2 and 19.5, and thus render claim 19 obvious.

[19.2] setting a picture order count (POC) of the non-existing picture to a POC of the reference picture;

268. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.2).

[19.3] setting an additional_picture_id of the non-existing picture to an additional_picture_id of the reference picture;

269. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 19.2 and 19.5. Element 19.3 is not addressed at this time.

[19.4] setting a temporal_id of the non-existing picture to a temporal_id of the reference picture; and

270. Mulroy-H.264/AVC-Raveendran renders obvious at least elements 19.2 and 19.5. Element 19.4 is not addressed at this time.

[19.5] marking the non-existing picture with a non-existing mark and used for reference mark.

271. Mulroy-H.264/AVC-Raveendran renders this element obvious. *Supra* Section IX.E (element 4.5).

XI. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

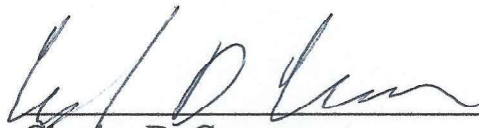
272. I am aware of no evidence that the method in the claims would result in a more efficient encoding and/or decoding process. Furthermore, the elements of

the '618 patent such as using absolute differences to identify reference pictures were previously disclosed by Mulroy. I am also unaware of any industry praise for the '618 patent or the process described in it. I also have no knowledge of any evidence that the claims of the '618 patent contributed to any successes, praise, or awards for the H.265/HEVC standard. The H.265/HEVC standard is the subject of thousands of patents, and I have no reason to think that the technology claimed in the '618 patent drove any of the success of the H.265/HEVC.

I, Charles D. Creusere, do hereby declare and state, that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: December 10, 2024

Respectfully submitted,



Charles D. Creusere

APPENDIX A

VITA

CHARLES D. CREUSERE

Klipsch School of Electrical & Computer Engineering

Mailing Address:

MSC 3-0

New Mexico State University

P.O. Box 30001

Las Cruces, NM 88003-8001

Phone: 575-646-3919

Fax: 575-646-1435

email: ccreuser@nmsu.edu

DISSERTATION TITLE

"Perfect Reconstruction Modulated Polyphase Filter Banks Using Reverse-Time Subfilters."

ACADEMIC TRAINING

- 1980-1985:** University of California at Davis, B.S. in Electrical and Computer Engineering.
- 1989-1990:** University of California at Santa Barbara, M.S. in Electrical and Computer Engineering.
- 1990-1993:** University of California at Santa Barbara, Ph.D. in Electrical and Computer Engineering.

PROFESSIONAL EXPERIENCE

2010-Present Holder of the Frank Carden Endowed Chair in Telemetry & Telecommunications and Full Professor. Current research interests include compressive sensing/sparse reconstruction for LIDAR and streaming sensor data as well as EEG brain analysis for audiovisual perceptual quality assessment and modeling.

October 2008 Selected for the International Foundation for Telemetry Endowed Professorship.

Jan. 2000-2008 Assistant/Associate professor in the Klipsch School of Electrical & Computer Engineering. My teaching areas include digital signal processing, image processing, pattern classification, and source coding (signal compression). I have done past research in areas of image, video, and audio compression as well as feature vector extraction for pattern classification. Currently, my research interests include distributed compression, polarimetric image processing for scene analysis, and nonstationary signal denoising.

1993-1999: Researcher & Team Leader, Naval Air Warfare Center, China Lake. My research efforts have focused on high speed image and video compression technologies which offer unique capabilities such as robustness to transmission errors and regional localization. My team (2 other people) and I have implemented a real-time (3 to 15 frames/second with 240x512 frames)

320C80-based system which uses a wavelet transform along with embedded coding techniques to compress a video input and stream it through the Internet via TCP/IP protocols. Our recent research focus has been to add more intelligence to the encoder so that the space-frequency information in the image that is most useful for image analysis is received with the highest fidelity. While most of my recent research has been in the area of embedded compression, I am still very much interested in other applications of time/space-frequency decompositions and of multirate digital signal processing concepts in general.

1999, Spring Quarter: Instructor at the University of California at Santa Barbara. Taught graduate class in Multirate Digital Signal Processing, ECE 258B.

1990-1993: Research Assistant, Department of Electrical and Computer Engineering, University of California, Santa Barbara. Worked under Prof. S.K. Mitra on subband coding and multirate filter bank theory. Also implemented real-time filter banks on a Motorola 56001 digital signal processor.

1992: Summer Employee, AT&T Bell labs, Murray Hill, NJ. Developed and simulated new methods of extremely low bit rate video coding for video telephone applications.

1985-1989: Design Engineer, Naval Weapons Center, China Lake. Designed, built, and tested the guidance electronics for the Laser Guided Training Round. This project included mixed analog and digital circuit design as well as the programming of an embedded DSP. Also developed software for an advanced video processor and studied ground target tracking.

FUNDED RESEARCH

- (2000) Office of Naval Research, *Compression of Digital Elevation Maps Using Non-linear Wavelets*, 2000-2003, \$94K
- (2001) Sandia National Labs, *Intelligent Compression for Remote Sensing*, 2001-2003, \$70K.
- (2002) National Science Foundation (Early Career Grant), *Efficient Audio Compression with Perceptually Embedded Scalability*, 2002-2007, \$350K.
- (2004) National Geospatial-Intelligence Agency, *Passive Polarimetric Imagery Classification Study*, 2004-2006, \$160K (joint with Dr. David Voelz).
- (2005) Los Alamos National Laboratories, *Signal Detection via Adapted Filter Banks and Geometric Dimensionality Reduction*, 2005-2006, \$15K (unburdened).
- (2006) Los Alamos National Laboratories, *Signal Detection via Adapted Filter Banks and Geometric Dimensionality Reduction*, 2006-2007, \$50K (unburdened).
- (2006) National Geospatial-Intelligence Agency, *Exploiting Polarization in Imaging Systems*, 2006-2009, \$304K (joint with Dr. David Voelz).
- (2006) Army Research Office, *Distributed Source Coding Using Bitstream-based Detection and Classification*, 2006-2009, \$326K.
- (2006) DARPA (Subcontract from LANL), *ADAM Project*, 2006-2007, \$104K (joint with Dr. Joe Lakey and Dr. Jaime Ramirez)

- (2009) NMSU IRG, *Perceptual audio quality evaluation by direct measurement of human brain responses*, 2009-2010, \$39K (joint with Dr. Jim Kroger, Psychology)
- (2011) National Science Foundation, *CIF:Medium:Assessment and modeling of temporal variation in perceived audio and video quality using direct brainwave measurement*, 2011-2015, \$917K (lead PI with Dr. Jim Kroger and Dr. Joerg Kliewer as co-PIs)
- (2011) NASA EPSCOR, *Proximity Operations for Near Earth Asteroid Exploration*, 2011-2014, \$750K (co-PI, with Dr. Eric Butcher (lead), others)
- (2012) National Geospatial Intelligence Agency (NGA), *Pulse Complexity Based LIDAR Scene Modeling for Sparse Reconstruction and Super-Resolution*, 2012-2013 (plus 3 1 year options), \$150K (\$75K/option year), co-PI Dr. David Voelz.
- (2018) Airforce Research Lab (AFRL), *Software Radio Design in LabView FPGA*, 2018-2019, \$140K.

PATENTS

- Patent titled "Parallel digital image compression system which exploits zerotree redundancies in wavelet coefficients," Patent Number 6,148,111.
- Patent titled "Efficient embedded image and video compression using lifted wavelets," Number: 6,466,698, granted October 15, 2002.

OTHER DISTINCTIONS

- Awarded the International Foundation for Telemetering Professorship, October 2008.
- Received an educational fellowship from the Department of Defense, 1989-1992.
- Certificate of Merit for the outstanding technical paper awarded at the AIAA Missile Sciences Conference for the paper "Automatic target recognition directed image compression," Nov. 1998.
- Patent (classified) "Notice of Allowability" titled, "Microcontroller-Based Laser Pulse Decoder," granted October 7, 1991.
- Associate editor for IEEE Trans. on Image Processing, 2002-2005, 2010-2014
- Associate editor for IEEE Trans. on Multimedia, 2008-2013.
- Guest Editor, "Issue on Advances in Hyperspectral Data Processing and Analysis", IEEE Journal of Selected Topics in Signal Processing, Vol. 5, Numbers: 5 & 6, August-September 2015,
- Co-general chair, IEEE Digital Signal Processing Workshop, August 2004, Taos, NM.
- Co-technical chair for the 2012 and 2014 Southwest Symposium on Image Analysis and Interpretation.
- Student Paper Contest Chair, 40th Asilomar Conf. on Signals, Systems, and Computers, October 2006.

- Organized special session entitled "Applications of Multirate DSP" at the 40th Asilomar Conf. on Signals, Systems, and Computers, October 2006.
- Member of technical program committees for the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), the IEEE International Conference on Image Processing (ICIP), and the IEEE Data Compression Conference (DCC).
- Senior Area Editor, IEEE Transactions on Image Processing, March 2016 to May 2022.
- Area Editor for IEEE Access, June 2022-Present.
- Deputy Editor-in-Chief, IEEE Transactions on Image processing, May 2023-present.

CONSULTING ACTIVITIES

- Video compression systems (technology consultant), Abba Tech, Albuquerque, NM, 2000.
- Expert witness in laser rangefinding technology, Asia Optical Inc. (through NY law firm of Osterlenk, Faber, Gerb & Soffen), Case: LTI versus Nikon/AOI, July 2001-2003. Case went to trial/ testified in court.
- Technical expert for defense; Case: Real-Time v. AT&T (byte.mobile), 2011-2012, case settled June 2012.
- Technical expert for defense; Case: Princeton Digital v. Dell, 2014-2015, case dismissed June 2015.
- Technical expert for defense; Noninfringement & IPR (6,597,812), Real-time v. SAP, 1/2016-6/2016.
- Technical expert for defense; IPRs (7,378,992 & 8,643,513), Real-time v. Riverbed, et al., 2/2016-2017
- Technical expert for defense; IPRs (7,415,530, 9,116,908, 7,161,506, & 9,054,728), Real-time v. Dell, et al., 2/2016-2017
- Technical expert for defense; Noninfringement, Real-time v. HP Enterprises, 4/2016-2018.
- Technical expert for defense; IPR (7,358,867, 7,161,506, & 9,054,728), Real-time v. Teradata 11/2016-2017
- Technical expert for defense; IPRs (7,415,530, 8,643,513 & 7,378,992), Real-time v. Veritas 12/2016-2017.
- Technical expert for defense; IPRs (7,075,917 & 6,304,612), UNILOC v. Apple 9/2018-2019.
- Technical expert for defense; IPR (7,558,730), Advanced Voice Recognition Systems v. Apple 7/2019-2020.
- Technical expert for defense; District Court; Noninfringement, Realtime Adaptive Systems v. YouTube/Google, 2018-2020.
- Technical expert for defense; ITC case; Nokia v. Lenovo, Oct. 2020-2021.
- Technical expert for defense; IPR (10,176,848), Maxell v. Apple, 2019-2021.

- Technical expert for plaintiff (rebutting invalidity); East Texas District Court; USAA v. PNC Bank, 2021-present. Deposed in Case 1 and Case 2; Attended trial in May 2022 for Case 1 and September 2022 for Case 2 but was not called to testify (USAA won on all counts, patents upheld as valid); I have also opined as an expert in 3 related IPRs (deposed in one of those cases, so far).
- Technical expert for defense; District Court; Gesture Tech Partners v. Apple/Lenovo&Motorola, 2021.

JOURNAL PUBLICATIONS

1. **C.D. Creusere and S.K. Mitra**, "A simple method for designing high-quality prototype filters for M-band pseudo-QMF banks," *IEEE Trans. on Signal Processing*, Vol. 43, No. 4, April 1995, pp. 1005-1007.
2. **C.D. Creusere and S.K. Mitra**, "Efficient audio coding using perfect reconstruction noncausal IIR filter banks," *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 2, March 1996, pp. 115-123.
3. **C.D. Creusere and S.K. Mitra**, "Image coding using wavelets based on perfect reconstruction IIR filter banks," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 5, Oct. 1996, pp. 447-458.
4. **C.D. Creusere**, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. on Image Processing*, Vol 6, No. 10, Oct. 1997, pp. 1436-1442.
5. **C.D. Creusere and A. Van Nevel**, "ATR-directed image and video compression," *Journal of Aircraft*, Vol. 36, No. 4, pp. 626-31, July-August 1999.
6. **C.D. Creusere**, "Fast embedded compression for video," *IEEE Trans. on Image Processing*, Vol. 8, No. 12, pp. 1811-16, December 1999.
7. **C.D. Creusere**, "Motion compensated video compression with reduced complexity encoding for remote transmission," *Signal Processing: Image Communications*, Vol. 16, pp. 627-42, April 2000.
8. **C.D. Creusere**, "Understanding perceptual distortion in MPEG scalable audio coding," *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 3, pp. 422-431, May 2005.
9. **L. E. Boucheron and C.D. Creusere**, "Lossless wavelet-based compression of digital elevation maps for fast and efficient search and retrieval," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 43, No. 5, pp. 1210-1214, May 2005.
10. **V. Thilak, D. Voelz, and C.D. Creusere**, "Polarization-based index of refraction and reflection angle estimation for remote sensing applications," *Applied Optics*, Vol. 46, Bo. 30, pp. 7427-7536, Oct. 2007.
11. **C.D. Creusere, K. Kallakuri, and R. Vanam**, "An Objective Metric of Human Subjective Audio Quality Optimized for a Wide Range of Audio Fidelities," *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, vol.16, no.1, pp.129-136, Jan. 2008
12. **S. Kandadai and C.D. Creusere**, "Scalable Audio Compression at Low Bitrates," *Audio, Speech, and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, vol.16, no.5, pp.969-979, July 2008
13. **S. Kandadai and C.D. Creusere**, "Reverse engineering and repartitioning vector quantizers using training set synthesis," *Signal Processing*, August 2008.
14. **V. Thilak, C.D. Creusere, and D. Voelz**, "Passive Polarimetric Imagery-Based Material Classification Robust to Illumination Source Position and Viewpoint," *Image Processing, IEEE Transactions on*, vol.20, no.1, pp.288-292, Jan. 2011.
15. **C.D. Creusere and J. Hardin**, "Assessing the Quality of Audio Containing Temporally Varying Distortions," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol.19, no.4, pp.711-720, May 2011.

16. **Castorena, J.; Creusere, C.D.**, "The Restricted Isometry Property for Banded Random Matrices," *Signal Processing, IEEE Transactions on* , vol.62, no.19, pp.5073-5084, Oct.1, 2014 doi: 10.1109/TSP.2014.2345350.
17. **Castorena, J.; Creusere, C.D.**, "Sampling of Time-Resolved Full-Waveform LIDAR Signals at Sub-Nyquist Rates," *Geoscience and Remote Sensing, IEEE Transactions on* , vol.53, no.7, pp.3791-3802, July 2015. doi: 10.1109/TGRS.2014.2383839.

REFEREED CONFERENCE PUBLICATIONS

1. **H. Babic, S.K. Mitra, C.D. Creusere, and A. Das**, "Perfect reconstruction recursive QMF banks for image subband coding," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1991, pp. 746-750.
2. **S.K. Mitra, C.D. Creusere, and H. Babic**, "A novel implementation of perfect reconstruction QMF banks using IIR filters," *Proc. IEEE Int. Symposium on Circuits and Systems*, San Diego, CA, May 1992, pp. 2312-2315.
3. **S.K. Mitra, C.D. Creusere, and H. Babic**, "Design of transmultiplexers using IIR filter banks," *Signal Processing VI: Theories and Applications*, Elsevier Science Publishers, 1992, pp. 223-226.
4. **C.D. Creusere and S.K. Mitra**, "Efficient image scrambling using polyphase filter banks," *Proc. International Conference on Image Processing*, Austin, TX, Nov. 1994, pp. 81-85.
5. **C.D. Creusere and G. Hewer**, "Wavelet-based nearest neighbor pattern classification using scale sequential matching," *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1994, pp. 1123-1127.
6. **C.D. Creusere**, "Embedded zerotree image coding using low complexity IIR filter banks," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995, pp. 2213-16.
7. **C.D. Creusere and Gary Hewer**, "Digital video compression for weapons control and bomb damage indication," *AGARD Conference Proceedings 576*, Chapter 16, Sept. 1995.
8. **C.D. Creusere**, "Image coding using parallel implementations of the embedded zerotree wavelet algorithm," *Proc. of the Digital Video Compression Conference (Algorithms and Technologies 1996)*, San Jose, CA, Jan. 28-Feb. 2, 1996, pp. 82-93.
9. **C.D. Creusere**, "A family of image compression algorithms which are robust to transmission errors," *Proceedings of the SPIE*, Vol. 2825, Denver, CO, August, 1996, pp. 890-900.
10. **C.D. Creusere**, "Perfect reconstruction time-varying IIR filter banks," *Conf. Rec. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1996, pp. 1319-23.
11. **C.D. Creusere**, "Out-of-loop motion compensation for reduced complexity video encoding," *Proc. of the Data Compression Conf.* (pp. 428) & *Data Compression Industry Workshop* (pp.28-37), March 1997, Snowbird, UT.
12. **C.D. Creusere**, "Periodic pan compensation for reduced complexity video compression," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol IV, pp. 2889-92, April 1997, Munich, Germany.
13. **C.D. Creusere**, "A new approach to global motion compensation which reduces video encoding complexity," *Proc. Int. Conf. on Image Processing*, Vol. III, pp. 634-7, October 1997, Santa Barbara, CA.
14. **C.D. Creusere**, "Spatially partitioned lossless image compression in an embedded framework," *Conf. Rec. 31st Asilomar Conf. on Signals, Systems, and Computers*, Nov. 1997, Pacific Grove, CA.
15. **C.D. Creusere**, "Adaptive embedding for reduced complexity image and video compression," *Proc.*

- of the *SPIE*, Vol 3309 (Visual Communications and Image Processing), pp. 48-57, Jan. 1998, San Jose, CA.
16. **C.D. Creusere**, "Successive coefficient refinement for embedded lossless image compression," *Proc. of the Data Compression Conf.*, pp. 539, March 1998, Snowbird, UT.
 17. **C.D. Creusere**, "Subband coding of speech and audio," *Proc. of the European Signal Processing Conf.* (invited paper), Sept. 1998, Isle of Rhodes, Greece.
 18. **C.D. Creusere**, "Fast embedded video compression using cache-based processing," *Proc. of the European Signal Processing Conf.*, Sept. 1998, Isle of Rhodes, Greece.
 19. **C.D. Creusere**, "Successive coefficient refinement for embedded lossless image compression," *Proc. Int. Conf. on Image Processing*, Oct. 1998, Chicago, IL.
 20. **C.D. Creusere** and A. Van Nevel, "Autonomous target recognition directed image compression," *Proc. of the AIAA*, Nov. 1998.
 21. **C.D. Creusere**, "Improved successive refinement for wavelet-based embedded image compression," *Proc. of the SPIE*, Denver, CO, July 1999.
 22. A. Van Nevel and **C.D. Creusere**, "Intelligent Bandwidth Compression," *Proc. of the SPIE*, Denver, CO, July 1999.
 23. **C.D. Creusere**, "Optimal refinement/significance map tradeoffs in SPIHT-based image compression," *Conf. Rec., 34th Asilomar Conf. on Signals, Systems, & Computers*, pp. 1026-30, Oct. 2000.
 24. **C.D. Creusere**, "Compression of digital elevation maps using nonlinear wavelets," *Proc. Int. Conf. on Image Processing*, pp. 824-7, October 2001.
 25. **C.D. Creusere and G. Dahman**, "Object detection and localization in compressed video," *Conf. Rec. 35th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2001, Pacific Grove, CA.
 26. **C.D. Creusere**, "An analysis of perceptual artifacts in MPEG scalable audio coding," *Proceedings of the Data Compression Conference*, pp. 152-161, April 2002, Snowbird, UT.
 27. **C.D. Creusere and N. Tolk**, "Combining wavelets and GLICBAWLS to achieve resolution-progressive lossless compression," *Proc. of the International Conference on Image Processing*, pp. III-229-32, October 2002.
 28. **L. Boucheron and C.D. Creusere**, "Compression of digital elevation maps for fast and efficient search and retrieval," *Proc. of the International Conference on Image Processing*, pp. 629-32, September 2003.
 29. **S. Kandadai and C.D. Creusere**, "An experimental study of object detection in the wavelet domain," *Conf. Rec. 37th Asilomar Conf. on Signals, Systems, and Computers*, pp. 1620-4, Nov. 2003, Pacific Grove, CA.
 30. **C.D. Creusere**, "Quantifying perceptual distortion in scalably compressed MPEG audio," *Conf. Rec. 37th Asilomar Conf. on Signals, Systems, and Computers*, pp. 265-9, Nov. 2003, Pacific Grove, CA.
 31. **C.D. Creusere and L. Zhou**, "Spatial object detection and classification in JPEG bitstreams," *Proceedings 11th Digital Signal Processing Workshop*, pp. 115-9, August 2004, Taos Ski Valley, NM.
 32. **L. Zhou and C.D. Creusere**, "Spatial object detection in JPEG bitstreams," *Proceedings European Conference on Signal Processing*, pp. 949-52, September 2004, Vienna, Austria.
 33. **V. Thilak and C.D. Creusere**, "Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter," *Proceedings European Conference on Signal Processing*, pp. 281-4, September 2004, Vienna, Austria.
 34. **S. Kandadai and C.D. Creusere**, "Reverse engineering vector quantizers by training set synthesis," *Proceedings European Conference on Signal Processing*, pp. 789-92, September 2004, Vienna, Austria.

35. **V. Thilak, J. Saini, D. G. Voelz, C. D. Creusere** , "Pattern recognition for passive polarimetric data using nonparametric classifiers," *Proc. of the SPIE Vol. 5888*, p. 337-344, Polarization Science and Remote Sensing II; Joseph A. Shaw, J. Scott Tyo; Eds.
36. **R. Vanam and C.D. Creusere**, "Evaluating low bitrate scalable audio quality using advanced version of PEAQ and energy equalization approach ," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. III, pp. 189-92, March 2005.
37. **S. Kandadai and C.D. Creusere**, "Reverse engineering vector quantizers for repartitioned signal spaces," *Proc. 39th Asilomar Conference on Signals, Systems, and Computers*, pp. 1208-12, Pacific Grove, CA, Nov. 2005.
38. **R. Vanam and C.D. Creusere**, "Scalable perceptual metric for evaluating audio quality," *Proc. 39th Asilomar Conference on Signals, Systems, and Computers*, pp. 319-23, Pacific Grove, CA, Nov. 2005.
39. **S. Kandadai and C.D. Creusere**, "Perceptually-weighted audio coding that scales to extremely low bitrates," *Proc. IEEE Data Compression Conference*, pp. 382-391, Snowbird, UT, March 2006.
40. **V. Thilak, D. Voelz, C. Creusere, S. Damarla**, "Estimating the refractive index and reflected zenith angle of a target using multiple polarization measurements," *Proc. SPIE Vol. 6240, 624004*, Polarization: Measurement, Analysis, and Remote Sensing VII; Dennis H. Goldstein, David B. Chenault; Eds., May 2006.
41. **V.M. Prasad and C.D. Creusere**, "Analyzing reversible lapped transformations using Reng probing," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 873-877, Oct. 2006.
42. **N. Balachandran and C.D. Creusere**, "Chirp classification using hidden Markov models," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 545-549, Oct. 2006.
43. **V. Thilak and C.D. Creusere**, "Estimating the complex index of refraction and view angle of an object using multiple polarization measurements," *Proc. 40th Asilomar Conference on Signals, Systems, and Computers*, pp. 1067-1071, Oct. 2006.
44. **V. Thilak, C.D. Creusere, and D.G. Voelz**, "Material classification using passive polarimetric imagery," *Proc. IEEE Int. Conf. on Image Proc.*, Vol. 4, pp. IV-121-124, Sept., 2007.
45. **A. Pamba, V. Thilak, D. G. Voelz, C. D. Creusere** "Estimation of incidence and reflection angles from passive polarimetric imagery: extension to out-of-plane scattering," *Proc. SPIE Vol 6682, 66820O*, Polarization Science and Remote Sensing III, Joseph A. Shaw; J. Scott Tyo, Editors, September 2007.
46. **V. Thilak, D. G. Voelz, C. D. Creusere**, "Image segmentation from multi-look passive polarimetric imagery, *Proc. SPIE 6682, 668206*, Polarization Science and Remote Sensing III, Joseph A. Shaw; J. Scott Tyo, Editors, October 2007.
47. **S. Kandadai and C.D. Creusere**, "Optimal Bit Layering for Scalable Audio Compression Using Objective Audio Quality Metrics," *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on* , pp.560-564, 4-7 Nov. 2007.
48. **C.D. Creusere and I. Mecimore**, "Bitstream-based overlap analysis for multi-view distributed video coding," *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 93-96, March 2008.
49. **V. Thilak, C.D. Creusere, and D.G. Voelz**, "Passive Polarimetric Imagery Based Material Classification For Remote Sensing Applications," *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 153-156 March 2008.
50. **V. Thilak, Q. Wang, D. G. Voelz, C. D. Creusere**, "Estimation of target geometry from Mueller matrix imagery," *Proc. SPIE 6972*, March 2008.
51. **C.D. Creusere and I. Mecimore**, "Bitstream-based correlation detector for multi-view distributed video coding applications," *Proc. Int. Conf. on Accoustics, Speech, and Signal Processing*, pp. 1001-1004, April 2008, Las Vegas, NV.

52. **S. Kandadai, J. Hardin, and C.D. Creusere**, "Audio quality assessment using the mean structural similarity measure," *Proc. Int. Conf. on Accoustics, Speech, and Signal Processing*, pp. 221-224, April 2008, Las Vegas, NV.
53. **S. Matta and C.D. Creusere**, "Efficient correlation extraction for distributed audio coding," *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, vol., no., pp.1272-1276, 26-29 Oct. 2008.
54. **J.C. Hardin and C.D. Creusere**, "Objective analysis of temporally varying audio quality metrics," *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, vol., no., pp.1245-1249, 26-29 Oct. 2008.
55. **S. Matta and C.D. Creusere**, "Distributed audio coding with efficient source correlation extraction," *Proceedings 13th Digital Signal Processing Workshop*, pp. 16-20, January 2009, Marco Island, FL.
56. **I. Mecimore and C.D. Creusere**, "Unsupervised bitstream based segmentation of images," *Proceedings 13th Digital Signal Processing Workshop*, pp. 642-647, January 2009, Marco Island, FL.
57. **Q. Wang, C.D. Creusere, V. Thilak, and D.G. Voeltz**, "Active polarimetric imaging for estimation of scene geometry," *Proceedings 13th Digital Signal Processing Workshop*, pp. 659-663, January 2009, Marco Island, FL.
58. **J.C. Hardin and C.D. Creusere**, "A temporally varying objective audio quality metric," *Proceedings 13th Digital Signal Processing Workshop*, pp. 21-25, January 2009, Marco Island, FL.
59. **I. Mecimore, W. Fahrenkrog, and C.D. Creusere**, "On bitstream based edge detection techniques," *ITEA Conference*, January 2009, El Paso, TX (best graduate student paper award).
60. **Mecimore, Ivan; Creusere, Charles D.**, "Low complexity multi-view distributed video coding based on JPEG," *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, vol., no., pp.165-168, 23-25 May 2010.
61. **Creusere, C.D.; Mehta, K.; Voelz, D.G.**, "Model-based estimation of surface geometry using passive polarimetric imaging," *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, vol., no., pp.4557-4560, 25-30 July 2010.
62. **Castorena, J.; Creusere, C.D.; Voelz, D.**, "Modeling lidar scene sparsity using compressive sensing," *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, vol., no., pp.2186-2189, 25-30 July 2010.
63. **Castorena, J.; Creusere, C.D.; Voelz, D.**, "Using finite moment rate of innovation for LIDAR waveform complexity estimation," *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, vol., no., pp.608-612, 7-10 Nov. 2010.
64. **Creusere, C.D.; Siddenki, S.; Hardin, J.; Kroger, J.**, "Early investigations into subjective audio quality assessment using brainwave responses," *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, vol., no., pp., Nov. 2011.
65. **Castorena, J.E.; Creusere, C.**, "Remote-sensed LIDAR using random sampling and sparse reconstruction," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2011.
66. **Davis, P; Creusere, C.**, "Quantifying the gains of compressive sensing for telemetering application," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2011.
67. **Creusere, C.D.; Kroger, J.; Siddenki, S.R.; Davis, P.; Hardin, J.**, "Assessment of subjective audio quality from EEG brain responses using time-space-frequency analysis," *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, vol., no., pp.2704-2708, 27-31 Aug. 2012.
68. **Castorena, J.; Creusere, C.D.**, "Compressive sampling of LIDAR: Full-waveforms as signals of finite rate of innovation," *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th*

European , vol., no., pp.984-988, 27-31 Aug. 2012.

69. **Castorena, J.; Creusere, C.D.;** "Random impulsive scan for LIDAR sampling," *Proc. IEEE Int. Conf. on Image Proc.*, October 2012.
70. **Creusere, C., Nelson, E., Critz, T., Butcher, E.;** "Analysis of communication interconnectedness in the proximity of near-earth asteroids," *Proc. International Telemetering Conference*, San Diego, CA, October 2012.
71. **Castorena, J.E.; Creusere, C.;** "Remote-sensed LIDAR using random impulsive scans," *Proc. International Telemetering Conference*, San Diego, CA, October 2012.
72. **Castorena, J.; Creusere, C.D.;** "Sub-spot localization for spatial super-resolved LIDAR," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* , vol., no., pp.2227,2231, 26-31 May 2013.
73. **Charles D. Creusere ; Juan Castorena,** "A unified framework for 3rd generation lidar pulse processing based on finite rate of innovations," *Proc. SPIE 8858, Wavelets and Sparsity XV*, 88580T (September 26, 2013); doi:10.1117/12.2022447.
74. **Nelson, E., Creusere, C., Critz, T., Butcher, E.;** "Analysis of communication rates in the proximity of near-earth asteroids," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2013.
75. **Castorena, J.E.; Creusere, C.;** "Full-waveform LIDAR recovery at sub-Nyquist rates," *Proc. International Telemetering Conference*, Las Vegas, NV, October 2013.
76. **Davis, P.; Creusere, C.D.; Kroger, J.;** "EEG and the human perception of video quality: Impact of channel selection on discrimination," *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE* , vol., no., pp.9,12, 3-5 Dec. 2013, doi: 10.1109/GlobalSIP.2013.6736798.
77. **Creusere, C.D.; McRae, N.; Davis, P.;** "Sample-based cross-frequency coupling analysis with CFAR detection," *Signals, Systems and Computers, 2014 48th Asilomar Conference on* , vol., no., pp.179,183, 2-5 Nov. 2014; doi: 10.1109/ACSSC.2014.7094423.
78. **Davis, P.; Creusere, C.D.; Kroger, J.;** "Classification of human viewers using high-resolution EEG with SVM," *Signals, Systems and Computers, 2014 48th Asilomar Conference on* , vol., no., pp.184,188, 2-5 Nov. 2014; doi: 10.1109/ACSSC.2014.7094424.
79. **Davis, P.; Creusere, C. D.; Tang, W.;** "ASIC implementation of the cross frequency coupling algorithm for EEG signal processing," *2014 14th International Symposium on Integrated Circuits (ISIC)*, pp.248-251, Dec. 2014 doi: 10.1109/ISICIR.2014.7029468
80. **Nelson, Evan; Creusere, Charles D.; Butcher, Eric,** "Determining position around an asteroid using communication relays and trilateration," *Aerospace Conference, 2015 IEEE* , vol., no., pp.1,6, 7-14 March 2015; doi: 10.1109/AERO.2015.7118955.
81. **Davis, Philip; Creusere, Charles D.; Tang, Wei,** "Window length effect on cross frequency coupling in an EEG processing circuit," in *Circuits and Systems (MWSCAS), 2015 IEEE 58th International Midwest Symposium on* , vol., no., pp.1-4, 2-5 Aug. 2015 doi: 10.1109/MWSCAS.2015.7282125.
82. **Davis, Philip; Creusere, Charles D.; Kroger, Jim,** "Subject Identification Based on EEG Response to Video Stimuli," *Proceedings International Conference on Image Processing, 2015 IEEE*, September 2015.
83. **Newtson, Kathy; Creusere, Charles D.;** "Histogram oriented gradients and map seeking circuits pattern recognition with compressed imagery," *Proc. Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, NM, March 2016.
84. **Davis, Philip; Creusere, Charles D.; Kroger, Jim,** " Assessing Cross Frequency Coupling in EEG Collected from Subjects Viewing Video using a Modified Metric ," *Proc. Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, NM, March 2016.
85. **Davis, Philip; Creusere, Charles D.; Kroger, Jim,** "The Effect of Perceptual Video Quality on EEG Power Distribution," *Proceedings International Conference on Image Processing, 2016 IEEE*,

Phoenix, AZ, September 2016.

86. **Creusere, Charles D.; Castorena, J.; Dragulin, I.; Voelz, D.**, "Quantifying the Accuracy of FRI based LIDAR Waveform Analysis," *Proceedings International Conference on Image Processing, 2016 IEEE*, Phoenix, AZ, September 2016.

87. **Kathy A. Newton, Charles D. Creusere**, "Compressed imagery detection rate through map seeking circuit, and histogram of oriented gradient pattern recognition", Proc. SPIE 10203, Pattern Recognition and Tracking XXVIII, 102030F (1 May 2017); doi:10.1117/12.2262919; <https://doi.org/10.1117/12.2262919>

88. **Erandi Wijerathna, Charles D. Creusere, David Voelz, Juan Castorena**, "Polarimetric LIDAR with FRI sampling for target characterization", Proc. SPIE 10407, Polarization Science and Remote Sensing VIII, 104070R (7 September 2017); doi: 10.1117/12.2272587; <https://doi.org/10.1117/12.2272587>

89. **Andrew J. Phillips, Charles D. Creusere**, "The effects of lossy EEG compression on ERP analysis," *International Telemetering Conference Proceedings*, November 2018, <http://hdl.handle.net/10150/631654>.

90. **Andrew J. Phillips, Charles D. Creusere**, "The effects of lossy frequency-domain EEG compression on cross-frequency coupling analysis," *International Telemetering Conference Proceedings*, October 2019.

91. **Sergio Lara, Charles D. Creusere**, "Software Defined Radio for Carrier and Symbol Timing Extraction," *International Telemetering Conference Proceedings*, October 2021.

92. **Sergio Lara, Charles D. Creusere**, "LabView for Software Defined Radio Development," *International Telemetering Conference Proceedings*, October 2021.

93. **Ayatelrahman Elsayed, Charles D. Creusere**, "The Feedback Dynamics of Brain-Computer Interfaces in a Distributed Processing Environment," *Proceedings of HICSS-55*, January 2022; <http://hdl.handle.net/10125/79821>.

94. **Joshua Tellez, Charles D. Creusere**, "Evaluation of JPEG-2000 Image Compression as Applied to Electroencephalography," *International Telemetering Conference Proceedings*, Glendale, AZ, October 2022, <https://repository.arizona.edu/handle/10150/666964>.

PRESENTATIONS & OTHER PUBLICATIONS

1. **C.D. Creusere**, "Generic Tracking Laboratory Study," *NWC Technical Paper 7018*, Nov. 1989, Naval Weapons Center, China Lake, CA.
2. **C.D. Creusere**, "Robust image coding using the embedded zerotree wavelet algorithm," presented at the Digital Compression Conference, Snowbird, UT, April 1996.
3. **C.D. Creusere**, "Creating a compressed and embedded image representation which is robust to transmission errors," presented at the Office of Naval Research Workshop on Error Resilient Compression, San Diego, CA, February, 1996.
4. **C.D. Creusere**, "Data Compression Project Final Report," NAWCWD Technical Paper, TP 8442.
5. **C.D. Creusere**: "Applications of multirate digital signal processing to communications systems," NATO/RTA lecture series 216 on *Applications of Mathematical Signal Processing Techniques to Mission Systems*, presented in Paris, France; Cologne, Germany; and Monterey, CA, November 1999.
6. **C.D. Creusere**, "Object detection and recognition in compressed video," invited presentation

and white paper for the Motion Imagery Workshop, sponsored by the National Imagery and Mapping Agency (NIMA) and the National Science Foundation (NSF).

7. **C.D. Creusere**, "Application specific compression: digital elevation map browsing, video surveillance, and scalable audio," invited seminar at the University of Arizona, Tucson, AZ, March 2003.

8. **C.D. Creusere**, "Compressing Digital Elevation Maps for Efficient Search and Retrieval," presented at the Office of Naval Research Workshop on Image Processing, Minneapolis, MN, May 2003.