

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SNAP INC. and HISENSE USA CORPORATION,
Petitioners

v.

NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2025-01014

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

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,050,321**

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING.....	1
III.	IDENTIFICATION OF CHALLENGE	2
	A. Prior Art.....	2
	B. Grounds for Challenge	2
IV.	'321 PATENT OVERVIEW	2
	A. Technology Overview	2
	B. The '321 Patent's Purported Invention	5
V.	LEVEL OF ORDINARY SKILL	7
VI.	CLAIM CONSTRUCTION	7
VII.	DETAILED EXPLANATION OF GROUNDS.....	8
	A. Ground 1: Kim Renders Obvious Claims 1, 3-4, 6-8, and 10-11	8
	1. Kim.....	8
	2. Claim 1	13
	3. Claim 3: "A method according to claim 1, further comprising: encoding into the video sequence an identifier value for the independent sequence."	23
	4. Claim 4	24
	5. Claim 6: "A video encoder according to claim 4, wherein the video encoder is configured to encode into the video sequence an identifier value for the independent sequence."	26
	6. Claim 7	27
	7. Claim 8	30
	8. Claim 10	48
	9. Claim 11	51
	B. Ground 2: Kim in view of Adolph Renders Obvious Claims 2, 5, and 9	56

TABLE OF CONTENTS
(continued)

	Page
1. Adolph.....	56
2. Motivation to Combine Kim and Adolph.....	59
3. Claim 2: “A method according to claim 1, further comprising: encoding the indication into the video sequence as a separate flag included in the header of a slice.”	62
4. Claim 5: “A video encoder according to claim 4, wherein the video encoder is configured to encode the indication into the video sequence as a separate flag included in the header of a slice.”	63
5. Claim 9: “A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.”	63
VIII. CONCLUSION.....	63
IX. MANDATORY NOTICES	63
A. Real Party in Interest	63
B. Related Matters.....	63
C. Notice of Counsel and Service Information.....	64

EXHIBIT LIST

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	C-Cube CL480PC MPEG-1 System Decoder Specification, available at https://web.archive.org/web/19970128081644/http://www.c-cube.com/prdctlst/cl480pc.pdf .
1008	U.S. Patent No. 5,838,265 ("Adolph")
1009	Ruby B. Lee et al., Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7100LC Processors, Hewlett-Packard J., Apr. 1995, available at https://archive.org/details/Hewlett-Packard_Journal_Vol._46_No._2_1995-04_Hewlett-Packard
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	Excerpt from Joan L. Mitchell et al., <i>MPEG Video Compression Standard</i> (Chapman & Hall, 1996)
1012	Excerpt from Roy Hoffman, <i>Data Compression in Digital Systems</i> (Chapman & Hall, 1997)
1013	Sony Corporation, Operating Instructions for SVT-S3050P Time Lapse Videocassette Recorder (1997), available at https://pro.sony/s3/cms-static-content/operation-manual/3193023022.pdf

Exhibit	Reference
1014	GYJR Inc., GYJR Products Application Guide (1998), <i>available at</i> https://web.archive.org/web/19980628195831/http://www.gyyr.com:80/pdf/appguide.pdf
1015	C-Cube CLM4111 MPEG-1 Video Encoder data sheet, available at https://web.archive.org/web/19970707143632/http://www.c-cube.com/products/CLM4111.pdf
1016	ISO/IEC 13818-2 (1st ed.) Information Technology – Generic coding of moving pictures and associated audio information: Video (“MPEG-2”)
1017	ISO/IEC TR 11172-5 (1st ed.) Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 5: Software simulation
1018	Original Complaint, <i>Nokia Techs. Oy v. Hisense Co. Ltd. and Hisense USA Corp.</i> , No. 1:25-cv-01871 (N.D. Ga. April 7, 2025), ECF No. 1.
1019	Complaint Under Section 337 of The Tariff Act of 1930, As Amended, Inv. No. 337-TA-3821 (Apr. 11, 2025)
1020	U.S. Patent No. 7,894,521
1021	IPR2024-00691, Paper 10: Decision Granting Institution of Inter Partes Review
1022	Thomas Sikora, <i>MPEG Digital Video-Coding Standards</i> , IEEE SIGNAL PROCESSING MAGAZINE (Sept. 1997)

LISTING OF CHALLENGED CLAIMS

Reference	Claim Limitation
Claim 1	
1[pre]	A method for encoding a video sequence comprising an independent sequence of image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the method comprising:
1[a]	encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;
1[b]	encoding identifier values for the image frames according to a numbering scheme; and
1[c]	resetting the identifier value for the indicated first image frame of the independent sequence.
Claim 2	
2	A method according to claim 1, further comprising: encoding the indication into the video sequence as a separate flag included in the header of a slice.
Claim 3	
3	A method according to claim 1, further comprising: encoding into the video sequence an identifier value for the independent sequence.
Claim 4	
4[pre]	A video encoder comprising at least one processor and at least one memory including computer program code, the at least one memory

Reference	Claim Limitation
	and the computer program code configured to, with the at least one processor, cause the video encoder to encode a video sequence comprising an independent sequence of image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence by causing the video encoder to at least:
4[a]	encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;
4[b]	encode identifier values for the image frames according to a numbering scheme; and
4[c]	reset the identifier value for the indicated first image frame of the independent sequence.
Claim 5	
5	A video encoder according to claim 4, wherein the video encoder is configured to encode the indication into the video sequence as a separate flag included in the header of a slice.
Claim 6	
6	A video encoder according to claim 4, wherein the video encoder is configured to encode into the video sequence an identifier value for the independent sequence.
Claim 7	
7[pre]	A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of

Reference	Claim Limitation
	image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the computer program product comprising:
7[a]	a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;
7[b]	a computer code for encoding identifier values for the image frames according to a numbering scheme; and
7[c]	a computer code for resetting the identifier value for the indicated first image frame of the independent sequence.
Claim 8	
8[pre]	A method for decoding a compressed video sequence, the method comprising:
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;
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;
8[c]	decoding identifier values for image frames according to a numbering scheme; and
8[d]	resetting the identifier value for the indicated first image frame of the independent sequence.

Reference	Claim Limitation
Claim 9	
9	A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.
Claim 10	
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:
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;
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; and
10[c]	decode identifier values for image frames according to a numbering scheme; and
10[d]	reset the identifier value for the indicated first image frame of the independent sequence.
Claim 11	
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:

Reference	Claim Limitation
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;
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;
11[c]	a computer program code for decoding identifier values for image frames according to a numbering scheme; and
11[d]	a computer program code for resetting the identifier value for the indicated first image frame of the independent sequence.

I. INTRODUCTION

Snap Inc. and Hisense USA Corporation (“Petitioners”) request *inter partes* review (“IPR”) of claims 1-11 (“Challenged Claims”) of U.S. Patent 8,050,321 (EX1001 or “’321 patent”).

This Petition relies primarily on prior art U.S. Patent No. 6,912,351 to Kim which was previously alleged to render obvious claims 8, 10, and 11 of the ’321 patent in IPR Case No. IPR2024-00691. That case was instituted (*see* EX1021) but subsequently terminated when Patent Owner settled with the prior petitioner. Petitioners here present substantially similar arguments based on Kim to those that were presented in the prior IPR for claims 8, 10, and 11. However, Petitioners also challenge the remaining claims of the ’321 patent.

II. GROUNDS FOR STANDING

Petitioners certify the ’321 patent is available for IPR, and Petitioners are not barred or estopped from requesting IPR on the grounds herein. Petitioner Snap has not been sued on the ’321 patent and this Petition is being filed much less than one year after service on Petitioner Hisense USA of Patent Owner’s district court and ITC complaints alleging infringement of the ’321 patent by Hisense. (*See* EX1018; EX1019).

III. IDENTIFICATION OF CHALLENGE

A. Prior Art

Petitioners respectfully request review and cancellation under 35 U.S.C. § 311 of the Challenged Claims in view of:

Prior Art	
“Kim,” U.S. Patent No. 6,912,351, filed on October 19, 1998; prior art at least under § 102(e) (EX1006)	
“Adolph,” U.S. Patent No. 5,838,265, filed May 23, 1996; prior art at least under § 102(b) (EX1008)	

B. Grounds for Challenge

Ground		Claims	Prior Art
1	103	1, 3-4, 6-8, 10-11	Kim
2	103	2, 5, 9	Kim, Adolph

IV. '321 PATENT OVERVIEW

A. Technology Overview

The '321 patent is generally concerned with encoding and decoding video sequences comprising independent sequences of image frames. (EX1001, Abstract.) As its title (“Grouping of Image Frames in Video Coding”) suggests, a key concept in the '321 patent is the grouping of these image frames. (EX1001.) The concept of grouping image frames into displayable sequences has been known at least since 1993 when International Standard ISO/IEC 11172-2 (“MPEG-1”) was released. (EX1003, ¶42.) That standard used the term “group of pictures” or “GOP” to refer

to a group of image frames. (EX1005, 23¹, 71 (explaining that “[a] picture corresponds to a single frame of motion video”).) The ’321 patent uses these terms in the same way. (*E.g.*, EX1001, 2:34-40.)

A specific type of GOP discussed in the ’321 patent is the “independently decodable [GOP]” or “independent sequence.” (EX1001, 11:26-30, Abstract.) An independently decodable GOP contains pictures that can be decoded independently of information outside the GOP. (EX1003, ¶43.) Not all GOPs are independently decodable (EX1001, 3:52-56) 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.

Temporal redundancy concerns the similarities between image frames. (EX1001, 2:1-3.) Because image frames in a video sequence are typically captured in quick succession, many aspects of the scene being captured do not change or only change in predictable ways from one image frame to the next. (EX1003, ¶44.) These aspects are considered to be temporally redundant.

¹ Page numbers for EX1005 refer to the page numbers added by Petitioners (and correspond to the pdf page number), unless otherwise stated.

Temporal prediction uses information about one frame to predict information about another frame and takes advantage of temporal redundancies by encoding only the changes that occur between image frames. (EX1003, ¶¶45-49.) Temporal prediction allows image frames to be encoded using less data because redundant information is excluded. (EX1003, ¶45.) Instead, the encoded information can contain only some information about the depicted scene and refer to another image frame for any missing information. (EX1003, ¶45.)

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. (EX1001, 2:16-23, EX1003, ¶ 46.)

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. (EX1003, ¶¶47-50; EX1001, 2:31-33 (“If a frame is lost, frames dependent on it can no longer be correctly decoded.”).)

There are different “types” of image frames that differ based on their dependence on other image frames. (E.g., EX1001, 3:50-52 (“Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames.”); EX1003, ¶48.) 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.)

As mentioned, 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 the same GOP. (EX1003, ¶49.)

Independently decodable GOPs existed long before the alleged invention of the '321 patent. (EX1003, ¶50.) As explained in MPEG-1 in 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.) 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.”).)

B. The '321 Patent's Purported Invention

The '321 patent's specification is largely devoted to describing a “scalable coding hierarchy” in which video data is coded using “scalability layers.” (EX1001,

5:12-25 (describing 10 of the patent's 14 figures as concerning a "scalable coding hierarchy" or "scalability"); EX1001, 7:43-55 (discussing "scalability layers").)

But while the '321 patent's parent includes claims that recite "scalable coding," that concept is absent from the '321 patent's claims. *Compare* EX1001, cls.1-11, *with* EX1020 (parent patent), cls.1-24 (referring to "scalable, compressed video sequence" and "scalability layers"). The '321 claims are instead directed to the simpler concepts of grouping and numbering image frames. (EX1003, ¶¶51-53.) According to the 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.)

The '321 patent states 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; EX1003, ¶53.)

V. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“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. (EX1003, ¶¶18-21.)

VI. CLAIM CONSTRUCTION

Claims should be construed “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

Petitioners have not proposed any terms for construction because no constructions are necessary to resolve any disputes identified in this Petition. *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

VII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Kim Renders Obvious Claims 1, 3-4, 6-8, and 10-11

1. Kim

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.) Kim describes a “recording method” that involves digitizing and compressing image frames from one or more cameras to generate a bit stream like the one shown in Fig. 5A. (EX1006, 3:33-5:7.) Kim also describes a “reproducing method” that involves decoding a bit stream 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.) (EX1003, ¶55.)

Kim describes a bitstream with pictures grouped into a “logical structure” based on their dependencies. Kim depicts this logical structure in Figure 5A:

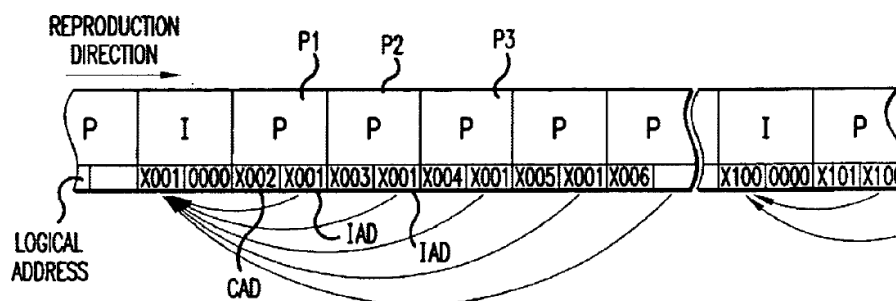


FIG.5A

(EX1006, Fig. 5A.) 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 are

similar to MPEG standard I- and P-pictures. (See EX1006, 4:15-19, 6:55-58; EX1003, ¶¶55-59; see also EX1005, 33.) 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. (EX1003, ¶56.) 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-pictures to be restored as full screen images. (EX1006, 5:55-59.)

Kim further 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:

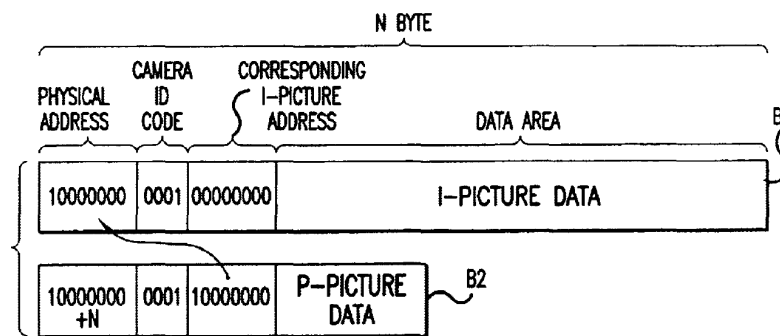
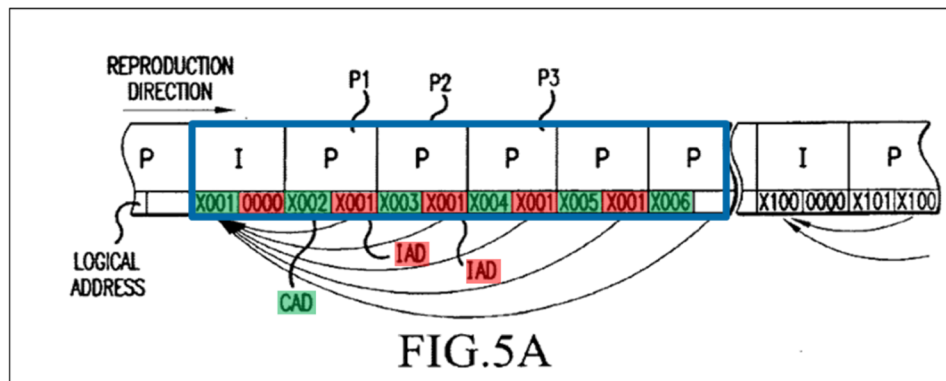


FIG.6

(EX1006, Fig. 6, 4:59-61.)

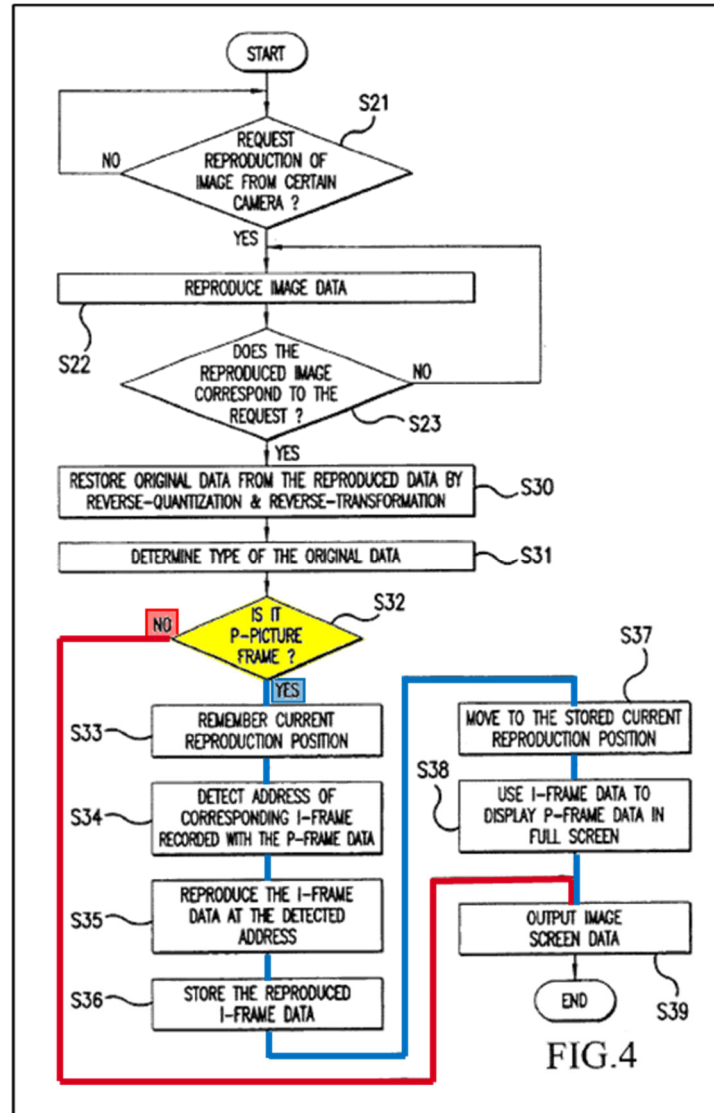
Kim explains that each P-picture has a current address (“CAD”) that indicates the current address of the P-picture, and an I-picture reference address (“IAD”) that

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 (EX1006, 4:65-5:1, Figs. 5A, 6 (“physical address”)), but the I-picture's IAD is zero (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).)

As shown below in Kim's Figure 4, Kim sets forth two paths for decoding the disclosed bitstream depending on whether the requested image corresponds to an I-picture or a P-picture. (EX1003, ¶¶59-62.) When an I-picture is requested, the I-picture is simply 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).)

In IPR2024-00691, Patent Owner tried to dispute that Kim is a “video” reference and suggested that Kim’s decoder can only display a single requested image rather than a sequence of images. Not so. Kim’s “Field of the Invention” expressly references “reproducing video data.” (EX1006, 1:7-12; *id.* at 1:17-20 (“A conventional Time Lapse Video Cassette Recorder (Time Lapse VCR) records and

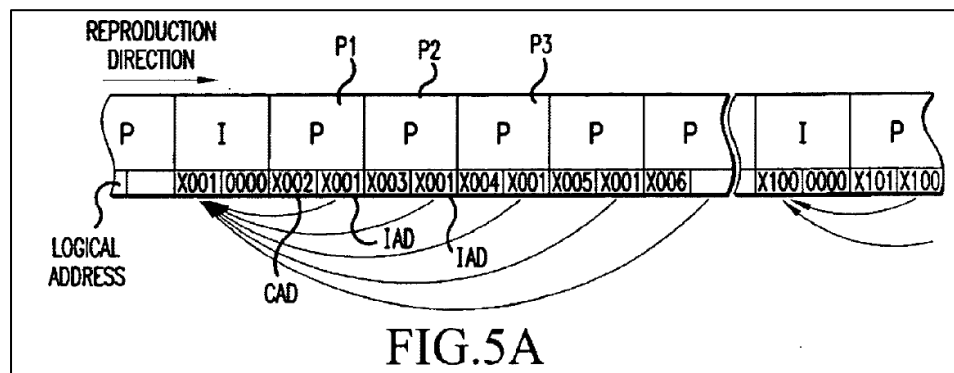
reproduces video signals on and from magnetic tapes.”).) The use of time lapse video for applications such as video surveillance and nature photography was well-known by the time of the alleged invention. (EX1003, ¶62.) For surveillance applications, a POSITA would have known it would have been impractical to require a user to manually select each image individually for viewing rather than watch a sequence of time lapse video frames to identify key events such as a break-in or property damage. (EX1003, ¶62.) Similarly, time lapse nature videos showing rapid progressions such as day becoming night, clouds moving in the sky, and flowers opening up have long been used in TV programs and movies to convey the passing of time or to quickly depict how things change over an extended period. (EX1003, ¶62.) Furthermore, a POSITA would have known that the time lapse video systems Kim describes not only recorded but also played back time-lapse video sequences. (EX1003, ¶61; *see e.g.*, EX1013 (1997 Sony SVT-S3050P manual) at 2 (“What is Time Lapse Video?”), 28-29 (discussing adjusting playback speeds for time lapse video); EX1014 (1998 GYYR time lapse video product guide) at 3 (section titled “Playback”).)

2. Claim 1

- a. 1[pre]: “A method for encoding a video sequence comprising an independent sequence of image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the method comprising:”**

To the extent the preamble is limiting,² Kim discloses or suggests it. (EX1003, ¶¶63-68.) Kim discloses encoding video sequences comprising image frames. For example, Kim discloses “an encoder 110 for generating I-picture data or P-picture data in frame units” (EX1006, 3:3-6.)

Kim also discloses the claimed “independent sequence” (EX1003, ¶¶65-68.) Kim’s Figure 5A (reproduced below) shows an example of a logical data structure of a digital data stream. (EX1006, 2:48-51.)



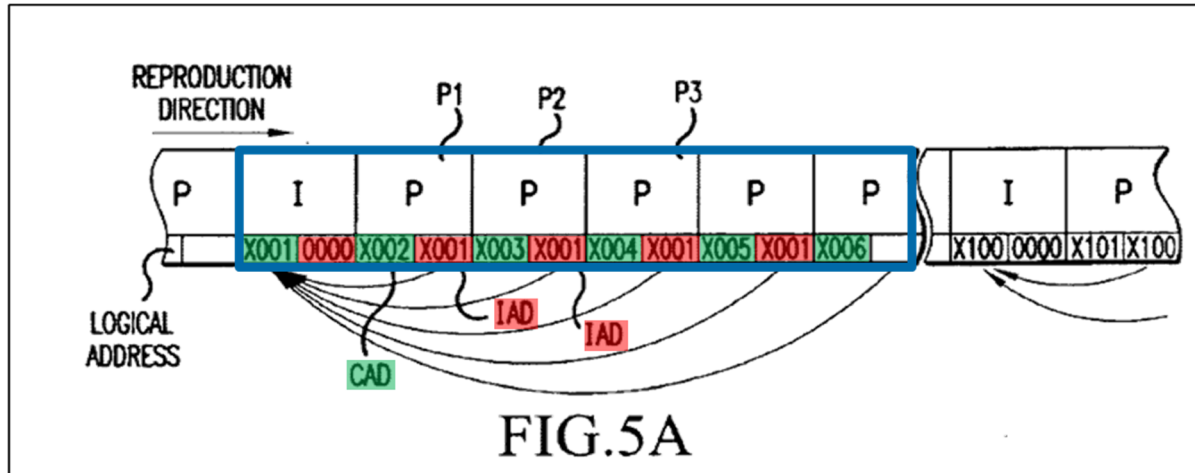
(EX1006, Fig. 5A.)

² Petitioners reserve the right to argue that the preamble is limiting in other proceedings.

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") 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." (EX1003, ¶66.)

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; EX1003, ¶67.) 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.) Thus, each P-picture's IAD identifies the I-picture on which the P-picture depends for motion compensation. (EX1003, ¶67.) This is further shown in annotated Figure 5A below, where every P-picture's IAD references the CAD of the preceding I-picture as confirmed by the 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); EX1003, ¶67.) Thus, every P-picture in Kim's bitstream depends on the preceding I-picture and only on that picture.



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

Because every P-picture following the I-picture depends only 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. (EX1003, ¶68.) Accordingly, Kim discloses or suggests a group of frames where every P-picture depends only on the preceding I-picture for motion prediction (“an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within the independent sequence”). In other words, each I-picture starts a new independent sequence consisting of the I-picture and some number of P-pictures that depend on it.

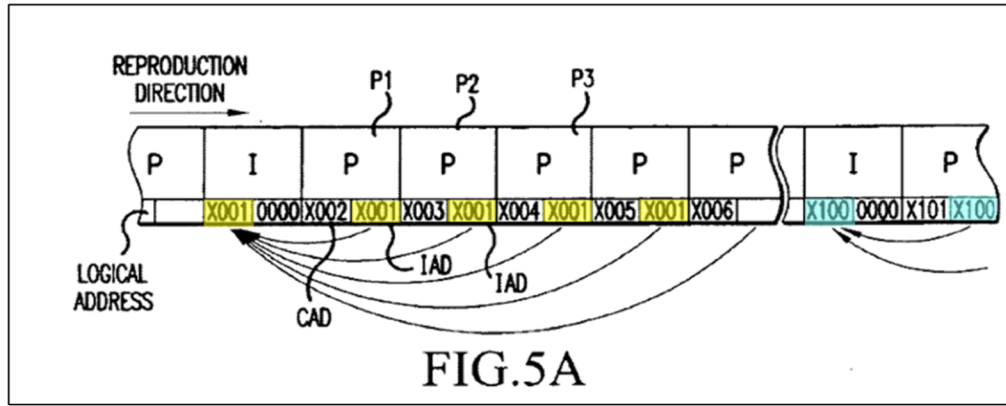
- b. 1[a]: “encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;”**

Kim teaches or suggests 1[b]. (EX1003, ¶¶69-74.)

First, Kim teaches the recited “indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence.” (EX1003, ¶70.) 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 the independent sequence.” Kim discloses that each image frame has a CAD and an IAD. (EX1006 4:46-52, 4:59-5:7, Figs. 5A, 6 (referring to CAD as “physical address”).) Kim’s Figure 5A shows that each I-picture has an IAD of 0000. (EX1006, Fig. 5A; EX1006, 4:59-5:1 (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 (or zero) indicates that that picture 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).)

A POSITA would have understood that the encoder disclosed in Kim encodes the IAD of the image frames. (EX1003, ¶72.) Kim’s Figure 5A provides an “example[] of a logical data structure of a digital data stream” (EX1006, 2:48-51), and “FIG. 6 shows an example of a data structure of the bit stream shown in FIG[. 5A” (EX1006, 4:59-61). Kim then explains the process for encoding this bitstream. (EX1006, 3:33-4:45, Fig. 3.) A POSITA would have understood that encoding the bitstream in Figure 5A would include encoding the components of that bitstream, including the IAD and CAD values. (EX1003, ¶72.) Additionally, Kim discloses that during the encoding process for each P-frame, “the position value of the corresponding (or previous) I-frame ... is added to the corresponding compressed P-frame output from the encoder 110.” (EX1006, 4:27-33; Fig. 5A.)

Accordingly, Kim teaches or suggests 1[a] by disclosing encoding the IAD value of 0000 for each I-picture as described and shown, for example, in Kim's Figure 5A. (EX1003, ¶73.)

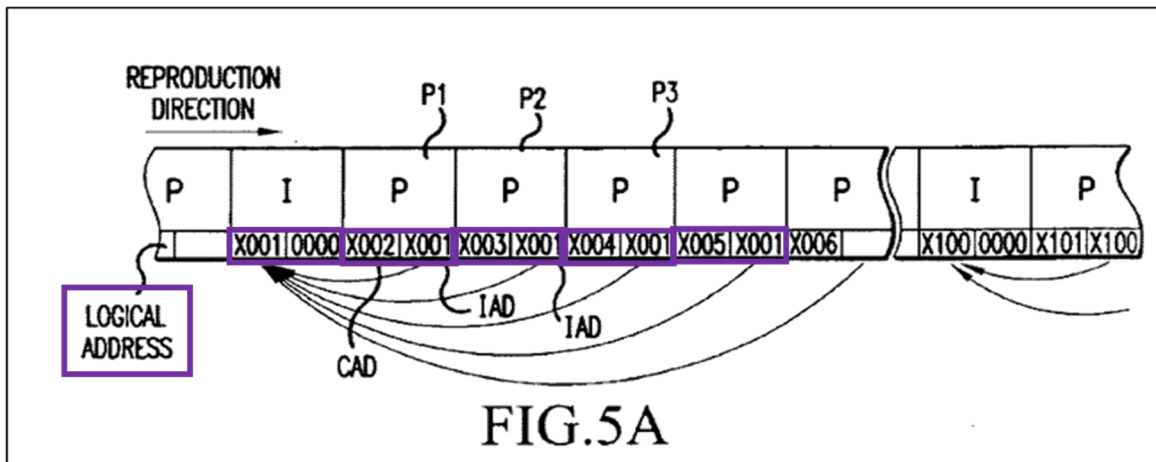
Taken together, Kim's disclosure of encoding the IAD value of 0000 for each I-picture and its disclosure of groupings consisting of an I-picture followed by P-pictures that depend on that I-picture teach or suggest "encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence." (EX1003, ¶74.)

c. 1[b]: "encoding identifier values for the image frames according to a numbering scheme; and"

Kim teaches or suggests 1[b]. (EX1003, ¶¶75-78.) Specifically, Kim's "logical addresses" are "identifier values for the image frames," and they are encoded "according to a numbering scheme."

Kim teaches or suggests that pictures have "logical addresses" that are assigned according to a numbering scheme. (EX1003, ¶77.) 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. (EX1006, 4:48-52.) The CAD increments by 1 for each picture in display order. (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 because the I-pictures do not depend on any earlier picture. (EX1006, Fig. 5A.) For P-pictures, the IAD references the CAD of the corresponding I-picture. (EX1006, 4:48-52.) Accordingly, the pictures (i.e., image frames) in Kim's disclosed bitstream have logical addresses that follow a numbering scheme. (EX1003, ¶77.)

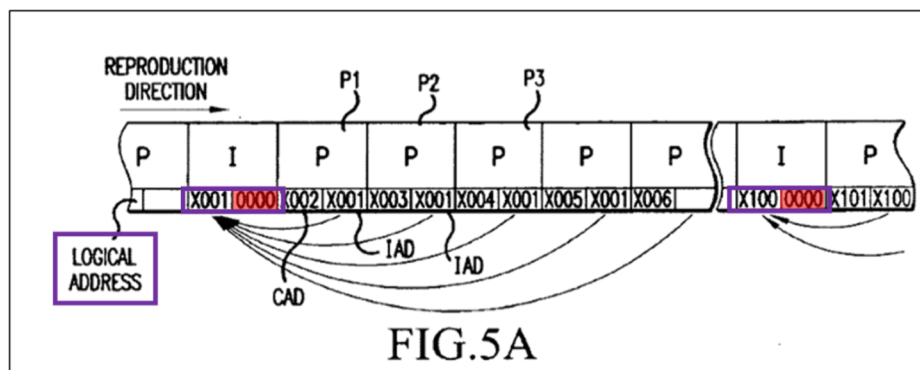


(EX1006, Fig. 5A (annotated).)

Moreover, given Kim's disclosure that the addresses for each image are part of the bitstream as shown in Fig. 5A, a POSITA would have understood that the address numbers are encoded. (EX1006, 4:27-33 ("the position value of the corresponding (or previous) I-frame ... is added to the corresponding compressed P-frame output from the encoder 110"); *see also* EX1003, ¶78.)

d. 1[c]: “resetting the identifier value for the indicated first image frame of the independent sequence.”

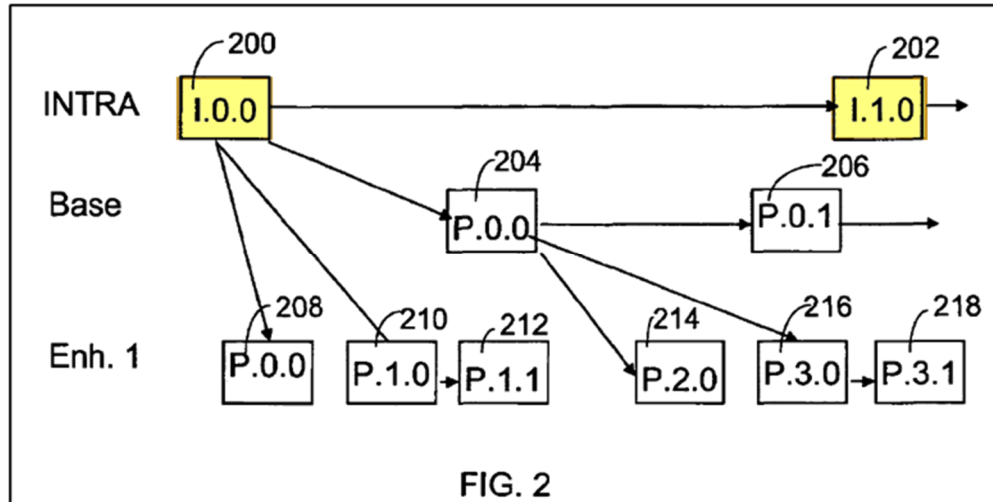
Kim teaches or suggests 1[c]. (EX1003, ¶¶79-85.) Specifically, Kim’s Figure 5A shows that the IAD component of each logical address is reset to zero for each I-picture. (EX1006, Fig. 5A (annotated below to show IAD component of each I-picture’s logical address in red and the whole logical address outlined in purple).) Because each I-picture is the first image frame of an independent sequence, Kim discloses or suggests that the IAD component of each logical address (“identifier value”) is reset for the indicated first image frame of the independent sequence. (EX1003, ¶80.)



(EX1006, Fig.5A.)

To the extent Patent Owner argues 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 ’321 patent’s own disclosure. (EX1003, ¶¶80-81.) In fact, the only two disclosures in the ’321 patent addressing

more than one independent sequence, Figures 2 and 3a, show only a component (*i.e.*, the image number) of the identifier being reset.



(EX1001, Fig.2 (annotated).)

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, only a component of the '321 patent's disclosed identifiers, the image number, resets. The subsequence identifier component does not reset. (EX1003, ¶82.)

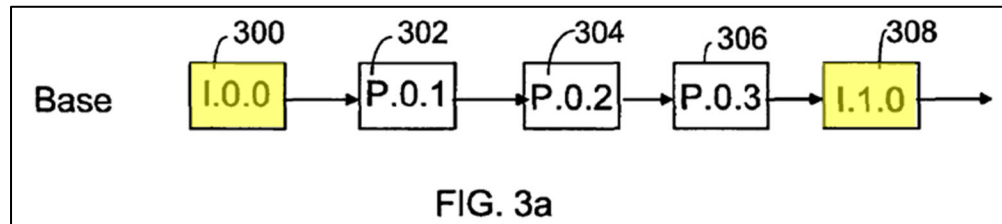


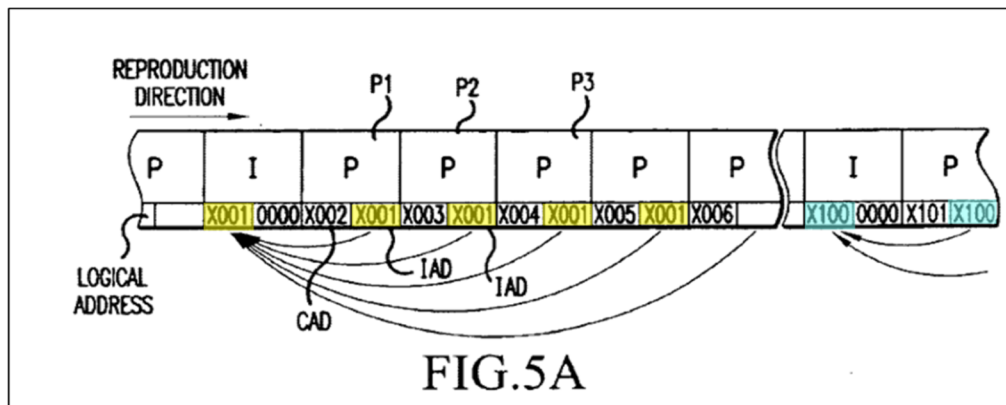
Figure 3a of the '321 patent likewise shows that, within the meaning of the '321 patent, “resetting an identifier value” includes resetting a component of an identifier value. (EX1003, ¶83.) The '321 patent states that “[t]he video sequence comprises a first image frame 300 which is an I-frame (I.0.0) 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.

Accordingly, Kim teaches or suggests 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. (EX1003, ¶84.)

For the reasons discussed above, claim 1 is unpatentable as obvious over Kim. (EX1003, ¶¶63-85.)

3. Claim 3: “A method according to claim 1, further comprising: encoding into the video sequence an identifier value for the independent sequence.”

As discussed above in Section VII(A)(2)(a), Kim discloses or suggests that each I-picture begins a new independent sequence. As also discussed, each I-picture has a unique CAD. For example, in Fig. 5A, the first I picture is numbered X001 and the later I frame shown is numbered X100.



(EX1006, Fig. 5A (annotated).) As also discussed above and shown in the figure, each P frame explicitly references its corresponding I frame in the IAD portion of its logical address. Thus every frame in every sequence includes an explicit reference to the CAD of the I-picture that begins the sequence. A POSITA would have understood that these explicit references to the CAD of the I-picture that marks the start of the sequence are “identifier value[s] for the independent sequence” as claimed. (EX1003, ¶88.) Again, using Fig. 5A as an example, every frame that is part of the independent sequence beginning with the first I-picture references the CAD for that I-picture (“X001”) in its logical address.

4. Claim 4

- a. 4[pre]: “A video encoder 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 encoder to encode a video sequence comprising an independent sequence ... by causing the video encoder to at least:”**

To the extent that the preamble is limiting, Kim teaches or suggests it. (EX1003, ¶¶89-93.) Kim discloses a Time Lapse recording/reproduction device that includes “an encoder 110 for generating I-picture data or P-picture data in frame units by using transformation and quantization of image data output from the frame memory 100”; “a recording-digital signal processor (DSP) 120 for extracting moving vectors from the image data output from the frame memory 100 and for generating P-picture data of the compressed image data”; and “a controller 150 for controlling the above-described components based on the commands from the user.” (EX1006, 3:3-27, Fig. 2.)

A POSITA would have understood that the disclosed controller and corresponding encoder and recording DSP 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 encoders to encode a compressed video sequence. (EX1003, ¶91.) A DSP is a specialized type of computer processor that executes a stored program of computer instructions, often for performing signal processing

operations. (EX1003, ¶91; *see also, e.g.*, EX1012 at 339 (discussing use of DSPs for compression).) Thus, Kim’s 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 encoder to encode a video sequence.” (EX1003, ¶91.)

Furthermore, Kim teaches or suggests encoding image frames in accordance with MPEG. (EX1006, 4:15-19.) By the time of the alleged invention, software-based MPEG encoders were known and commercially available. (EX1003, ¶92; EX1011 at 363-381; EX1015; EX1022 at 99 (discussing availability of MPEG “[e]ncoder and decoder PC boards”).) By 1998, source code was available for a software-based reference encoder for MPEG-2. (*See* EX1017.) Accordingly, it would have been obvious to a POSITA to implement an MPEG encoder with a processor, memory and computer code. By the same token, it would have been obvious to a POSITA to similarly implement Kim’s encoder with a processor, memory and computer code as claimed. (EX1003, ¶92.)

Claim 4[pre] recites the same “independent sequence” as 1[pre] which is disclosed by Kim as discussed above. (EX1003, ¶¶93.)

- b. 4[a]: “encode into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;”**

This limitation is substantially identical to 1[a] and is obvious for the same reasons discussed in Section VII(A)(2)(b). (*See also* EX1003, ¶95.)

- c. 4[b]: “encode identifier values for the image frames according to a numbering scheme; and”**

This limitation is substantially identical to 1[b] and is obvious for the same reasons discussed in Section VII(A)(2)(c). (*See also* EX1003, ¶97.)

- d. 4[c]: “reset the identifier value for the indicated first image frame of the independent sequence.”**

This limitation is substantially identical to 1[c] and is obvious for the same reasons discussed in Section VII(A)(2)(d). (*See also* EX1003, ¶99.)

- 5. Claim 6: “A video encoder according to claim 4, wherein the video encoder is configured to encode into the video sequence an identifier value for the independent sequence.”**

Claim 6 adds a substantially identical limitation to claim 3 and is obvious for the same reasons discussed in Section VII(A)(3). (*See also* EX1003, ¶101.)

6. Claim 7

- a. 7[pre]: “A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence comprising an independent sequence of image frames, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, the computer program product comprising:”**

To the extent that the preamble is limiting, Kim teaches or suggests it. (EX1003, ¶¶102-07.) As discussed in Section VII(A)(2)(a) for 1[pre], Kim teaches “encoding a compressed video sequence comprising an independent sequence of image frames” Kim also teaches the additional elements of 7[pre].

Kim discloses a Time Lapse recording/reproduction device that includes “an encoder 110 for generating I-picture data or P-picture data in frame units by using transformation and quantization of image data output from the frame memory 100”; “a recording-digital signal processor (DSP) 120 for extracting moving vectors from the image data output from the frame memory 100 and for generating P-picture data of the compressed image data”; and “a controller 150 for controlling the above-described components based on the commands from the user.” (EX1006, 3:3-27, Fig. 2.)

A POSITA would have understood these components to be “data processing devices,” given Kim’s disclosure that they all process data in the form of image data or commands. (EX1003, ¶105.)

And as Dr. Havlicek explains, a POSITA would have found it obvious to implement Kim’s encoder 110, recording DSP 120, and controller 150 as “[a] computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence” (EX1003, ¶¶106-107.) As discussed in Section VII(A)(4)(a), a DSP is a specialized type of computer processor that executes a stored program of computer instructions, often for performing signal processing operations. A POSITA would have understood that Kim’s recording DSP 120 performs its portion of the encoding process by executing a “computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for encoding a video sequence” (EX1003, ¶106.) Likewise, a POSITA would have found it obvious to implement the remaining video encoding functions performed by Kim’s encoder 110 and controller 150 as a stored computer program running on a general purpose processor that was available at the time. (EX1003, ¶¶106-107.) 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. (EX1003, ¶106.)

Furthermore, Kim discloses encoding image frames in accordance with MPEG. (EX1006, 4:15-19.) By the time of the alleged invention, software-based MPEG encoders were known and commercially available. (EX1003, ¶107; EX1011 at 363-381; EX1015.) By 1998, source code was available for a software-based reference encoder for MPEG-2. (*See* EX1017.) It would have been obvious to a POSITA to implement an MPEG encoder with a processor, memory and computer code as claimed. By the same token, it would have been obvious to a POSITA to similarly implement Kim's encoder with a processor, memory and computer code as claimed. (EX1003, ¶107.)

- b. 7[a]: “a computer program code for encoding into the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence;”**

This limitation is substantially identical to 1[a] and is obvious for the same reasons discussed in Section VII(A)(2)(b). (*See also* EX1003, ¶109.)

- c. 7[b]: “a computer code for encoding identifier values for the image frames according to a numbering scheme; and”**

This limitation is substantially identical to 1[b] and is obvious for the same reasons discussed in Section VII(A)(2)(c). (*See also* EX1003, ¶111.)

- d. 7[c]: “a computer code for resetting the identifier value for the indicated first image frame of the independent sequence.”**

This limitation is substantially identical to 1[c] and is obvious for the same reasons discussed in Section VII(A)(2)(d). (*See also* EX1003, ¶113.)

7. Claim 8

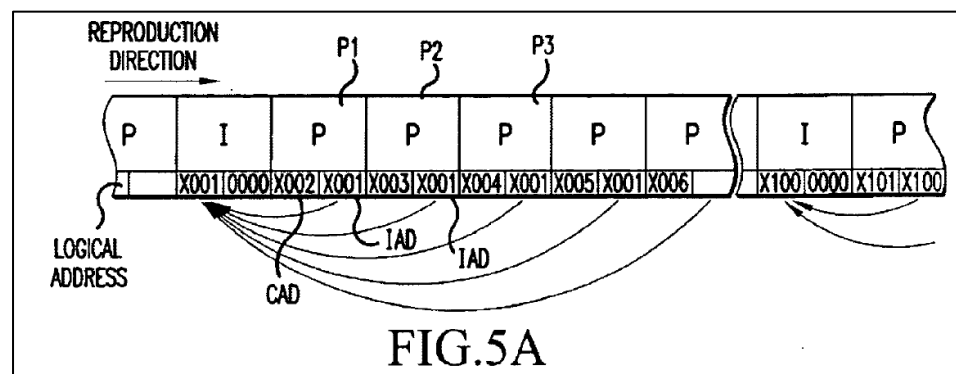
Claim 8 is the first of the '321 patent's decoder-side claims. The decoder claims recite similar limitations to the encoder claims discussed above and are obvious for similar reasons.

- a. 8[pre]: “A method for decoding a compressed video sequence, the method comprising:”**

To the extent the preamble is limiting, Kim discloses or suggests it. (EX1003, ¶¶115-16.) 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 or suggests a “method for decoding” its compressed intermittent data (“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;”

Kim teaches or suggests 8[a]. (EX1003, ¶¶117-25.) Kim’s Figure 5A (reproduced below) shows an example of a logical data structure of a digital data stream. (EX1006, 2:48-51.) As explained below, this figure shows the recited “indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.”

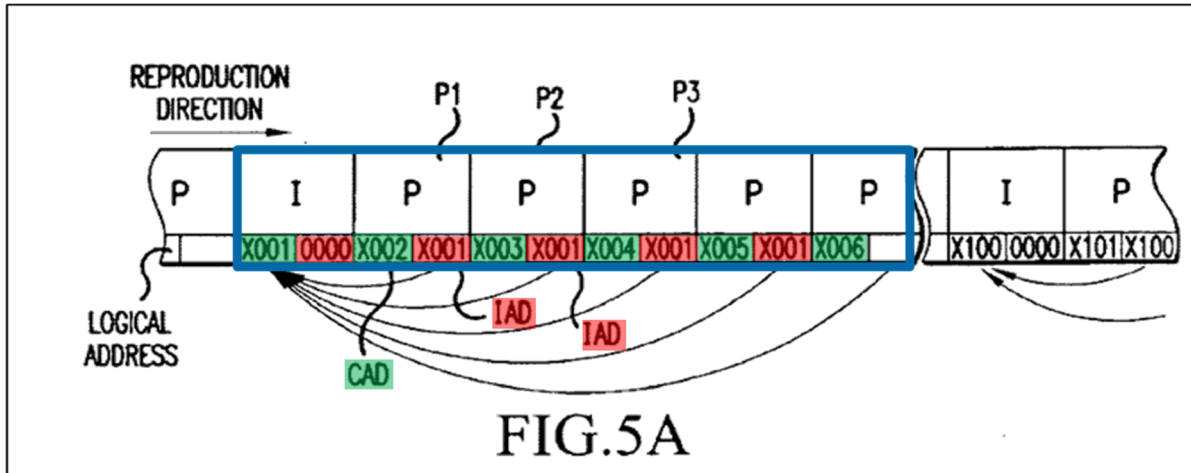


(EX1006, Fig. 5A.)

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”) 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." (EX1003, ¶¶119-22.)

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; EX1003, ¶120.) 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.) Thus, each P-picture's IAD identifies the I-picture on which the P-picture depends for motion compensation. (EX1003, ¶120.) This is further shown in annotated Figure 5A below, where every P-picture's IAD identifies the address of the preceding I-picture as further shown by the 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); EX1003, ¶120.) Thus, every P-picture in Kim's bitstream depends on the preceding I-picture.

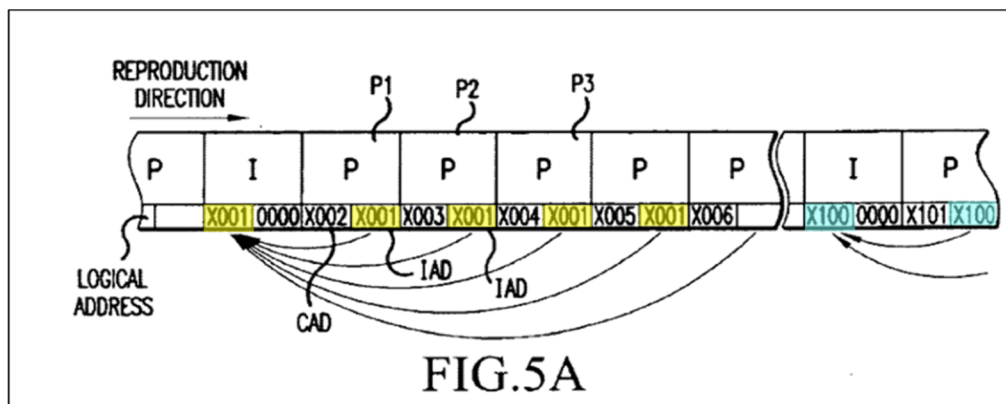


(EX1006, Fig. 5A (annotated).)

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. (EX1003, ¶121.) Accordingly, Kim discloses or suggests a group of frames where every P-picture depends only on the preceding I-picture for motion prediction (“an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within the independent sequence”).

Kim also teaches the recited “indication.” (EX1003, ¶122.) 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.” Kim discloses that each image frame has a CAD and an IAD. (EX1006 4:46-52, 4:59-5:7, Figs. 5A, 6.) Kim’s Figure 5A shows that each I-picture has an

IAD of 0000. (EX1006, Fig. 5A; EX1006, 4:59-5:1 (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 or zero indicates that a given I-picture 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).)

A POSITA would have understood that the decoder disclosed in Kim decodes the IAD of the image frames. (EX1003, ¶123.) Kim’s Figure 5A provides an “example[] of a logical data structure of a digital data stream” (EX1006, 2:48-51), and “FIG. 6 shows an example of a data structure of the bit stream shown in FIG[[]].

5A” (EX1006, 4:59-61). Kim then explains the process for decoding this bitstream. (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. (EX1003, ¶¶122-23.)

Accordingly, Kim teaches or suggests 8[a] by disclosing the decoding the IAD value of 0000 for each I-picture as shown, for example, in Kim’s Figure 5A.

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 that group 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.” (EX1003, ¶125.)

- 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;”**

Kim discloses or suggests 8[b]. (EX1003, ¶¶126-37.)

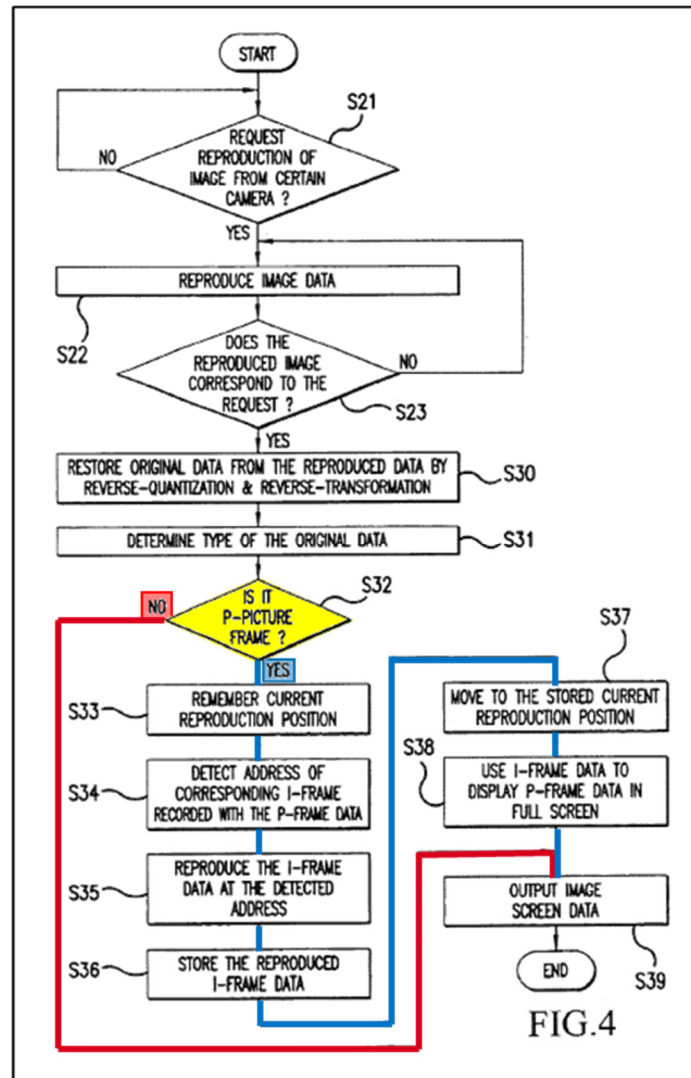
As discussed in Section VII(A)(7)(b) for 8[a], Kim’s disclosures of a grouping of an I-picture followed by P-pictures teach the recited “independent sequence.” As to 8[b], Kim teaches that the decoding of that grouping begins with the I-picture

(“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].

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. (EX1003, ¶127.) A POSITA would have known that video playback involves displaying a sequence of still images over time. (EX1001, 1:55-58.) Further, as discussed in Section VII(A)(1), a POSITA would have known that time lapse video sequences can be played back and that the time lapse video systems Kim describes included that capability. Kim discloses that the flowchart in Figure 4 depicts a reproducing method for a data structure, such as 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 would have understood that the decoding steps in Figure 4 can be repeated for each image in an independent sequence shown in Figure 5A. *See Perfect Web Techs., Inc. v. InfoUSA, Inc.*, 587 F.3d 1324, 1330 (Fed. Cir. 2009) (finding that a POSITA’s common sense taught repeating an element existing in the

prior art until success was achieved); *see also B/E Aerospace, Inc. v. C&D Zodiac, Inc.*, 962 F.3d 1373, 1380-81 (Fed. Cir. 2020) (same). Moreover, as noted above, decoding video sequences including independent sequences (*e.g.*, closed GOPs in MPEG-1) was well known at the relevant time. (EX1003, ¶129.) 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. (EX1003, ¶129.)

Kim's Figure 4 is reproduced below:



(EX1006, Fig. 4 (annotated).)

Kim's first path discloses that the first image decoded and output for display is an I-picture. (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. (EX1006, Figs. 5A, 6.) As explained above in Section VII(A)(7)(b), Kim's Figure 5A shows that each independent sequence begins with an I-picture. Accordingly, this path when applied to Figure 5A's independent sequence, which begins by decoding an I-picture, discloses "starting the decoding of the video sequence from said first image frame of the independent sequence." (EX1003, ¶¶130-31.)

Although Kim's disclosure of the first path is sufficient to teach or suggest this limitation, the second path also discloses "starting the decoding of the video sequence from said first image frame of the independent sequence."

The second path reflects the system's behavior when the user requests reproduction of a P-picture. (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.) A POSITA would have understood this second path when applied to Figure 5A's independent sequence 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. (EX1003, ¶¶132-33.) Thus, based on either path from Step S32 in Kim’s Figure 4 when applied to Figure 5A’s independent sequence, Kim discloses or suggests “starting the decoding of the video sequence from said first image frame of the independent sequence.”

As discussed 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. Alternatively, 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. (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. (EX1003, ¶¶133-34.) This is sufficient to teach “starting the decoding of the video sequence from said first image frame of the independent sequence.”

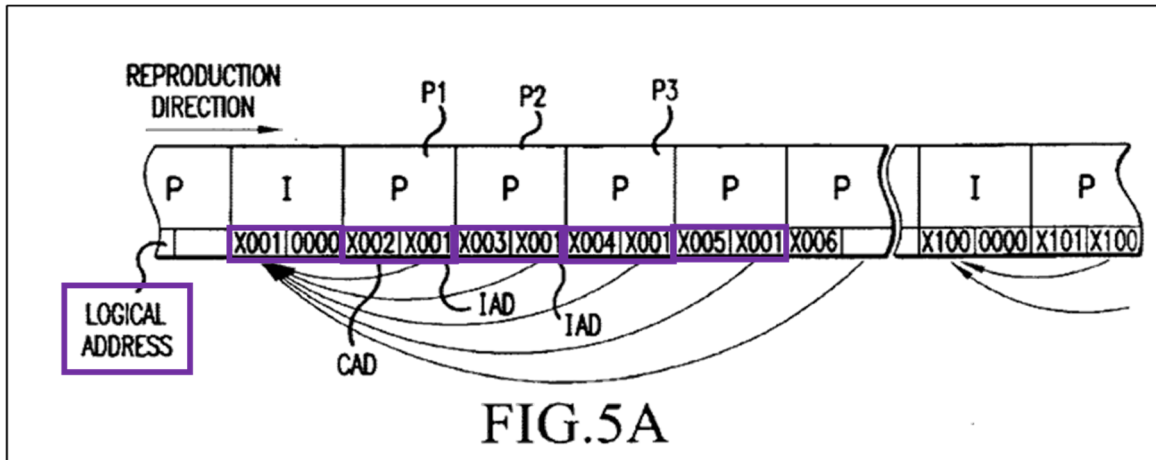
Kim also discloses or suggests that the video sequence is decoded without prediction from any image frame decoded prior to said first image frame, as recited in 8[b]. As explained in Section VII(A)(7)(b), each independent sequence in Kim's bitstream consists of an I-picture followed by a series of P-pictures that depend on the I-picture for prediction. (EX1006, Fig. 5A.) The I-picture that begins the sequence is intra-coded and thus does not depend on any other picture at all. (EX1006, 3:60-61.) And because every P-picture refers to the first I-picture in each independent sequence, no P-picture requires prediction from any image frame decoded prior to the first I-picture. (EX1003, ¶¶135-36.) Accordingly, Kim discloses that "the video sequence is decoded without prediction from any image frame decoded prior to said first image frame."

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." (EX1003, ¶¶127-37.)

d. 8[c]: “decoding identifier values for image frames according to a numbering scheme; and”

Kim teaches or suggests 8[c]. (EX1003, ¶¶138-45.) Specifically, Kim’s “logical addresses” are “identifier values for image frames,” and they are decoded “according to a numbering scheme.”

Kim teaches or suggests that pictures have “logical addresses” that are assigned according to a numbering scheme. (EX1003, ¶139.) 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. (EX1006, 4:48-52.) CAD increments by 1 for each picture in reproduction order. (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. (EX1006, Fig. 5A.) And for P-pictures, the IAD provides the CAD of the corresponding I-picture. (EX1006, 4:48-52.) Accordingly, the pictures (i.e., image frames) in Kim’s disclosed bitstream have logical addresses that follow a numbering scheme. (EX1003, ¶139.)

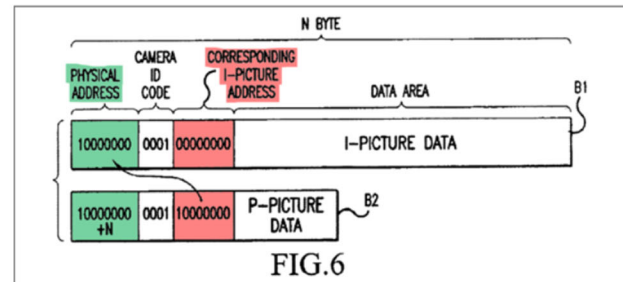
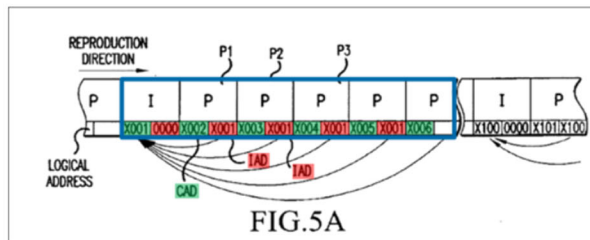


(EX1006, Fig. 5A (annotated).)

Kim discloses or suggests that logical addresses are decoded according to a numbering scheme. (EX1003, ¶140.) Specifically, Kim discloses that when a P-picture is decoded, “the decoder detects the address of the corresponding I-picture frame (e.g., data stored in the area (3) of the bit stream B2 shown in Fig. 6) ... and reproduces based on the detected address” (EX1006, 5:41-46.) Kim thus teaches or suggests that 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. (EX1003, ¶140.) Accordingly, Kim discloses that logical addresses are decoded according to a numbering scheme.

In IPR2024-00691, Patent Owner tried to argue that Kim discloses decoding the “physical address” depicted in Fig. 6 rather than the “logical address” depicted in Fig. 5A. That argument lacks merit because the “physical address” in Fig. 6 clearly refers to the same CAD in Fig. 5A. (EX1003, ¶¶141-42.) Consistent with

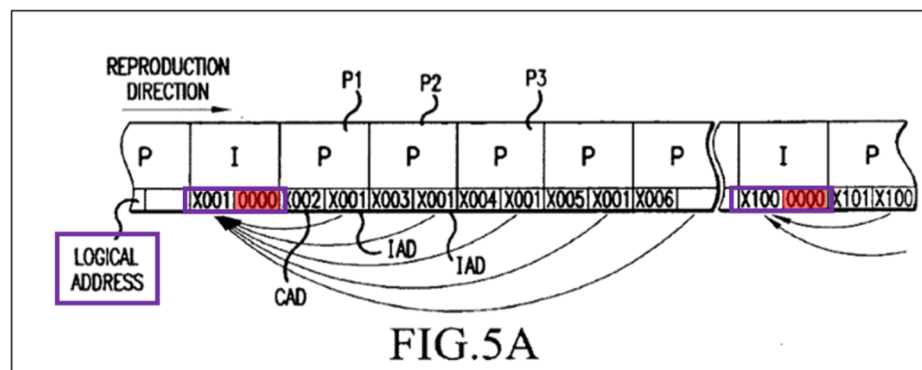
Kim's Fig. 5A and its description of that figure, Kim's Fig. 6 shows how the I-Picture Address (IAD) for the P-picture references the "Physical Address" (or CAD) of the corresponding I-picture.



(EX1006, Figs. 5A & 6.) Moreover, Kim teaches assigning picture addresses during the encoding process before the bitstream is ever written to the storage medium and nothing in Kim suggests the encoder has or needs advance knowledge of specific disk addresses where the pictures will be stored. (EX1003, ¶144.) If the bitstream relied on specific disk addresses for each picture, the bitstream would need to be reencoded every time it was stored on a different medium. (EX1003, ¶144.) Finally, even if Kim's lone reference to a "physical address" was understood to refer to a specific disk address, a POSITA would have understood the CAD in Fig. 5A to use the same address such that the CAD would still be decoded as discussed above. (EX1003, ¶145.) Patent Owner's attempt to distinguish between the CAD in Fig. 5A and the physical address in Fig. 6 should be rejected.

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

Kim teaches or suggests 8[d]. (EX1003, ¶¶146-51.) Specifically, Kim’s Figure 5A shows that the IAD component of each logical address is reset to zero for each I-picture. (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 or suggests that the IAD component of each logical address (“identifier value”) is reset for the indicated first image frame of the independent sequence. (EX1003, ¶146.)



(EX1006, Fig.5A.)

As discussed above for 1[c], Kim’s resetting the IAD portion of the logical address is similar to the ’321 patent’s disclosure of resetting the image number portion of the identifier. (*See supra* § VII(A)(2)(d); EX1003, ¶¶147-48.)

In IPR2024-00691, Patent Owner argued that the claim language requires the *decoder* to do the resetting whereas Kim discloses that the Temporal Reference for

the first picture in the GOP is set by the *encoder*. But to the extent Patent Owner suggests that the claimed “resetting” step requires the decoder to modify the identifier assigned by the encoder, the ’321 patent provides no support for that interpretation. The only time that the patent describes “reset[ting]” the identifier value for the indicated first picture of an independent sequence according to a numbering scheme, it is referring to an *encoding* process. (EX1001, 4:42-45; *see also id.* claim 1.) As to decoding, the patent states “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.)³ Importantly, however, the patent repeatedly and consistently teaches that the image number of the first image in each sub-sequence is always zero. (EX1001, 8:19-22 (“Next, *within each sub-sequence*, the video frames of the sub-sequence are given image numbers, using for example consecutive numbering that starts with *the number 0 given to the first video frame of the sub-sequence.*”); 9:21-22 (“Since the P-frame 210 is the first image frame in the sub-sequence, the image number of the P-frame 210 is 0.”); 8:38-40; 8:57-59; 9:13-15; 9:35-37.) Every I frame and every

³ Although this disclosure uses the term “set” rather than “reset,” Patent Owner identified this disclosure (and only this disclosure) during prosecution as the support for the resetting limitation in the decoder claims. (EX1002, 776.)

P frame that starts a sub-sequence in the figures has an image number of zero. (Ex1001, Figs. 2-6.) Accordingly, when the “decoder ... sets the identifier of the first image frame” to zero (EX1001, 11:37-40), it need not change the identifier value of zero set by the encoder. (EX1003, ¶149.) Patent Owner’s contrary interpretation would exclude the preferred embodiment and the sole disclosure Patent Owner identified as the support for the resetting limitation. *See Adams Respiratory Therapeutics, Inc. v. Perrigo Co.*, 616 F.3d 1283, 1290 (Fed. Cir. 2010) (“A claim construction that excludes the preferred embodiment ‘is rarely, if ever, correct and would require highly persuasive evidentiary support.’”) (citation omitted).

A POSITA would have understood that, consistent with the disclosure of the ’321 patent itself, Kim’s decoder resets the identifier for the first image frame (an I-frame) of the exemplary closed GOP during decoding. Kim’s decoder stores I-pictures in frame memory 101 for use in decoding subsequent P-pictures. (EX1006, 5:48-52, 5:55-59.) It would have been obvious to a POSITA for Kim’s decoder to store the IAD along with the I-picture so the decoder could determine whether the I-picture needed to decode a given P-picture is already present in frame memory. (EX1003, ¶¶150.) Setting the IAD for the I-picture (*e.g.*, 0000 for the first I-picture in Fig. 5A) in frame memory 101 sets (or resets) the identifier in the same manner described by the ’321 patent. (EX1001, 11:37-40; EX1003, ¶150.)

Accordingly, Kim teaches or suggests 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. (EX1003, ¶¶146-50.)

For the reasons discussed above, claim 8 is unpatentable as obvious over Kim. (EX1003, ¶¶114-51.)

8. Claim 10

Claim 10 is substantively the same as claim 8, except 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.” (EX1003, ¶153.)

- 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:”**

To the extent that the preamble is limiting, Kim teaches or suggests it. (EX1003, ¶¶154-57.) As discussed in Section VII(A)(7)(a) for 8[pre], Kim teaches “decoding a compressed video sequence.” Kim teaches the additional elements of 10[pre].

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.)

A POSITA would have understood that the disclosed controller and corresponding decoder and reproducing DSP 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. (EX1003, ¶¶155-56.) As discussed in Section VII(A)(4)(a), a DSP is a specialized type of computer processor that executes a stored program of computer instructions for performing signal processing operations. (EX1003, ¶156.) Thus, Kim’s 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.”

Furthermore, Kim discloses encoding image frames in accordance with MPEG (EX1006, 4:15-19) and subsequently decoding those image frames (EX1006, 5:8-68). By the time of the alleged invention, software-based MPEG decoders were

known and commercially available. (EX1003, ¶157; EX1011 at 363-381; EX1007; EX1022 at 99 (discussing availability of MPEG “[e]ncoder and decoder PC boards”).) For example, source code was available for a software-based reference decoder for MPEG-2. (*See* EX1017.) Accordingly, it would have been obvious to a POSITA to implement Kim’s decoder with a processor, memory and computer code as claimed. (EX1003, ¶157.)

- 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;”**

This limitation is substantially identical to 8[a] and is obvious for the same reasons discussed in Section VII(A)(7)(b). (*See also* EX1003, ¶158.)

- 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; and”**

This limitation is substantially identical to 8[b] and is obvious for the same reasons discussed in Section VII(A)(7)(c). (*See also* EX1003, ¶159.)

d. 10[c]: “decode identifier values for image frames according to a numbering scheme; and”

This limitation is substantially identical to 8[c] and is obvious for the same reasons discussed in Section VII(A)(7)(d). (*See also* EX1003, ¶160.)

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

This limitation is substantially identical to 8[d] and is obvious for the same reasons discussed in Section VII(A)(7)(e). (*See also* EX1003, ¶161.)

For the reasons discussed above, claim 10 is unpatentable as obvious over Kim. (EX1003, ¶¶156-62.)

9. Claim 11

Claim 11 is substantively the same as claim 8, except 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.” (EX1003, ¶163.)

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:”

To the extent that the preamble is limiting, Kim teaches or suggests it. (EX1003, ¶¶164-72.) As discussed in Section VII(A)(7)(a) for 8[pre], Kim teaches

“decoding a compressed video sequence.” Kim teaches the additional elements of 11[pre].

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.)

A POSITA would have understood these components to be “data processing devices,” given Kim’s disclosure that they all process data in the form of image data or commands. (EX1003, ¶166.)

And as Dr. Havlicek explains, 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, for decoding a compressed video sequence.” (EX1003, ¶¶167-71.) As discussed in Section VII(A)(4)(a), 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.” (EX1003, ¶¶167-68.) Likewise, a POSITA would have found it obvious to implement the remaining video decoding functions performed by Kim’s decoder 111 and controller 150 as a stored computer program running on a general purpose processor that was available at the time. (EX1003, ¶¶169-71.) 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. (EX1003, ¶170.)

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. By the time of the alleged invention, software-based MPEG decoders were known and commercially available. (EX1003, ¶172; EX1011 at 363-381; EX1007.) For example, source code was available for a software-based reference decoder for MPEG-2. (*See* EX1017.) Accordingly, 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.” (EX1003, ¶172.)

- 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;”**

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 supra* Section VII(A)(7)(b).) As explained in Section VII(A)(7)(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 explained in Section VII(A)(9)(a), a POSITA would have recognized 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. (EX1003, ¶173.)

- 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;”**

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. As explained in Section VII(A)(7)(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 explained in Section VII(A)(9)(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. (EX1003, ¶174.)

- d. **11[c]: “a computer program code for decoding identifier values for image frames according to a numbering scheme; and”**

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. As explained in Section VII(A)(7)(d), Kim teaches “decoding identifier values for image frames according to a numbering scheme.” And as explained in Section VII(A)(9)(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. (EX1003, ¶175.)

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

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. As explained in Section VII(A)(9)(e), Kim teaches “resetting the identifier value for the indicated first image frame of the independent sequence.” And as explained in Section VII(A)(9)(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. (EX1003, ¶176.)

For the reasons discussed above, claim 11 is unpatentable as obvious over Kim. (EX1003, ¶¶163-77.)

B. Ground 2: Kim in view of Adolph Renders Obvious Claims 2, 5, and 9

1. Adolph

Adolph discloses a video coding technique for reducing resynchronization delays caused by data loss during decoding by inserting important picture-level information into the slice headers of a video sequence. (EX1008, Abstract.) Adolph explains that issues such as channel interference can cause loss of picture level data

that leads to “undesirably long failure times [can] arise in the decoder which is used, as a result of consequential errors occurring.” (EX1008, 1:11-21.) (EX1003, ¶178.)

Adolph proposes adding certain important data to the slice headers in an MPEG bit stream “to achieve a shorter resynchronization time and a reduction in consequential errors.” (EX1008, 1:32-35.) “[A]dditional information is inserted, according to the invention, at an advantageous point into the slice header, which information contains the *most important information about the respective frame and thus allows decoding* even in the event of loss of the picture start code or extension start code” (EX1008, 2:3-11.) The additional information can “relate[] ... to the frame type (I-, P-, or B-frame) and/or to important parameters for decoding of movement vectors.” (EX1008, 2:41-45.) Adolph’s technique “complies with the MPEG standard, in particular the MPEG2 standard.” (EX1008, 2:31-33.) (EX1003, ¶179.)

Adolph’s Figure 3 (below) shows the insertion of picture-related information in the slice header including the “temporal[] reference, in which an integer number indicates the chronological arrangement of the encoded frame” and the “picture coding type, which indicates whether the frame is I-, P-, or B-encoded.” (EX1008, 4:1-15.) (EX1003, ¶180.)

<u>EXTRA INFORMATION SLICE</u>	
<u>PICTURE HEADER:</u>	BITS
TEMPORAL_REFERENCE	8
PICTURE_CODING_TYPE	2
<u>PICTURE CODING EXTENSION:</u>	
FORWARD_HORIZONTAL_f_CODE	4
FORWARD_VERTICAL_f_CODE	4
BACKWARD_HORIZONTAL_f_CODE	4
BACKWARD_VERTICAL_f_CODE	4
INTRA_DC_PREDICTION	2
CONCEALMENT_MOTION_VECTORS	1
q_SCALE_TYPE	1
INTRA_VLC_FORMAT	1
ALTERNATE_SCAN	1

INFORMATION CONTENTS OF THE SLICE HEADER

FIG. 3

(EX1008, Fig. 3 (annotated); 4:1-15.) Adolph teaches including picture-level information in the MPEG standard “extra_information_slice” field in the slice header. (EX1008, 3:60-63; Fig. 2.) To increase coding efficiency, the information may be included only in the headers of certain slices (*e.g.*, every other macroblock line or every third macroblock line). (EX1008, 2:37-41.)

A decoder can use the additional information provided in the slice header to resynchronize even if the picture header that normally provides this information is lost. (EX1008, 5:19-27.) As a result, “[l]ong resynchronization times are avoided ...

and consequential errors in the decoding resulting from fault predictions are reduced.” (EX1008, 5:26-29.) (EX1003, ¶182.)

2. Motivation to Combine Kim and Adolph

It would have been obvious to a POSITA to combine the teachings of Kim and Adolph. (EX1003, ¶¶183-90.) Kim teaches or suggests the use of MPEG in encoding video data. Moreover, by the time of the alleged invention, MPEG was well-established as a leading video compression standard. (EX1003, ¶184; *see also* EX1022, *e.g.*, at 83 (“The MPEG-1 and MPEG-2 video-compression techniques have developed into important and successful coding standards world-wide, with an increasing number of MPEG-1 and MPEG-2 VLSI chip-sets and products becoming available on the market.”).)

Adolph describes a straightforward error resilience technique to shorten resynchronization time by inserting “the most important information about the respective frame[s]” in slice headers in an MPEG bitstream. Kim discloses sequences of video frames consisting of an I frame followed by P frames that are differentially encoded relative to the I frame and distinguishes the I frames with an IAD of 0000. Kim further discloses picture-level address information that indicates the corresponding I frame for each P frame and that this information is needed during decoding. In particular, as discussed above in Sections VII(A)(1) and VII(A)(7)(d), a given P frame cannot be decoded without first decoding its corresponding I frame

and that I frame is indicated in the address information for the P frame. (EX1003, ¶¶ 55-62; 146-50; 185.) Because in Kim's approach every P-picture in the independent sequence relies on the same I-picture, data loss related to the I-frame (e.g., it's CAD) would have drastic impacts on the decoding of the rest of the frames in the sequence. Adolph's error resilience technique would provide information that might allow the decoder to decode the sequence properly even if the CAD stored at the picture-level is lost or corrupted. (EX1003, ¶185.)

It would have been obvious to a POSITA to apply Adolph's teachings by inserting "most important" Kim's picture-level information (*i.e.*, IAD and CAD) in the slice headers as taught by Adolph. (EX1003, ¶186.) Inserting a picture IAD with a value of 0000 (as Kim discloses) into the slice header (as Adolph discloses) would constitute a separate flag as claimed. (EX1003, ¶186.) To reduce the level of redundancy and improve coding efficiency, the information could be included only in a subset of slice headers as Adolph teaches. (EX1003, ¶186.)

Even if Patent Owner argues that the claimed "flag" must be a *single-bit* flag, this also would have been obvious. Adolph specifically teaches inserting the picture type in the slice header. (EX1008, 2:41-45, 4:5-7; Fig. 3.) Adolph further teaches minimizing the number of bits used for the information added to the slice header. (EX1008, 2:8-11, 2:45-51, 4:7-15.) Given that Kim's coding method uses only two picture types (I- and P- pictures), it would have been obvious to a POSITA that

Kim's picture type could be represented with a one-bit flag. (EX1003, ¶186.) For example, a value of '1' could indicate the current picture is an I-picture and a value of '0' could indicate that the current picture is a P-picture.⁴ This approach would efficiently allow for the picture type to be coded with a single bit in the slice header in order to minimize the overhead associated with the information added to the slice headers. (EX1003, ¶187.) A POSITA would have understood that such an "I-picture" flag would provide an equivalent, but more efficient, way of indicating an I-frame than using multiple zero bits to indicate the I-frame as Kim describes and shows in Figs. 5A, 5B, and 6. (EX1003, ¶187.)

A POSITA would have had a reasonable expectation of success in combining the teachings of Kim and Adolph as proposed. As mentioned, Kim suggests the use of MPEG standard and Adolph's MPEG-compliant technique exploits an already-available slice header field provided for in the MPEG standard to insert the additional information. (See EX1016, 30 (showing `extra_information_slice` as part of Picture Header), 59 (describing `extra_information_slice` field).) If a generic

⁴ One-bit flags like this were commonplace in the MPEG video standards. (See, e.g., EX1016, 43 (`constrained_parameters_flag`), 44 (`low_delay`, `colour_description`), 49 (`mux_to_progressive_sequence`, `drop_frame`, `closed_gop`, `broken_link`).)

MPEG decoder is not designed to make use of the additional information in the slice header, Adolph teaches the decoder will simply ignore the additional information. (EX1008 at 5:29-32; EX1003, ¶188.) The proposed combination would provide the same information Kim describes to facilitate Kim’s decoding method. (EX1003, ¶188.)

Finally, Kim and Adolph are directed to similar digital video devices that utilize optical disks. Kim describes time lapse recording systems which record digital video to and/or read digital video from optical disks. (EX1006, 1:8-11.) Adolph suggests application of its error resilience techniques to systems using “digital storage media such as DVD.” (EX1008, 5:36-38; EX1003, ¶¶189-90.)

3. Claim 2: “A method according to claim 1, further comprising: encoding the indication into the video sequence as a separate flag included in the header of a slice.”

As just discussed, it would have been obvious from Kim and Adolph to include the picture-level IAD (i.e., 0000) that indicates the first picture in the independent sequence as a flag in the slice header. Even if Patent Owner argues that the claim requires a *one-bit* flag, it would have been obvious to insert a one-bit flag (e.g., an “I-picture” flag) in the slice header to distinguish between Kim’s two picture types. This would have been consistent with Adolph’s teachings of (1) indicating the frame type in the slice header (EX1008, 2:41-45, 4:5-7; Fig. 3), and (2)

minimizing the number of bits used for the information added to the slice header (EX1008, 2:8-11, 2:45-51, 4:7-15). (EX1003, ¶¶191-193.)

- 4. Claim 5: “A video encoder according to claim 4, wherein the video encoder is configured to encode the indication into the video sequence as a separate flag included in the header of a slice.”**

Claim 5 adds a substantially identical limitation to claim 2 and is obvious for the same reasons discussed in Section VII(B)(3). (*See also* EX1003, ¶194.)

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

Claim 9 adds a substantially identical limitation to claim 2 and is obvious for the same reasons discussed in Section VII(B)(3). (*See also* EX1003, ¶195.)

VIII. CONCLUSION

Inter Partes Review of the Challenged Claims is respectfully requested.

IX. MANDATORY NOTICES

A. Real Party in Interest

Petitioners identify themselves as real parties in interest. Out of an abundance of caution, Petitioner Hisense USA Corporation also identifies Hisense Co., Ltd., which entity was also sued by Nokia for alleged infringement of the '321 patent.

B. Related Matters

To the best of Petitioners' knowledge, the '321 patent is presently or has been involved in the following related matters:

Case	Filing Date
<i>Certain Video-Capable Laptop, Desktop Computers, Handheld Computers, Tablets, Televisions, Projectors, and Components and Modules Thereof</i> Inv. No. 337-TA-3821 (ITC)	4/11/2025
<i>Nokia Technologies Oy v. Hisense Co. Ltd. et al</i> No. 1-25-cv-01871 (NDGA)	4/7/2025
<i>Nokia Technologies Oy v. ASUSTeK Computer Inc. et al</i> No. 2-25-cv-03053 (CDCA)	4/7/2025
<i>Nokia Technologies Oy v. Acer Inc. et al</i> No. 1-25-cv-00523 (WDTX)	4/7/2025
<i>Element Television Company, LLC et al v. Nokia Corporation a/k/a Nokia of America Corporation et al</i> No. 0-24-cv-04269 (DMN)	11/25/2024
<i>Amazon.com, Inc. v. Nokia Technologies Oy</i> , IPR2024-00691 (PTAB)	3/19/2024
<i>Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof</i> Inv. No. 337-TA-1380 (ITC)	10/31/2023
<i>Nokia Corporation et al v. Amazon.com, Inc.</i> No. 1-23-cv-01232 (DDE)	10/27/2023
<i>Nokia Technologies Oy v. HP Inc. f/k/a Hewlett-Packard Company</i> No. 1-23-cv-01237 (DDE)	10/31/2023
<i>Nokia Technologies Oy v. Amazon.com, Inc. et al</i> No. 1-23-cv-01236 (DDE)	10/31/2023
<i>Element TV Company, LP et al. v. Nokia Corporation et al.</i> , IPR2025-01068 (PTAB)	6/4/2025

C. Notice of Counsel and Service Information

LEAD COUNSEL
Patrick J. McKeever (Reg. No. 66,019) PERKINS COIE LLP 11452 El Camino Real, Suite 300

San Diego, CA 92130-2080 Telephone: (858) 720-5722 mckeever-ptab@perkinscoie.com	
BACK-UP COUNSEL	
Brian E. Ferguson (Reg. No. 36,801) WINSTON & STRAWN LLP 1901 L Street, N.W. Washington, D.C. 20036 T: 202-282-5276 beferguson@winston.com	Wei Yuan (Reg. No. 71,772) PERKINS COIE LLP 11452 El Camino Real, Suite 300 San Diego, CA 92130-2080 Telephone: (858) 720-5798 yuan-ptab@perkinscoie.com
Joseph C. Masullo (<i>pro hac vice to be submitted</i>) WINSTON & STRAWN LLP 1901 L Street, N.W. Washington, D.C. 20036 T: 202-282-5897 jmasullo@winston.com	Nicholas Lipperd (Reg. No. 83,084) PERKINS COIE LLP 11452 El Camino Real, Suite 300 San Diego, CA 92130-2080 Telephone: (858) 720-5798 lipperd-ptab@perkinscoie.com
Eimeric Reig (<i>pro hac vice to be submitted</i>) WINSTON & STRAWN LLP 101 California Street 21st Floor San Francisco, CA 94111 T: 415-591-6808 ereigplessis@winston.com	Hisense-NokiaIPR@perkinscoie.com
Lauren M. Rennecker (<i>pro hac vice to be submitted</i>) WINSTON & STRAWN LLP 1901 L Street, N.W.	Attorneys for Petitioner Hisense USA Corporation

Washington, D.C. 20036 T: 202-282-5897 lrennecker@winston.com Attorneys for Petitioner Snap Inc.	
---	--

Respectfully submitted,

/ Patrick J. McKeever /

Patrick J. McKeever
Reg. No. 66,019

PERKINS COIE LLP
11452 El Camino Real, Suite 300
San Diego, CA 92130-2080

Date: June 23, 2025

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(d)

Pursuant to 37 C.F.R. §42.24(a), Petitioners hereby certifies that portions of the above-captioned Petition for *Inter Partes* Review of U.S. Patent 8,050,321, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 11,948. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ *Patrick J. McKeever* /

Patrick J. McKeever
Reg. No. 66,019

Date: June 23, 2025
PERKINS COIE LLP
11452 El Camino Real, Suite 300
San Diego, CA 92130-2080

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *Inter Partes* Review of U.S. Patent No. 8,050,321 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

Nokia
c/o Muetting Raasch Group
111 WASHINGTON AVE. S., SUITE 700
Minneapolis, MN 55401

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Brian E. Farnan
bfarnan@farnanlaw.com
Michael J. Farnan
mfarnan@farnanlaw.com
Warren H. Lipschitz
wlipschitz@mckoolsmith.com
Alexandra F. Easley
aeasley@mckoolsmith.com
Joshua Budwin
jbudwin@mckoolsmith.com
R. Mitch Verboncoeur
mverboncoeur@mckoolsmith.com
Josh Newcomer
jnewcomer@mckoolsmith.com
Kevin Burgess
kburgess@mckoolsmith.com
Theodore Stevenson, III
ted.stevenson@alston.com
John D. Haynes
john.haynes@alston.com

Nicholas T. Tsui
nick.tsui@alston.com
Shawn Gannon
shawn.gannon@alston.com

Respectfully submitted,

/ Patrick J. McKeever /

Patrick J. McKeever
Reg. No. 66,019

Date: June 23, 2025