

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and AMAZON.COM SERVICES LLC,

PETITIONERS

v.

NOKIA TECHNOLOGIES OY,

PATENT OWNER.

IPR2024-00691

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

**DECLARATION OF DR. JOSEPH P. HAVLICEK
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
	A. Educational Background.....	2
	B. Professional Experience.....	2
	C. Publications.....	6
III.	LEVEL OF ORDINARY SKILL IN THE ART	8
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	15
VII.	OVERVIEW OF THE '321 PATENT.....	16
	A. Technology Overview.....	16
	B. The Challenged Claims.....	20
	C. '321 Patent Prosecution History	24
VIII.	CLAIM CONSTRUCTION	25
IX.	DETAILED EXPLANATION OF GROUNDS	26

A.	Ground 1: MPEG-1 Renders Obvious Claims 8, 10, and 11.....	26
1.	MPEG-1	26
2.	Claim 8.....	35
3.	Claim 10.....	45
4.	Claim 11.....	51
B.	Ground 2: Kim Renders Obvious Claims 8, 10, and 11	57
1.	Kim	57
2.	Claim 8.....	60
3.	Claim 10.....	75
4.	Claim 11.....	78
C.	Ground 3: MPEG-1 and Yagasaki Render Obvious Claim 9.....	85
1.	Yagasaki	85
2.	Claim 9: A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.	86
3.	Motivation to Combine MPEG-1 and Yagasaki	90
X.	CONCLUSION	92

I, **Dr. Joseph P. Havlicek**, 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 EX1004. The following is a brief summary of my relevant qualifications and professional experience.

A. Educational Background

8. I received a Bachelor of Science Degree in electrical engineering with minors in mathematics and computer science from Virginia Tech in 1986. I also received a Master of Science degree in electrical engineering from Virginia Tech in 1988. I received a Ph.D. in Electrical and Computer Engineering from the University of Texas at Austin in 1996. My Ph.D. research was in the field of image processing.

B. Professional Experience

9. From December 1984 to May 1987, I was a software engineer at Management Systems Laboratories, Blacksburg, VA. My main job responsibilities

included developing software for nuclear materials management under contract with the United States Department of Energy.

10. From June 1987 to January 1997, I was an electrical engineer at the United States Naval Research Laboratory. For the period of June 1987 through August 1989, I was an on-site contractor affiliated with SFA, Inc., Landover, Maryland. From August 1989 through January 1997, I was a regular government employee. I was on leave without pay from August 1987 through July 1988 while completing my Master of Science degree. I was also on leave without pay for much of the period from August 1990 through January 1997 while I completed my Ph.D. My main job responsibilities at the United States Naval Research Laboratory included designing digital and analog circuits to process real-time video signals and designing and implementing target detection, tracking, and identification algorithms. I developed missile warning receivers for Navy aircraft including the Navy's first two-color infrared system. I was a recipient of the 1990 Department of the Navy Award of Merit for Group Achievement for this work. The production version of this system was deployed on Navy helicopters and saw combat in Afghanistan and Iraq.

11. From January 1993 through December 1993, I was an on-site contractor at International Business Machines (IBM) Corporation, Austin, TX. My main job responsibilities included designing and implementing image compression and decompression algorithms (CODECs) for IBM products.

12. Since January 1997, I have been a regular faculty member in the School of Electrical and Computer Engineering at the University of Oklahoma, Norman, OK. I was an Assistant Professor from January 1997 through June 2002. I was promoted to the rank of Associate Professor and granted tenure in July 2002. I was promoted to the rank of Professor in July 2007. I was appointed to the Williams Companies Foundation Presidential Professorship in April 2009. In April 2017 I was appointed to the Gerald Tuma Presidential Professorship. My main job responsibilities at the University of Oklahoma include conducting academic research in electrical and computer engineering, teaching graduate and undergraduate courses in electrical and computer engineering, and performing professional and institutional service.

13. I am a member of several professional societies and organizations, including the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Signal Processing Society, the IEEE Computer Society, and the IEEE Intelligent Transportation Society. I am a Senior Member of the IEEE. From November 2015 through February 2018, I served as a Senior Area Editor for the IEEE Transactions on Image Processing. I was formerly an Associate Editor for the IEEE Transactions on Image Processing from December 2010 through October 2015. I served as a Technical Area Chair for the IEEE International Conference on Image Processing in the area of Image & Video Analysis, Synthesis, and Retrieval (2012, 2013) and have served on the organizing committee of that conference (2007). I have also served as a Technical Area

Chair for the IEEE International Conference on Acoustics, Speech, and Signal Processing in the area of Image, Video, and Multidimensional Signal Processing (2012-2014).

14. For over 35 years, I have conducted research and taught classes in the field of image processing and image analysis. My main scholarly contributions have been in the areas of modulation domain image models and image processing (AM-FM image models), video target tracking, and distributed control of video networks for intelligent transportation systems. I have supervised or co-supervised 12 Ph.D. students to completion and I am currently supervising two Ph.D. students. I have been a member of 67 additional doctoral dissertation committees. I have supervised 27 M.S. students to completion, and I am currently supervising two additional M.S. students. I have served as a committee member on 68 additional M.S. committees.

15. I teach a variety of courses at the University of Oklahoma, including the required junior-level Signals and Systems course ECE 3793 (taught 21 times), the graduate level Digital Image Processing course ECE 5273 (taught 23 times), and the graduate level Digital Signal Processing course ECE 5213 (taught 17 times). I have received a number of teaching awards, including the University of Oklahoma College of Engineering Outstanding Faculty Advisor Award (2005-2006) and the University of Texas Engineering Foundation Award for Exemplary Engineering Teaching while Pursuing a Graduate Degree (1992).

16. Since joining the University of Oklahoma in January 1997, I have been Principal Investigator or Co-Principal Investigator on over 110 externally funded grants and contracts with a total value of \$27,061,653. My main research contributions have been in the areas of signal, image, and video processing, video target tracking, and intelligent transportation systems. I have been author or co-author on over 130 scholarly publications in these areas. I was a recipient of the 1990 Department of the Navy Award of Merit for Group Achievement for my work in video target tracking. My research group at the University of Oklahoma originated the Virtual Traffic Management Center concept featured in a December 2014 FHWA technical report (Guidelines for Virtual Transportation Management Center Development) and a November 2014 FHWA national webinar with the same title. I am co-founder and director of the University of Oklahoma Center for Intelligent Transportation Systems. Under my supervision, the Center has collaborated with the Oklahoma Department of Transportation since 1998 to design and implement the Oklahoma Statewide Intelligent Transportation System, including a geographically distributed video network that is currently deployed on major highways and interstates across the entire State of Oklahoma.

C. Publications

17. A complete listing of my publications is found in my curriculum vitae (EX1004). I highlight some of the publications relevant to the subject matter of the '321 patent below:

- J.P. Havlicek, T.N. Arian, H. Soltani, T. Przebinda, and M. Özaydın, “A preliminary case for Hirschman transform video coding,” in Proc. IEEE Southwest Symp. Image Anal. & Interp., Santa Fe, NM, Mar. 29-31, 2020, pp. 104-107.
- E. Vorakitolan, J.P. Havlicek, R.D. Barnes, and A.R. Stevenson, “Simple, Effective Rate Control for Video Distribution in Heterogeneous Intelligent Transportation System Networks,” in Proc. IEEE Southwest Symp. Image Anal. & Interp., San Diego, CA, Apr. 6-8, 2014, pp. 37-40.
- V. DeBrunner, J.P. Havlicek, T. Przebinda, and M. Özaydın, “Entropy-Based Uncertainty Measures for $L_2(R_n)$, $l_2(Z)$, and $l_2(Z/NZ)$ with a Hirschman Optimal Transform for $l_2(Z/NZ)$,” IEEE Trans. Signal Process., vol. 53, no. 8, pp. 2690-2699, Aug. 2005.
- O. Alkhouli, V. DeBrunner, and J. Havlicek, “Hirschman Optimal Transform (HOT) DFT Block LMS Algorithm,” in Adaptive Filtering, L. Garcia, ed., ISBN: 978-953-307-158-9, InTech, Sep. 2011, pp. 135-152.
- V. DeBrunner, M. Özaydın, T. Przebinda, and J. Havlicek, “The optimal solutions to the continuous- and discrete-time versions of the Hirschman uncertainty principle,” in Proc. IEEE Int’l. Conf. Acoust., Speech, Signal Process., Istanbul, Turkey, Jun. 5-9, 2000, vol. 1, pp. 81-84.

III. LEVEL OF ORDINARY SKILL IN THE ART

18. 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.

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

20. 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.

21. 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

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

23. 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. Joseph Havlicek
1004	Curriculum Vitae of Dr. Joseph Havlicek
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

Exhibit	Reference
1010	Ketan Patel, et al., Performance of a Software MPEG Video Decoder, Multimedia '93: Proceedings of the First ACM Int'l Conf. on Multimedia, August 2-6, 1993
1011	Original Complaint, Nokia Techs. Oy v. Amazon.com, Inc., No. 1:23-cv-01236 (D. Del. Oct. 31, 2023), ECF No. 1
1012	First Amended Complaint, Nokia Corp. v. Amazon.com, Inc., No. 1:23-cv-01232 (D. Del. Oct. 31, 2023), ECF No. 5
1013	Declaration of June Ann Munford
1014	U.S. Patent No. 7,894,521
1015	U.S. Patent No. 5,786,858 ("Yagasaki")

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

25. 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

26. 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.

27. 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

28. 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.

29. 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.

30. 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.

31. 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

32. 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

33. 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

34. 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.

35. 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.

36. 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.

37. 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 so as 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.

38. 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.

39. 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.

40. 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

41. For the reasons I discuss below, it is my opinion that claims 8-11 of the '321 patent are disclosed in the prior art, and are rendered obvious by the prior art.

VII. OVERVIEW OF THE '321 PATENT

A. Technology Overview

42. The '321 patent is generally concerned with encoding and decoding video sequences comprising independent sequences of image frames.¹ (EX1001, Abstract.) As its title “Grouping of Image Frames in Video Coding” suggests, a key concept discussed in the '321 patent is the grouping of these image frames. (EX1001.) The

¹ The challenged claims of the '321 patent use the term “image frames,” so I follow that convention here. The '321 patent uses the terms “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 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.”).) These terms are all ways to 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”).)

concept of grouping image frames into displayable sequences has existed 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.)

43. A specific type of GOP discussed in the ’321 patent is the “independently decodable [GOP]” or “independent sequence.” (EX1001, 11:26-30, Abstract.) As its name suggests, an independently decodable GOP is one containing pictures that can be decoded independently of information outside of the GOP. Not all GOPs are independently decodable. (See EX1001, 3:52-56.) This is because image frames may depend on information from other frames, either within the same GOP or outside of it. This dependency arises from the use of a compression technique called “temporal prediction,” which takes advantage of “temporal redundancy” across image frames.

44. Temporal redundancy concerns the similarities between image frames. (See EX1001, 2:1-3 (“[T]emporal redundancy refers to the changes taking place in specific image objects in subsequent image frames.”).) Because image frames in a

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

video sequence are typically captured in quick succession, many aspects of the scene being captured do not change, or change in predictable ways from one image frame to the next. These aspects are considered to be temporally redundant.

45. One way to take advantage of temporal redundancies is to encode only the changes that occur between image frames. This approach is called temporal prediction because information about one frame is used to predict information about another frame from a different time. Temporal prediction allows image frames to be encoded using less data because encoding does not have to include redundant information. Instead, the encoded information can contain only some information about the depicted scene and refer to another image frame for any missing information.

46. A specific type of temporal prediction mentioned in the '321 patent is “motion-compensated temporal prediction.” (*See* EX1001, 2:18-24.) Motion compensated-temporal prediction is accomplished by predicting 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”).)

47. When an image frame is encoded using temporal prediction, additional information outside of the encoded image frame is needed to decode it, namely, 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.”).)

48. 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. “Intra-frames” or “I-frames” do not depend on information from other image frames. (EX1001, 2:23-29.) “Inter-frames,” or “P-frames,” depend on information from previous I- or P-frames. (EX1001, 2:27-31.) “Bi-directionally predicted frames,” or “B-frames,” depend on both a preceding and succeeding reference frame. (EX1001, 2:43-48.)

49. As mentioned above, an independently decodable GOP or independent sequence is one that can be decoded without information from image frames in another GOP. In other words, an independently decodable GOP is one in which any image frames encoded using temporal prediction depend only on reference frames within that same independently decodable GOP.

50. Independently decodable GOPs existed long before the alleged invention of the ’321 patent. As explained in MPEG-1 at least as early as 1993, “[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.) To this end, MPEG-1 described ways to encode groups of pictures to ensure that they were independently decodable

and still 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

51. Petitioner’s Petition challenges (and in this Declaration I address) claims 8-11 of the ’321 patent. Claims 8, 10, and 11 are independent. Claim 9 depends from claim 8. I have reproduced the claims below:

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.

52. The Challenged Claims are directed to the grouping and numbering image frames. According to the '321 patent, this grouping and numbering helps 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 describes the problem as follows: “When 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.)

53. The '321 patent states that this problem 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

54. The '321 patent issued from U.S. Application No. 11/338,934, which was filed on January 25, 2006, and claims priority to a continuation application filed on November 29, 2002, and a foreign application filed on January 23, 2002. MPEG-1 and Kim were not 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 rejecting then-pending claim 11. (EX1002, 750-51.) Then-pending claim 11 recited “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 that 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 the Examiner’s finding with respect to Yagasaki and did not distinguish Yagasaki. Instead, the Applicant amended the parent of claim 11, then-

pending claim 10 and argued that amended claim 10 distinguished Apostolopoulos. (EX1002, 770-778.) Then-pending claim 11 ultimately issued as claim 9. (EX1002, 806.)

55. In the notice of allowance, the Examiner found that the prior art reference considered, 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.) As discussed in detail below, the references cited in this Declaration teach all of the limitations of the Challenged Claims, including the limitations the Examiner found lacking in the art applied during prosecution.

VIII. CLAIM CONSTRUCTION

56. I have given all the claim terms of the Challenged Claims their plain and ordinary meaning, as would have been understood by a POSITA, at the time of the alleged invention.

IX. DETAILED EXPLANATION OF GROUNDS

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

1. MPEG-1

a. Overview

57. 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.) When the Standard was approved in 1993, it consisted of four parts: Systems, Video, Audio, and Compliance Testing. (EX1005, 11.) ISO/IEC later published a fifth part, which consisted of 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>.)

58. Historically, the term “MPEG-1” referred to all parts of the Standard. But I understand that Petitioner’s Petition uses the term “MPEG-1” to specifically refer to the second part of the Standard (ISO/IEC 11172-2), which concerns video. (*See* EX1005, 11.) I adopt the Petition’s naming convention and refer to the second part of the Standard as MPEG-1.

59. MPEG-1 was developed by the Moving Pictures Expert Group in response 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.) It “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

60. MPEG-1 discloses five key concepts 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)

61. In MPEG-1, “[a] picture corresponds to a single frame of motion video or to a movie frame.” (EX1005, 71.) A “picture” in MPEG-1 is equivalent 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 noted above in footnote 1, the ’321 patent itself interchangeably uses the terms “pictures” and “image frames.” (*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.”).)

62. MPEG-1 describes three picture types that are relevant to this Petition: I-pictures, P-pictures, and B-pictures. (*See* EX1005, 71.)

63. First, 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).

64. Second, 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.”).)

65. Third, 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.)

66. Each picture has a header with certain information about the picture. (*See* EX1005, 87.) Included in that information is the picture’s picture coding type. (*See* EX1005, 88.)

ii. Picture Ordering

67. MPEG-1 discusses two “picture orderings”: “display order” and “bitstream order.” (*See* EX1005, 84.) The display order is “[t]he order in which the decoded pictures should be displayed.” (EX1005, 22, 87.) The bitstream order is the “the order in which pictures are transmitted” in the bitstream. (EX1005, 70.) Because the decoder decodes pictures in the order 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.”).)

68. As explained in MPEG-1, 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.) MPEG-1 explains that “[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.) For example, Figure D.3 shows that picture P3 is fourth in display order, while Figure D.4 shows that it is second

in decoding order. This is because pictures B1 and B2 depend on picture P3 to be decoded.

69. It is worth noting that the '321 patent uses the term “decoding order.” A POSITA would have understood the plain and ordinary meaning of “decoding order” to be the order in which pictures are decoded. MPEG-1’s “bitstream order” and the '321 patent’s “decoding order,” therefore, refer to the same picture ordering (which a POSITA would have understood is distinct from the “display order”). (EX1005, 20.)

iii. Groups of Pictures (“GOPs”)

70. A group of pictures (“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.)

71. 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.)

72. Because GOPs must end with an I- or a P-picture in display order, no picture in a GOP may depend on a reference picture in a subsequent GOP for motion compensation prediction. Neither the I- nor the P-pictures may 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 no B-picture in the GOP may 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.) Since GOPs must end with an I- or a P-picture, any B-picture may not refer to a picture beyond that final picture.

iv. Closed GOPs

73. 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.) Although MPEG-1 does not require that every GOP be independently decodable, MPEG-1 explains that “encoders can still construct groups of pictures which are independent of one another.” (EX1005, 85.)

74. MPEG-1 discloses the concept of a closed GOP, which is an independently decodable sequence of pictures. MPEG-1 explains that a closed GOP is a “group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures.” (EX1005, 43.) A closed GOP, therefore, “can be decoded without using decoded pictures of the previous group for motion compensation.”³ (EX1005, 85.) And

³ As I have already noted above in ¶ 72, it is forbidden in MPEG-1 for *any* picture in *any* GOP to be encoded with a motion vector pointing to a reference picture in a later GOP. Therefore, the additional requirement for a closed GOP to be “encoded without motion vectors pointing to the previous [GOP]” (EX1005, 42) implies that, in a closed

as I explained in the previous section, all GOPs can be decoded without using decoded pictures of a subsequent group for motion compensation. (*See* Section IX.A.1.b.iii.) The closed GOPs described in MPEG-1 are thus independently decodable sequences of pictures.

v. Temporal Reference

75. As explained above, each picture begins with a header with certain information about the picture. (EX1005, 87.) Included in this information is the picture’s “Temporal Reference.” (EX1005, 87.) The Temporal Reference is a “number which can be used to define the order in which the pictures must be displayed.” (EX1005, 87.) MPEG-1 specifies that “[t]he first picture, in display order, in each group must have Temporal Reference equal to zero,” and Temporal Reference is “incremented by one for each picture in the group.” (EX1005, 87, 44.) Temporal Reference thus gives the display order of pictures. (EX1005, 44 (“The temporal-reference is assigned (in sequence) to the pictures in display order.”).)

c. Exemplary Closed GOP

76. MPEG-1 provides multiple ways in which an encoder can “construct groups of pictures which are independent of one another”—i.e., closed GOPs. (*See*

GOP, it is forbidden for any picture to be encoded with a motion vector pointing to a reference picture in a different GOP. In other words, all temporal references within a closed GOP must be to other pictures in that same GOP.

EX1005, 85.) For example, “[o]ne way to do this is to omit any B-pictures preceding the first I-picture.” (EX1005, 85.) A GOP encoded in this manner 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 and this Declaration use the term “exemplary closed GOP” to refer to a GOP encoded in this manner.

77. MPEG-1 discloses multiple examples of GOPs that have been encoded without B-pictures preceding the first I-picture. For example, the following figure shows “[a] typical example of a closed group [of pictures]” presented in display order:

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)

(EX1005, 86.)

78. As I noted above, a feature of the exemplary closed GOP is that a single I-picture must be the first picture of the GOP 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 specifies that the first picture in decoding order of any GOP must be an I-picture. (EX1005, 84 (“*Property 1*. A group of pictures, in bitstream order, must start with an I-picture.”).) 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 specifies 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—necessarily has a Temporal Reference of zero.

79. This feature is shown in the following example in 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 indicated in the figure, the top row shows a series of pictures in an exemplary GOP in display order, while the bottom row shows the same exemplary GOP in decoding order. The numbers in the figure indicate the Temporal Reference values for each picture. 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[pre]: A method for decoding a compressed video sequence, the method comprising

80. In my opinion, MPEG-1 teaches a “method for decoding a compressed video sequence.”

81. MPEG-1 discloses a “compressed video sequence.” As MPEG-1 explains, it “specifies a coded representation that can be used for compressing video sequences to bitrates around 1.5 Mbit/s.” (EX1005, 12; *see also* EX1005, 68 (explaining that MPEG-1 “defines a format for compressed digital video”).) This coded representation of a compressed video sequence is called a “coded video bitstream.” (EX1005, 16 (“[MPEG-1] specifies a syntax for a coded video bitstream.”), 21 (defining coded bitstream as “[a] coded representation of a series of one or more pictures as specified in [MPEG-1]”), 69 (explaining that MPEG-1 describes the “structure of an [MPEG-1] bitstream” and “the coding operations that compress the video”).)

82. MPEG-1 further discloses a method for decoding this coded video bitstream (“compressed video sequence”). As MPEG-1 explains, “[t]o view . . . moving pictures, the decoder accesses the [MPEG-1] bitstream, and decodes it as described in [Section] D.7.” (EX1005, 69; *see also* EX1005, 15 (“The decoding process is defined by [MPEG-1].”), 22 (“decoding (process): The process defined in [MPEG-1] that reads an input coded bitstream and produces decoded pictures or audio samples.”), 14 (“[MPEG-1] specifies the syntax and semantics of the bitstream and the signal

processing in the decoder.”.) Section D.7 of MPEG-1 discloses the processes for decoding MPEG-1 video. (EX1005, 120-122.)

83. Accordingly, it is my opinion that MPEG-1 teaches a “method for decoding a compressed video sequence.”

- b. 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**

84. In my opinion, MPEG-1 teaches this limitation. Below, I explain 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 will then explain how MPEG-1’s 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 will also explain how a Temporal Reference of zero teaches an alternative “indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence.” It is my opinion that either of these teachings, considered together with MPEG-1’s closed GOP, teaches the limitation as a whole.

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

85. 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.”

86. As I explained above in Section IX.A.1.b.iv, a closed GOP is decoded without using decoded pictures of the previous or subsequent group for motion compensation. This is because a closed GOP is defined to be a “group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures.” (EX1005, 43.) And all closed GOPs “must end with an I or a P-picture” and therefore cannot depend on pictures from a subsequent group of pictures for motion compensation. (See EX1005, 84, 23 (explaining that an I-picture is “[a] picture coded using information only from itself”), 25 (explaining that a P-picture is “[a] picture that is coded using motion compensation prediction from the past reference picture”), 32 (explaining that “[t]he 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”).)

87. Accordingly, MPEG-1’s closed GOP is “an independent sequence” (i.e., it does not depend on other sequences), wherein all motion vectors (“motion-compensated temporal prediction references”) point only to image frames within that closed GOP

and not to image frames outside of the closed GOP (“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”

88. MPEG-1’s `closed_gop` flag teaches the remainder of 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 [the] independent sequence.”

89. In an MPEG-1 bitstream, “a group of pictures begins with a [GOP] header.” (EX1005, 85.) The GOP header begins with a “`group_start_code`,” which “identifies the beginning of a group of pictures.” (EX1005, 43, 85.)

90. The GOP header contains a `closed_gop` flag that indicates whether the GOP to which it corresponds is open or closed. (EX1005, 43, 86.) As explained above in Section IX.A.2.b.ii, a closed GOP teaches an “independent sequence.” Accordingly, when the flag indicates that the GOP is closed, it indicates that the GOP is an independent sequence.

91. The GOP header precedes first picture in the GOP in decoding order. (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”).) Thus, the GOP header is an indication of the first image frame in decoding order of the GOP because it precedes that image frame.

92. 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 of the GOP. (See EX1005, 120 (describing the decoding process for forward playback).)

93. When the `closed_gop` flag indicates that the GOP is closed, it indicates that the next image frame decoded is the first image frame in decoding order of an independent sequence. Thus, when the `closed_gop` flag indicates that the GOP is closed, the `closed_gop` flag is “an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.”

94. MPEG-1 discloses decoding this indication because it discloses decoding the GOP header including 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”).)

95. Alternatively, a Temporal Reference of zero in MPEG-1 teaches the remainder of 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 [the] independent sequence.” As discussed above in Section IX.A.1.c, MPEG-1 discloses an exemplary closed GOP that is encoded without B-pictures preceding the first I-picture in display order. As further explained in that Section, the first picture in decoding order of the exemplary closed GOP will always be an I-frame with a Temporal Reference of zero (i.e., because that I-frame is the first frame in display order and so has a Temporal

Reference of zero). Thus, 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."

96. As I discuss in more detail below for limitation 8[c], MPEG-1 discloses that the decoder decodes Temporal Reference. (*See* EX1005, 120 (disclosing that picture header gets decoded during forward playback).)

97. Accordingly, 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 an independent sequence" through its disclosure of decoding the closed_gop flag or alternatively through its disclosure of decoding the Temporal Reference of zero for the first I-frame of the closed GOP.

98. It is therefore my opinion that, taken together, MPEG-1's disclosure of decoding a closed_gop flag or the Temporal Reference of zero for the first I-frame of the closed GOP and its disclosure of a closed GOP teach or suggest "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."

- c. **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**

99. In my opinion, MPEG teaches this limitation.

100. MPEG-1 teaches starting the decoding of a video sequence from the first picture of a closed GOP (i.e., “first image frame of the independent sequence”). MPEG-1 explains that “[i]f 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.) A closed GOP, like any GOP, begins with a group of pictures start code. (See EX1005, 43 (explaining that the group of pictures start code “identifies the beginning of a group of pictures”).) Thus, MPEG-1’s disclosure that a decoder should discard all bits until it finds the beginning of a GOP includes discarding all bits until it finds the beginning of a closed GOP. A POSITA would have understood this to teach starting the decoding of the video sequence from the first picture of the closed GOP (“independent sequence”) because a closed GOP is “a series of one or more coded pictures.” (See EX1005, 32.) In other words, when the found group of pictures start code is the start code of a closed GOP, MPEG-1 will “start the decoding of the video sequence” from the first I-picture that begins that GOP (i.e., “said first image frame of the independent sequence”).

101. MPEG-1 also teaches that “the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.” Specifically,

MPEG-1 discloses that closed GOPs are “decoded without using decoded pictures of the previous group for motion compensation.” (EX1005, 86.)

102. Accordingly, MPEG-1 teaches starting decoding with the first picture of a closed GOP (“starting the decoding of the video sequence from the first image frame of an independent sequence”) and that a closed GOP is decoded without using decoded pictures of the previous group for motion compensation (“decoded without prediction from any image frame decoded prior to the first image frame”).

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

103. In my opinion, MPEG-1 teaches this limitation.

104. MPEG-1 discloses that each picture has an identifier value called a Temporal Reference. (See EX1005, 87, 44.) The Temporal Reference “is a ten-bit number which can be used to define the order in which the pictures must be displayed.” (EX1005, 87, 44.)

105. MPEG-1 discloses that Temporal Reference is decoded. MPEG-1 discloses, for example, that during forward playback, after decoding the group of pictures header, the decoder “will decode the first picture header in the group of pictures.” (EX1005, 120.) Because the Temporal Reference is part of the picture header, the decoder decodes the Temporal Reference when it decodes the first picture header in the group of pictures. This is because decoding the header includes decoding the components of the header. Furthermore, a POSITA would have understood that

each picture in a GOP has a header (EX1005, 71) and that each of these headers would have been decoded as each picture was decoded. A POSITA therefore would have understood that each picture's Temporal Reference would have been decoded.

106. MPEG-1 also discloses that Temporal Reference values are decoded according to a numbering scheme. MPEG-1 explains that during the decoding process “slices and macroblocks in the picture are decoded and written into a display buffer” and then “displayed in the order defined by the temporal reference.” (EX1005, 119.) A POSITA would have understood that MPEG-1's disclosure that pictures are “displayed in the order defined by temporal reference” means that the decoder outputs pictures in the order of their Temporal Reference. MPEG-1 explains that pictures are assigned a Temporal Reference according to the following numbering scheme: “The 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 in which it should output the pictures, a POSITA would have understood that the pictures and their corresponding Temporal Reference values are decoded according to a numbering scheme.

107. Accordingly, MPEG-1 teaches “decoding identifier values for image frames according to a numbering scheme.”

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

108. In my opinion, MPEG-1 teaches this limitation.

109. As explained in Section IX.A.1.c, MPEG-1 discloses an exemplary closed GOP that begins with an I-frame in both decoding and display order. MPEG-1 discloses that “[f]or the earliest picture (in display order) in each group of pictures, the temporal-reference is reset to zero.” (EX1005, 44; *see also* EX1005, 87 (“The first picture, in display order, in each group must have Temporal Reference equal to zero.”).) As shown in the MPEG-1 figure below, the Temporal Reference is zero for the I-frame and then increments by one for 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 every exemplary closed GOP in a bitstream. A POSITA would therefore have understood that when the MPEG-1 decoder decodes the next exemplary closed GOP, the Temporal Reference for the first picture in decoding order of the exemplary closed GOP would be reset to zero, as it would be the first picture in display order for that GOP.

110. Thus, the Temporal Reference (“identifier value”) for the first picture in decoding order of every exemplary closed GOP (“the indicated first image frame of the independent sequence”) in a bitstream is reset to zero because that picture is also the first picture in the GOP in display order. Accordingly, MPEG-1 discloses “resetting the identifier value for the indicated first image frame of the independent sequence.”

111. For the reasons discussed above, it is my opinion that MPEG-1 teaches claim 8.

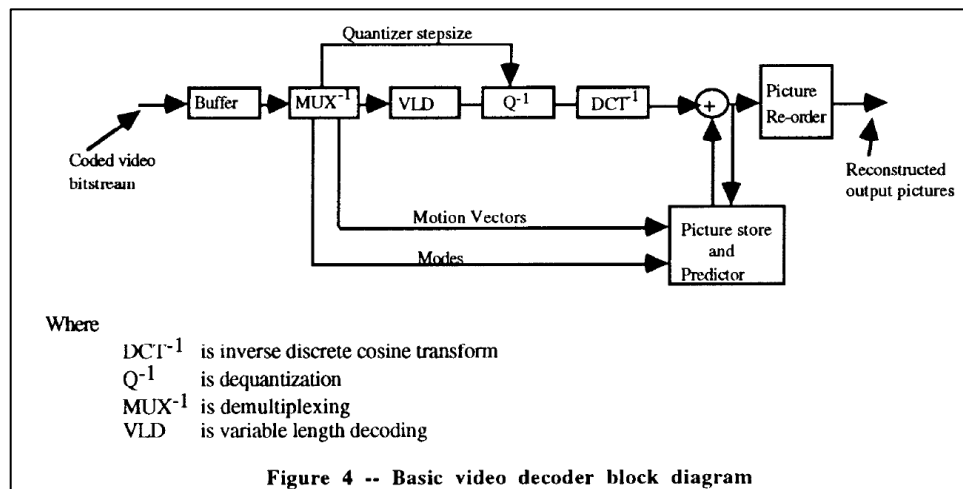
3. Claim 10

112. 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.” In my opinion, MPEG-1 teaches or suggests Claim 10.

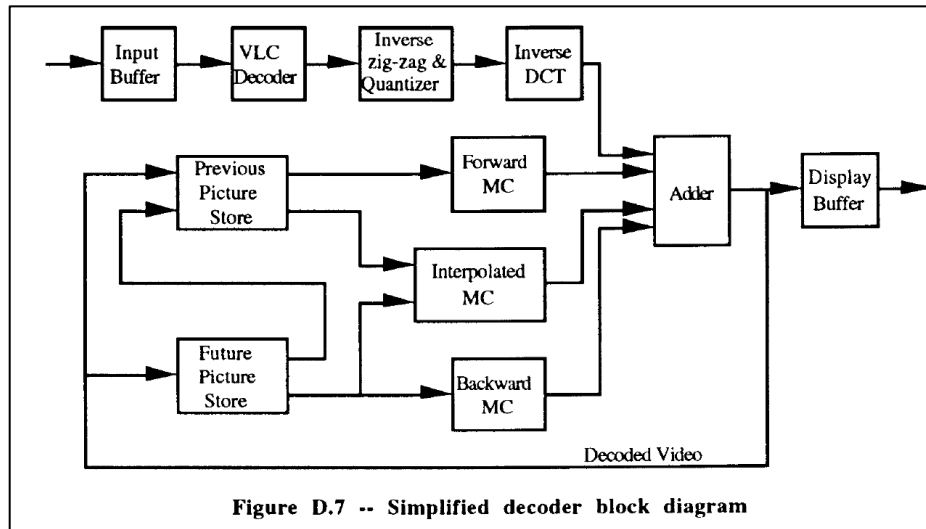
- a. 10[pre]: A video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence, by causing the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

113. In my opinion, MPEG-1 discloses the preamble of Claim 10. As discussed above in Section IX.A.2.a, MPEG-1 discloses “decoding a compressed video sequence.” MPEG-1 also discloses the additional elements of 10[pre].

114. MPEG-1 discloses multiple decoders. For example, MPEG-1 discloses:



(EX1005, 16.)



(EX1005, 73.)

115. In my opinion, a POSITA would have understood and found it obvious that an MPEG-1's decoders would have been implemented as a system comprising at least one processor and at least one memory including computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence based on their background knowledge at least by the time of the purported invention. As part of that background knowledge, a POSITA would have been aware of decoders with the recited hardware that existed before the priority date of the '321 patent.

116. One example of a decoder that would have been in a POSITA's background knowledge was C-Cube's CL480PC MPEG-1 audio/video decoder. (EX1007, 1.) The CL480PC decoded video using a combination of processors. (EX1007, 3.) One of the processors used code stored in a DRAM that was loaded into

on-chip memory. (EX1007, 3) Given that C-Cube's CL480PC was an operating MPEG-1 decoder, a POSITA would have had a reasonable expectation of success in implementing its configuration.

117. The C-Cube CL480, which was publicly available by 1997, was just one example of an MPEG-1 decoder that featured one or more processors implementing code stored in a DRAM for the decoding process. Indeed, by January 23, 2002, the date of the foreign patent to which the '321 patent claims priority, this configuration was a common and obvious configuration for an MPEG-1 decoder.

118. MPEG-1 itself also suggests implementing its decoder using the processor, memory, and computer program code of a personal computer. (EX1005, 12 (MPEG-1 "provides flexibility for use with personal computer and workstation displays").) Thus, a POSITA would have recognized that MPEG-1 suggests decoding compressed video using a processor in personal running computer code stored in memory.

119. Indeed, numerous disclosures predating the priority date of the '321 patent describe processors running software to decode MPEG-1 video. For example, U.S. Patent No. 7,263,273 ("Lapstun"), which was filed on November 25, 2000, discloses multiple ways to decode MPEG-1 video. (*See* EX1008, 6:56-65, 7:53-58.) One of these ways is to use a processor to decode the video in software. (EX1008, 7:56-57.) Likewise, 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.) As another example, in 1993, engineers implemented a decoder in 12,000 lines of C code using the X Window⁴ System (EX1010, 3) which a POSITA would have understood to comprise a processor and computer memory to implement the code.

120. In my opinion, the above 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.

121. It thus would have been obvious to a POSITA to implement any of MPEG-1’s disclosed decoders 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.”

⁴ Exhibit 1010 incorrectly uses the term “X Windowing System” instead of “X Window System,” which was also known as X11.

- b. 10[a]: 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**

122. In my opinion, this limitation is substantively identical to limitation 8[a].

See Section IX.A.2.b. MPEG-1 therefore teaches limitation 10[a] for the same reasons that it teaches 8[a].

- c. 10[b]: 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**

123. In my opinion, this limitation is substantively identical to limitation 8[b].

See Section IX.A.2.c. MPEG-1 therefore teaches limitation 10[b] for the same reasons that it teaches 8[b].

- d. 10[c]: decode identifier values for image frames according to a numbering scheme**

124. In my opinion, this limitation is substantively identical to limitation 8[c].

See Section IX.A.2.d. MPEG-1 therefore teaches limitation 10[c] for the same reasons that it teaches 8[c].

- e. 10[d]: reset the identifier value for the indicated first image frame of the independent sequence**

125. This limitation is substantively identical to limitation 8[d]. *See* Section IX.A.2.e. MPEG-1 therefore teaches limitation 10[d] for the same reasons that it teaches 8[d].

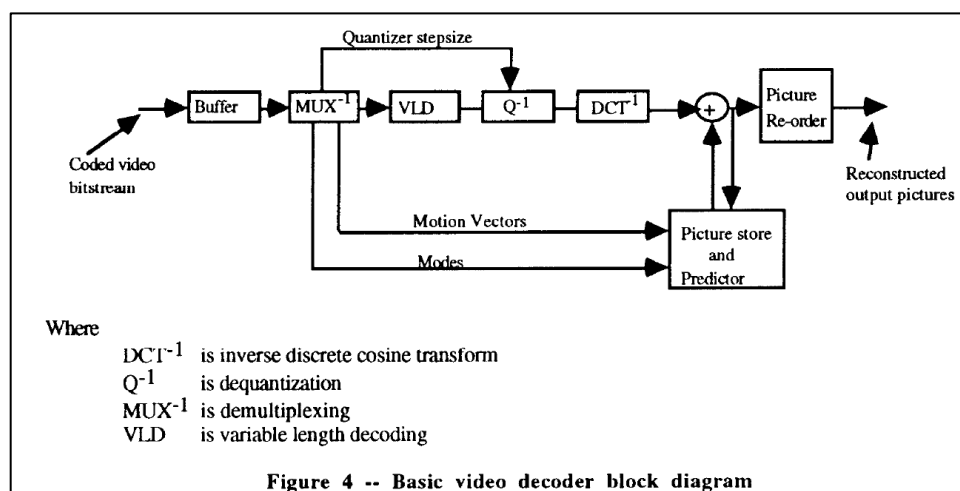
126. For the reasons discussed above, it is my opinion that MPEG-1 teaches or suggests claim 10.

4. Claim 11

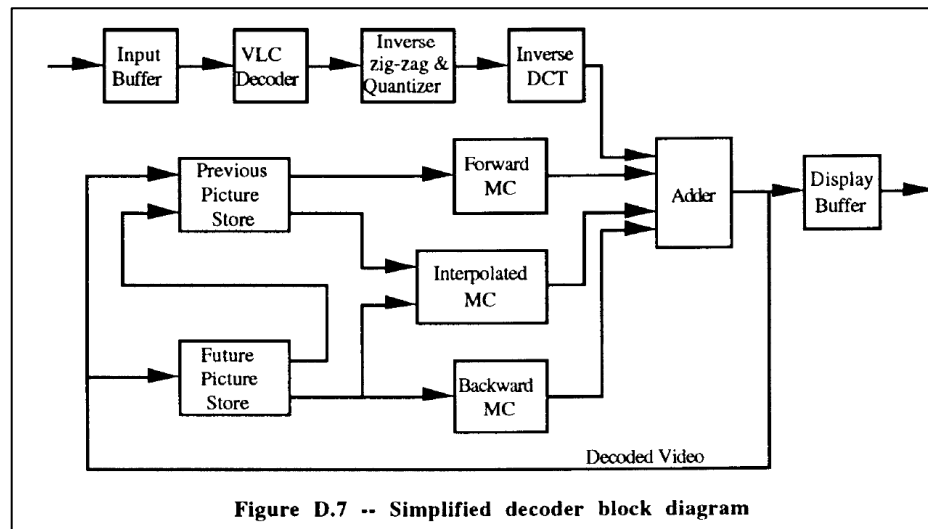
127. Claim 11 is substantively the same as claim 8, except instead of being written in terms of “[a] method for decoding a compressed video sequence” it is written in terms of “[a] computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.” In my opinion, MPEG-1 teaches or suggests Claim 11.

- a. **11[pre]: 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**

128. In my opinion, MPEG-1 teaches or suggests 11[pre]. As discussed above in Section IX.A.2.a, MPEG-1 teaches “decoding a compressed video sequence.” MPEG-1 further teaches the additional elements of 11[pre]. For example, MPEG-1 discloses multiple decoders:



(EX1005, 16.)



(EX1005, 73.)

129. A POSITA would have found it obvious to implement the disclosed video decoder in MPEG-1 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 in above Section IX.A.3.a, MPEG-1 decoders were well known in the art before the priority date of the '321 patent. For example, C-Cube's CL480PC MPEG-1 audio/video decoder decoded video using a combination of processors. (EX1007, 3.) One of the CL480PC's processors used microcode stored in a DRAM that was loaded into on-chip memory. (EX1007, 3.) A POSITA would have understood that this microcode was "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."

130. The C-Cube CL480, which was publicly available by 1997, was just one example of a system that used “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.” Indeed, by the priority date of the ’321 patent, a POSITA would have understood the configuration of C-Cube’s CL480PC to be a common and obvious configuration existing in the art.

131. MPEG-1 also suggests implementing its decoder as a “computer program product” using the processor, memory, and computer program code of a personal computer. (EX1005, 12 (MPEG-1 “provides flexibility for use with personal computer and workstation displays”).) A POSITA would have recognized that MPEG-1 suggests decoding a compressed video sequence using a computer program product on the personal computer itself. Because MPEG-1 discloses operations that would have routinely been carried out by computer program products available in personal computers at the time of the purported invention, a POSITA would have had a reasonable expectation of success in implementing this configuration.

132. Indeed, numerous disclosures predating the priority date of the ’321 patent describe software implementations of MPEG-1 decoders implemented as computer program products. For example, U.S. Patent No. 7,263,270, which was filed on November 25, 2000, discloses multiple ways to decode MPEG-1 video. (*See* EX1008, 6:59-65, 7:53-58.) One of these ways is to use “software” running on a processor.

(EX1008, 7:56-57.) Likewise, 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” and that it would have been “stored on a non-transitory computer readable medium and executable in a data processing device.” As another example, in 1993, engineers implemented a decoder in 12,000 lines of C code using the X Window System (EX1010, 3) which a POSITA would have understood to comprise a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.

133. Accordingly, it is my opinion that it would have been obvious to a POSITA to implement any of MPEG-1’s disclosed decoders 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.”

- b. 11[a]: 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**

134. In my opinion, this limitation is substantively identical to limitation 8[a], except that it is directed to “a computer program code for” decoding an indication rather than the act of decoding an indication. (*See* Section IX.A.2.b.) As explained in Section IX.A.2.b, 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 an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence.” And as explained in Section IX.A.4.a, a POSITA would have recognized that the decoding would be carried out by “a computer program product,” which would include “computer program code” such as microcode or C-code. Accordingly, it is my opinion that MPEG-1 teaches or suggests this limitation.

- c. **11[b] 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**

135. In my opinion, this limitation is substantively identical to limitation 8[b], except that it is directed to “a computer program code for” starting the decoding rather than the act of starting the decoding. (See Section IX.A.2.c.) As explained in Section IX.A.2.c, MPEG-1 teaches “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.” And as explained in Section IX.A.4.a, a POSITA would have recognized that the decoding process would be carried out by “a computer program product,” which would include “computer program code” such as microcode or C-code. In my opinion, MPEG-1 therefore teaches or suggests this limitation.

d. 11[c] a computer program code for decoding identifier values for image frames according to a numbering scheme

136. In my opinion, this limitation is substantively identical to limitation 8[c], except that it is directed to “a computer code for” decoding identifier values rather than the act of decoding identifier values. (*See* Section IX.A.2.d.) As explained in Section IX.A.2.d, MPEG-1 teaches “decoding identifier values for image frames according to a numbering scheme.” And as explained in Section IX.A.4.a, a POSITA would have recognized that the decoding process would be carried out by “a computer program product,” which would include “computer program code” such as microcode or C-code. In my opinion, MPEG-1 therefore teaches or suggests this limitation.

e. 11[d] a computer program code for resetting the identifier value for the indicated first image frame of the independent sequence

137. In my opinion, this limitation is substantively identical to limitation 8[d], except that it is directed to “a computer program code for” resetting the identifier value rather than the act of resetting the identifier value. (*See* Section IX.A.2.e.) As explained in Section IX.A.2.e, MPEG-1 teaches “resetting the identifier value for the indicated first image frame of the independent sequence.” And as explained in Section IX.A.4.a, a POSITA would have recognized that the decoding process would be carried out by “a computer program product,” which would include “computer program code” such as microcode or C-code. In my opinion, MPEG-1 therefore teaches or suggests this limitation.

138. For the reasons discussed above, it is my opinion that MPEG-1 teaches or suggests claim 11.

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

1. Kim

139. Kim discloses “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.) Relevant to the Petition and this Declaration, Kim describes a “reproducing method” that involves decoding a bitstream of compressed video data in response to a user’s request and outputting an analog image signal. (EX1006, 5:8-17, 5:29-34, 5:55-6:8.) To support its reproducing method, Kim describes a bitstream with pictures grouped into a “logical structure” based on their dependencies. Kim depicts this logical structure in Figure 5A:

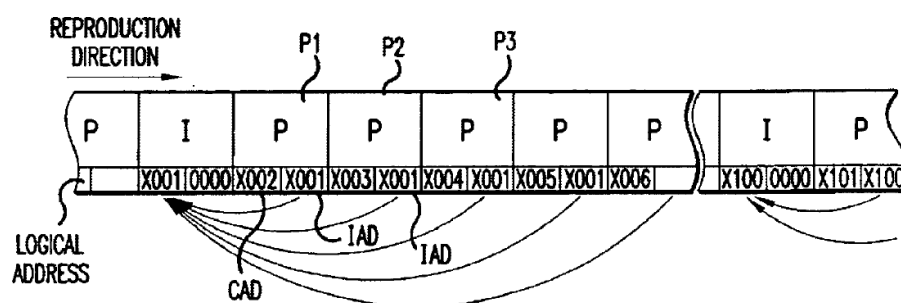


FIG.5A

(EX1006, Fig. 5A.)

140. The logical structure comprises an I-picture followed by P-pictures that depend on it. (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.) Specifically, in Kim, each I-picture is an “intra-picture” and therefore “carries the entire image information of a frame.” (EX1006, 3:60-61.) Kim’s I-pictures therefore do not depend on other pictures to be decoded and output as an analog image signal. In addition, in Kim, each P-picture is a “predictive-picture” (EX1006, 2:22) that is encoded based on “the moving vector and the position value of the corresponding (or previous) I-frame” (EX1006, 4:28-33). Kim’s P-pictures therefore require the data from their corresponding I-picture moving vectors to be restored as full screen images. (EX1006, 5:55-59.)

141. To further support its reproducing method, Kim discloses that the bitstream is comprised of data structures that associate each picture with address fields used to identify the picture and its corresponding reference picture. Kim depicts these data structures in Figure 6:

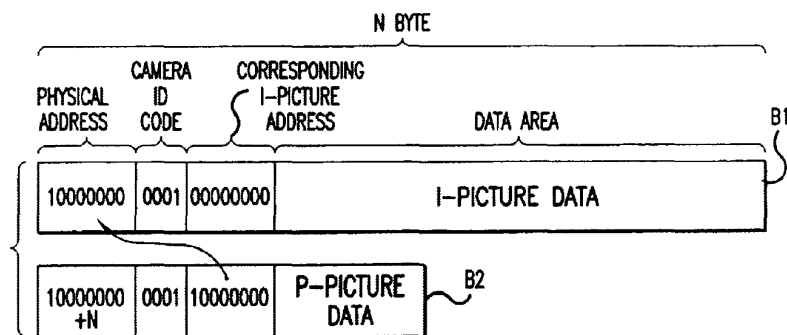
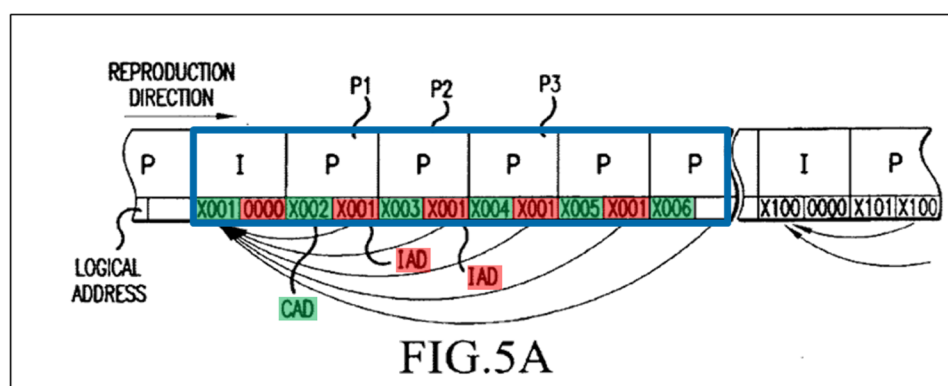


FIG.6

(EX1006, Fig. 6.)

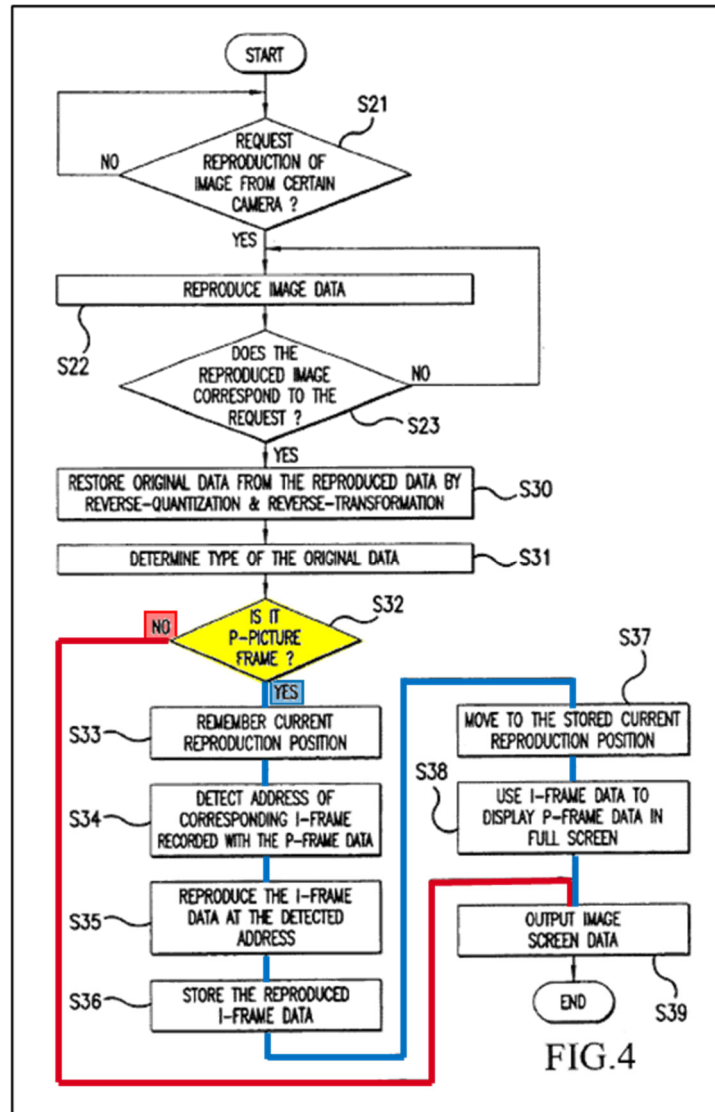
142. Kim explains that each P-picture has a current address (“CAD”) that indicates the current address of the P-picture, and an I-picture reference address (“IAD”)

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”)). Figure 5A below is annotated to highlight the IAD values in red and CAD values in green.



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

143. As shown below in Kim's Figure 4, Kim sets forth two paths for decoding the disclosed bitstream depending on whether the user requests image data that corresponds to 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).) When the request is directed 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

```
[pre]: A method for decoding a compressed video sequence, the method comprising
```

144. In my opinion, Kim teaches the preamble of claim 8.

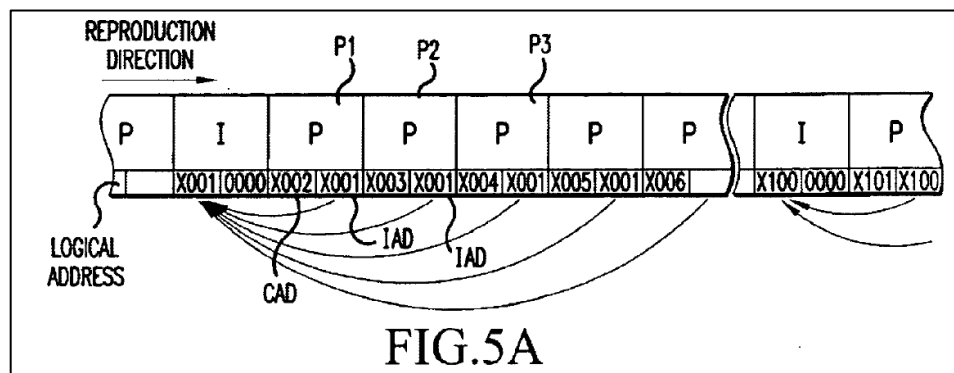
145. Kim is directed to “reproducing video data on and from an optical disk using a Time Lapse recording/reproduction process.” (EX1006, 1:11-12.) Kim “provide[s] a recording method for compressing and intermittently recording image

data on a recording medium with a maximum recording capacity.” (EX1006, 2:3-6.) Kim further “provide[s] a reproduction method for reproducing [the] compressed intermittent data from a recording medium.” (EX1006, 2:7-9.) A step in the reproduction method includes decoding image data in a decoder. (EX1006, 5:29-31.) Accordingly, Kim discloses a “method for decoding a compressed video sequence.”

- b. **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**

146. In my opinion, Kim teaches this limitation.

147. Kim’s Figure 5A (reproduced below) shows an example of a logical data structure of a digital data stream. (EX1006, 2:48-51.) In my opinion, this figure and corresponding disclosures teach the recited “indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.”

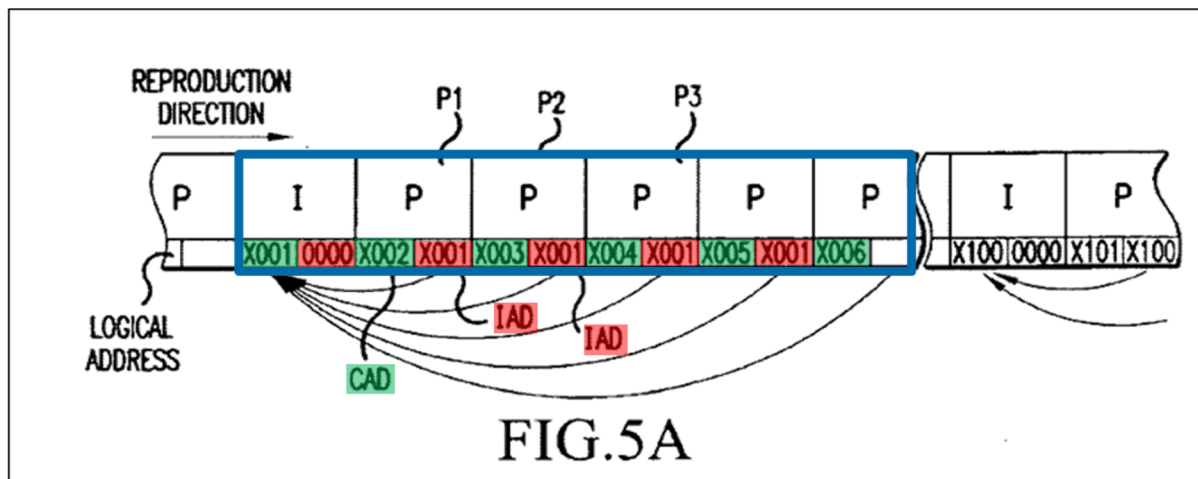


(EX1006, Fig. 5A.)

148. The bitstream disclosed in Kim includes intra-pictures (I-pictures) and predictive-pictures (P-pictures). (EX1006, Fig. 5A, 2:17-27.) Kim's I-pictures are coded based only on information within themselves. (EX1006, 3:60-61 ("The I-picture frame (intra-frame) data carries the entire image information of a frame.")) The P-pictures (also referred to in Kim as "P-frames") may be P-pictures similar to the P-pictures described in MPEG-1 and are coded with "the moving vector and the position value of the corresponding (or previous) I-frame." (EX1006, 4:15-27.) Kim's grouping of an I-picture followed by P-pictures that refer to it teaches the recited "independent sequence."

149. Kim discloses an independent sequence in which every P-picture following an I-picture depends on that I-picture for motion prediction. (EX1006, Fig. 5A, 4:46-52.) Kim explains that "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.) Accordingly, Kim discloses that each P-picture's IAD identifies the I-picture on which the P-picture depends for motion compensation. This is further shown in annotated Figure 5A below, where every P-picture's IAD identifies the address of the preceding I-picture and shows these dependencies as arrows pointing from each P-picture back to the preceding I-picture. (EX1006, Fig. 5A (IAD highlighted in red; CAD highlighted

in green; independent sequence outlined by blue box).) Thus, every P-picture in Kim's bitstream depends on the preceding I-picture.

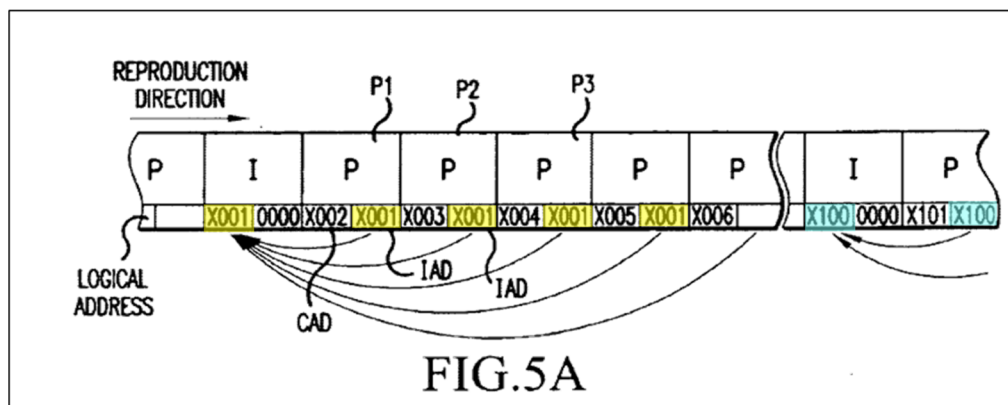


(EX1006, Fig. 5A (annotated).)

150. Because every P-picture following the I-picture depends on that I-picture for motion prediction in the sequence shown in Figure 5A, all motion-compensated temporal prediction references of the sequence shown in Figure 5A refer only to image frames within the sequence. It is therefore my opinion that Kim teaches “an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within the independent sequence.”

151. In my opinion, Kim also teaches the recited “indication.” Specifically, as shown in Kim's Figure 5A, an IAD of 0000 or zero is “an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.” As explained in Kim, each image frame in the bitstream has a CAD and an

IAD. (See EX1006, 4:46-52, 4:59-5:7, Figs. 5A, 6.) Kim’s Figure 5A shows that each I-frame has an IAD of 0000. (See EX1006, Fig. 5A; see also 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”).) The IAD for each P-picture, on the other hand, is the CAD (current address) of the preceding I-picture. (EX1006, 4:48-52.) Because each I-picture starts an independent sequence, the IAD of 0000 indicates that a given frame starts an independent sequence (i.e., “is the first image frame, in decoding order, of an independent sequence”). Annotated Figure 5A below shows the CAD for the I-picture and IAD for the P-pictures of the first independent sequence highlighted in yellow and shows the CAD for the I-picture and IAD for the P-pictures of the second independent sequence highlighted in light blue.



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

152. A POSITA would have understood that the decoder disclosed in Kim decodes the IAD of the image frames. Kim discloses that Figure 5A provides an

“example[] of a logical data structure of a digital data stream” (EX1006, 2:48-51) and that “FIG. 6 shows an example of a data structure of the bit stream shown in FIG[. 5A” (EX1006, 4:59-61.) Kim then explains the process for decoding this bitstream. (*See* EX1006, 5:8-6:8, Fig. 4.) A POSITA would have understood that decoding the bitstream in Figure 5A would include decoding the components of that bitstream, including the IAD values.

153. Accordingly, Kim discloses limitation 8[a] by decoding the IAD value of 0000 for the I-frame in Kim’s Figure 5A.

154. It is therefore my opinion that, taken together, Kim’s disclosure of decoding the IAD value of 0000 for the I-picture and its disclosure of a grouping of I-pictures followed by P-pictures that depend on the corresponding I-picture for each group teach “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.”

- c. **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**

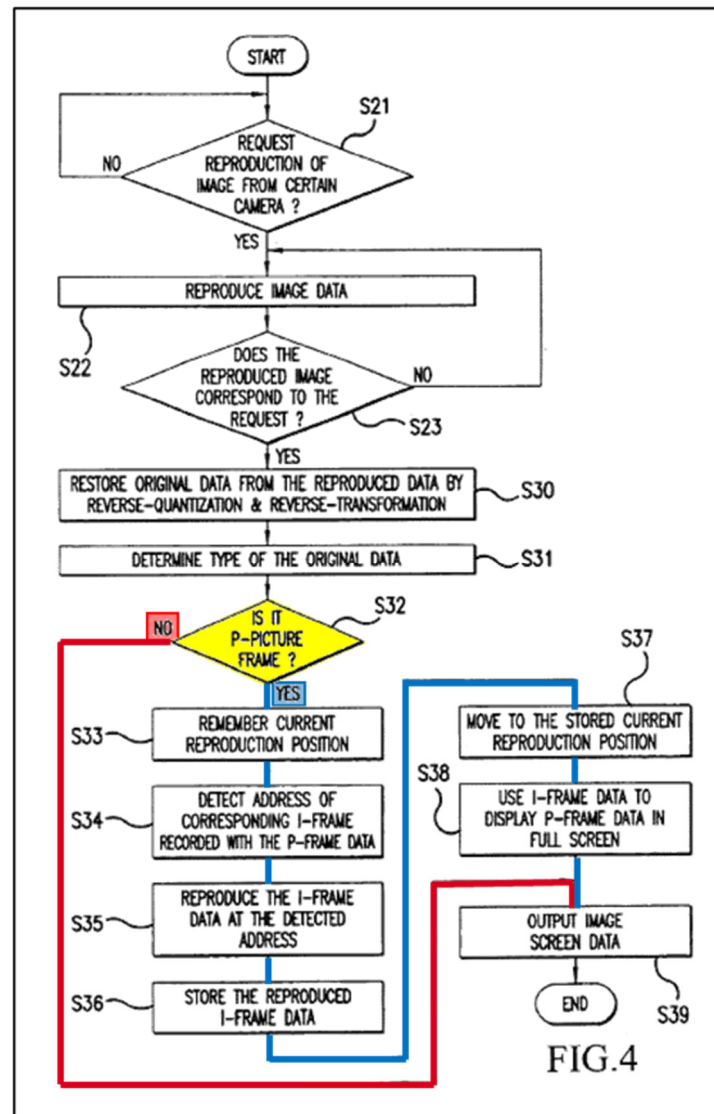
155. In my opinion, Kim teaches this limitation.

156. As discussed in Section IX.B.2.b for 8[a], Kim’s disclosures of a grouping of an I-picture followed by P-pictures teaches the recited “independent sequence.” As

to 8[b], Kim teaches that it starts 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”). Specifically, as shown in Figure 4, Kim discloses two paths for decoding an image in response to a user requesting playback. Either path discloses or suggests 8[b].

157. A POSITA would have understood that Kim’s Figure 4 process is applied, or would have found it obvious to apply Kim’s Figure 4 process, to each image frame in a video sequence, including the sequence in Kim’s Figure 5A. Specifically, a POSITA would have been motivated to apply Kim’s Figure 4 process to Kim’s Figure 5A video data rather than a single image, so that Kim’s Figure 4 process could be applied to more types of input data. For example, Kim discloses that the flowchart in Figure 4 depicts a reproducing method for a data structure, such as the one shown in Figure 6. (EX1006, 6:8-10.) Kim further discloses that Figure 6 shows an example of a data structure of the bitstream shown in Figure 5A. (EX1006, 4:59-60.) A POSITA therefore would have understood that the decoding steps in Figure 4 can be repeated for each image in an independent sequence shown in Figure 5A. As explained for Ground 1, decoding video sequences including closed GOPs was well known at the relevant time. A POSITA would have found this combination to be no more than applying prior art elements (Kim’s Figure 4 process to Kim’s Figure 5A data structure) according to their known functions to achieve predictable results.

158. Kim's Figure 4 is reproduced below:



(EX1006, Fig. 4 (annotated).)

159. Kim's first path discloses that the first image decoded and output for display is an I-picture. (See EX1006, Fig. 4 (showing "no" branch at Step S32 annotated in red).) Kim explains that "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.) Kim's use of the phrase "not a P-picture" means that the picture is an I-picture because Kim's sequence includes only I- and P-pictures. (*See, e.g.*, EX1006, Figs. 5A, 6.) As I explained previously, Kim's Figure 5A shows that each independent sequence begins with an I-picture. Accordingly, this path, which begins by decoding an I-picture, discloses "starting the decoding of the video sequence from said first image frame of the independent sequence."

160. The second path reflects the system's behavior when the user requests reproduction of a P-picture. (*See* EX1006, Fig. 4 (showing "yes" branch at Step S32 annotated in blue).) In the second path, "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.) The system then detects the address of the corresponding I-picture on which the P-picture depends for motion compensation. (EX1006, 5:41-48.) The system then decodes the I-picture. (EX1006, 5:49-52.) After the I-picture is decoded, the P-picture is restored as a full-screen image using the I-picture frame data and moving vector, and is ultimately output as a reproduction signal. (EX1006, 5:55-65.)

161. A POSITA would have understood this second path to disclose "starting the decoding of the video sequence from said first image frame of the independent sequence" because the system decodes the I-picture (the first image frame of the independent sequence) before it can produce the P-picture that depends on it. In other

words, a POSITA would have understood the P-picture to have been decoded only after the information on which it depends was available and used to decode the P-picture. This is consistent with the '321 patent's disclosure that a dependent frame "can no longer be correctly decoded" if the frame on which it depends is lost. (EX1001, 2:31-33.)

162. As I explained above, a POSITA would have found it obvious to apply Kim's Figure 4 process to each image frame in a video sequence, including the sequence in Kim's Figure 5A. But it is also worth noting in the alternative that Figure 4 itself 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 then 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 (showing "yes" branch at Step S32 annotated in blue), 5:38-52.) Thus, even without applying Figure 4's process to all of the pictures of an independent sequence in Figure 5A, a POSITA would have understood that Figure 4's process applies to an independent sequence consisting of a P-picture and a corresponding I-picture. This is sufficient to teach "starting the decoding of the video sequence from said first image frame of the independent sequence."

163. Kim also discloses that the video sequence is decoded without prediction from any image frame decoded prior to said first image frame, as recited in 8[b]. As I

explained in Section IX.B.2.b, each independent sequence in Kim's bitstream consists of an I-frame followed by a series of P-pictures that depend on the I-frame for prediction. (See EX1006, Fig. 5A.) It is my opinion that because every P-picture refers to the first I-frame in each independent sequence, no P-picture requires prediction from any image frame decoded prior to the first I-picture.

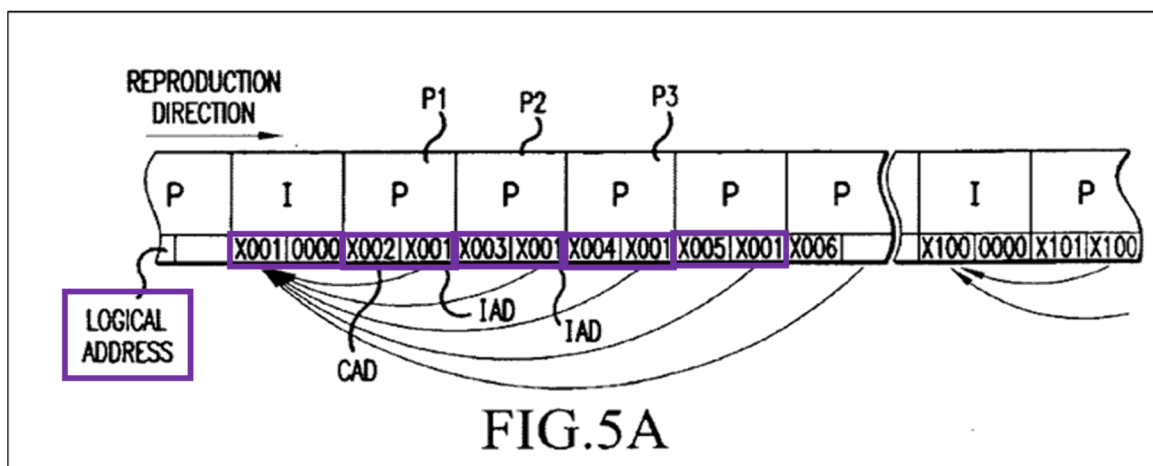
164. In view of the foregoing discussion, it is my opinion that Kim teaches "the video sequence is decoded without prediction from any image frame decoded prior to said first image frame."

165. In sum, either of the decoding paths described in Kim's Figure 4 when applied to Figure 5A's independent sequence, or alternatively the "yes" path from Step 32 in Kim's Figure 4 when applied to an independent sequence with a P-picture and corresponding I-picture, discloses "starting the decoding of the video sequence from said first image frame of the independent sequence," and under either path, Kim discloses decoding the independent sequence "without prediction from any image frame decoded prior to said first image frame."

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

166. In my opinion, Kim teaches this limitation. Specifically, Kim's "logical addresses" are "identifier values for image frames," and they are decoded "according to a numbering scheme."

167. Kim teaches that pictures have “logical addresses” that are assigned according to a numbering scheme. As shown in Kim’s Figure 5A, each picture has a “logical address” comprised of the picture’s IAD and CAD. (EX1006, Fig. 5A (annotated below with logical addresses outlined in purple).) For I-pictures and P-pictures, the CAD component of the logical address provides the current address of a picture. (See EX1006, 4:48-52.) CAD increments by 1 for each picture in reproduction order. (See EX1006, Fig. 5A (showing that the first I-picture has a CAD of 0001, that the following P-picture has a CAD of 0002, and that CAD continues to increment by 1 for each subsequent P-picture in the group).) For I-pictures, the IAD component is zero. (See EX1006, Fig. 5A.) And for P-pictures, the IAD provides the CAD of the corresponding I-frame. (See EX1006, 4:48-52.) Thus, the pictures (i.e., image frames) in Kim’s disclosed bitstream have logical addresses that follow a numbering scheme.



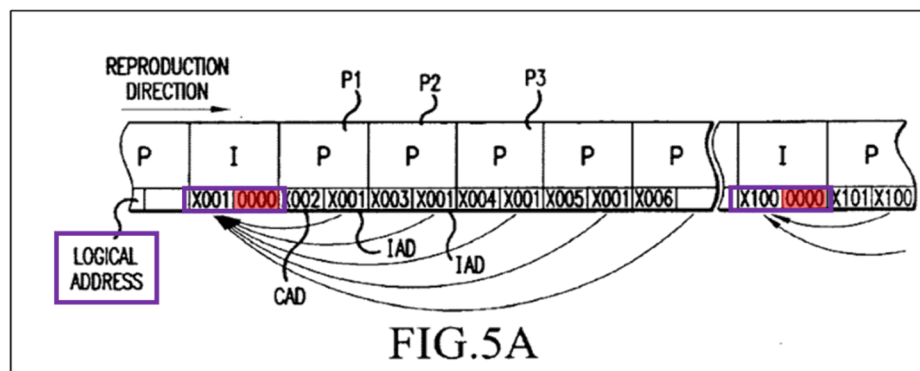
(EX1006, Fig. 5A (annotated).)

168. Kim also teaches that the logical addresses are decoded according to a numbering scheme. Specifically, Kim discloses that when a P-picture is decoded, “the

decoder detects the address of the corresponding I-picture frame ... and reproduces based on the detected address.” (EX1006, 5:41-46.) Thus, the decoder uses the logical addresses to determine the order in which pictures are decoded and reproduced. In reproducing the pictures in this order, the decoder also decodes the logical addresses. Accordingly, Kim discloses that logical addresses are decoded according to a numbering scheme.

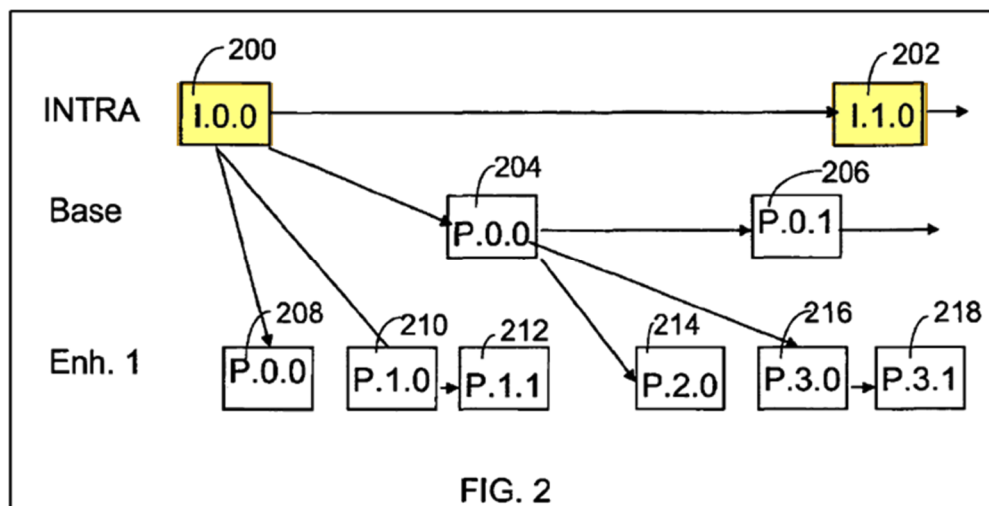
e. 8[d]: resetting the identifier value for the indicated first image frame of the independent sequence

169. In my opinion, Kim teaches this limitation. Specifically, Kim’s Figure 5A shows that the IAD component of each logical address is reset to zero for each I-picture. (See EX1006, Fig. 5A (annotated below to show IAD component of each I-picture’s logical address in red and the whole logical address outlined in purple).) Because each I-picture is the first image frame of an independent sequence, Kim discloses that the IAD component of each logical address (“identifier value”) is reset for the indicated first image frame of the independent sequence.



(EX1006, Fig. 5A.)

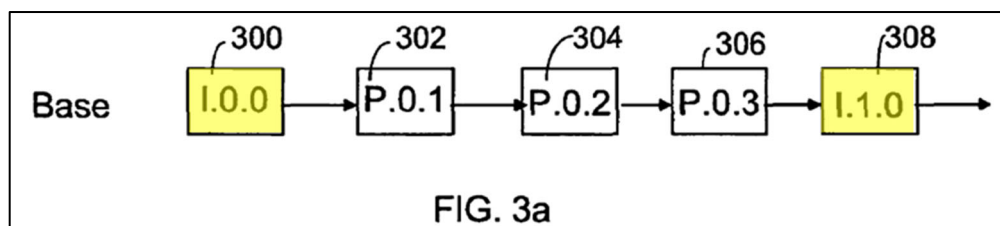
170. I understand Patent Owner may argue that Kim's disclosure of resetting a component of an identifier value does not teach or suggest this limitation. That argument is incorrect because resetting a component of the identifier value of an indicated first image frame is consistent with the disclosure of the '321 patent. In fact, the only two disclosures in the '321 patent addressing more than one independent sequence, Figures 2 and 3a, show only a component of an identifier value being reset.



(EX1001, Fig. 2 (annotated).)

171. In Figure 2 of the '321 patent, I-frames 200 and 202 reflect the first image frames of independent sequences. (EX1001, Fig. 2.) I-frame 200 has a type I and a subsequence identifier and image number of (0.0). (EX1001, 8:22-25.) I-frame 202 has a type I and subsequence identifier and image number of 1.0. (EX1001, 8:36-40.) The '321 patent explains that the identifier for I-frame 202 is therefore I.1.0, a combination of the type, subsequence identifier, and image number. (EX1001, 8:41.) The '321 patent further explains that the next I-frame would have an identifier of I.2.0.

Thus, it is my opinion that only a component of the '321 patent's disclosed identifiers, the image number, resets. The subsequence identifier component does not reset.



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

172. Figure 3a of the '321 patent likewise shows that the inventor intended the phrase “resetting an identifier value” to include resetting a component of an identifier value. The '321 patent states that “[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.) According to the '321 patent, “a second subsequence” is initiated by “an I-frame (I.1.0) coded therein, i.e. an I-frame 308.” (EX1001, 12:31-34.) Thus, again, only a component of the identifier value is reset when a new subsequence is initiated. This confirms that Kim teaches “resetting an identifier value” in the same way as the '321 patent discloses.

173. Accordingly, in my opinion Kim teaches resetting the identifier value for the indicated first image frame of the independent sequence through its disclosure that the IAD portion of a picture's logical address is reset for the first frame of an independent sequence.

174. For the reasons discussed above, it is my opinion that Kim teaches claim 8.

3. Claim 10

175. 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.”

- a. **10[pre] 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**

176. In my opinion, Kim teaches or suggests 10[pre]. As discussed in the Section IX.B.2.a for 8[pre], Kim teaches “decoding a compressed video sequence.” Kim teaches the additional elements of 10[pre].

177. Kim discloses a Time Lapse recording/reproduction device that 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.)

178. A POSITA would have understood that the disclosed controller 150 and corresponding decoder 111 and reproducing DSP 121⁵ comprise at least one processor and at least one memory including computer program code configured to, with the at least one processor, cause the video decoders to decode a compressed video sequence. For example, a DSP is a specialized type of computer processor that executes a stored program of computer instructions for performing signal processing operations. Thus, the disclosure of the DSP by itself is sufficient to teach “at least one processor and at least one memory including computer program code configured to, with the at least one processor, cause the video decoders to decode a compressed video sequence.”

179. Furthermore, Kim discloses encoding image frames in accordance with MPEG (EX1006, 4:15-19) and subsequently decoding those image frames (EX1006, 5:8-68). As noted above in Section IX.A.3.a, 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

⁵ DSP stands for “digital signal processor,” as disclosed in the Kim. (EX1006, 3:7.)

A POSITA therefore would understand that Kim’s DSP 121 contains a processor, and that, as of the alleged priority date of the ’321 patent, it could be programmable.

memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence.” Kim thus discloses decoding video that conforms to the MPEG standard. Accordingly, it is my opinion that for the reasons above and discussed in Section IX.A.3.a, it would have been obvious to a POSITA to implement Kim’s 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.”

- b. 10[a] 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**

180. In my opinion, this limitation is substantively identical to limitation 8[a]. (See Section IX.B.2.b.) Kim therefore teaches limitation 10[a] for the same reasons that it teaches 8[a].

- c. 10[b] 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**

181. In my opinion, this limitation is substantively identical to limitation 8[b]. (See Section IX.B.2.c.) Kim therefore teaches limitation 10[b] for the same reasons that it teaches 8[b].

d. 10[c] decode identifier values for image frames according to a numbering scheme

182. It is my opinion that this limitation is substantively identical to limitation 8[c]. (*See* Section IX.B.2.d.) Kim therefore teaches limitation 10[c] for the same reasons that it teaches 8[c].

e. 10[d] reset the identifier value for the indicated first image frame of the independent sequence

183. It is my opinion that this limitation is substantively identical to limitation 8[d]. (*See* Section IX.B.2.e.) Kim therefore teaches limitation 10[d] for the same reasons that it teaches 8[d].

184. For the reasons discussed above, it is my opinion that claim 10 is unpatentable as obvious over Kim.

4. Claim 11

185. In my opinion, 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 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.”

- a. **11[pre] 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**

186. In my opinion, Kim teaches or suggests 11[pre]. As discussed in Section IX.B.2.a for 8[pre], Kim teaches “decoding a compressed video sequence.” Kim also teaches the additional elements of 11[pre].

187. Kim discloses a Time Lapse recording/reproduction device that 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.)

188. A POSITA, considering Kim’s Figure 2, would have understood the disclosed decoder 111, reproducing DSP 121,⁶ and controller 150 together or individually to be “data processing devices,” given Kim’s disclosure that they all process data in the form of image data or commands.

189. A POSITA would have found it obvious 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,

⁶ DSP stands for “digital signal processor,” as disclosed in Kim. (EX1006, 3:7.)

for decoding a compressed video sequence.” More specifically, a POSITA would have understood that Kim’s decoder 111 is “for performing a reverse-quantization and reverse-transformation of the demultiplexed image data” (EX1006, 3:18-20) and that 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 under control of controller 150 which is “for controlling the above-described components based on the commands from the user.” (EX1006, 3:25-27).

190. Because a DSP is a specialized type of computer processor that executes a stored program of computer instructions for performing signal processing operations, a POSITA would have understood that Kim’s reproducing DSP 121 necessarily performs its portion of the decoding process by executing 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.”

191. A POSITA also would have found it obvious to implement the remaining video decoding functions performed by Kim’s decoder 111 and controller 150 in either of two ways: as dedicated custom integrated circuit hardware or as a stored computer program running on a general-purpose processor that was available at the time. Both of these implementation options would have been obvious to a POSITA and both would have been well within the capabilities of a POSITA. Therefore, in my opinion it would

have been obvious to a POSITA to implement Kim's decoder 111, reproducing DSP 121, and controller 150 in one or more computer processors executing "[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."

192. A POSITA would have been motivated to implement Kim in this way and would have had an expectation of success at least because, at the time, there were already well-known examples of implementing real-time MPEG video decoders in software computer program products running on general purpose computer processors. For example, a real-time software MPEG-1 decoder was available for free download from the University of California since at least 1993 (*see generally* EX1010) and engineers at Hewlett-Packard had implemented a real-time software MPEG-1 video decoder on HP workstations and personal computers at least by 1995 (EX1009, 60-68), with timing information for their software implementation of the video decoding functions performed by Kim's decoder 111 given in Table 1 (EX1009 at 63). Therefore, 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).

193. Thus, based on the foregoing, it is my opinion that a POSITA would have understood Kim to disclose 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.”

194. Furthermore, Kim discloses encoding image frames in accordance with MPEG (EX1006, 4:15-19) and subsequently decoding those image frames (EX1006, 5:8-68). Kim thus discloses decoding video that conforms to the MPEG standard. As I explained in Section IX.A.4.a, it would have been obvious to a POSITA to implement an MPEG-1 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.” Thus, in my opinion, it would have been obvious to a POSITA to implement 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.”

- b. 11[a] 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**

195. In my opinion, this limitation is substantively identical to limitation 8[a], except that it is directed to “a computer program code for” decoding an indication rather than the act of decoding an indication. (*See* Section IX.B.2.b.) As I explained above in Section IX.B.2.b, Kim 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.” And as I explained in Section IX.B.4.a, a POSITA would recognize that the decoding would be carried out by “a computer program product,” which would include “computer program code.” Kim therefore teaches or suggests this limitation.

- c. 11[b] 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**

196. In my opinion, this limitation is substantively identical to limitation 8[b], except that it is directed to “a computer program code for” starting the decoding rather than the act of starting the decoding. (*See* Section IX.B.2.c.) As I explained above in Section IX.B.2.c, Kim teaches “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.” And as I explained above in Section IX.B.4.a, a POSITA would recognize that the decoding process would be carried out by “a computer program product,” which would include “computer program code.” Therefore, it is my opinion that Kim teaches or suggests this limitation.

d. 11[c] a computer program code for decoding identifier values for image frames according to a numbering scheme

197. It is my opinion that this limitation is substantively identical to limitation 8[c], except that it is directed to “a computer code for” decoding identifier values rather than the act of decoding identifier values. (See Section IX.B.2.d.) As I explained above in Section IX.B.2.d, Kim teaches “decoding identifier values for image frames according to a numbering scheme.” And as I explained above in Section IX.B.4.a, a POSITA would recognize that the decoding process would be carried out by “a computer program product,” which would include “computer program code.” Kim therefore teaches or suggests this limitation.

e. 11[d] a computer program code for resetting the identifier value for the indicated first image frame of the independent sequence

198. In my opinion, this limitation is substantively identical to limitation 8[d], except that it is directed to “a computer program code for” resetting the identifier value rather than the act of resetting the identifier value. (See Section IX.B.2.e.) As explained

in Section IX.B.2.e., it is my opinion that Kim therefore teaches “resetting the identifier value for the indicated first image frame of the independent sequence.” And as I explained above in Section IX.B.4.a, a POSITA would recognize that the decoding process would be carried out by “a computer program code product,” which would include “computer program code.” Kim therefore teaches or suggests this limitation.

199. For the reasons discussed above, it is my opinion that Kim teaches or suggests claim 11.

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

1. Yagasaki

200. Yagasaki discloses methods and apparatuses for encoding and decoding an image signal. (EX1015, Title.) Yagasaki describes the MPEG as a “highly efficient encoding system[] which rel[ies] upon the correlation of video signals.” (EX1015, 1:36-39.) Given the timing of Yagasaki, a POSITA would have understood that Yagasaki refers to the same “MPEG” described by MPEG-1. Yagasaki claims priority to foreign applications that were filed in 1993, around the time of MPEG-1’s release.

201. Further, Yagasaki describes a video signal encoded by an MPEG system as having a “video sequence ... divided into a frame group (GOP),” where each of the frames is classified as an I-picture, P-picture, or B-picture. (EX1015, 1:58-67.) Consistent with 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.)

202. According to Yagasaki, hardware limitations make it difficult to “select a frame freely and reproduce it at a high speed.” (EX1015, 2:14-15, 2:23-25.) This is because P- and B-pictures “cannot be decoded in the absence of decoded images of past and/or future frames.” (EX1015, 2:12-13.) Even I-pictures “can be reproduced at high speed ... only with rough motion.” (EX1015, 2:19-20.)

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

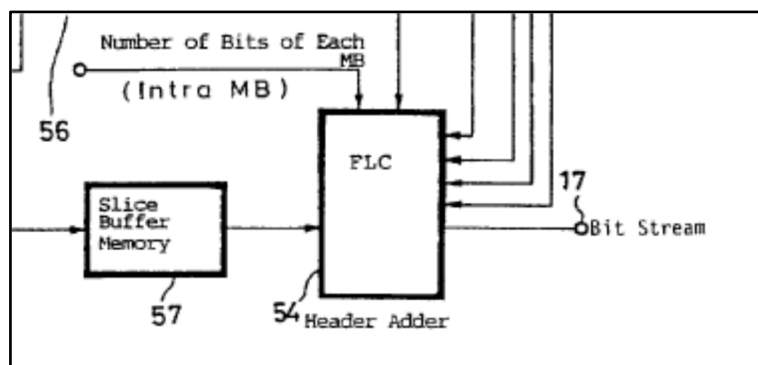
2. Claim 9: A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.

204. MPEG-1 discloses “[a] method according to claim 8,” which in turn recites “a method for decoding a compressed video sequence,” for the reasons described above in Section IX.A.2.a. Yagasaki discloses “wherein the indication is a separate flag included in the header of a slice.” Yagasaki further describes decoding a slice header that contains such a flag. (EX1015, 14:30-40.)

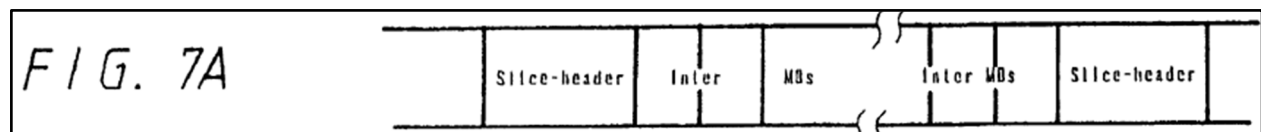
205. First, Yagasaki discloses that an image may be formed of multiple slices. Yagasaki discloses an example image composed of “macroblocks,” each of which is “a unit range to be encoded.” (EX1015, 9:27-30.) “A number of macroblocks arranged in

the order in which they are transmitted are called a slice.” (EX1015, 9:30-33.) A POSITA would have recognized that the slices described by Yagasaki are also described 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].”).)

206. Yagasaki also discloses slice headers. It describes a “hybrid encoder 102” that includes a “fixed length coder” (FLC) 54. (EX1015, 4:56, 6:20-22.) Yagasaki refers to FLC 54 as a “header adder,” which adds a header to image data stored in a slice buffer memory 57:

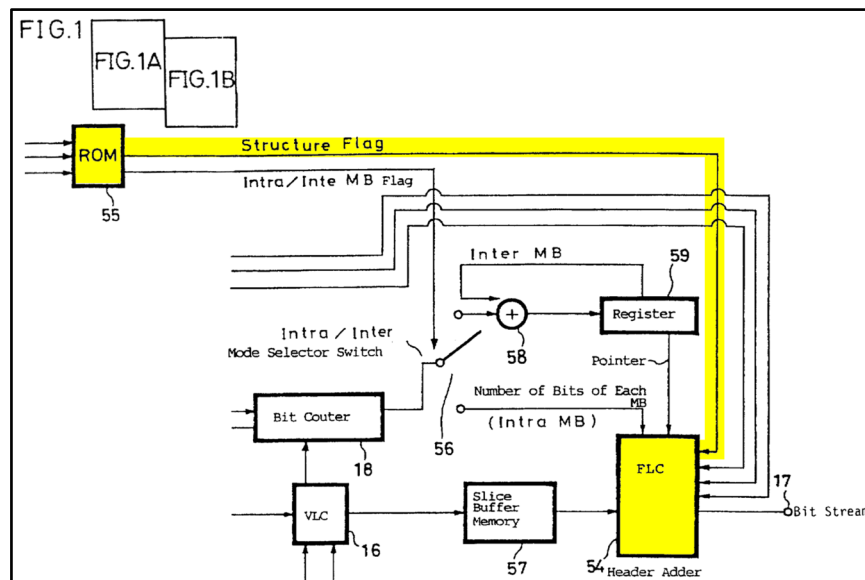


(EX1015, 6:15-24, 6:39-41, 7:3-4, Fig. 1B (excerpt).) As Yagasaki claim 38 recites, the “header adder” is “for adding slice headers.” (EX1015, 23:4-5.) Yagasaki further shows slices containing these slice headers:



(EX1015, Fig. 7A.) A POSITA would have recognized that the slice header described by Yagasaki is also described in MPEG-1. (EX1005, 91-92 (describing slice headers).).

207. The information added to the slice header includes a separate flag. A “ROM 55 holds a distribution pattern of intraframe coding macroblocks for high-speed reproduction which are assigned to each of the frames of the GOP.” (EX1015, 6:42-44.) “[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.) As shown below, ROM 55 passes the structure flag to FLC 54, which in response “adds information to a slice header.” (EX1015, 7:1-4.)



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

208. The information added to the slice header includes “a flag representing a structure of a slice.” (EX1015, 9:55-61.) This separate flag renders obvious an

indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence. Specifically, the flag is “indicative of 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).)

209. Accordingly, Yagasaki teaches a method of decoding a compressed video sequence that includes separate flags included in the header of a slice, which is part of a frame. Yagasaki discloses that these flags identify when a slice is composed entirely of intra-frame coded macroblocks. (EX1015, 10:26.)

210. From the disclosure of Yagasaki, a POSITA would have found it obvious to place flags marking intra-coding status in slice headers. As a fully intra-coded frame is the beginning of an independent sequence, it would also have been obvious to a POSITA that flags indicating this same characteristic from the MPEG-1 grounds (closed_GOP or Temporal Reference) could similarly be included in a slice header.

211. For the reasons discussed above, it is my opinion that claim 9 is unpatentable as obvious over the combination of MPEG-1 and Yagasaki.

3. Motivation to Combine MPEG-1 and Yagasaki

212. A POSITA would have considered implementing the teachings of Yagasaki within the encoding and decoding methods disclosed by MPEG-1. Yagasaki provides video coding methods that may be performed in an MPEG-1 system, with predictable results. Indeed, Yagasaki expressly discloses that the “basic coding process in the present invention is the same as the encoding process in MPEG.” (EX1015, 7:5-6.) Additionally, Yagasaki describes the same GOPs and I-pictures taught in MPEG-1. (EX1015, 1:58-2:9; EX1005, 71.) I-pictures are intra-coded in their entirety and thus contain slices with all intra-coded macroblocks; Yagasaki and MPEG-1 both confirm these are the only kind of frame that may be directly accessed and decoded independently. (EX1015, 2:7-18; EX1005, 23, 71.) The decoding process for full intra-coded slices taught by Yagasaki may also be performed in an MPEG-1 system—Yagasaki explains that slices “with all intra macroblocks ... can be decoded by the normal MPEG decoding process.” (EX1015, 17:32-36.)

213. Yagasaki further recognizes the desirability of efficient video coding, which reduces recording size. (EX1015, 1:31-36.) Yagasaki notes that frames cannot all be entirely intra-coded I-pictures; they require too much time to read and process. (EX1015, 2:16-28.) But Yagasaki teaches that intra-frame coded regions are critical 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 understand this is

because non-intra-frame coded regions do not contain sufficient data to decode a full image.

214. For “high-speed reproduction” during decoding, Yagasaki describes outputting only slices containing intra-frame coded macroblocks to a buffer while discarding others. (EX1015, 14:30-40.) Thus, a POSITA would recognize that an indication of the intra-coding status for slices, as taught by Yagasaki, allows for efficient decoding. Yagasaki teaches doing so using separate flags in the slice headers, as discussed in Section IX.C.2.

215. A POSITA would further recognize that the indication method taught by Yagasaki could be used to provide other indications of intra-coding status disclosed by MPEG-1 for other image regions, including closed_GOP or Temporal Reference. The modification would improve the efficiency of decoding 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.

216. A POSITA would have also found it obvious to include the indications of intra-coding status disclosed by MPEG-1 in the slice header because it is obvious to try. There are only a finite number of header locations in which such indications can be placed and a POSITA would have found it obvious that the slice header is one of these locations.

217. In my opinion, a POSITA would have had a reasonable expectation of success that placing indications of intra-coding status in the slice headers would be successful. Yagasaki teaches that such notifications can be placed in slice headers (*see* Section IX.C.2) and successfully decoded in the normal MPEG-1 decoding process.

218. In my opinion, a POSITA would thus have been motivated to modify the methods disclosed by MPEG-1 using these teachings of Yagasaki.

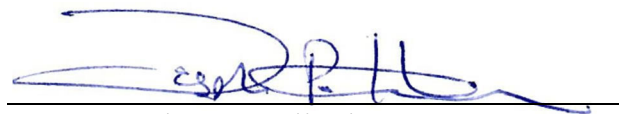
X. CONCLUSION

219. 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.

220. 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: March 19, 2024

By:


Dr. Joseph P. Havlicek