

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ELEMENT TV COMPANY, LP, AND
ELEMENT TELEVISION COMPANY, LLC,

Petitioners

v.

NOKIA CORPORATION AND NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2025-01068

U.S. Patent No. 8,050,321

**DECLARATION OF DR. ELI SABER IN SUPPORT OF PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 8,050,321**

TABLE OF CONTENTS

I.	INTRODUCTION AND COMPENSATION	1
II.	BACKGROUND AND QUALIFICATIONS	2
III.	LEVEL OF ORDINARY SKILL IN THE ART	7
IV.	MATERIALS REVIEWED	9
V.	LEGAL STANDARDS	10
A.	Priority Date	11
B.	Claim Interpretation	12
C.	Prior Art	12
D.	Obviousness	12
VI.	SUMMARY OF OPINIONS	16
VII.	OVERVIEW OF THE '321 PATENT.....	16
A.	Technology Overview.....	16
B.	The Challenged Claims.....	20
C.	'321 Patent Prosecution History	25
VIII.	CLAIM CONSTRUCTION	26
IX.	DETAILED EXPLANATION OF GROUNDS.....	27
A.	Ground 1: MPEG-1 Renders Obvious Claims 8, 10, and 11.....	27
1.	MPEG-1	27
2.	Claim 8.....	36

3.	Claim 10.....	46
4.	Claim 11.....	49
B.	Ground 2: Kim Renders Obvious Claims 8, 10, and 11	52
1.	Kim	52
2.	Claim 8.....	56
3.	Claim 10.....	74
4.	Claim 11.....	75
C.	Ground 3: MPEG-1 and Yagasaki Render Obvious Claim 9	78
1.	Yagasaki	78
2.	Claim 9: <i>A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.</i>	80
3.	Motivation to Combine MPEG-1 and Yagasaki	83
X.	CONCLUSION.....	85

I, **Eli Saber**, hereby declare as follows:

I. INTRODUCTION AND COMPENSATION

1. I have been retained by Amazon.com, Inc., as an independent expert consultant in this inter partes review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”).

2. I have been asked by Counsel for Petitioner to consider whether certain references teach or suggest the features cited in claims 8-11 of U.S. Patent No. 8,050,321 (“the ’321 patent”) (EX1001). My opinions and the bases for those opinions are set forth below.

3. I am being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I incur during the course of this work. My compensation in no way depends on the outcome of this matter. I have no other financial interest in this proceeding.

4. My analysis here is based on my years of education, research and experience, as well as my investigation and study of relevant materials, including those cited herein. I may rely upon these materials, my knowledge and experience, and/or additional materials to rebut arguments raised by the Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

5. My analysis of the materials in this proceeding is ongoing, and I will continue to review any new material as it is provided. This Declaration represents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

II. BACKGROUND AND QUALIFICATIONS

6. I am over the age of 18 and am competent to write this declaration. I have personal knowledge, or have developed knowledge of these technologies, based upon education, training, or experience, of the matters set forth herein.

7. My qualifications are set forth in my curriculum vitae, a copy of which is filed as Exhibit EX-1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I am a Professor in the Department of Electrical and Microelectronic Engineering (EME) and the Chester F. Carlson Center for Imaging Science at the Rochester Institute of Technology (RIT) in Rochester, New York. I also serve as the Director of the Image, Video and Computer Vision Laboratory. I joined RIT's full-time faculty in 2004. I have over thirty-four (34) years of industry and academic experience, thirty (30) of which are in the fields of Computer Vision, Digital Image/Video Processing, Pattern Recognition, and Machine Learning.

9. I received a Ph.D. in Electrical Engineering from the University of

Rochester in 1996. My concentration was in Signal/Image/Video Processing, Pattern Recognition, and Computer Vision. I received a Master of Science Degree in Electrical Engineering from the University of Rochester in 1992 in the same concentration as listed above, and I received a Bachelor of Science Degree in Electrical and Computer Engineering, Summa Cum Laude, from the State University of New York at Buffalo in 1988.

10. Before becoming a full-time professor, I worked for Xerox Corporation for 16 years, from 1988 to 2004. During my years at Xerox, I was responsible for delivering color management, image processing innovations, architectures, and algorithms; xerographic sub-systems for a variety of color products; and control systems for toner production facilities. One of several roles I held at Xerox was Advanced Development Scientist and Manager. In that capacity, I established the Advanced Design Laboratory—an Imaging/Xerographics lab—and provided technical and managerial leadership for the Electrical, Imaging and Xerographics Department. In another role, as Product Development Scientist and Manager, I led the research and development of image quality metrics for various product platforms. I also led the Image Science, Analysis, and Evaluation area, with 12 to 15 direct reports and a budget of approximately \$2 million.

11. From 1997 until 2004, I was an adjunct faculty member at the Electrical Engineering Department of the Rochester Institute of Technology and at the

Electrical & Computer Engineering Department of the University of Rochester. I was responsible for teaching undergraduate and graduate coursework such as Digital Signal Processing, Digital Image Processing, Digital Video Processing, Pattern Recognition, Detection and Estimation Theory and Communication Systems; and performing research in multimedia applications, pattern recognition, image understanding and color engineering.

12. Since joining RIT full time, I have been responsible for teaching several undergraduate and graduate courses in Digital Signal Processing, Digital Image Processing, Digital Video Processing, Advanced Neural Networks and Deep Learning, Pattern Recognition, Engineering Analysis, Random Signal & Noise, Advanced Engineering Mathematics, Matrix Methods, Communications, Modern Control Theory, and Linear Systems.

13. I proposed, developed, and introduced the Digital Video Processing and the Advanced Neural Network and Deep Learning courses; and I reestablished the Pattern Recognition course at the Electrical Engineering Department at RIT. These graduate courses are instrumental in providing MS/PhD students with the proper foundation and knowledge in all aspects of the subject areas in order to perform in-depth research in the field and continue to advance the state of the art.

14. The Digital Video Processing course is designed to provide a thorough and comprehensive coverage of the field. Topics covered include the mathematical

representation of digital video including motion and time-varying image formation models, spatio-temporal sampling/reconstruction, sampling structure conversions, motion estimation using optical flow, block-based, pel-recursive, Bayesian and point correspondence type methodologies, segmentation, motion tracking, filtering, restoration, super-resolution and compression.

15. The Pattern Recognition course, on the other hand, covers pertinent concepts in Bayes decision theory, linear and nonlinear classifiers, artificial neural networks, feature selection/generation/extraction and various types of clustering algorithms and techniques.

16. The “Advanced Neural Networks and Deep Learning: Theory, Mathematics and Algorithms” course is intended to provide an in-depth comprehensive study of the theory, mathematics and algorithms for advanced neural networks and their usage in deep learning. It is primarily targeted for upper graduate students that are planning to pursue advanced studies/research in the field. The course begins with an in-depth study of the mathematical methodologies for neural networks and optimization (zero-, first- and second-order) techniques. It then progresses to provide a thorough study of shallow and deep neural networks architectures and algorithms, activation functions, regression, two- and multi-class classification, unsupervised learning, feature engineering/selection, backpropagation along with applications to a variety of engineering problems.

17. I am currently working to develop a follow-on course (Part 2) to the Advanced Neural Networks and Deep Learning course described above. This course is intended to cover convolutional and recurrent style neural networks, principles of feature engineering, kernel methods, tree-based learners, attention mechanisms/transformers, reinforcement learning and generative adversarial networks. This course is being targeted for the first time in Fall 2026.

18. I am also currently working to revamp the graduate course titled: “Image and Video Compression.” The updated version of this course will start by providing a brief history of video coding and video coding standards and then progress to discussing video basics, formats and fundamentals of video compression. It will then progress to cover JPEG, JPEG 2000, H.261, MPEG-1, MPEG-2, H.263, MPEG-4 visual, H.264 and H.265 and if time permits, AV-1 and H.266. The course is intended to debut in Fall 2027.

19. My current research focuses on machine learning for object tracking/remote sensing applications; image/video processing for multimedia, military & biomedical applications; and computer vision and three-dimensional scene reconstruction. As a principal investigator (PI) or co-principal investigator, I have acquired almost \$6 million in research funding since joining RIT and have managed multiple government grants from the Department of Defense as well as several corporate grants from Hewlett-Packard, Lenel, and Data Physics. I am

currently managing (as PI) a multiyear government grant from the Department of Defense titled “Target Detection/Tracking and Activity Recognition from Multimodal Data.” In this project, we propose to develop operational target detection/tracking techniques and activity recognition/understanding methods that can leverage multimodal data in a fusion framework using deep learning algorithms while providing accurate and near real-time performance.

20. In 2012, I was awarded the prestigious Trustees Scholarship Award, the highest award at RIT with regard to research recognition.

21. I am the author or co-author of thirty-eight (38) peer-reviewed journal publications.

22. I have also authored or co-authored 102 conference and workshop publications and a book entitled Advanced Linear Algebra for Engineers with MATLAB, published by CRC Press in February 2009, and am a named inventor on multiple U.S. and foreign patents.

23. I am a senior member of the Institute of Electrical and Electronic Engineers and a member of the IEEE Signal Processing Society, the Electrical Engineering Honor Society, Eta Kappa Nu.

III. LEVEL OF ORDINARY SKILL IN THE ART

24. I understand that a person of skill in the art (“POSITA”) is a hypothetical person who is presumed to have known the relevant art at the time of

the invention. To determine the level of skill held by a person of ordinary skill in the art, I understand that a number of factors may be considered, including the following:

- The type of problems encountered in the art;
- Prior art solutions to those problems;
- Rapidity with which innovations are made;
- Sophistication of the technology; and
- Educational level of active workers in the field.

25. I understand that not all of the above factors will be present in every case and that one or more factors may predominate.

26. In view of the above factors, it is my opinion that a POSITA at the time of the alleged invention of the '321 patent would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study, such as physics and (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa.

27. As of the January 23, 2002, I met and exceeded the qualifications of a POSITA. My opinions are offered from this perspective, even if they do not specifically refer to the perspective of a POSITA.

IV. MATERIALS REVIEWED

28. My opinions contained in this Declaration are based on the documents I reviewed, my professional judgment, as well as my education, experience, and knowledge.

29. In forming my opinions expressed in this Declaration, I have reviewed and considered the following materials:

Exhibit	Reference
1001	U.S. Patent No. 8,050,321
1002	U.S. Patent No. 8,050,321 File History
1003	Declaration of Dr. Eli Saber
1004	Curriculum Vitae of Dr. Eli Saber
1005	ISO/IEC 11172-2 (1st ed.) Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video ("MPEG-1")
1006	U.S. Patent No. 6,912,351 ("Kim")
1007	CL480PC MPEG-1 System Decoder Specification, <i>available at</i> https://web.archive.org/web/19970128081644/http://www.c-cube.com/prdctlst/cl480pc.pdf .
1008	U.S. Patent No. 7,263,270
1009	Ruby B. Lee et al., <i>Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7100LC Processors</i> , Hewlett-Packard J., Apr. 1995, <i>available at</i> https://archive.org/details/Hewlett-Packard_Journal_Vol._46_No._2_1995-04/Hewlett-Packard
1010	Ketan Patel, et al., <i>Performance of a Software MPEG Video Decoder</i> , Multimedia '93: Proceedings of the First ACM Int'l Conf. on Multimedia, August 2-6, 1993
1011	Complaint, <i>Element Television Co. v. Nokia Corp.</i> , No. 24-cv-04269 (D. Mn. Nov. 25, 2024), ECF No. 1
1012	RESERVED
1013	Declaration of June Ann Munford
1014	U.S. Patent No. 7,894,521

Exhibit	Reference
1015	U.S. Patent No. 5,786,858 (“Yagasaki”)

30. I have also considered any materials cited in this Declaration, even if not included in the list above.

31. My opinions have also been guided by my appreciation of how a POSITA would have understood the claims and the specification of the ’321 patent at the time of the alleged invention, which I have been asked to initially consider to be January 23, 2002 (the filing date of Finnish Application No. 20020127). My opinions reflect how one of ordinary skill in the art would have understood the ’321 patent, the prior art to the patent, and the state of the art at the time of the alleged invention.

V. LEGAL STANDARDS

32. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a POSITA.

33. For the purposes of this Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Priority Date

34. I understand that, to gain benefit of the filing date of an earlier patent application, each application in the chain leading back to the earlier application must comply with the statutory written description requirement.

35. I understand that, to satisfy the written description requirement, a patent's specification must reasonably convey to a POSITA that the inventor had possession of the claimed subject matter when the patent application was filed. I understand that, to show "possession" of the invention, the disclosure must describe the invention, including all of its claimed limitations. It is not enough that a claimed invention is an obvious variant of that which is disclosed in the specification.

36. I further understand that entitlement to a filing date extends only to subject matter that is disclosed and not to subject matter that is not disclosed but would be obvious over what is expressly disclosed. Thus, a prior application itself must describe an invention, and do so in sufficient detail that a POSITA can clearly conclude that the inventor invented the claimed invention as of the filing date sought.

37. I also understand that a patent's priority date is a legal determination based on underlying factual issues. For that reason, I am not rendering a legal opinion on the ultimate legal question of priority date. Rather, my testimony addresses the underlying facts and factual analysis that would support a legal determination of the priority date.

B. Claim Interpretation

38. I understand from Counsel that during an IPR proceeding, claims are to be construed in light of the specification as would be read by a POSITA at the time the application was filed. I understand that claim terms are given their ordinary and customary meaning as would be understood by a POSITA in the context of the entire disclosure. A claim term, however, will not receive its ordinary meaning if the patentee acted as his or her own lexicographer and clearly set forth a definition of the claim term in the specification. In that case, the claim term will receive the definition set forth in the patent.

C. Prior Art

39. I understand from Counsel that certain patents and printed publications can be considered prior art in an IPR proceeding. I understand that prior art must predate the effective filing date of the claims of the patent-at-issue.

D. Obviousness

40. I understand that a claim is unpatentable if the claim limitations, as arranged in a claim, would have been obvious to a person of ordinary skill in the art as of the date of invention. For example, a combination of familiar elements according to known methods is likely to be obvious if the combination does no more than produce predictable results.

41. In determining obviousness, I understand that the analysis may take into account multiple prior art references, as well as the common knowledge possessed

by a POSITA (as discussed above). I further understand that the POSITA is not an automaton but one having common sense and ordinary creativity. I also understand that a claim is not obvious simply because a combination of prior art references discloses each claim element collectively.

42. In determining obviousness, I understand that one should consider the following.

- The scope and content of the prior art;
- The differences between the claimed invention and the prior art;
- The level of ordinary skill in the pertinent art; and
- Secondary considerations of obviousness.

43. I understand that a POSITA is assumed to have knowledge of all prior art references. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be rigidly applied, but that the test can be important to avoiding such hindsight.

44. I understand that a POSITA can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine the references may be implicit in the prior art, and there is no requirement that there be an actual

or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a POSITA would employ to combine the known elements in the prior art in the manner claimed by the patent-at-issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. This does not mean, however, that a POSITA, for purposes of the obviousness inquiry, does not have recourse to common sense.

45. I understand that the following are examples of approaches and rationales that may be considered in determining whether a prior art reference could have been combined with other prior art or with other information within the knowledge of a POSITA:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known techniques to improve similar devices, methods, or products in the same way;
- Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;

- That it was “obvious to try”—i.e., that it was one choice from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and
- Some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art reference or to combine multiple prior art reference teachings to arrive at the claimed invention.

46. I also understand that obviousness is a legal conclusion based on the underlying factual issues of the scope and content of the prior art, the differences between the claimed invention and the prior art, the level of ordinary skill in the art, and any objective indicia of non-obviousness. For that reason, I am not rendering a legal opinion on the ultimate legal question of obviousness. Rather, my testimony addresses the underlying facts and factual analysis that would support a legal conclusion of obviousness, and when I use the term obvious, I am referring to the perspective of a POSITA at the time of invention.

VI. SUMMARY OF OPINIONS

47. For the reasons I discuss below, it is my opinion that claims 8-11 of the '321 patent would have been obvious to a POSITA in view of the prior art, and if asked to do so, I can identify disclosure in the references that corresponds to each limitation of each Challenged Claim. In the interest of keeping my declaration focused, however, I only specifically address claim elements/limitations or key points that may not be readily apparent on the face of the prior art but would nevertheless have been understood by a POSITA reading that art. For avoidance of doubt, however, I have reviewed and I agree with the full limitation-by-limitation analyses presented in the Petition.

VII. OVERVIEW OF THE '321 PATENT

A. Technology Overview

48. The '321 patent is generally directed to encoding and decoding video sequences with independent sequences of image frames.¹ (EX1001, Abstract.) A

¹ The Challenged Claims use the term “image frames,” so I will do likewise here, and the '321 patent uses “image frame,” “frame,” “video frame,” and “picture” interchangeably. (See EX1001, 10:25-30 (“If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image

key concept discussed in the '321 patent is the grouping of these image frames. But the concept of grouping image frames into displayable sequences has been well-known since at least 1993, when International Standard ISO/IEC 11172-2 ("MPEG-1") was released. That standard used the term "group of pictures" or "GOP" to refer to a group of image frames. (EX1005, 23² (defining group of pictures), 71 (explaining that "[a] picture corresponds to a single frame of motion video").) The '321 patent uses these terms in the same way. (*See, e.g.*, EX1001, 2:34-40.)

presentation can be used."), 8:26-29 ("FIG. 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion compensated temporal prediction."), 4:20-23 ("An indication of at least one image frame is encoded into the video sequence, which indicated image frame is the first picture, in decoding order, of the independent sequence.")) All of these terms refer to a single still image. (*See* EX1001, 2:55-58 (explaining that "video files ... comprise a great number of still image frames, which are displayed rapidly in succession ... to create an impression of a moving image").)

² Page numbers for non-patent references refer to page numbers added by Petitioners (and correspond to the pdf page number) unless otherwise stated.

49. A specific type of GOP discussed in the '321 patent is the “independently decodable [GOP]” or “independent sequence.” (EX1001, 11:26-30, Abstract.) An independently decodable GOP contains pictures that can be decoded independently of information outside of the GOP. But not all GOPs are independently decodable (*see* EX1001, 3:52-56) because image frames may depend on information from other frames, either within the same GOP or outside of it. Such dependency arises from the use of a “temporal prediction” compression technique that takes advantage of “temporal redundancy” across image frames.

50. As the patent explains, “temporal redundancy refers to the changes taking place in specific image objects in subsequent image frames.” (*See* EX1001, 2:1-3.) Because video sequence image frames are usually captured in quick succession, many parts of the captured scene either do not change or change in predictable ways from one image frame to the next. These parts are considered to be temporally redundant.

51. Encoding only the differences between image frames is one way to take advantage of temporal redundancies. This is called “temporal prediction” because information about one frame is used to predict information about another frame from a different time. It allows image frames to be encoded using less data by omitting redundant information and instead only encoding some information about the

depicted scene and then referring to another image frame for any omitted information.

52. The '321 patent mentions “motion-compensated temporal prediction” (EX1001, 2:18-24), which predicts the location of an object in a given image frame based on its location in other image frames and information about its movement. (See EX1001, 2:16-23 (describing “motion-compensated temporal prediction” as follows: “the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames”).)

53. When temporal prediction is used to encode an image frame, additional information outside the encoded image frame is needed to decode it—e.g., the reference image frame on which it depends. (See EX1001, 2:31-33 (“If a frame is lost, frames dependent on it can no longer be correctly decoded.”).)

54. Image frames are given “types” based on their dependence on other image frames. (See, e.g., EX1001, 3:50-52 (“Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames.”).) The '321 patent discusses three of these types: (1) “Intra-frames” or “I-frames,” which do not depend on information from other image frames (EX1001, 2:23-29); (2) “Inter-frames” or “P-frames,” which depend on information from previous I- or P-frames (EX1001, 2:27-31); and (3) “Bi-directionally predicted

frames” or “B-frames,” which depend on both a preceding and succeeding reference frame (EX1001, 2:43-48).

55. As discussed above, an independently decodable GOP or independent sequence can be decoded without information from image frames in another GOP. This means that an independently decodable GOP includes image frames encoded using temporal prediction that depend only on reference frames within that same independently decodable GOP.

56. Independently decodable GOPs long predate the ’321 patent. For example, MPEG-1 (from 1993) explained that “[t]he original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group.” (EX1005, 85.) MPEG-1 therefore described encoding groups of pictures to ensure that they were independently decodable while still being compliant with the MPEG-1 standard. (EX1005, 85 (“One way to do this is to omit any B-pictures preceding the first I-picture. Another way is to allow such B-pictures, but to code them using only backward motion compensation.”).)

B. The Challenged Claims

57. The Petition challenges claims 8-11 of the ’321 patent (reproduced below for reference). Claims 8, 10, and 11 are independent, and claim 9 depends from claim 8.

8. A method for decoding a compressed video sequence,
the method comprising:

decoding from the video sequence an indication of at
least one image frame, which is the first image frame, in
decoding order, of an independent sequence, wherein all
motion-compensated temporal prediction references of
the independent sequence refer only to image frames
within said independent sequence;

starting the decoding of the video sequence from said
first image frame of the independent sequence, whereby
the video sequence is decoded without prediction from
any image frame decoded prior to said first image frame;

decoding identifier values for image frames according to
a numbering scheme; and

resetting the identifier value for the indicated first image
frame of the independent sequence.

9. A method according to claim 8, wherein the indication
is a separate flag included in the header of a slice.

10. A video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence, by causing the video decoder to at least:

decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;

start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and

decode identifier values for image frames according to a numbering scheme; and

reset the identifier value for the indicated first image frame of the independent sequence.

11. A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, the computer program product comprising:

a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;

a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

a computer program code for decoding identifier values
for image frames according to a numbering scheme; and

a computer program code for resetting the identifier
value for the indicated first image frame of the
independent sequence.

58. The Challenged Claims are directed to the grouping and numbering of image frames that, according to the patent, helps to resolve “[a] problem in the numbering of image frames [that] arises when [a] user wishes to start browsing a video file in the middle of a video sequence.” (EX1001, 11:11-13.) The ’321 patent says that, “[w]hen the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering,” which “[t]he decoder typically interprets ... as [an] unintentional loss of image frames.” (EX1001, 11:20-25.)

59. This can be “avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at the random point of the video file, and the number of the initiation image is set at zero.” (EX1001, 11:26-30.) In other words, “when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero.” (EX1001, 11:37-40.) This allows

the decoder to “find the correct image frame that initiates the subsequence from the video sequence of image frames.” (EX1001, 11:46-49.)

C. '321 Patent Prosecution History

60. The '321 patent issued from U.S. Application No. 11/338,934 (filed on January 25, 2006) and claims priority to a continuation application that was filed on November 29, 2002, and a foreign application that was filed on January 23, 2002. Neither MPEG-1 nor Kim was before the Examiner during prosecution. (See EX1002.) Yagasaki was cited by the Examiner as part of an obviousness combination with U.S. Patent No. 7,103,669 (“Apostolopoulos”) to reject then-pending claim 11 (EX1002, 750-51), which read “A method according to claim 10, wherein the indication is a separate flag included in the header of a slice.” (EX1002, 735.) The Examiner found Yagasaki “discloses putting a flag for indicating an intra-frame coded block in the slice header (column 10, line 65 – column 11, line 2).” (EX1002, 750.) The Applicant did not dispute this finding and did not distinguish Yagasaki. Instead, the Applicant amended then-pending claim 10 and argued that the amendment distinguished the claim from Apostolopoulos. (EX1002, 770-778.) Then-pending claim 11 ultimately issued as claim 9. (EX1002, 806.)

61. In the Notice of Allowance, the Examiner explained that Apostolopoulos “discloses a similar method but fails to disclose starting the decoding of the video sequence from said first image frame of the independent

sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; decoding identifier values for image frames according to a numbering scheme; and resetting the identifier value for the indicated first image frame of the independent sequence.” (EX1002, 794.) But it is my opinion that the references cited in this Declaration teach the limitations of the Challenged Claims, including the limitations the Examiner found missing from the art applied during prosecution.

VIII. CLAIM CONSTRUCTION

62. I have given all the claim terms of the Challenged Claims their plain and ordinary meaning that a POSITA would have understood at the time of the alleged invention.

63. I understand that in a separate proceeding before the Board, Patent Owner proposed to construe the term *indication*³ in claim 8 to mean “information in an encoded bitstream that necessarily confirms something to a decoder.” See IPR2024-00691, Paper 20, 34. I disagree with this construction because I do not believe it comports with the plain and ordinary meaning of *indication* to a POSITA at the time of the ’321 patent in view of the claim language. In my opinion, the Challenged Claims do not require the *indication* to “necessarily confirm[]” anything;

³ I have italicized claim language throughout this declaration.

instead, the recited indication need only show that *at least one image frame*—including the first image frame in decoding order—is present. (EX1001, 23:51-24:2; EX2005, 77:20-80:12, 83:10-85:4, 87:18-88:22.) In any event, my opinions do not depend upon the Board adopting a specific construction of the term *indication* as I believe the Challenged Claims would nevertheless have been obvious to a POSITA in view of the prior art discussed below.

IX. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: MPEG-1 Renders Obvious Claims 8, 10, and 11

1. MPEG-1

a. Overview

64. International Standard ISO/IEC 11172 (“the Standard”) is titled “Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1,5 Mbit/s.” (EX1005, 1.) The Standard was approved in 1993 and consisted of four parts at the time: Systems, Video, Audio, and Compliance Testing. (EX1005, 11.) ISO/IEC later published a fifth part that included a software simulation. (See ISO/IEC TR 11172-5:1998: Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s Part 5: Software simulation, <https://www.iso.org/standard/25029.html>.)

65. The term “MPEG-1” historically referred to all parts of the Standard; however, I understand that the Petition uses the term “MPEG-1” specifically to refer

to the second part of the Standard (ISO/IEC 11172-2) regarding video. (See EX1005, 11.) To avoid any confusion, I will use that same naming convention in this declaration.

66. The Moving Pictures Expert Group developed MPEG-1 to respond to “the growing need for a common format for representing compressed video on various digital storage media such as CDs, DATs, Winchester disks and optical drives.” (EX1005, 12, 68.) MPEG-1 “specifies a syntax for a coded video bitstream,” (EX1005, 15) and defines the process for decoding that bitstream (EX1005, 15). (See also EX1005, 18 (“This part of ISO/IEC 11172 specifies the coded representation of video for digital storage media and specifies the decoding process.”).)

b. Key Concepts

67. Among other things, MPEG-1 teaches five concepts that are particularly relevant to Ground 1: (1) picture type; (2) picture ordering; (3) groups of pictures (GOPs); (4) closed GOPs; and (5) Temporal Reference.

i. Picture Type (I, P, and B Pictures)

68. MPEG-1 teaches that “[a] picture corresponds to a single frame of motion video or to a movie frame.” (EX1005, 71.) A “picture” in MPEG-1 corresponds to the claimed *image frames* in the ’321 patent. (See EX1001, 2:55-58 (explaining that “video files ... comprise a great number of still image frames, which

are displayed rapidly in succession ... to create an impression of a moving image”).)

As I observed above, the ’321 patent uses the terms “pictures” and “image frames” interchangeably. (*See, e.g.*, EX1001, 4:20-23 (“An indication of at least one image frame is encoded into the video sequence, which indicated image frame is the first picture, in decoding order, of the independent sequence.”).)

69. MPEG-1 describes three picture types that are relevant my analysis: I-pictures, P-pictures, and B-pictures. (*See* EX1005, 71.)

70. An “I-picture” (intra coded picture) is “[a] picture coded using information only from itself” (EX1005, 23) and therefore “without reference to any other pictures” (EX1005, 71).

71. A “P-picture” (predictive coded picture) is “[a] picture that is coded using motion compensated prediction from the past reference picture.” (EX1005, 25.) “[T]he past reference picture” is defined to be the nearest adjacent I- or P-picture that occurs at an earlier time in display order. (*See* EX1005, 25 (defining “past reference picture”), 26 (defining “reference picture”); *see also* EX1005, 71 (“P-pictures ... are coded using motion compensation from a previous I or P-picture.”).)

72. A “B-picture” (bidirectionally predictive coded picture) is “[a] picture that is coded using motion compensated prediction from a past and/or future reference picture.” (EX1005, 20; *see also* EX1005, 71.)

73. Each picture has a header with certain corresponding information that includes, for example, picture coding type. (See EX1005, 87-88.)

ii. Picture Ordering

74. MPEG-1 discusses two “picture orderings” that it refers to as “display order” and “bitstream order.” (See EX1005, 84.) Display order is “[t]he order in which the decoded pictures should be displayed.” (EX1005, 22, 87.) Bitstream order is the “the order in which pictures are transmitted” in the bitstream. (EX1005, 70.) Because the decoder decodes pictures in the order in which it receives them, “bitstream order” also reflects the order in which pictures are decoded. (EX1005, 32 (“The order of the pictures in the coded stream is the order in which the decoder processes them in normal playback.”), 71 (explaining that bitstream order is “the order which the decoder requires them to decode the bitstream”), 21 (“coded order [video]: The order in which the pictures are stored and decoded. This order is not necessarily the same as the display order.”).)

75. MPEG-1 explains that these picture orderings may be different because of picture dependencies. (EX1005, 71.) For example, MPEG-1 provides the following example of a sequence of pictures in display order:

I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Figure D.3 -- Typical sequence of pictures in display order

(EX1005, 71.) The same sequence in bitstream order would be:

I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P	B	B
0	3	1	2	6	4	5	9	7	8	12	10	11	15	13	14	18	16	17

Figure D.4 -- Typical sequence of pictures in bitstream order

(EX1005, 71.) According to MPEG-1, “[b]ecause the B-pictures depend on the following (in display order) I or P-picture, the I or P-picture must be transmitted and decoded before the dependent B-pictures.” (EX1005, 71.) In Figure D.3, for example, picture P3 is fourth in display order, but it is second in decoding order in Figure D.4. This is because pictures B1 and B2 depend upon picture P3 to be decoded.

76. Notably, the ’321 patent uses the term “decoding order.” A POSITA would have understood the plain and ordinary meaning that phrase to be the order in which pictures are decoded. “Decoding order” and MPEG-1’s “bitstream order” thus refer to the same picture ordering—which a POSITA would have understood to be distinct from “display order”. (EX1005, 20.)

iii. Groups of Pictures (“GOPs”)

77. A GOP is “[a] series of one or more coded pictures intended to assist random access.” (EX1005, 23.) “[I]t is the smallest coding unit that can be independently decoded within a sequence, and consists of a header and some number of pictures.” (EX1005, 71.)

78. Pictures in a GOP in *bitstream* order “must start with an I-picture and may be followed by any number of I, P or B-pictures in any order.” (EX1005, 84.) Pictures in a GOP in *display* order are “contiguous” and “must begin ... with an I or a B-picture, and must end with an I or a P-picture.” (EX1005, 84.)

79. Given that a GOP must end with an I- or a P-picture in display order, no picture in a GOP can depend on a reference picture in a subsequent GOP for motion compensation prediction. I- and P-pictures cannot depend on reference pictures in a subsequent GOP because I-pictures “are coded without reference to any other pictures,” and P-pictures “are coded using motion compensation from a previous I or P-picture.” (EX1005, at 71.) And a B-picture in the GOP cannot refer to a picture beyond the last I- or P-picture because reference pictures are “the nearest adjacent I- or P-pictures to the current picture in display order.” (EX1005, 26.) Thus, because GOPs must end with an I- or a P-picture, it follows that any B-picture cannot refer to a picture beyond that final picture.

iv. Closed GOPs

80. In MPEG-1, “[t]he original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group.” (EX1005, 85.) MPEG-1 does not require that every GOP be independently decodable, but “encoders can still construct groups of pictures which are independent of one another.” (EX1005, 85.)

81. MPEG-1 teaches the concept of a closed GOP—i.e., an independently decodable sequence of pictures—and MPEG-1 explains that it is a “group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures.” (EX1005, 43.) Accordingly, a closed GOP “can be decoded without using decoded pictures of the previous group for motion compensation.”⁴ (EX1005, 85.) And as noted previously, all GOPs can be decoded without using decoded pictures of a subsequent group for motion compensation. (*See* §IX.A.1.b.iii.) The MPEG-1 closed GOPs are therefore independently decodable sequences of pictures.

v. Temporal Reference

82. Each MPEG-1 picture begins with a header that includes the picture’s “Temporal Reference,” which is a “number which can be used to define the order in which the pictures must be displayed.” (EX1005, 87.) In MPEG-1, “[t]he first picture, in display order, in each group must have Temporal Reference equal to zero,” which is “incremented by one for each picture in the group.” (EX1005, 87,

⁴ As noted above, MPEG-1 does not allow any picture in a GOP to be encoded with a motion vector pointing to a reference picture in a later GOP. So the additional requirement that a closed GOP be “encoded without motion vectors pointing to the previous [GOP]” (EX1005, 42) implies that all temporal references within a closed GOP must be to other pictures in that same GOP.

44.) Temporal Reference thus provides the display order of pictures. (EX1005, 44 (“The temporal-reference is assigned (in sequence) to the pictures in display order.”).)

c. Exemplary Closed GOP

83. MPEG-1 teaches several ways for an encoder to “construct groups of pictures which are independent of one another” (closed GOPs). (*See* EX1005, 85.) “One way to do this is to omit any B-pictures preceding the first I-picture.” (EX1005, 85.) Such a GOP will necessarily begin with an I-picture in both decoding and display order. (EX1005, 84 (“A group of pictures, in bitstream order must start with an I picture” and “must begin, in display, order with an I or a B-picture.”).) The Petition uses the term “exemplary closed GOP” to refer to a GOP encoded in this manner, and I will do likewise in this declaration for consistency.

84. MPEG-1 describes several example GOPs encoded without B-pictures preceding the first I-picture. For example, Figure D.18(a) below depicts “[a] typical example of a closed group [of pictures]” presented in display order. (EX1005, 86.)

I	B	B	P	B	B	P	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12
(a) closed group												

MPEG-1, Fig. D.18(a)

85. In the exemplary closed GOP, a single I-picture must be the first picture in both decoding order and display order, and it must therefore have a Temporal

Reference of zero. It is first in decoding order because MPEG-1 requires the first picture in decoding order of any GOP to be an I-picture. (EX1005, 84 (“*Property 1*. A group of pictures, in bitstream order, must start with an I-picture.”).) And it is first in display order because the first picture of a GOP in display order must be an I- or a B-picture, and the example expressly says that any B-pictures preceding the first I-picture have been omitted. (See EX1005, 84 (“*Property 2*. Another property of a group of pictures is that it must begin, in display order, with an I or a B-picture.”).) Because this I-picture is first in display order, it has a Temporal Reference of zero. (EX1005, 87 (“The first picture, in display order, in each group must have Temporal Reference equal to zero.”).) Thus, the first picture in decoding order of every exemplary closed GOP (*i.e.*, every GOP encoded with an I-picture as the first picture in display order) must necessarily have a Temporal Reference of zero.

86. This is illustrated in the following example from MPEG-1:

Example (a) in display order	I	B	P	B	P
	0	1	2	3	4
Example (a) in decoding order	I	P	B	P	B
	0	2	1	4	3

MPEG-1, Fig. D.19(a)
 (“Examples of Groups of Pictures and Temporal References”)

(EX1005, 87.) As shown, the top row presents a series of pictures in an exemplary GOP in display order, and the bottom row presents the same exemplary GOP in

decoding order. The numbers in the figure indicate the Temporal Reference values for each picture, where the I-picture in the exemplary GOP is first in both decoding and display order and has a Temporal Reference of zero. (EX1005, 87.)

2. Claim 8

- a. **8[a]:** *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*

87. I first explain below how MPEG-1's closed GOP teaches *an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*. I then explain how the MPEG-1 closed_gop flag teaches *an indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence*. I also explain how a Temporal Reference of zero provides *an alternative indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence*. In my opinion, either of these teachings, when considered with MPEG-1's closed GOP, discloses the full limitation to a POSITA.

- i. ***an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence***

88. A POSITA would have understood MPEG-1's closed GOP to be *an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence.*

89. As I explained above, a closed GOP is decoded without using pictures from a previous or subsequent group for motion compensation because a closed GOP is, by definition, a “group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures.” (EX1005, 43.) All closed GOPs therefore “must end with an I or a P-picture” and cannot depend on pictures from a subsequent group of pictures for motion compensation. (See EX1005, 84, 23 (an I-picture is “[a] picture coded using information only from itself”), 25 (a P-picture is “[a] picture that is coded using motion compensation prediction from the past reference picture”), 32 (“The order of the pictures in the coded stream is the order in which the decoder processes them in normal playback” and “[t]he last coded picture, in display order, of a group of pictures is either an I-Picture or a P-Picture”).)

90. A POSITA would have therefore understood MPEG-1's closed GOP to be *an independent sequence* because it does not depend on other sequences, and a POSITA would have further understood that all MPEG-1 motion vectors (which

correspond to *motion-compensated temporal prediction references*) point only to image frames within that closed GOP (meaning they *refer only to image frames within said independent sequence*).

ii. *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*

91. In a MPEG-1 bitstream, “a group of pictures begins with a [GOP] header” that, in turn, begins with a “group_start_code” that “identifies the beginning of a group of pictures.” (EX1005, 43, 85.)

92. The GOP header contains a closed_gop flag that indicates whether the GOP to which it corresponds is open or closed. (EX1005, 43, 86.) And as I explained above, a POSITA would have understood a closed GOP to be an *independent sequence* such that when the closed GOP flag indicates that the GOP is closed—rather than open—it is also *an indication of...an independent sequence*.

93. Furthermore, because the GOP header precedes the first picture in the GOP in decoding order, a POSITA would have understood that the GOP header is *an indication of...the first image frame, in decoding order*, of a closed GOP because it precedes that image frame. (See EX1005, 120 (explaining that “the decoder will decode the group of pictures header, including the closed_gop and broken-link information” and then “decode the first picture header in the group of pictures”).)

94. And because the closed_gop flag is in the GOP header, the first image frame decoded after the closed_gop flag is *the first image in decoding order* in a closed GOP. (See EX1005, 120 (describing the decoding process for forward playback).) That is, when the closed_gop flag indicates that the GOP is closed, it also indicates that the next image frame decoded is the first image frame in decoding order of *an independent sequence*.

95. A POSITA would have understood that MPEG-1 decodes this *indication* because it decodes the GOP header, which includes the closed_gop flag. (EX1005, 120 (explaining that “the decoder will decode the group of pictures header, including the closed_gop and broken-link information”).)

96. Alternatively, a Temporal Reference of zero in MPEG-1 teaches *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence*. As I discuss above, MPEG-1 describes an exemplary closed GOP that is encoded without B-pictures preceding the first I-picture in display order, and the first picture in decoding order of the exemplary closed GOP will always be an I-frame with a Temporal Reference of zero (because that I-frame is the first frame in display order and so has a Temporal Reference of zero). A POSITA would therefore have understood that in the exemplary closed GOP, the I-frame’s Temporal Reference of

zero is an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.

97. And as I explain below for limitation 8[c], an MPEG-1 decoder decodes Temporal Reference. (See EX1005, 120 (disclosing that picture header gets decoded during forward playback).)

98. A POSITA would thus have understood MPEG-1 to disclose *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence* based on decoding the closed_gop flag or alternatively decoding the Temporal Reference of zero for the first I-frame of the closed GOP.

99. It is therefore my opinion that MPEG-1's disclosure of decoding either a closed_gop flag or the Temporal Reference of zero for the first I-frame of a closed GOP teaches *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence.*

100. I understand that in a separate proceeding regarding the '321 patent, the Patent Owner argued that the MPEG-1 closed_gop flag is only an indication of whether the GOP is open or closed, not of a particular image frame, and because it is in the header, it is a syntax element that only applies to the entire GOP. IPR2024-

00691, Paper 20, 32-33. I disagree with this argument because it appears to ignore the exemplary closed GOP, in which the closed_gop flag indicates that the next image frame decoded is the first one in decoding order of the independent sequence, as I explain above.

101. I further understand that in a separate proceeding, the Patent Owner argued that closed_gop “does not confirm to a decoder any type of image as a first image frame”; rather, “an MPEG-1 decoder that receives a closed_gop flag with a value of ‘0’ or ‘1’ has no certainty about what frames follow in the bitstream.” IPR2024-00691, Paper 20, 34. This argument appears to be based on Patent Owner’s proposed construction for “indication” that I discuss in the Claim Construction section above. Even applying that construction, however, a POSITA would have understood the closed_gop flag to confirm to the decoder that the next image frame decoded will be the first image frame in decoding order in the exemplary closed GOP because, after decoding the closed_gop flag in the exemplary closed GOP header, the decoder recognizes that the next image frame it receives will be an I-picture that is the first image frame in decoding order of the independent sequence, as MPEG-1 illustrates in Example D.19(a). (EX1005, 87.)

Example (a) in display order	I 0	B 1	P 2	B 3	P 4
Example (a) in decoding order	I 0	P 2	B 1	P 4	B 3

102. Finally, I understand that in the separate proceeding, the Patent Owner further argued that a Temporal Reference of zero indicates the first picture in display order, not necessarily in decoding order. IPR2024-00691, Paper 20, 41. I disagree. In the exemplary closed GOP of MPEG-1 that I discuss above, an I-picture is the first image frame in both decoding and display order and has a Temporal Reference of zero.

- b. **8[b]: *starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame***

103. MPEG-1 starts the decoding of a video sequence from the first picture of a closed GOP (the *first image frame of the independent sequence*). “If playback begins from a random point in the bitstream, the decoder should discard all the bits until it finds a sequence start code, a group of pictures start code, or a picture start code which introduces an I-picture.” (EX1005, 120.) And a closed GOP begins with a group of pictures start code. (See EX1005, 43 (the group of pictures start code “identifies the beginning of a group of pictures”).) Consequently, a POSITA

would have understood that a MPEG-1 decoder will discard all bits until it finds a group of pictures start code at the beginning of a closed GOP (*independent sequence*), which means the decoder begins decoding the video sequence from the first picture in that closed GOP (*said first image frame of the independent sequence*). (See EX1005, 32.)

104. In addition, based on the MPEG-1 requirement that closed GOPs be “decoded without using decoded pictures of the previous group for motion compensation” (EX1005, 86), a POSITA would have understood not only that decoding begins with the first picture of a closed GOP (*starting the decoding of the video sequence from the first image frame of an independent sequence*), but also that the closed GOP is decoded without using pictures from a prior group for motion compensation (*decoded without prediction from any image frame decoded prior to the first image frame*).

c. 8[c]: *decoding identifier values for image frames according to a numbering scheme*

105. Each picture in MPEG-1 has an *identifier value* called a Temporal Reference that “is a ten-bit number which can be used to define the order in which the pictures must be displayed.” (EX1005, 87, 44.) During forward playback, after decoding the GOP header, the decoder “will decode the first picture header in the group of pictures” (EX1005, 120), and because the Temporal Reference is part of the picture header, and each picture in a GOP has a header (EX1005, 71), the decoder

decodes the Temporal Reference when it decodes the first picture header in the GOP. A POSITA therefore would have understood that each picture's Temporal Reference would be *decod[ed]* in a MPEG-1 GOP.

106. Furthermore, MPEG-1 Temporal Reference values are decoded according to a numbering scheme. During decoding, “slices and macroblocks in the picture are decoded and written into a display buffer” and “displayed in the order defined by the temporal reference.” (EX1005, 119.) A POSITA would have understood that “displayed in the order defined by temporal reference” means that the decoder outputs pictures in the order of their Temporal Reference. MPEG-1 pictures are assigned a Temporal Reference using the following numbering scheme: “[t]he first picture, in display order, in each [GOP] must have Temporal Reference equal to zero,” and Temporal Reference is then “incremented by one for each picture in the group.” (EX1005, 87, 44.) Because pictures are assigned Temporal Reference according to a numbering scheme and use that numbering scheme to determine the order to output pictures, a POSITA would have understood that the pictures and their corresponding Temporal Reference values are *decod[ed]...according to a numbering scheme*.

d. 8[d]: *resetting the identifier value for the indicated first image frame of the independent sequence*

107. As I explained above, MPEG-1 presents an exemplary closed GOP that begins with an I-frame in both decoding and display order. And “[f]or the earliest

picture (in display order) in each group of pictures, the temporal-reference is reset to zero.” (EX1005, 44, 87.) Accordingly, Figure D.19(a) from MPEG-1 shows a Temporal Reference of zero for the I-frame that increments by one in each subsequent frame in display order.

Example (a) in display order	I	B	P	B	P
	0	1	2	3	4
Example (a) in decoding order	I	P	B	P	B
	0	2	1	4	3

MPEG-1, Fig. D.19(a)

(“Examples of Groups of Pictures and Temporal References”)

This would be true for each and every exemplary closed GOP in a given bitstream. So a POSITA would therefore have understood that when a MPEG-1 decoder decodes the next exemplary closed GOP, the Temporal Reference (*identifier value*) for the first picture in decoding order of the exemplary closed GOP (*the indicated first image frame of the independent sequence*) would be *reset* to zero because it would be the first picture in display order for that GOP.

108. I understand that in a separate proceeding involving the ’321 patent, the Patent Owner argued MPEG-1 does not teach this claim limitation because: (1) the encoder, not the decoder, resets the identifier; (2) only the identifier for a subsequent video sequence is reset; and (3) MPEG-1 does not always require that an I-picture

(the first image frame in decoding order) have a Temporal Reference of zero.

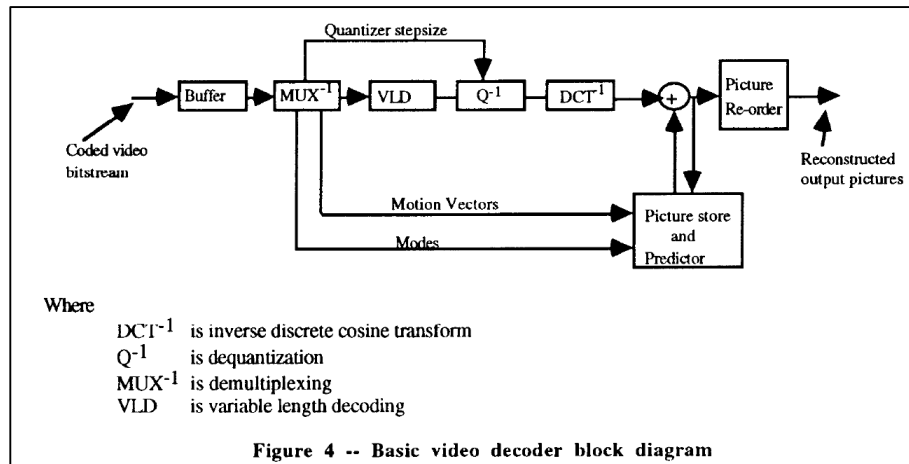
IPR2024-00691, Paper 20, 50-53. I disagree with all three arguments.

109. As explained above, the Temporal Reference in MPEG-1 is reset for the first image frame of each independent sequence, and the encoder informs the decoder of the value of the Temporal Reference for each frame (including the first image frame in decoding order). This happens for every exemplary closed GOP, and I am not aware of any scenario in MPEG-1 in which a compliant decoder would change the value of a syntax element set by the encoder. Consequently, it would have been obvious to a POSITA that an MPEG-1 decoder likewise resets the Temporal Reference to zero when it decodes the first image frame in decoding order of the exemplary closed GOP. If this were not the case—e.g., if a decoder unilaterally changed a Temporal Reference value—the decoder could not correctly determine the proper display order.

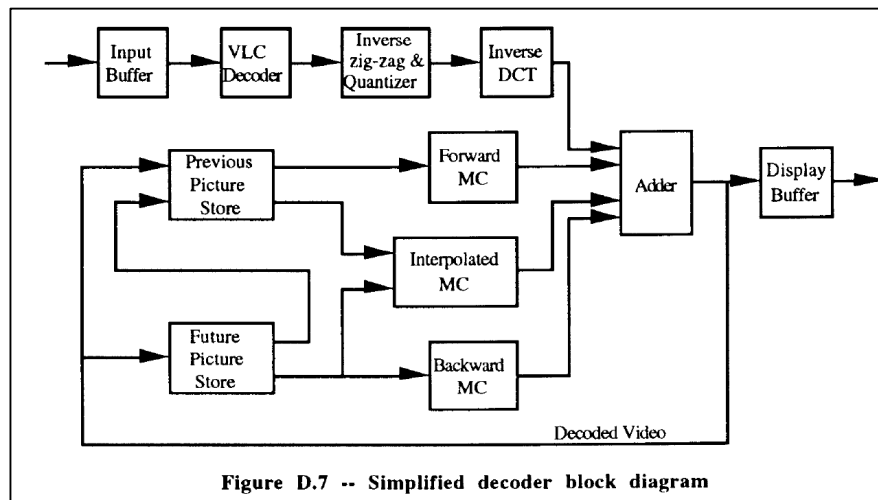
3. Claim 10

110. Claim 10 is substantively the same as claim 8, except instead of being written in terms of “[a] method for decoding a compressed video sequence” it is written in terms of a “[a] video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence.”

111. MPEG-1 discloses multiple decoders, including, for example, the following:



(EX1005, 16.)



(EX1005, 73.)

112. It would have been obvious to a POSITA that an MPEG-1 decoder would be implemented using at least one processor and at least one memory that

stores computer program code configured to cause the video decoder to decode a compressed video sequence. This would have been obvious to a POSITA based on background knowledge such as awareness of well-known decoders with such hardware that pre-date January 23, 2002 (the date of the foreign patent to which the '321 patent claims priority).

113. One example of this common and obvious configuration is C-Cube's CL480PC MPEG-1 audio/video decoder (EX1007, 1), which was publicly available by 1997 and decoded video using a combination of processors. (EX1007, 3.) One of the processors used code saved in DRAM that was loaded into on-chip memory. (EX1007, 3.) Because C-Cube's CL480PC was an operational MPEG-1 decoder, a POSITA would have had a reasonable expectation of success in implementing its configuration.

114. Many references that predate the '321 patent likewise describe software decoding of MPEG-1 video. For example, U.S. Patent No. 7,263,273 (filed November 25, 2000) describes software decoding of MPEG-1 video using a processor. (See EX1008, 6:56-65, 7:53-58.) And engineers in 1995 wrote about an entry-level enterprise workstation that could "achieve[] real-time MPEG-1 ... video decompression via software running on a general-purpose processor." (EX1009, 60.) Even back in 1993, engineers implemented a decoder in 12,000 lines of C code

using the X Window⁵ System (EX1010, 3), which a POSITA would have understood to include a processor and memory.

115. Even MPEG-1 itself suggests implementing a decoder using a processor, memory, and software code on a personal computer. (See EX1005, 12 (MPEG-1 “provides flexibility for use with personal computer and workstation displays”).) Thus, MPEG-1 suggests to a POSITA decoding compressed video using a processor in personal running computer code stored in memory.

116. Such disclosures confirm that the use of *decoder[s] comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence* were well known in the art before the earliest priority date of the '321 patent. And it would have been obvious to a POSITA to implement a MPEG-1 decoder using that structure and configuration.

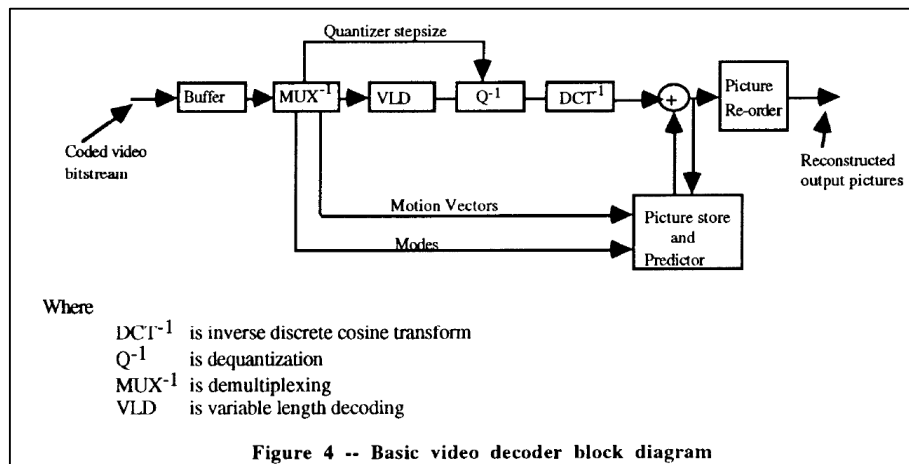
4. Claim 11

117. Claim 11 is substantively the same as claim 8, except instead of being written in terms of *a method for decoding a compressed video sequence* it is written

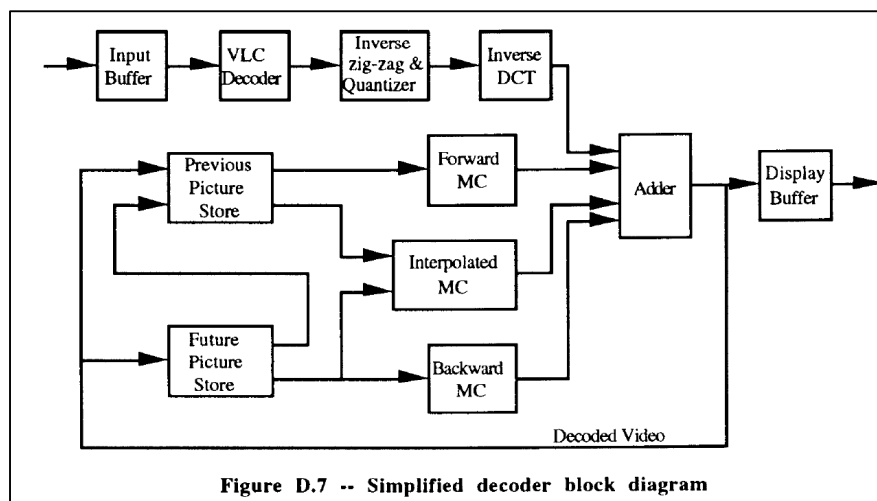
⁵ Exhibit 1010 incorrectly uses the term “X Windowing System” instead of “X Window System,” which was also known as X11.

in terms of a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.

118. As discussed above, MPEG-1 describes several decoders:



(EX1005, 16.)



(EX1005, 73.)

119. A POSITA would have found it obvious to implement a MPEG-1 video decoder as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. As I explained above, MPEG-1 decoders were well known in the art long before the '321 patent (*e.g.*, C-Cube's CL480PC MPEG-1 audio/video decoder (EX1007, 3)). For instance, a CL480PC processor (publicly available by 1997) would use microcode stored in DRAM that was loaded into on-chip memory. (EX1007, 3.) A POSITA would have understood such microcode to be a common and obvious configuration corresponding to *a computer program product ... executable in a data processing device, for decoding a compressed video sequence*" and that the DRAM on which it was stored was "*a non-transitory computer readable medium*."

120. A POSITA would likewise have understood MPEG-1 to suggest a decoder that uses the processor, memory, and program code of a personal computer. (*E.g.*, EX1005, 12 (MPEG-1 "provides flexibility for use with personal computer and workstation displays").) And because MPEG-1 operations were routinely carried out by computer program products available on personal computers before the '321 patent, a POSITA would have had a reasonable expectation of success in implementing MPEG-1 decoding in software.

121. For example, U.S. Patent No. 7,263,270 (filed November 25, 2000) describes several ways to decode MPEG-1 video, including using “software” running on a processor. (See EX1008, 6:59-65, 7:53-58.) As another example, in 1995, engineers wrote about an entry-level enterprise workstation that could “achieve[] real-time MPEG-1 ... video decompression via software running on a general-purpose processor.” (EX1009, 60.) A POSITA would have understood “software” to be *a computer program product that would be stored on a non-transitory computer readable medium and executable in a data processing device*. And to provide yet another example, engineers in 1993 implemented a decoder in 12,000 lines of C code using the X Window System (EX1010, 3), which a POSITA would have understood to include a *computer program product, stored on a non-transitory computer readable medium and executable in a data processing device*.

B. Ground 2: Kim Renders Obvious Claims 8, 10, and 11

1. Kim

122. Kim describes “a device and method for recording and reproducing video data on and from an optical disk using a Time Lapse recording/reproduction process.” (EX1006, 1:10-12.) Notably, Kim’s “reproducing method” decodes a bitstream of compressed video data in response to a user’s request and outputs an analog image signal. (EX1006, 5:8-17, 5:29-34, 5:55-6:8.) Kim illustrates the bitstream with using a “logical structure” based on dependencies:

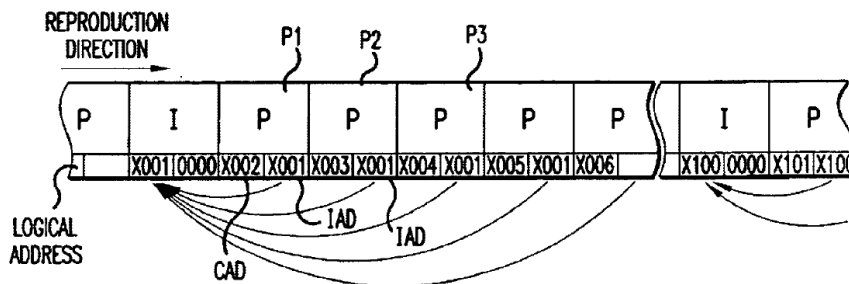


FIG.5A

(EX1006, Fig. 5A.)

123. As shown, the logical structure includes an I-picture followed by P-pictures that depend on the I-picture. (EX1006, Fig. 5A, 4:46-52.) Kim's I- and P-pictures have similar characteristics to the MPEG-1 I- and P-pictures discussed above. (See EX1006, 4:15-19, 6:55-58.) For example, each of Kim's I-pictures is an "intra-picture" and therefore "carries the entire image information of a frame." (EX1006, 3:60-61.) The I-pictures do not depend on other pictures to be decoded, and they output as an analog image signal. And each of Kim's P-pictures is "predictive" (EX1006, 2:22) and encoded based on "the moving vector and the position value of the corresponding (or previous) I-frame" (EX1006, 4:28-33). P-pictures thus require data from a corresponding I-picture moving vectors to be restored as full images. (EX1006, 5:55-59.)

124. Kim's bitstream also includes data structures that associate each picture with address fields that identify the picture and its corresponding reference picture, as illustrated in Figure 6:

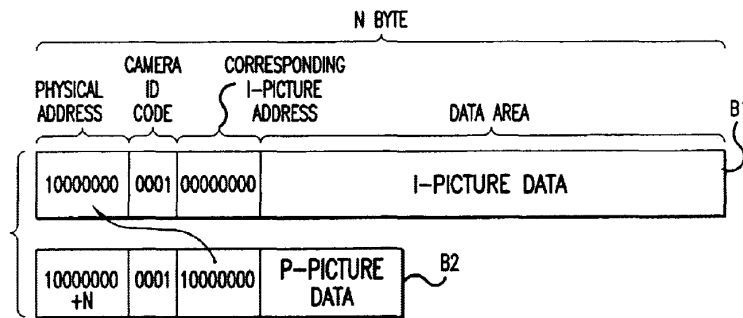


FIG.6

(EX1006, Fig. 6.)

125. Each P-picture has a current address (“CAD”) and an I-picture reference address (“IAD”) that indicates the address of the corresponding (or previous) I-frame. (EX1006, 4:46-52.) I-pictures also have a CAD that indicates the picture’s current address (*see* EX1006, 4:65-5:1, Figs. 5A, 6 (“physical address”)), but the I-picture’s IAD is zero (*see* EX1006, Figs. 5A (showing “0000”), 6 (showing “00000000” for the “corresponding I-Picture address”)). This is illustrated in Figure 5A below, which I have annotated to indicate IAD values in red and CAD values in green.

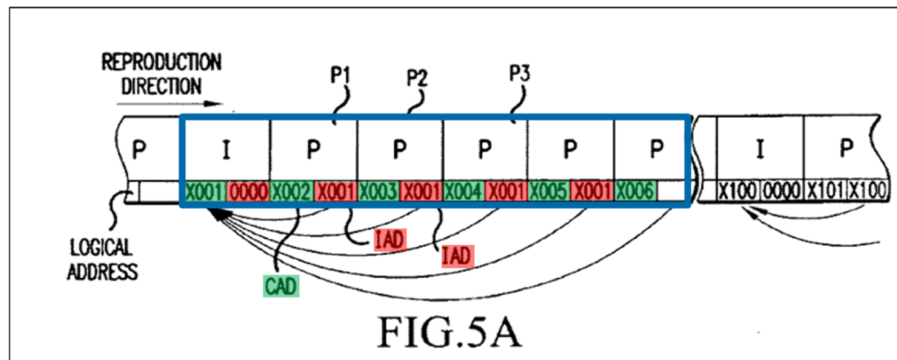
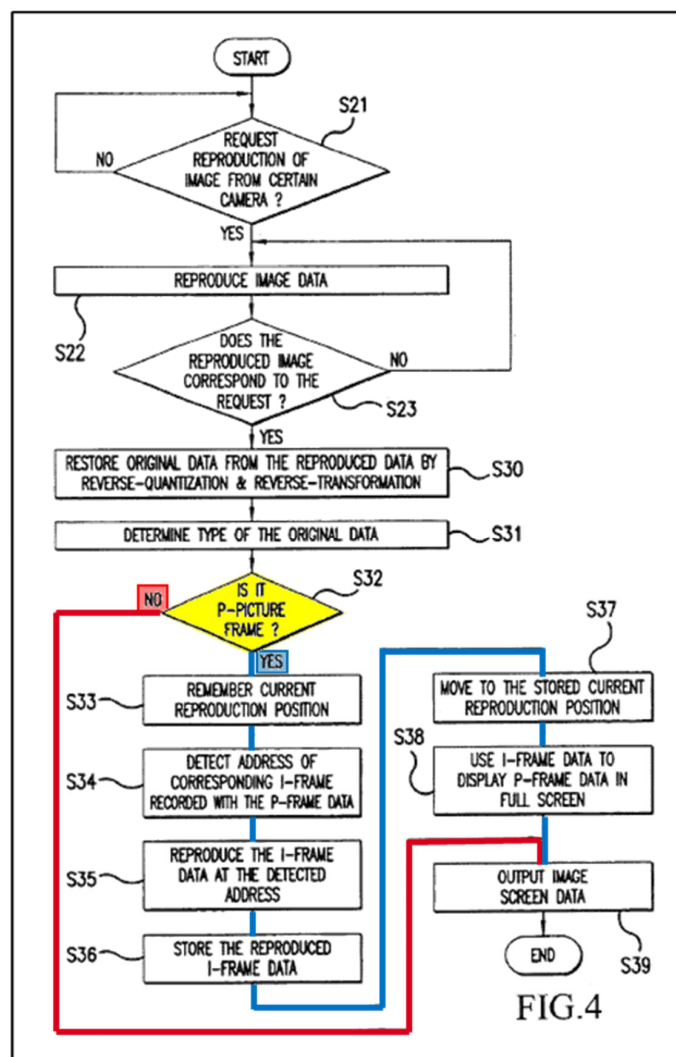


FIG.5A

(EX1006, Fig. 5A (annotated).)

126. Figure 4 of Kim illustrates two paths for decoding a bitstream that depend upon whether the user requests image data for an I-picture or a P-picture. When the request corresponds to an I-picture, the picture is decoded and output as an image signal. (EX1006, 5:66-6:3, Fig. 4 (“no” path annotated in red).) But when the request corresponds to a P-picture, the system “detects the address of the corresponding I-picture frame ... and reproduces based on the detected address [of that] I-picture.” (EX1006, 5:38-48; Fig. 4 (“yes” path annotated in blue).)

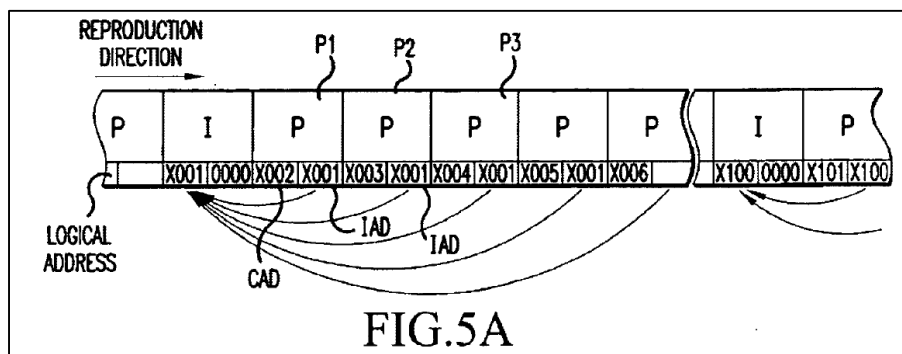


(EX1006, Fig. 4 (annotated).)

2. Claim 8

- a. **8[a]:** *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*

127. Figure 5A of Kim illustrates a logical structure of a digital data stream (EX1006, 2:48-51) that a POSITA would have understood to correspond to *an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.*

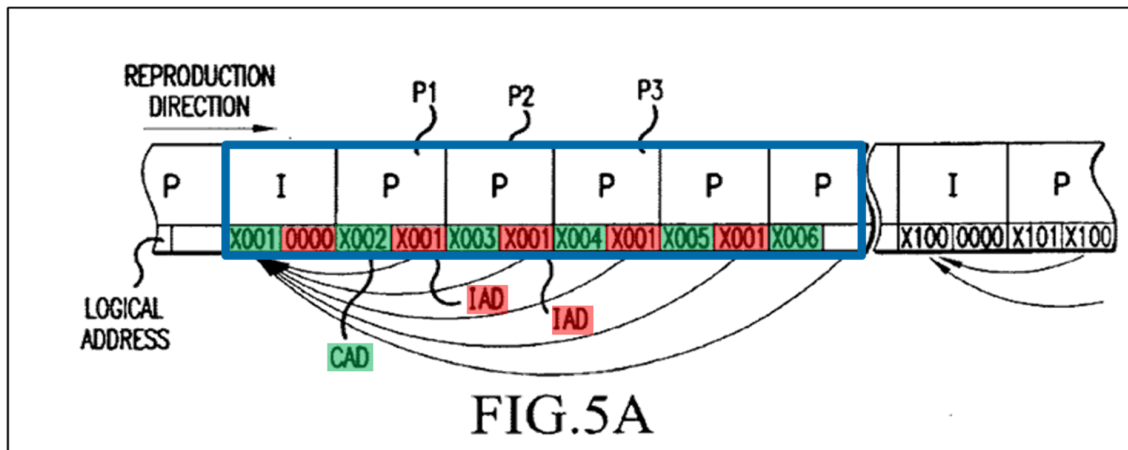


(EX1006, Fig. 5A.)

128. The bitstream includes intra-pictures (I-pictures) and predictive-pictures (P-pictures) (EX1006, Fig. 5A, 2:17-27), and while the I-pictures are coded based only on information within themselves (EX1006, 3:60-61), the P-pictures (also called “P-frames”) are coded with “the moving vector and the position value

of the corresponding (or previous) I-frame” (EX1006, 4:15-27). Kim’s I-picture followed by P-pictures is thus *an independent sequence*.

129. In Kim’s independent sequence, every P-picture that follows an I-picture depends on that I-picture for motion prediction. (EX1006, Fig. 5A, 4:46-52.) Accordingly, “each P-picture frame . . . has a current address CAD indicating the current address of the P-picture, and an I-picture reference address IAD indicating the address of the corresponding (or previous) I-frame.” (EX1006, 4:46-52.) Each P-picture’s IAD thus identifies the I-picture on which the P-picture depends for motion compensation. This is illustrated in Figure 5A, where each P-picture IAD provides the address of the preceding I-picture, and arrows from each P-picture to a preceding I-picture indicate dependencies. (EX1006, Fig. 5A (IAD in red; CAD in green; independent sequence in blue box).)

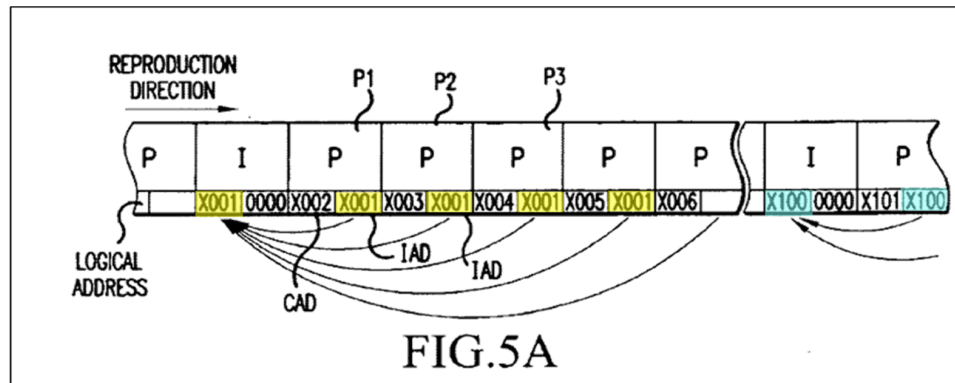


(EX1006, Fig. 5A (annotated).)

130. Given that each P-picture following an I-picture depends on that I-picture for motion prediction, all motion-compensated temporal prediction references shown in Figure 5A refer only to image frames within the sequence. A POSITA would therefore have understood Kim to disclose or at least suggest *an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within the independent sequence.*

131. A POSITA would have also understood an IAD of 0000 or zero in Figure 5A of Kim to be *an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.* Kim explains that each image frame in the bitstream has a CAD and an IAD (*see* EX1006, 4:46-52, 4:59-5:7, Figs. 5A, 6), and each I-frame in Figure 5A has an IAD of 0000. (*See* EX1006, 4:59-5:1 (describing “a data structure of the bit stream shown in FIG[. 5A” and explaining that for I-pictures “the address of the corresponding I-picture frame [(i.e., the IAD)] ... is *zero* or absent”).) On the other hand, the IAD for each P-picture is the CAD of the preceding I-picture. (EX1006, 4:48-52.) And because each I-picture begins an independent sequence, the IAD of 0000 indicates that a given frame starts the independent sequence. This is again illustrated in Figure 5A, where I have highlighted the CAD for the I-picture and IAD for the P-pictures of the

first independent sequence in yellow, and I have highlighted the CAD for the I-picture and IAD for the P-pictures of the second independent sequence in blue.



(EX1006, Fig. 5A (annotated).)

132. A POSITA would have further understood that Kim’s decoder decodes the IAD of image frames. Figure 5A is an “example[] of a logical data structure of a digital data stream” (EX1006, 2:48-51), and “FIG. 6 shows an example of a data structure of the bit stream shown in FIG[. 5A” (EX1006, 4:59-61.) Given that context, along with Kim’s explanation of the decoding process for the bitstream (EX1006, 5:8-6:8, Fig. 4), a POSITA would have understood that process to include decoding the components of the bitstream, which would include IAD values—namely the IAD value of 0000 for the I-frame in Figure 5A of Kim.

133. A POSITA would therefore have understood that Kim decoding the IAD value of 0000 for the I-picture and grouping of I-pictures followed by P-pictures that depend on the corresponding I-picture for each group to correspond to *decoding from the video sequence an indication of at least one image frame, which is the first*

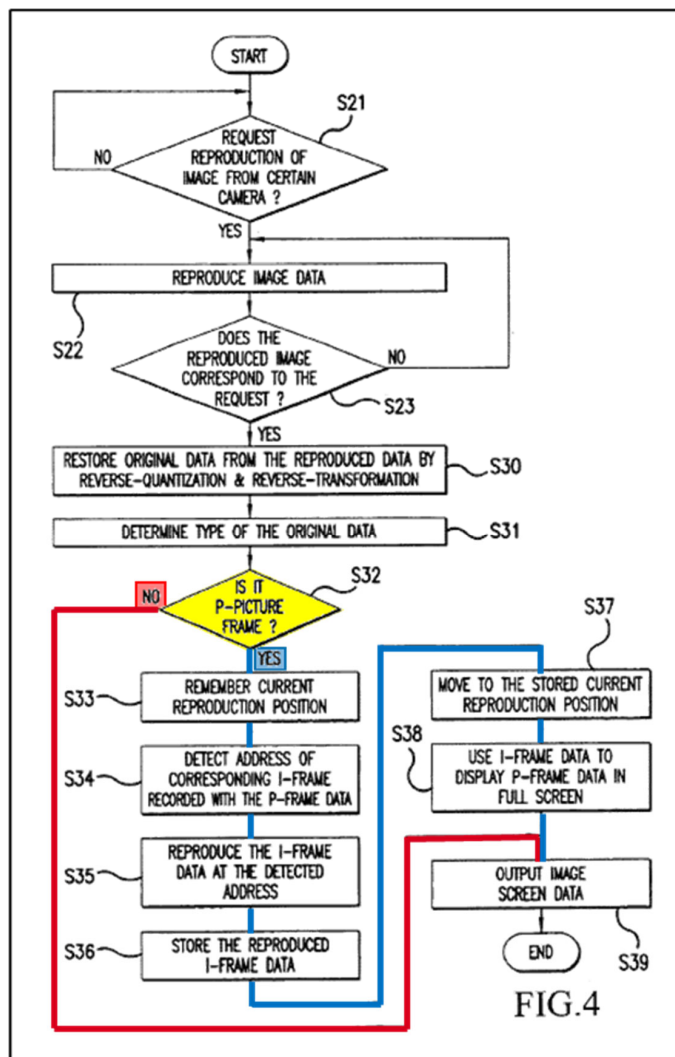
image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence.

134. I understand that in a separate proceeding involving the '321 patent, the Patent Owner argued that Kim does not render this limitation obvious because: (1) if a P-picture (not an I-picture) is selected first, it does not have an IAD of 0000 but is the first image frame in decoding order; (2) if an I-picture is selected first, the IAD is not decoded; and (3) Kim discloses searching an optical medium using a physical address rather than the logical address that includes the IAD. IPR2024-00691, Paper 20, 56-59. I disagree with all three arguments.

135. The first argument appears to interpret decoding order based on where the decoder starts decoding frames rather than where the decoder completes decoding frames. *See id.*, 56-57. That interpretation is not consistent with my reading of the '321 patent or with the plain-and-ordinary meaning of “decoding order” (or the synonym, “bitstream order”) to a POSITA—which is “the order which the decoder requires them to decode the bitstream.” (EX1005, 71.) As I explained above, Kim’s I-picture is decoded before the P-picture that depends on it, even if decoding the P-picture starts but cannot successfully finish. And a POSITA would have understood that the I-picture is the first image frame in decoding order, even when a P-picture is selected such that in the process flow, the decoder decodes the

IAD value 0000 (*indication*) when it decodes the I-picture on which the P-picture depends.

136. Patent Owner's second argument appears to misread Kim, which explains (with reference to Figure 4) that the decoder does not know until step S30—where image data is transmitted to the decoder—whether the frame is an I-picture or P-picture. If the image data identifies an I-picture, the IAD of 0000 is decoded. And if a P-picture is selected, the IAD of the I-picture it depends on is still decoded so that the P-picture can be fully decoded.

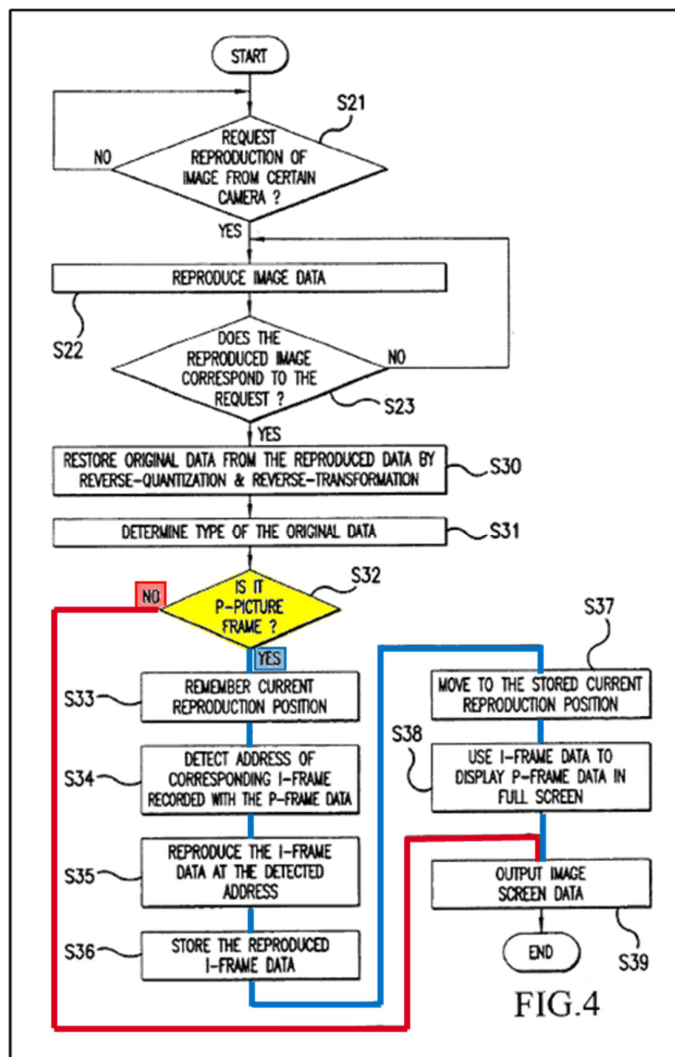


137. Finally, I disagree with Patent Owner's third argument because Kim must decode the logical address (IAD) before it can populate a data structure at a given physical address.

- b. **8[b]: *starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame***

138. As I explain above, a POSITA would have understood Kim's grouping of an I-picture followed by P-pictures to be an *independent sequence*. Furthermore, Kim begins the decoding of that grouping with the I-picture (*starting the decoding of the video sequence from said first image frame of the independent sequence*).

139. Figure 4 of Kim illustrates two paths for decoding an image in response to a user requesting playback, and a POSITA would have understood—or would have found it obvious—that Figure 4 applies to each image frame in a video sequence, including the sequence in Figure 5A.



(EX1006, Fig. 4 (annotated).)

140. In particular, a POSITA would have applied Kim's Figure 4 process to Figure 5A video data rather than a single image to enlarge the scope of input data types to which the process illustrated in Figure 4 could be applied. As an example, the flowchart in Figure 4 depicts a reproducing method for a data structure (e.g., the one shown in Figure 6 (EX1006, 6:8-10)), and Figure 6 depicts an example data structure of the bitstream in Figure 5A (EX1006, 4:59-60). A POSITA would have

therefore understood the decoding steps in Figure 4 could be repeated for each image in the independent sequence of Figure 5A. As I explained for Ground 1, it was well known before the '321 patent how to decode video sequences such as closed GOPs. A POSITA would therefore have found the use of Kim's Figure 4 process on the Figure 5A data structure to be no more than applying prior art elements according to their known functions to achieve predictable results.

141. The first path in Figure 4 of Kim (annotated in red above) illustrates that the first image decoded and output for display is an I-picture. (*See* EX1006, Fig. 4 (showing “no” branch at Step S32).) In Step S32, “if the detected image data is not a P-picture frame, then the detected image data is decoded by the decoder 111 as described above, and output as an analog image signal from the D/A converter 91 in Step S39.” (EX1006, 5:66-6:3.) A POSITA would have understood the phrase “not a P-picture” to mean that the picture is an I-picture since Kim includes only I- and P-pictures in the sequence. (*See, e.g.*, EX1006, Figs. 5A, 6.) Consequently, this path begins by decoding an I-picture, which corresponds to *starting the decoding of the video sequence from said first image frame of the independent sequence*.

142. The second path in Kim's Figure 4 (annotated in blue above) illustrates reproduction of a P-picture. (*See* EX1006, Fig. 4 (showing “yes” branch at Step S32).) Kim explains that, “if the currently restored image data is detected to be P-picture frame data in Step S32, the controller 150 stores the current reproduction

position in Step S33.” (EX1006, 5:38-41.) Kim next detects the address of the corresponding I-picture on which the P-picture depends for motion compensation, and then the decoder decodes the I-picture. (EX1006, 5:41-52.) Thereafter, Kim generates the P-picture as a full-screen image using the I-picture frame data and moving vector before outputting the reproduction signal. (EX1006, 5:55-65.)

143. A POSITA would have understood Kim’s second path to be *starting the decoding of the video sequence from said first image frame of the independent sequence* because Kim decodes the I-picture (*the first image frame of the independent sequence*) before it can produce the P-picture that depends on that I-picture. This behavior is consistent with the statement in the ’321 patent that a dependent frame “can no longer be correctly decoded” if the frame on which it depends is lost. (EX1001, 2:31-33.)

144. I noted above that a POSITA would have found it obvious to apply Kim’s Figure 4 process to each image in a video sequence, including the sequence in Figure 5A, but notably, Figure 4 contemplates a video sequence that includes an I-picture and a P-picture (because at Step 32, if the user requests reproduction of a P-picture, the system detects the address of the corresponding I-picture on which the P-picture depends for motion compensation and decodes that I-picture). (*See* EX1006, Fig. 4, 5:38-52.) Thus, even without applying Figure 4 to all pictures of an independent sequence in Figure 5A, a POSITA would have understood that

Figure 4 applies to an independent sequence with a P-picture and a corresponding I-picture (*starting the decoding of the video sequence from said first image frame of the independent sequence*).

145. Additionally, a POSITA would have understood that Kim's video sequence is decoded without prediction from any image frame before the first image frame. This is because every P-picture refers to the first I-frame in each independent sequence, so no P-picture requires prediction from an image frame decoded before the first I-picture.

146. Thus, either decoding path described in Kim's Figure 4 (when applied to the independent sequence of Figure 5A) or alternatively the blue "yes" path (when applied to an independent sequence with a P-picture and corresponding I-picture) would *start the decoding of the video sequence from said first image frame of the independent sequence*. And either path decodes the independent sequence *without prediction from any image frame decoded prior to said first image frame*.

147. I understand that in a separate proceeding relating to the '321 patent, the Patent Owner asserted that Kim does not render this limitation obvious because the decoder starts decoding the P-picture before it decodes the I-picture on which it depends. IPR2024-00691, Paper 20, 60-61. I disagree. As I explained above, a POSITA would have understood that Kim's I-picture is the first frame in decoding order, even if the user selects a P-picture.

148. I further understand that in a separate proceeding, the Patent Owner asserted that Kim does not render this limitation obvious because: (1) if the user selects an I-picture, no sequence is decoded; and (2) if Kim only decoded I-pictures, the IAD would not be unique. IPR2024-00691, Paper 20, 62. I disagree with both arguments. With reference to Kim’s Figure 5A—which shows one I-picture followed by P-pictures—if a user selects the I-picture, followed by P-pictures, Kim’s decoder would start decoding the sequence in Figure 5A with the first image frame (the I-picture).

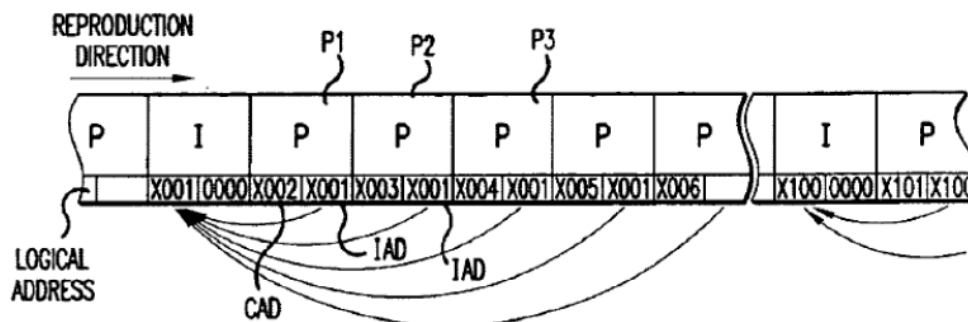


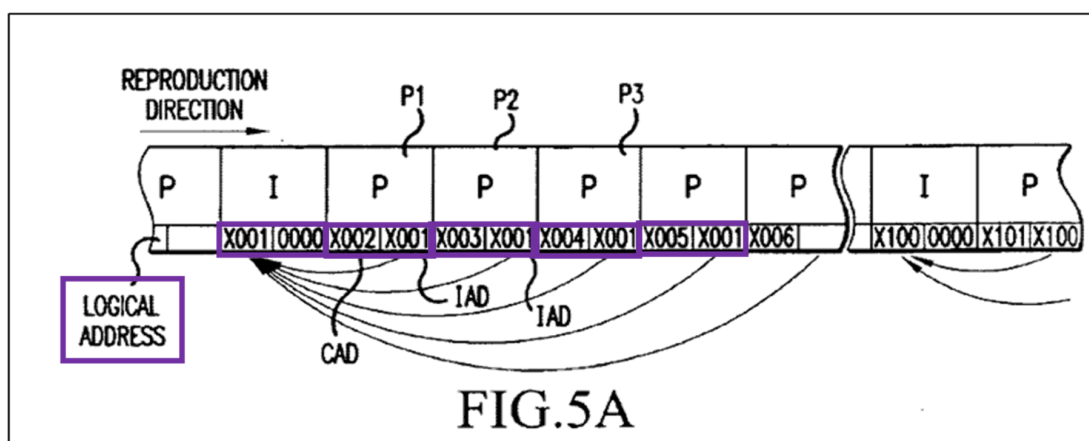
FIG.5A

c. 8[c]: *decoding identifier values for image frames according to a numbering scheme*

149. A POSITA would have understood Kim’s “logical addresses” to be *identifier values for image frames decoded according to a numbering scheme*.

150. Kim’s “logical addresses”—which, as shown in Figure 5A, include the picture’s IAD and CAD—are assigned according to a numbering scheme. The CAD component of the logical address provides the current address of a picture and

increments by 1 for each picture in reproduction order. (See EX1006, 4:48-52, Fig. 5A (the first I-picture has a CAD of 0001, the following P-picture has a CAD of 0002, and that CAD increments by 1 for each subsequent P-picture).) For I-pictures, the IAD component is zero, and for P-pictures, the IAD provides the CAD of the corresponding I-frame. (See EX1006, 4:48-52, Fig. 5A.) A POSITA would have therefore understood Kim to provide logical addresses that follow a numbering scheme in the bitstream.



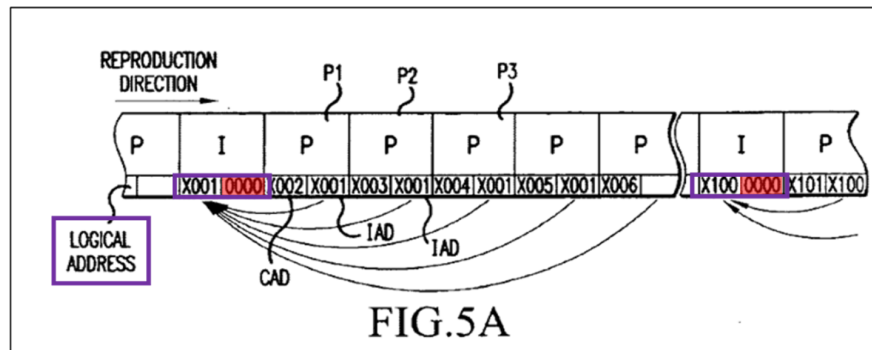
(EX1006, Fig. 5A (annotated).)

151. The logical addresses are also decoded according to the numbering scheme. For a P-picture, “the decoder detects the address of the corresponding I-picture frame ... and reproduces based on the detected address.” (EX1006, 5:41-46.) In this way, the decoder decodes and uses logical addresses to determine the order for decoding and reproducing pictures. Accordingly, Kim’s logical addresses are decoded according to a numbering scheme.

152. I understand that in a separate proceeding involving the '321 patent, the Patent Owner asserted that Kim does not render this limitation obvious because it teaches decoding physical addresses—which Patent Owner says “are not according to a numbering scheme”—rather than logical ones. IPR2024-00691, Paper 20, 63-65. This argument appears to be directed to the “optical disc” example from Kim. *Id.* (citing EX1006, 5:41-44). I disagree. As I explain above, Kim’s logical addresses (rather than physical addresses) correspond to the *identifier values*. Furthermore, a POSITA would have understood that Kim must decode the logical address in order to identify a corresponding physical address.

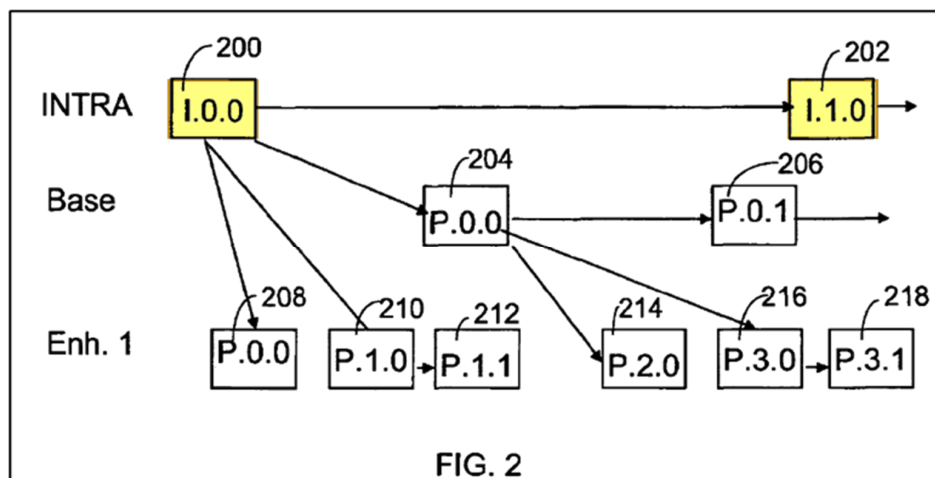
d. 8[d]: *resetting the identifier value for the indicated first image frame of the independent sequence*

153. Figure 5A of Kim shows the IAD component of each logical address reset to zero for an I-picture. (See EX1006, Fig. 5A (IAD of each I-picture logical address annotated in red and full logical address outlined in purple).) Each I-picture is the first image frame of an independent sequence, so a POSITA would have understood that the IAD component of each logical address (*identifier value*) is reset for the first image frame of the independent sequence.



(EX1006, Fig. 5A.)

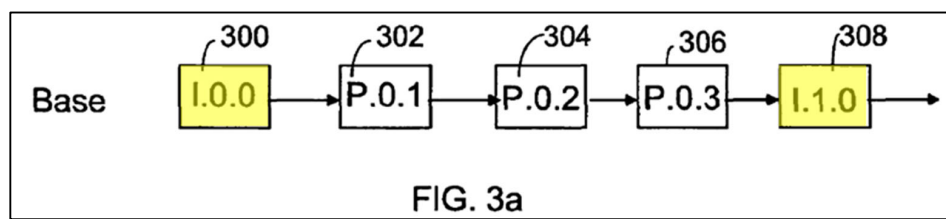
154. To the extent Patent Owner argues that resetting a component of an identifier value does not meet this limitation, I disagree because such a mapping is consistent with the '321 patent: the only two disclosures in the patent addressing more than one independent sequence (Figures 2 and 3a) illustrate resetting only a component of an identifier value, as highlighted in my annotations below.



(EX1001, Fig. 2 (annotated).)

155. In Figure 2, for example, I-frames 200 and 202 are first image frames of independent sequences. (EX1001, Fig. 2.) I-frame 200 has a type I and a

subsequence identifier and image number 0.0. (EX1001, 8:22-25.) I-frame 202 has a type I and subsequence identifier and image number 1.0. (EX1001, 8:36-40.) The identifier for I-frame 202 is thus I.1.0, which is a combination of the type, subsequence identifier, and image number. (EX1001, 8:41.) And the next I-frame would have an identifier of I.2.0. A POSITA would have therefore understood that only a component of the '321 patent's identifiers—the image number—resets; the subsequence identifier does not.

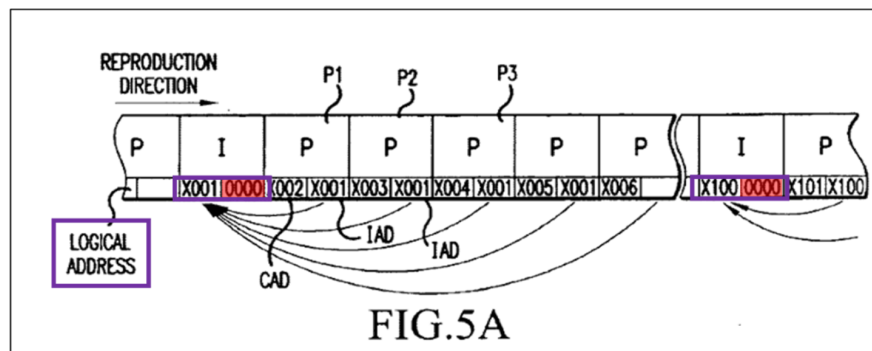


(EX1001, Fig. 3a (annotated).)

156. Figure 3a from the '321 patent further indicates that *resetting an identifier value* encompasses resetting a component of an identifier value. As the patent explains, “[t]he video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence” (EX1001, 12:24-26), and “a second subsequence” is initiated by “an I-frame (I.1.0) coded therein, i.e. an I-frame 308.” (EX1001, 12:31-34.) Here again, a POSITA would have understood that only a component of the identifier value is reset when a new subsequence is initiated—similar to the manner in which Kim *reset[s] an identifier value*.

157. I understand that in a separate proceeding involving the '321 patent, the Patent Owner asserted that Kim does not render this limitation obvious for two reasons: (1) Kim only discloses that its encoder resets the identifier value; and (2) Kim does not teach that its reproducing method resets the IAD. IPR2024-00691, Paper 20, 65-68. I disagree with both assertions.

158. The MUX of Kim's encoder outputs the address of I- and P-pictures in the bitstream (EX1006, 4:38-42), but that does not preclude Kim's decoder from resetting the IAD for each I-picture (*first image frame*). As explained above, Kim teaches that the bitstream transmitted to recording/reproducing unit 140 has a logical data structure shown in Figure 5A. (EX1006, 4:46-48.)



(EX1006, Fig.5A.) When Kim's decoder "reproduces" the bitstream, it resets the IAD component of each logical address for an I-picture to preserve the logical structure produced during encoding. If that were not the case, the decoder could not reproduce the original bitstream.

3. Claim 10

159. Claim 10 is substantively identical to claim 8, except instead of reciting *a method for decoding a compressed video sequence*, it recites *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence.*

160. Kim's Time Lapse reproduction device includes "a decoder 111 for performing a reverse-quantization and reverse-transformation of ... demultiplexed image data"; "a reproducing-DSP 121 for generating original image frames based on the P-picture data output from the decoder 111 and moving vector output from the de-MUX 131; and "a controller 150 for controlling the above-described components based on the commands from the user." (EX1006, 3:18-27, Fig. 2.)

161. Because a DSP⁶ is a type of processor that executes a stored program to perform signal processing operations, a POSITA would have understood that controller 150, decoder 111, and reproducing DSP 121 are *at least one processor and at least one memory including computer program code configured to, with the*

⁶ DSP in Kim stands for "digital signal processor." (EX1006, 3:7.)

at least one processor, cause the video decoders to decode a compressed video sequence.

162. Kim further describes encoding and then decoding MPEG image frames (EX1006, 4:15-19, 5:8-68). And as I explained previously, it would have been obvious to a POSITA to implement an MPEG-1 decoder as *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence.* It is therefore my opinion that for the same reasons, it would have been obvious to a POSITA to implement Kim using *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence.*

4. Claim 11

163. Claim 11 is substantively the same as claim 8, except instead of reciting *a method for decoding a compressed video sequence*, it recites *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.*

164. Kim's Time Lapse reproduction device includes "a decoder 111 for performing a reverse-quantization and reverse-transformation of ... demultiplexed

image data”; “a reproducing-DSP 121 for generating original image frames based on the P-picture data output from the decoder 111 and moving vector output from the de-MUX 131”; and “a controller 150 for controlling the above-described components based on the commands from the user.” (EX1006, 3:18-27.) A POSITA would have understood decoder 111, reproducing DSP 121, and controller 150 together or individually to be *data processing devices* because they all process image data or commands.

165. It would have been obvious to a POSITA to implement Kim’s decoder 111, reproducing DSP 121, and controller 150 as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. Decoder 111 is “for performing a reverse-quantization and reverse-transformation of the demultiplexed image data” (EX1006, 3:18-20), and reproducing DSP 121 is “for generating original image frames based on P-picture data output from the decoder 111 and the moving vector output from the de-MUX 131” (EX1006, 3:21-23), both using controller 150 to “control[] the above-described components based on the commands from the user.” (EX1006, 3:25-27).

166. A DSP is a special kind of processor that executes a stored program to perform signal processing operations, so a POSITA would have understood that DSP 121 in Kim executes a *computer program product, stored on a non-transitory*

computer readable medium and executable in a data processing device, for decoding a compressed video sequence.

167. It further would have been obvious to a POSITA to implement decoder 111 and/or controller 150 in either of two ways: (1) as a dedicated custom IC or (2) as a program running on a general-purpose processor available at the time. Either implementation would have been well-known and obvious to a POSITA and would have been well within the capabilities of the POSITA. It therefore would have been obvious to a POSITA to implement Kim's decoder 111, reproducing DSP 121, and controller 150 as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.*

168. A POSITA would have been motivated to implement Kim in this manner and would have expected to succeed because there were already well-known examples of real-time MPEG video software decoders running on general purpose processors. As one example, the University of California offered a freely downloadable, real-time software MPEG-1 decoder beginning no later than 1993 (*see generally* EX1010), and as another example, Hewlett-Packard implemented a real-time software MPEG-1 video decoder on workstations and PCs by no later than 1995 (EX1009, 60-68), with timing information for the video decoding functions in Table 1 (EX1009 at 63). A POSITA would have found it obvious to implement them

in computer program software on a computer processor such that, together with reproducing DSP 121 (also a computer processor), they would meet the limitation “[a] computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, the computer program product.” Finally, a POSITA would have understood that these computer program products would have been stored on a non-transitory computer readable medium so that they could be stored and then executed by the one or more processors to decode a video sequence “based on the commands from the user.” (EX1006, 3:25-27).

169. Moreover, Kim expressly encodes and decodes image frames using MPEG (EX1006, 4:15-19, 5:8-68). And as I explained previously, it would have been obvious to a POSITA to implement an MPEG-1 decoder—such as Kim’s decoder—as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*.

C. Ground 3: MPEG-1 and Yagasaki Render Obvious Claim 9

1. Yagasaki

170. Yagasaki describes the encoding and decoding of an image signal and identifies MPEG as a “highly efficient encoding system[] which rel[ies] upon the correlation of video signals.” (EX1015, 1:36-39.) Based on the date of Yagasaki—

it claims priority to foreign applications filed in 1993, very close in time to the release of MPEG-1—a POSITA would have understood it to refer to MPEG-1.

171. Yagasaki also describes an MPEG video signal as having a “video sequence ... divided into a frame group (GOP),” where each frame is classified as an I-picture, P-picture, or B-picture. (EX1015, 1:58-67.) Like MPEG-1, Yagasaki describes an I-picture as an “[i]ntra-coded frame and “the only type of frame that can be directly accessed and decoded” without reference to other frames. (EX1015, 1:63, 2:10-17.)

172. Yagasaki explains that hardware limitations make it difficult to “select a frame freely and reproduce it at a high speed” because P- and B-pictures “cannot be decoded in the absence of decoded images of past and/or future frames.” (EX1015, 2:12-15, 2:23-25.) Even I-pictures “can be reproduced at high speed ... only with rough motion.” (EX1015, 2:19-20.)

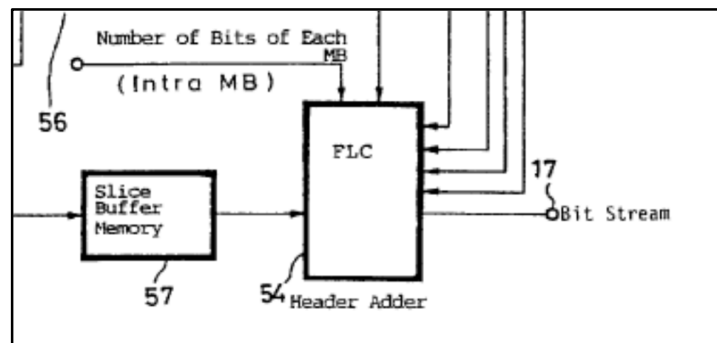
173. Yagasaki addresses these problems with a “decoding apparatus capable of reproducing a moving image smoothly at a high speed” by providing predictive-coded pictures (i.e., P- or B-pictures) with designated slices that “consist entirely of intra-coded macroblocks.” (EX1015, 2:29-32, 2:41-44.) “An indication of the slices designated is provided in the coded picture signals.” (EX1015, 2:44-45.)

2. **Claim 9: *A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.***

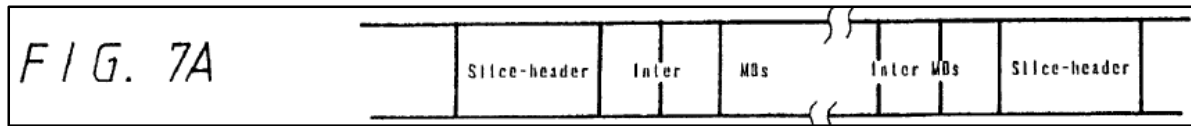
174. As I explained for Ground 1, a POSITA would have understood MPEG-1 to teach *a method according to claim 8*. Yagasaki teaches the rest of this claim, including decoding a slice header that contains a flag. (EX1015, 14:30-40.)

175. An image in Yagasaki contains multiple slices made up of “macroblocks,” each of which is “a unit range to be encoded.” (EX1015, 9:27-30.) “A number of macroblocks arranged in the order in which they are transmitted are called a slice.” (EX1015, 9:30-33.) A POSITA would have recognized that the slices Yagasaki describes correspond to slices in MPEG-1. (*See, e.g.*, EX1005, 26 (“slice [video]: A series of macroblocks. It is one of the layers of the coding syntax defined in [MPEG-1].”).)

176. Yagasaki’s slices further include headers. Yagasaki’s “hybrid encoder 102” includes a “fixed length coder” (FLC) 54 that Yagasaki refers to as a “header adder” that adds a header to image data in slice buffer memory 57. (EX1015, 4:56, 6:20-22.)

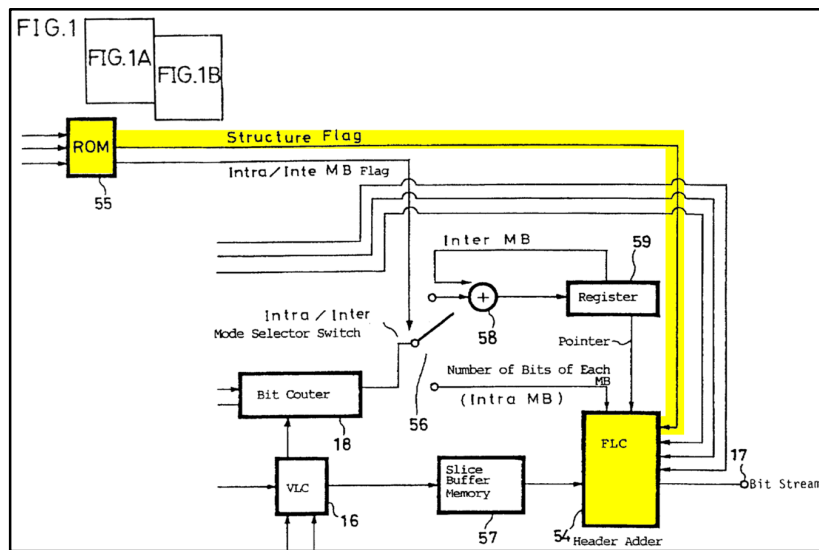


(EX1015, 6:15-24, 6:39-41, 7:3-4, Fig. 1B (excerpt).) The “header adder” is thus “for adding slice headers” (EX1015, 23:4-5), as illustrated in Figure 7A:



(EX1015, Fig. 7A.) A POSITA would have recognized that Yagasaki’s slice header corresponds to the slice header in MPEG-1. (EX1005, 91-92 (describing slice headers).).

177. Information added to the slice header includes “a flag representing a structure of a slice.” (EX1015, 9:55-61.) “[T]he ROM 55 outputs, to a macroblock mode selector switch 56, a flag which indicates whether a structure flag of the slice and the macroblock are an intra-frame coding macroblock for high-speed reproduction.” (EX1015, 6:44-50.) “[T]he intra-frame coded macroblocks can be extracted using only the flag information in the slice header, without decoding the entire bit stream.” (EX1015, 10:67-11:3.) ROM 55 passes the structure flag to FLC 54, which then “adds information to a slice header.” (EX1015, 7:1-4.)



(EX1015, Figs. 1A, 1B (annotated excerpt).)

178. A POSITA would have understood Yagasaki’s separate flag to “indicat[e] [] whether there is an intra-frame coded macroblock to be decoded within any particular slice upon reproduction at a high speed in both forward and reverse directions, and is also indicative of the structure of such a macroblock if it exists.” (EX1015, 10:1-5.) The slice may be “entirely composed of intra-frame coded macroblocks.” (EX1015, 10:14-15.) A POSITA would have recognized that a frame composed of such slices functions as the beginning of an independent sequence. (EX1015, 2:7-9 (I-picture is intra-coded in its entirety).)

179. Yagasaki describes decoding a compressed video sequence with separate flags in the slice header (which is part of a frame) to identify when a slice is composed entirely of intra-frame coded macroblocks. (EX1015, 10:26.)

180. Based on this description, it would have been obvious to a POSITA to put flags indicating intra-coding status in slice headers. Moreover, because a fully intra-coded frame is the beginning of an independent sequence, it would have been obvious that MPEG-1 flags indicating this same characteristic (closed_GOP or Temporal Reference) could be included in a slice header.

3. Motivation to Combine MPEG-1 and Yagasaki

181. A POSITA would have considered implementing Yagasaki with MPEG-1 encoding and decoding methods. Yagasaki's video coding methods could be performed in an MPEG-1 system with predictable results, as Yagasaki expressly suggests when it states that the "basic coding process in the present invention is the same as the encoding process in MPEG." (EX1015, 7:5-6.) And Yagasaki even describes the same GOPs and I-pictures as MPEG-1. (EX1015, 1:58-2:9; EX1005, 71.) Because I-pictures are entirely intra-coded, they contain slices with only intra-coded macroblocks. This is the only kind of frame that can be directly accessed and independently decoded. (EX1015, 2:7-18; EX1005, 23, 71.) Such slices can be decoded in an MPEG-1 system. (EX1015, 17:32-36 (slices "with all intra macroblocks ... can be decoded by the normal MPEG decoding process").)

182. Yagasaki also recognizes the value of efficient video coding to reduce recording size. (EX1015, 1:31-36.) Frames cannot all be entirely intra-coded I-pictures since they would require too much time to process. (EX1015, 2:16-28.) But

as Yagasaki notes, intra-frame coded regions are important for decoding, must “cover at least the entire image area” in one GOP, and should not “leave gaps on the image area.” (EX1015, 8:22-28.) A POSITA would have understood that this is because non-intra-frame coded regions lack sufficient data to produce a complete image.

183. For “high-speed reproduction,” Yagasaki outputs only slices with intra-frame coded macroblocks to a buffer while discarding others. (EX1015, 14:30-40.) A POSITA would therefore have recognized that Yagasaki’s indication of the intra-coding status for slices—using separate flags in the slice headers, as discussed above—allows for efficient decoding.

184. Furthermore, a POSITA would have recognized that Yagasaki’s indication could be used to provide other indications of intra-coding status for other image regions, including closed_GOP or Temporal Reference. This would improve decoding efficiency by identifying for the decoder the intra-coded image regions that provide the underlying image data from which the decoder may derive the remaining images of a GOP.

185. And a POSITA would have found it obvious to include such indications of intra-coding status (from MPEG-1) in the slice header because it is obvious to try. There are only a finite number of header locations for such indications, and the slice header is one of those few locations.

186. A POSITA would also have had a reasonable expectation of success in placing indications of intra-coding status in the slice headers because Yagasaki expressly teaches it (as I discuss above), and MPEG-1 decoding also involves successfully decoding such status indicators (as discussed above for Ground 1).

X. CONCLUSION

187. I currently hold the opinions expressed in this Declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

188. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct to the best of my knowledge. I understand that willful false statements are punishable by fine or imprisonment or both. *See* 18 U.S.C. § 1001.

Date: June 4th, 2025

By:

A handwritten signature in blue ink, appearing to read 'Eli Saber', written over a horizontal line.

Eli Saber, PhD