UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC., AMAZON.COM SERVICES, LLC,
Petitioner

v.

NOKIA TECHNOLOGIES OY,
Patent Owner

Case: IPR2024-00691
U.S. Patent No. 8,050,321

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 7

A. TECHNICAL BACKGROUND OF THE '321 PATENT 7

B. THE PRIOR ART 13

1. MPEG-1 13

2. Kim 20

C. ASSERTED GROUNDS 26

D. POSITA 27

E. CLAIM CONSTRUCTION 27

F. CLAIM TABLE 27

III. ARGUMENT 30

A. PETITIONER HAS NOT SHOWN THAT THE CHALLENGED CLAIMS ARE UNPATENTABLE IN LIGHT OF GROUND 1 (MPEG-1) 30

1. Petitioners fail to demonstrate that limitations 8[a], 10[a], and 11[a] are obvious over MPEG-1 30

2. Petitioner fails to demonstrate that limitations 8[b], 10[b], and 11[b] are obvious over MPEG-1 45

3. Petitioner fails to demonstrate that limitations 8[d], 10[d], and 11[d] are obvious over MPEG-1 because MPEG-1 does not teach or suggest resetting the identifier value as required 50

B. PETITIONER HAS NOT SHOWN THAT THE CHALLENGED CLAIMS ARE UNPATENTABLE IN LIGHT OF GROUND 2—KIM. 53

1. Petitioner fails to demonstrate that limitations 8[a], 10[a], and 11[a] are obvious over Kim 55

2. Petitioner fails to demonstrate that limitations 8[b], 10[b], and 11[b] are obvious over Kim 60

3. Petitioner fails to demonstrate that limitations 8[c], 10[c], and 11[c] are obvious over Kim 63

4. Petitioner fails to demonstrate that limitations 8[d], 10[d], and 11[d] are obvious over Kim because Kim does not teach or suggest resetting the identifier value as required 65

C. PETITIONER HAS NOT ESTABLISHED A REASONABLE LIKELIHOOD THAT THE CHALLENGED CLAIMS ARE UNPATENTABLE IN LIGHT OF GROUND 3.....68

1. Petitioner fails to demonstrate that Claim 9 is obvious over MPEG-1 in combination with Yagasaki 68

IV. CONCLUSION 68

TABLE OF AUTHORITIES

Page(s)

OTHER AUTHORITIES

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

PATENT OWNER's UPDATED EXHIBIT LIST

Exhibit[1]	Description
2001	Expert Declaration of Dr. Ian Richardson in Support of Patent Owner's Preliminary Response
2002	Biography of Steven Udick
2003	Declaration of Steven Udick in Support of Motion for *Pro Hac Vice*
2004	Expert Declaration of Dr. Ian Richardson in Support of Patent Owner's Response
2005	Deposition Transcript of Dr. Joseph Havlicek
2006	ITU-T Rec. H.264 (03/2005)

[1] Bolded exhibit numbers indicate exhibits used in this Patent Owner Response.

I. INTRODUCTION

Following the Board's Decision Granting Institution of *Inter Partes* Review, Patent Owner Nokia Technologies Oy sets forth the following substantive errors, omissions, and shortcomings underlying the grounds that Petitioner has asserted for invalidity argued in its Petition for *Inter Partes* Review of U.S. Patent No. 8,050,321 (Petition or Pet.). All three grounds are fatally flawed.

MPEG-1 does not invalidate Claims 8, 10, or 11 of the '321 Patent. Petitioners rely on the MPEG-1 video coding standard to attempt to show that the inventions of the '321 Patent are not novel, and, despite numerous evolutions of video coding with working groups of experts each step of the way, the claims of the '321 Patent are nothing more than what was done in MPEG-1. This is wrong, for numerous reasons.

Most devastating to Petitioner's argument is that MPEG-1 does not disclose the [d] limitations of the challenged claims—the decoder "resetting the identifier value" limitation. The reason is simple. Petitioners have failed to identify ***any*** disclosure in MPEG-1 that shows the ***decoder*** resetting the identifier value. Instead, Petitioners attempt to point to an identifier value of a second Group of Picture ("GOP")—one which the ***encoder*** sets to zero for each new GOP—to argue that the identifier value of the current GOP is ***reset*** to zero by a ***decoder***. Petitioner's

argument points to the wrong process of video coding (the encoder instead of the decoder) and to the wrong parameter.

But this fatal law is not the only one. Petitioner also argues that the [a] limitations (which require an indication of a first image frame in decoding order) is met by the closed_GOP flag that is a syntax element within the Group of Pictures header. This header precedes any image frame in the bitstream. And closed_gop flag does not confirm the presence of any image frame to a decoder. Rather, closed_gop flag provides information only about the group of pictures layer, not any information about any picture. To circumvent this fact, Petitioner and its expert craft a modified MPEG-1 decoder that excludes any possibility that the decoder would ever successfully decode bitstreams other than ones with an I-frame after every GOP header. As a threshold matter, this argument confuses what a POSITA would understand an "indication" to be. In video coding, an indication confirms something to a decoder with certainty. In Petitioner's modified decoder that it posits would encounter only a controlled universe of bitstreams, the fact an I-frame follows a GOP header does not mean the GOP header indicated the I-frame. Rather, even in Petitioner's contrived scenario, the modified decoder would only confirm the presence of an I-frame in the bitstream through picture layer syntax (not the closed_gop flag). Moreover, this convenient modification to an MPEG-1 decoder

(which Petitioner's expert admits was not clearly laid out in his declaration) does not reflect the teachings of MPEG-1, which defines a complete decoding process.

Petitioner's third flawed argument is that MPEG-1 discloses starting the decoding from said first image frame in decoding order (where first said image frame is the one identified by the indication). This is wrong. Instead, MPEG-1 discloses starting the decoding at the first instance it reaches a sequence start code (one layer above group of pictures), the group start code, or the picture start code of the first I-frame (one layer below the group layer, syntactically). If a user randomly seeks to a portion of the bitstream after the group start code (and therefore after the GOP header that includes the closed_GOP flag), it will move to the first I-frame it finds and decode from there, even if an I-frame in the GOP preceded it. Thus, an MPEG-1 decoder will not start decoding from the indicated first image frame in decoding order but merely the first I-frame it finds. The MPEG-1 decoder would not have ever decoded the indication, either.

It is unsurprising that Petitioner's arguments on MPEG-1 are flawed. The '321 Patent sought to improve prior art random access, including the limited random access provided by MPEG-1's Group of Pictures. What the '321 Patent inventors sought to improve was a numbering scheme to facilitate randomly accessing within any independent sequence and not confuse the decoder with numbering schemes that would make it appear as some frames were lost (e.g. starting at frame 15 in display

order or decoding order, where MPEG-1 would attempt to reconstruct frames 1-14). In resetting the identifier value to zero at the decoder, the decoder can then process the independent sequence without error. Put simply, the ’321 Patent improved the very random access techniques used in MPEG-1 that Petitioners incorrectly allege disclose the claims of the ’321 patent.

The Time-Lapse Kim device does not invalidate Claims 8, 10, or 11 of the ’321 Patent. Next, Petitioners attempt to argue that the improved random access inventions of the ’321 Patent for video streaming are disclosed by Kim’s intermittent data recording and reproducing time-lapse device. This argument fails.

Most devastating to Petitioner’s argument is its failure to acknowledge that Kim’s decoder (or reproducer) does not reset any identifier value as required by the [d] limitations. There is no disclosure in Kim that, during decoding, the identifier value that Petitioner points to (an IAD that is set to 0000) is reset by the decoder. Instead, Kim is clear that during the recording process (an encoding process), the encoder sets that value and inserts it into the bitstream for the decoder/reproducer to receive. Nothing more. This point alone dooms this ground.

Next, Kim fails to disclose the [a] limitations (an indication of a first image frame in decoding order) and/or the [b] limitations (starting the decoding from said first image frame in decoding order). Petitioner identifies Figure 4 as the decoding process of Kim and identifies two different scenarios where a user requests a

particular image frame: one path for when a user selects an I-frame and another for when the user selects a P-frame. Petitioner argues both paths support its position. Neither do.

When a user selects a frame, Kim decodes the image data of that frame and then attempts to determine, from that image data, whether it is an I-frame or a P-frame. In doing so, Kim does not disclose looking to either the logical IAD or CAD of that frame. As for the first path, when the frame the user selected is an I-frame, Kim's device is done looking for data and then produces the image back. In this instance, Kim never looks to the IAD or the CAD for any reason. This dooms Petitioner because this means the IAD of 0000 that Petitioner claims is the indication is never decoded or used by Kim. Thus, the I-frame path cannot disclose decoding an indication of a first image frame in decoding order.

Next, turning to the scenario where a user requests a P-frame (the second path), the first image frame that Kim decodes is the user-selected frame. Figure 4, and the accompanying disclosure, make this clear. Even Petitioner's expert admits that decoding begins with the user selected frame. During this second path, there is a portion where Petitioner argues Kim looks to the logical addresses (IAD and CAD) to find the I-frame that holds remaining image data that can be combined with the decoded P-frame . This I-frame is the one that Petitioner claims is indicated as the first image frame in decoding order. But this I-frame is not decoded first. Rather, the

user-selected P-frame is the first frame to have its image data decoded. Only after decoding the data of the user-selected P-frame does Kim retrieve the I-frame image data to decode it. For this reason, Kim does not disclose starting the decoding with said first image frame in decoding order as required by the [b] limitations.

Finally, Petitioners fail to show limitation [c], which is decoding identifier values for image frames according to a numbering scheme. Petitioner points to the logical addresses, IAD and CAD, disclosed in Kim's Figure 5A as the identifier values that have a numbering scheme. But a close reading of Kim indicates that these logical addresses are not decoded by Kim's decoder. Kim's specification, when referring to obtaining the address of the image frame makes clear that the obtained address refers to the physical addresses disclosed in Figure 6 (and specifically area 3). When attempting to reach the user selected frame, both Petitioner's expert and Nokia's expert agree that the physical address would be used. And when attempting to reach the I-frame in the case where the user-selected frame is a P-frame, Kim discloses using the physical address. There is no disclosure in Kim that indicates that the logical address IAD and CAD are used or decoded. Kim's physical addresses cannot be contorted into a numbering scheme. Because the size of the N byte for each frame is variable and depends on the encoded data, there is no predictability as to any numbering (where the numbering is last address + N byte). The variability precludes any ability for the physical address to be a numbering scheme.

II. BACKGROUND

A. Technical Background of the ’321 Patent

The ’321 Patent relates to an improvement in video decoding that enables a decoder to identify where to begin decoding, especially when a user randomly selects a frame to start viewing video when rewinding or fast-forwarding. This is known as random access in video coding, which is what the ’321 Patent is directed to. ’321 Patent, 3:52-56, 3:62-4:4, 4:48-60[2]. EX2004, ¶41

According to the ’321 Patent states, “[a] problem in the numbering of image frames arises when the user wishes to start browsing a video file in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point, or when a video files that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error.” ’321 Patent, 11:11-20. because the issue arises most notably “[w]hen the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering. The decoder

[2] The ’321 Patent is EX1001 but will be referred to throughout as the ’321 Patent.

typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost." '321 Patent, 11:20-25. EX2004, ¶42

"[T]his 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 to zero. . . . Consequently, 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." '321 Patent, 11:26-40; EX2004, ¶43

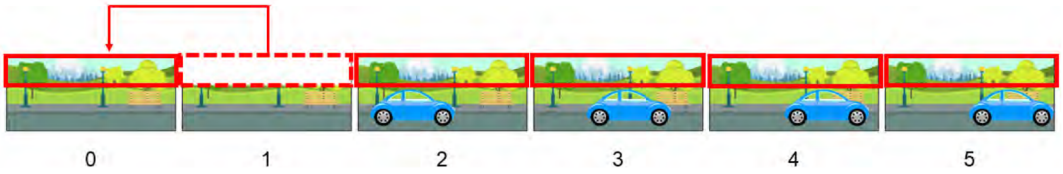
This improved method of the '321 Patent "enable[s] detection of image frames from which a decoder can start the decoding process." '321 Patent, 4:8-10. "The invention is based on the idea of encoding a video sequence comprising an independent sequence of image frames, wherein at least one reference picture is predictable from at least one previous image frame that is earlier than the previous reference image frame in decoding order." '321 Patent, 4:16-20; EX2004, ¶44

In his accompanying Declaration, Dr. Iain Richardson explains the illustrations below to further describe the random-access problems addressed by the '321 Patent. As an introduction, temporal redundancy allows non-changing content in a series of video images (e.g., the red boxes) to be predicted from earlier pictures

that include the redundancy and avoids the need to encode the same content repetitively.

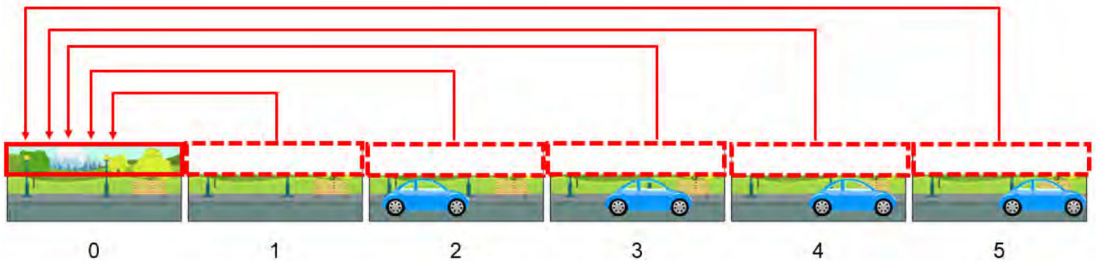
EX2004, ¶45

This is illustrated below, where the top of Picture 1 (dashed rectangle) is predicted from the top of Picture 0 (solid rectangle); Picture 0 contains the video content necessary for Picture 1.



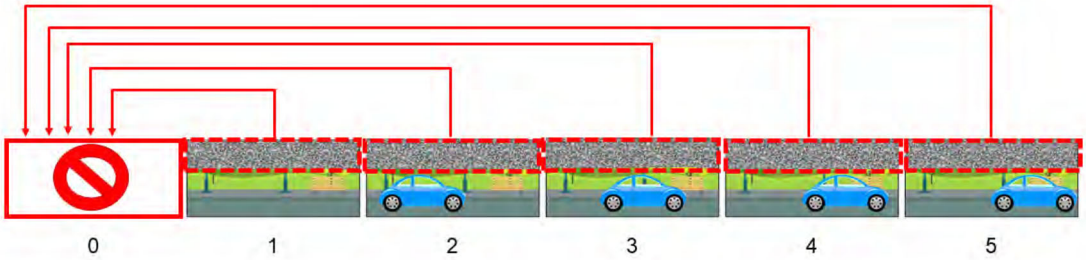
EX2004, ¶46

The more this can be achieved, the more compression is improved. For example, here the top of Picture 0is only transmitted once, and Pictures 1-5 predict that content (arrows) from Picture 0.



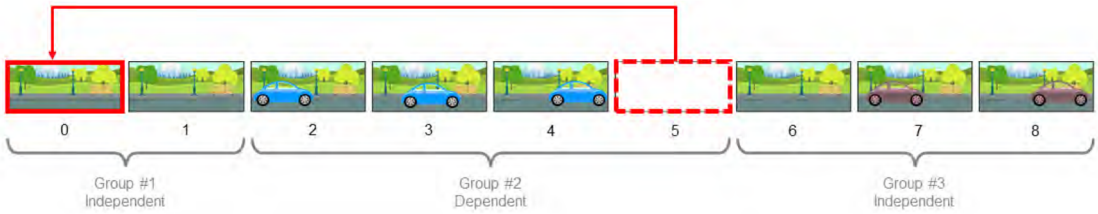
EX2004, ¶47

But rrrors occur when the (Picture 0, which is a reference picture) is missing; the predicted pictures (Pictures 1-5) cannot reference the video content.



EX2004, ¶48

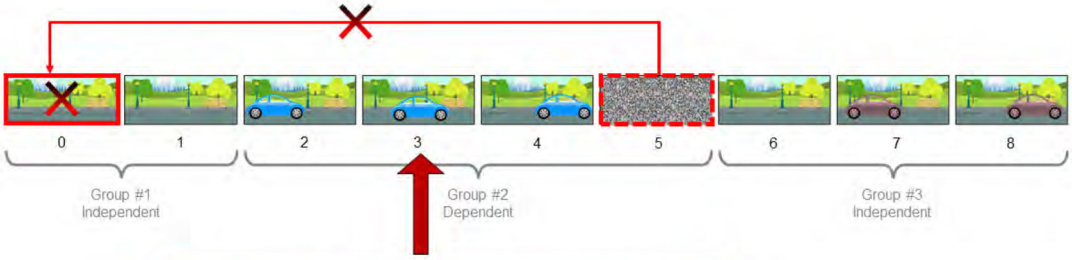
More errors can occur as well, when multiple groups of pictures are present. For example, the video sequence (Pictures 0-8) displays two independent sequences: Group #1 (Pictures 0-1) and Group #3 (Pictures 6-8). But between those, it also includes a dependent sequence: Group #2 (Pictures 2-5). An independent sequence includes pictures referring only to pictures within the same independent sequence, while a dependent sequence includes at least one picture that refers to another picture not within that dependent sequence. For example, Picture 5 in Group #2 predicts from Picture 0 in Group #1, so Group #2 is a dependent sequence. EX2004, ¶49.



EX2004, ¶49.

In these examples, when a user randomly accesses a specific picture within the middle of a video sequence, the decoder does not know where to begin decoding to permit the randomly-accessed picture to be displayed without any prediction errors. '321 Patent, 3:52-56, 3:62-4:4. As explained in the '321 Patent, these issues present challenges with random access because the decoder does not know from which image frames it can begin decoding. *Id.*; EX2004, ¶50.

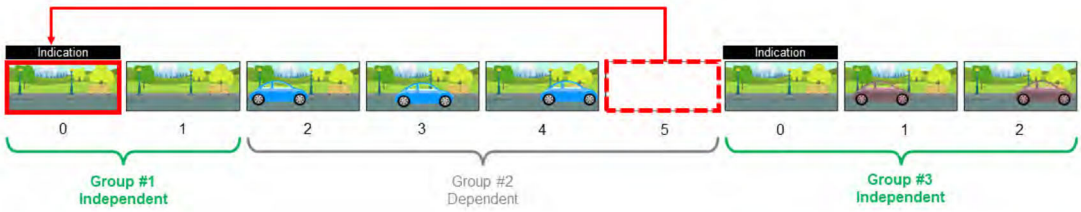
Dr. Richardson's example below illustrates this problem. Let's say the user selects Picture 3 as the random-access point. If the decoder starts decoding at Picture 3, an error occurs. This is because when decoding Picture 5, Picture 0 was never decoded but it is needed to predict Picture 5.



EX2004, ¶51.

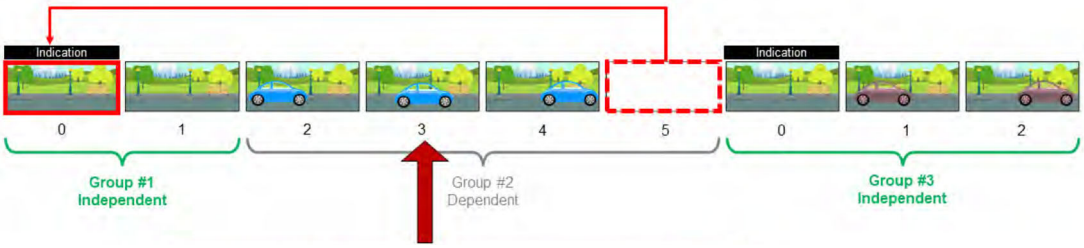
The ’321 Patent fixes these prior art problems,.’321 Patent, 4:48-60, through the identification of an initiation frame and resetting the numbering of that frame to zero solves the issue so that the decoder does not interpret missing frames. ’321 Patent, 11:11-40; EX2004, ¶52.

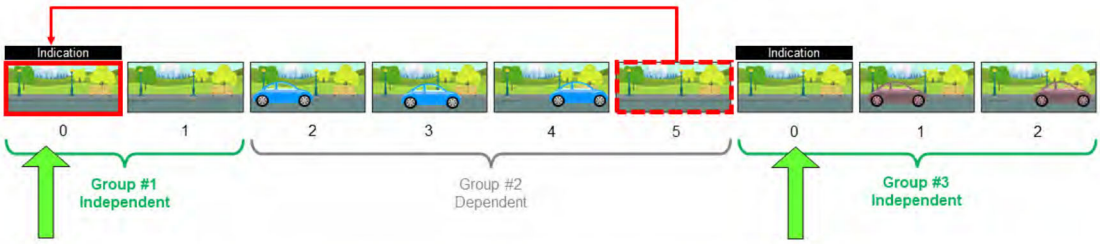
Reading the requirements of the claims, a decoder decodes an indication of the first image frame of an independent sequence. As an example, below are two independent sequences (Groups #1 and #3), each including an indication (black box) for each first image frame that informs the decoder where decoding can begin to avoid random-access problems.



EX2004, ¶53.

Here, as Dr. Richardson explains, if the user selects Picture 3 in Group #2 as the random-access point (red arrow), decoding begins at the first image of one of the independent sequences: either Picture 0 in Group #1 (left green arrow) or Picture 0 in Group #3 (right green arrow).





EX2004, ¶54.

B. The Prior Art

1. MPEG-1

MPEG-1 Video ((ISO/IEC 11172-2, "MPEG-1") is a standard developed by the Moving Pictures Expert Group that was approved in 1993. MPEG-1 does not enable random access at a user-specified playback point and thus did not solve the prior-art problem identified and claimed by the '321 Patent. Instead, MPEG-1 describes that after a user specifies a random access point, the decoder looks for the next sequence start code, group of pictures (GOP) start code, or picture start code that introduces an I-picture (collectively, "start codes") to start decoding. Ex. 1005, 43, 87, 120; EX2004, ¶¶73-78. Thus, in MPEG-1, decoding and playback starts from the next start code rather than from a specific user-identified frame (i.e., the random access point). *Id.*

Several properties of groups of pictures in MPEG-1 are important to keep in mind. MPEG-1 defines a Group of Pictures or GOP below:

D.5.2 Group of pictures

Two distinct picture orderings exist, the display order and the bitstream order (as they appear in the video bitstream). A group of pictures (gop) is a set of pictures which are contiguous in display order. A group of pictures must contain at least one I-picture. This required picture may be followed by any number of I and P-pictures. Any number of B-pictures may be interspersed between each pair of I or P-pictures, and may also precede the first I-picture.

Property 1. A group of pictures, in bitstream order, must start with an I-picture and may be followed by any number of I, P or B-pictures in any order.

Property 2. Another property of a group of pictures is that it must begin, in display order, with an I or a B-picture, and must end with an I or a P-picture. The smallest group of pictures consists of a single I-picture, whereas the largest size is unlimited.

The original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group. In the final version of this part of ISO/IEC 11172 this is not always true, and any B-pictures preceding (in display order) the first I-picture in a group may require the last picture in the previous group in order to be decoded. Nevertheless encoders can still construct groups of pictures which are independent of one another. One way to do this is to omit any B-pictures preceding the first I-picture. Another way is to allow such B-pictures, but to code them using only backward motion compensation.

Property 3. From a coding point of view, a concisely stated property is that a group of pictures begins with a group of pictures header, and either ends at the next group of pictures header or at the next sequence header or at the end of sequence, whichever comes first.

Some examples of groups of pictures are given below:

```
I
I  P  P
I  B  P  B  P
B  B  I  B  P  B  P
B  B  I  B  B  P  B  B  P  B  B  P
B  I  B  B  B  B  P  B  I  B  B  I  I
```

Figure D.17 -- Examples of groups of pictures in display order

These examples illustrate what is possible, and do not constitute a suggestion for structures of groups of pictures.

EX1005, 84-85 (explaining and providing examples of Group of Pictures as defined in MPEG-1); EX2004, ¶¶60-61.

Petitioners' arguments refer to MPEG-1's "closed_gop flag." MPEG-1's closed_gop flag is part of the "Group of pictures layer." That is, closed_gop flag is part of the syntax layer that applies to an entire GOP. EX1005, at 43.

closed_gop -- This is a one-bit flag which may be set to "1" if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures.

EX2004, ¶64.

As specified in MPEG-1, the closed_gop flag is received in the group of pictures header. MPEG-1 defines "Closed GOPs" as follows:

Closed GOP

A one bit flag follows the time code. It denotes whether the group of pictures is open or closed. Closed groups can be decoded without using decoded pictures of the previous group for motion compensation, whereas open groups require such pictures to be available.

A typical example of a closed group is shown in figure D.18a.

I	B	B	P	B	B	P	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12

(a) closed group

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

(b) open or closed group

Figure D.18 -- Example groups of pictures in display order

A less typical example of a closed group is shown in figure D.18b. In this example, the B-pictures which precede the first I-picture must use backward motion compensation only, i.e. any motion compensation must be based only on picture number 2 in the group.

If the closed_gop flag is set to 0 then the group is open. The first B-pictures that precede the first I-picture in the group may have been encoded using the last picture in the previous group for motion compensation.

EX1005, at 86; EX2004, ¶65. The "one bit flag" referred to is the closed_gop flag. EX2004, ¶65. As the closed_gop flag is a flag in the group of pictures header, it is a syntax element that applies to an entire GOP and not any one specific coded frame. As a result, the closed_gop flag is therefore not an indication of any specific image frame. EX2004, ¶66.

For example, in Figure D.18 above, the value of the closed_gop flag (0 or 1)

defines a property of the entire GOP, such as the 13-picture GOP shown in Fig. D.18a or the 15-picture GOP shown in Fig. D.18b. Unlike the closed_GOP flag, there are other syntax elements in an MPEG-1 video bitstream that pertain to a specific coded picture, namely syntax elements found at the Picture Layer and lower syntax layers. EX2004, ¶68. For example,. MPEG-1 states that "The picture layer contains all the coded information for one picture". Ex. 1005 at 87. As a result, the closed_gop flag is not an indication of an image frame in the. As shown in Figure D.18 above, a closed_gop flag does not indicate any particular image frame. EX2004, ¶¶67-68.

As Dr. Richardson explains with an example in his Declaration, the presence of an indication in a bitstream necessarily confirms to a decoder the thing indicated. EX2004, ¶¶102-103 At the time a closed_gop flag is received, a decoder does not have an indication of any image frame. EX2004, ¶104. Any indication occurs later in the bitstream. Indeed, this is the reason for the picture layer in any image frames that follow the closed_gop flag. The picture layer (not the closed_gop flag) includes indication of those image frames.

Consider "Temporal Reference" in MPEG-1. Unlike closed_gop flag, Temporal Reference is encoded as a syntax element temporal_reference that is part of the Picture Layer syntax. That is, Temporal Reference is a syntax element that is associated with a specific coded picture. EX1005, at 38. MPEG-1 discloses that the

"Temporal Reference" can be used to define "the order in which the pictures must be displayed." For a given group of pictures (whether open or closed), the first picture in display order has a Temporal Reference of zero:[3]

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in display order	I	B	P	B	P							
	0	1	2	3	4							
Example (a) in decoding order	I	P	B	P	B							
	0	2	1	4	3							
Example (b) in display order	B	B	I	B	B	P	B	B	P	B	B	P
	0	1	2	3	4	5	6	7	8	9	10	11
Example (b) in coded order	I	B	B	P	B	B	P	B	B	P	B	B
	2	0	1	5	3	4	8	6	7	11	9	10
Example (c) in display order	B	I	B	B	B	B	P	B	I	B	B	I	I
	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in coded order	I	B	P	B	B	B	B	I	B	I	B	B	I
	1	0	6	2	3	4	5	8	7	11	9	10	12

Figure D.19 -- Examples of groups of pictures and temporal references

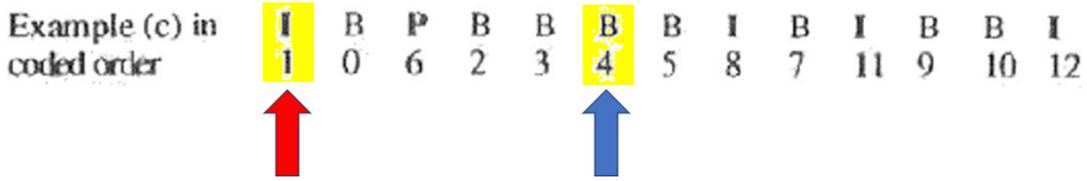
EX1005, 87-88; EX2004, ¶¶70-71. Display order is different from the "coded order" of the pictures, which defines the order that the pictures are to be decoded. As

[3] Temporal reference is also set to zero by the encoder when there are more than 1024 pictures in a GOP. EX1005, 87. This is a limitation of the number of bits used to express numbers.

illustrated in Figure D.19, the coded order for a picture can be different from the display order.

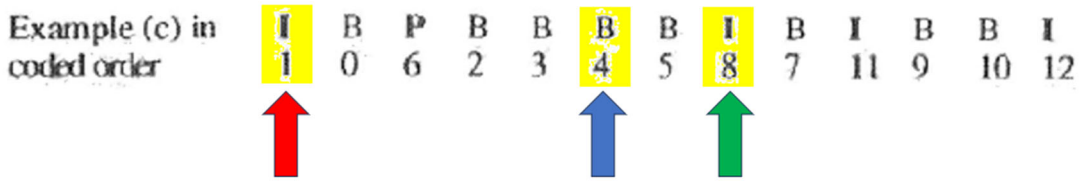
MPEG-1 suffers from the prior art problems discussed in the ’321 Patent because it does not enable random access at a user-specified playback point. MPEG-1 makes clear that after a user specifies a random-access point, the decoder looks for the next sequence start code, group of pictures (GOP) start code, or a picture start code that introduces an I-picture (collectively, “start codes”) to start decoding. EX1005, 43, 87, 120. Then, all prior frames are discarded, and decoding and playback starts from that next start code (rather than from a specific identified frame), which is not the user-specified random-access point. EX2004, ¶73.

Applying the teachings of MPEG-1 to Example (c) of Figure D.19 highlights the issues with MPEG-1 compared to the claimed inventions of the ’321 Patent. As Dr. Richardson explains and illustrates below, the first frame of the GOP in decoding order is I1 (red arrow) and the user selects, for example, frame B4 (blue arrow) as the playback point. EX2004, ¶74



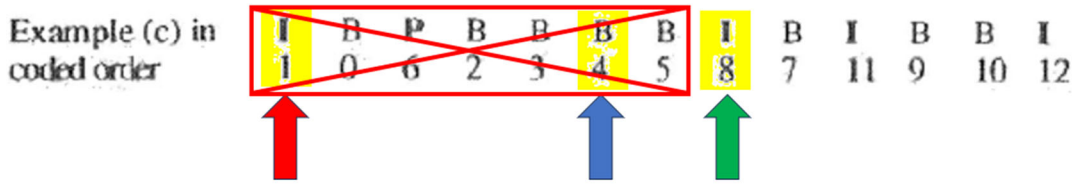
Going on, Dr. Richardson shows that MPEG-1 starts decoding from frame I8 (green arrow), which is not the first frame I1 (red arrow) of the independent

sequence. MPEG-1 starts decoding from I8 (green arrow) because MPEG-1 teaches that after a user specifies a random-access point, the decoder looks for the next start code to start decoding. EX1005, 43, 87, 120. In Example (c), the next start code is a picture start code which introduces an I-picture I8. *See id.*



EX2004, ¶75.

Next, Dr. Richardson confirms that the decoder discards all frames until frame I8 and starts decoding and playback from I8 because MPEG-1 teaches that all prior frames are discarded and decoding and playback starts from that next start code (rather than from a specific identified frame), which is not the user-specified random-access point. EX1005, 43, 87, 120.



EX2004, ¶76.

As another issue, MPEG-1 does not disclose the decoder resetting any identifier value of any indicated first image frame. In the above example, MPEG-1 never teaches resetting the any identifier value of those frames. EX2004, ¶77.

MPEG-1 does not disclose decoding any indication of a first image frame of an independent sequence, starting the decoding from the indicated first image frame, and resetting an identifier value of the indicated first image frame. EX2004, ¶78. That is, when the user selects a playback point, MPEG-1 discloses that it discards all frames and start decoding at a start code (not an indicated first image frame). So, MPEG-1 suffers from the same random-access problems discussed above, and the '321 Patent acknowledged MPEG-1's GOP concept as a problem in random access and sought to address it. '321 Patent, 2:34-55, 3:52-56; EX2004, ¶78. For these reasons, MPEG-1 fails to disclose several limitations of the asserted claims that enable random access from the user-specified point. EX2004, ¶78.

2. Kim

Kim discloses a Time Lapse recording/reproduction device and method. Specifically, Kim describes recording, compressing and reproducing intermittently recorded image data. Kim is not related to video streaming. EX2004, ¶79. Dr. Havlicek acknowledged that the device of Kim is "not your average consumer video," "not 30-frames-a-second consumer video," and the input video (camera video) "might run at 30fps, frames per second, but Kim's device isn't going to record that." Ex. 2005, 39:14-40:22. Examples of where this type of technology may be applied is in banks or other security-sensitive areas. In such implementations,

pictures may be recorded intermittently and then stored, for example on magnetic tapes. EX2004, ¶79.

As shown by Kim's Figure 4 these individual images can be individually and incrementally reproduced when needed. However, Kim does not enable random access of streaming video and does not address the problems solved by the '321 patent. (Ex 1006 at 1:8-56). EX2004, ¶¶79-80.

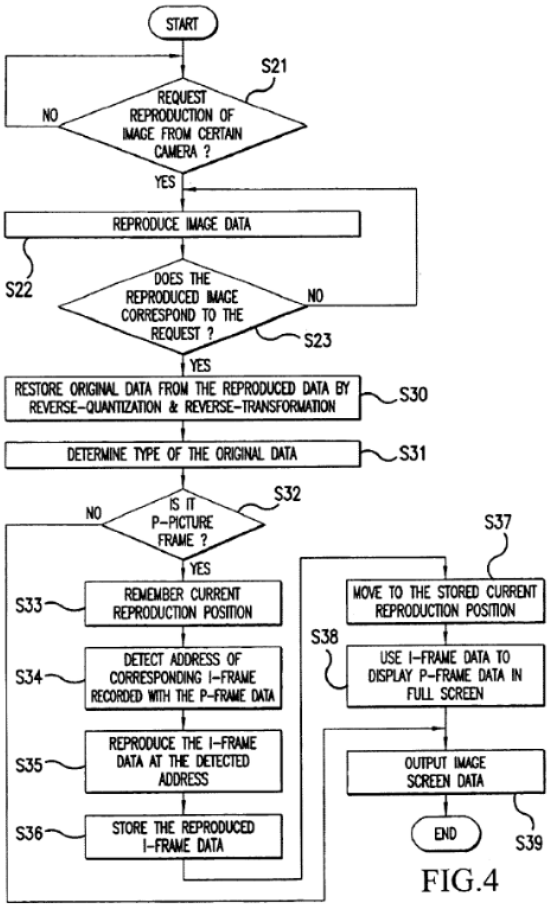


FIG.4

Petitioner relies on Figure 5A of Kim. Kim describes Figures 5A and 5B as illustrating "examples of a logical data structure of a digital data stream compressed and recorded on the optical disk according to [Kim's] present invention." EX2004, ¶81. In FIG. 5A, each "P-picture frame P1, P2, P3 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 (shown below).

> The bit stream generated and transmitted to the recording/reproducing unit **140** as described above has a logical structure as shown in, e.g., FIG. **5**A. As shown therein, each P-picture frame **P1**, **P2**, **P3** 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. If image signals are

EX2004, ¶¶80-81.

Petitioner relies on Figure 5A and Kim at 4:46-52 to show "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." Petition at 46. However, Petitioner is incorrect that Figure 5A and 4:46-52 of Kim show the claimed "indication".

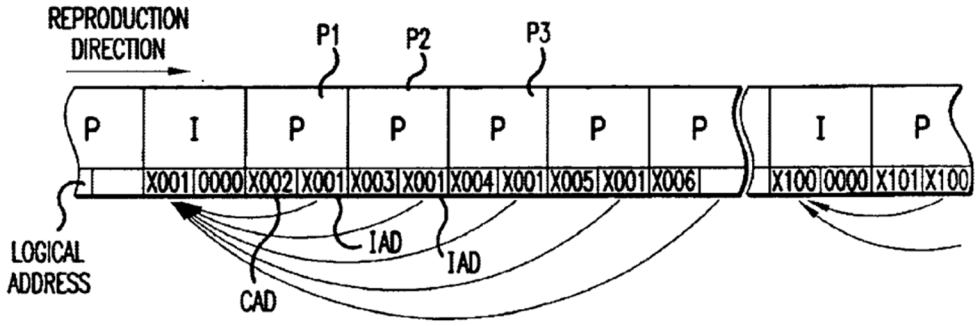


FIG.5A

EX2004, ¶¶81-90.

Specifically, Petitioners points to an IAD of 0000 as the claimed "indication" and argues Figure 5A discloses an independent sequence because the P pictures in

the figure refer to the first I frame. Pet., 46. There are several problems with this argument. First, the 0000 of the I frame is not the indication of the first image frame in decoding order because—in the scenario Petitioners identify for their Kim Ground—the user-selected P frame (not the I frame) is decoded first. EX2004, ¶90. Second, in a scenario in which a user selected an I frame, the 0000 of the I frame would not serve as an indication of any first image frame from the video sequence because the user would have already selected the I frame (and Kim's system would as a result go to location of the corresponding data in memory). EX1006, Figure 4, 4:59-65, 5:41-44; EX2004, ¶¶83-85. Third, Petitioners also identify Kim's IAD address as an identifier value, but the IAD address is not reset by the decoder, as required by the claims. EX2004, ¶90. In Kim's method, the IAD address for an I-picture is "zero or absent" and Kim discloses it as such referring to Figure 6. Kim never teaches that the decoder resets an IAD address. EX2004, ¶90.

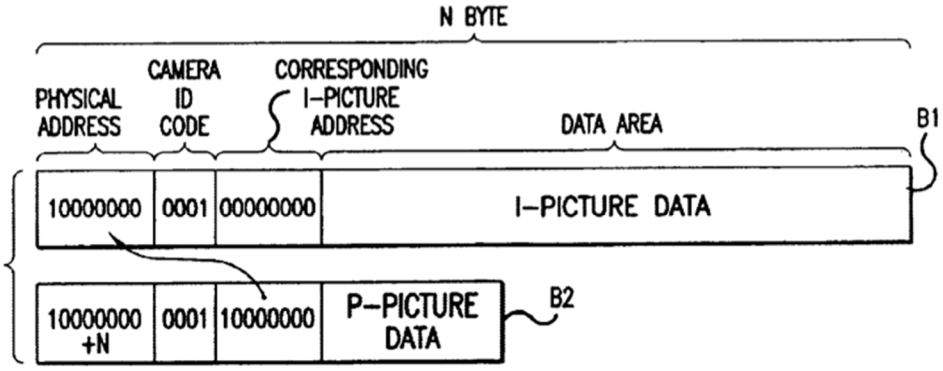


FIG.6

As Dr. Richardson explains, since an I-picture does not require a reference to another I-picture, the IAD field can be "zero or absent" and a POSITA reading the disclosure would understand that the field could take any value. Kim, 4:66-5:1; EX2004, ¶91.

Because the IAD field for an I-picture can be "zero or absent" in Kim, it there is no functional use and therefore a POSITA would recognize that it is not disclosed as an indication that is "decoded from [a] video sequence." The sum of support for Petitioner's statement that, "A POSITA would have understood that decoding the bitstream in Figure 5A would include decoding… the IAD values" (Petition at 47) is Dr Havlicek's statement that "Kim … explains the process for decoding this bitstream," referring to Kim at 5:8-6:8 and Figure 4 (Ex. 1003, at 152-154). However, Kim Figure 4 and 5:8-6:8 say nothing about decoding or even detecting the IAD field for an I-picture. Thus, there is no independent support for Petitioner's statement (Pet. at 47). EX2004, ¶92.

Kim's decoder, shown in Figure 4, decodes and displays a single picture, based on a user request for one image. If the requested image is an I-picture, the corresponding coded image data is simply read from the optical disk at the relevant location, decoded, and displayed. As shown in the annotated figure below (Figure 4 alongside column 5 of Kim), the first image data decoded is the image data for the frame the user selected. As shown by the presence of steps S21 and S22 before steps

S31 and S32, Kim is unaware at that point whether the user-selected frame is an I-frame or a P-frame.

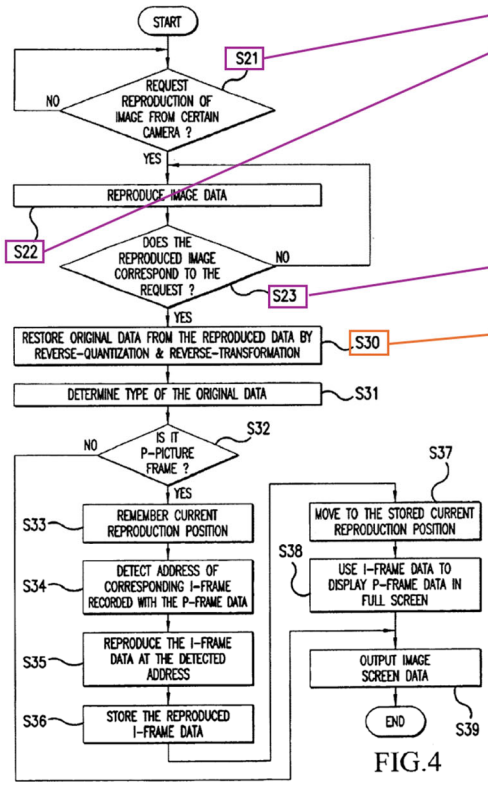


FIG.4

As shown in Steps S21 and S22 of FIG. 4, if the user requests reproduction of an image data from a particular camera or channel among the intermittent image data recorded on the optical disc **1**, the controller **150** controls the recording/reproducing unit **140** to reproduce the corresponding image data from the optical disc **1**. The image data reproduced from the optical disc **1** is multiplexed data, which is demultiplexed by the de-MUX **131**. By demultiplexing, the camera (channel) ID code CHID is separated from the reproduced image data and output to the controller **150**. Based on the camera ID code CHID, the controller **150** determines whether or not the reproduced image data corresponds to the selected camera or channel, in Step S23. This ensures that the correct image data is processed to comply with the user's request. If incorrect image data has been reproduced, the controller **150** controls the recording/reproduction unit **140** to reproduce the correct image data from the optical disc **1**. In Step S30, the image data corresponding to the user's request is transmitted and decoded in the decoder **111**. The decoding process of the decoder **111** involves reverse-quantization and reverse-transformation to restore the original image data prior to data compression carried out during recording. Then the property (type) of the currently restored image data is determined in Step S31 by the reproducing DSP **121**, and the determination result is transmitted to the controller **150**. More specifically, 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. Then in Step S34, the controller **150** 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**) from the reproduced image data output from the de-MUX **131**, and reproduces based on the detected address the corresponding (or previous) I-picture frame data recorded on the optical disc **1** by controlling the recording/reproduction unit **140**.

In Step S36, the reproduced corresponding I-picture frame data is processed by the reproducing DSP **121**, and decoded by the decoder **111** to be stored in the frame memory **101**. The controller **150**, after Step S36, controls the recording/reproducing unit **140** to move the video heads to the original recording position stored in Step S33.

Annotated Kim, Figure 4 and 5:1-41; EX2004, ¶¶93-94.

As Dr. Richardson explains, if the requested image is a P-picture, the P-picture data is first read and decoded, then the corresponding I-picture is located, read, and decoded; then the P-picture is reconstructed (by combining the decoded P-picture and necessary I-picture data), then displayed. EX1006, Fig 4, 5:8-6:8. Thus, the 0000 IAD value cannot be the indication of the first image frame in decoding order; it is never decoded. EX2004, ¶95.

Kim fails to disclose decoding any indication of a first image frame of an independent sequence, starting the decoding from the indicated first image frame, and resetting an identifier value of the indicated first image frame. EX2004, ¶96. When the user selects an image for output, Kim teaches to start decoding at the user-selected image (not an indicated first image frame of an independent sequence). EX2004, ¶96. Kim does not teach that the decoder resets the IAD value of the user-selected image or any other image. EX2004, ¶96. Kim's output of a single image frame does not solve the random-access problems discussed above. EX2004, ¶96. Kim fails to disclose several limitations of the asserted claims that avoid errors and enable random access from a user-specified image in a video sequence.

C. Asserted Grounds

Petitioners assert the following three grounds. Petitioners have not proven any ground by a preponderance of the evidence.

Ground		Claims	Prior Art
1	103	8, 10, 11	MPEG-1
2	103	8, 10, 11	Kim
3	103	9	MPEG-1, Yagasaki

Pet. 2.

D. POSITA

Petitioner identified that the POSITA "would have had a (1) Bachelor's degree in electrical engineering, computer engineer, 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 . . . [and] [a]dditional experience can substitute for the level of education, and vice-versa." Pet. 9. Because this Response does not depend on the level of ordinary skill in the art, Patent Owner has applied Petitioner's definition of a POSITA.

E. Claim Construction

Petitioner identifies no claims for construction. Pet. 9. Likewise, Patent Owner identifies no claims for construction at this juncture based on Petitioner's representation that it has not sought to construe any claim terms.

F. Claim Table

For the sake of this analysis, Patent Owner will adopt Petitioner's limitation labels reproduced below:

Reference	Claim Limitation
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
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:
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.

III. ARGUMENT

A. Petitioner Has Not Shown That The Challenged Claims Are Unpatentable In Light Of Ground 1 (MPEG-1).

MPEG-1 does not anticipate or render obvious the asserted claims. First, MPEG-1 does not disclose an indication of the first image frame in decoding order. Instead, MPEG-1 describes a closed_gop flag that indicates whether a group of pictures (GOP) is closed or open. EX2004, ¶¶97-122. Further, MPEG-1 does not disclose the *decoder* resetting the identifier value for the indicated first image frame ***in decoding order***. Rather, for MPEG-1-compliant bitstreams, the *encoder* sets the temporal reference of the first frame ***in display order*** at the beginning of each new independent sequence (or if there are more than 1024 pictures in a group of pictures). EX2004, ¶¶138-148.

1. Petitioners fail to demonstrate that limitations 8[a], 10[a], and 11[a] are obvious over MPEG-1

Each of limitations 8[a], 10[a], and 11[a] are written with similar language and require "[decoding/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." Petitioner[4] has failed to show that MPEG-1 teaches or suggests these limitations. EX2004, ¶97.

Petitioners asserts two distinct arguments for how the "[decoding/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 limitations" are taught or suggested—pointing to MPEG-1's closed GOP or, alternatively, to MPEG-1's Temporal Reference.

a) MPEG-1's Closed_GOP Flag is Not the Claimed "Indication"

Petitioner asserts that MPEG-1's closed_gop teaches or suggests "[decoding/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

---

[4] Because Petitioners put forth an expert declaration that is substantively identical to the limitation-by-limitation analysis, references to "Petitioner" as used in these claim sections refers to both Petitioner's argument and the substantively identical expert declaration substance.

sequence." This is wrong for at least two reasons: (1) the closed_gop flag is a single-bit flag that only indicates whether the *entire* GOP is closed or open, it does notconfirm the presence of any image frame to a decoder. So, closed_gop flag is not an indication according to the claims; (2) not only does closed_gop flag fail to indicate any frame to a decoder, it does not indicate ***which frame is the first***. EX2004, ¶99. As Dr. Richardson explains in his declaration, the mere fact that closed GOPs happen to have first image frames does not mean that closed_gop flag indicates a first image frame in decoding order to a decoder. EX2004, ¶100.

MPEG-1's closed GOP flag is only an indication of whether or not the group of pictures is open or closed, it flag does not indicate ***any*** image frame (whether or not the image frame is first in decoding order). EX1005, 37; EX2004, ¶100. The closed GOP information at its own structural level of the coded video bitstream data and is not related to any particular frame:

0.5 Structure of the coded video bitstream

This part of ISO/IEC 11172 specifies a syntax for a coded video bitstream. This syntax contains six layers, each of which either supports a signal processing or a system function:

Layers of the syntax	Function
Sequence layer	Random access unit: context
Group of pictures layer	Random access unit: video
Picture layer	Primary coding unit
Slice layer	Resynchronization unit
Macroblock layer	Motion compensation unit
Block layer	DCT unit

EX1005, 16; EX2004, ¶100. As Dr. Richardson explain, Syntax elements in the GOP layer apply to an **entire GOP**, whereas syntax elements in the Picture layer apply to

a specific coded picture. This means that when a decoder processes the bitstream and decodes the closed_gop flag, the decoder does not have confirmation about the existence of any particular coded frame in the bitstream. Rather, by decoding closed_gop flag, the decoder instead receives an indication whether a GOP is open or closed based on the value of the flag. EX2004, ¶101.

There is no confirmation of the existence of any frame when the decoder processes the closed_gop flag. EX2004, ¶101. The decoder remains unaware of any particular frame. For this reason, the closed_gop flag cannot indicate any specific coded frame in an independent sequence, let alone the first one. EX2004, ¶101. The fact that image frames may follow a closed_gop flag does not mean that the closed_gop flag has served as an indication of whatever frames happen to follow the closed_gop flag. Indeed, other syntax (in the picture layer) serves to indicate any frames to the decoder. Under Petitioners' interpretation of closed_gop flag, a decoder should treat that additional syntax as superfluous, which a POSITA knows is not the case. EX2004, ¶102

Put another way, just because a driver gets into their car to go to work does not mean that every time a driver gets into their car, they are going to work. The "getting into the car" cannot be an indication of going to work just because in some instances that occurs.

Further, as Dr. Richardson explains, a POSITA would understand "indication" in the claims refers to information in an encoded bitstream that necessarily confirms something to a decoder. EX2004, ¶102. Here, the indication is "of at least one image frame, which is the first image frame, in decoding order, of an independent sequence." Thus, as Dr. Richardson confirms, a POSITA would understand that this "indication" cannot merely be information that coincidentally appears alongside other information in the bitstream. EX2004, ¶102. That is, in video coding, an indication in a bitstream must inform a decoder of something with certainty. EX2004, ¶102.

As Dr. Richardson explains, an indication takes the following form in a coded bitstream:

> If indication is present, then thing indicated is confirmed necessarily to the decoder.

EX2004, ¶103. Nonetheless, Petitioners' MPEG-1 Ground relies on a closed_gop flag in MPEG-1 that does not indicate what Petitioners allege to be the first image frame. EX2004, ¶103. In MPEG-1 the fact that a closed_gop flag has a value of "0" or "1" does not confirm to a decoder any type of image as a first image frame. Rather, an MPEG-1 decoder that receives a closed_gop flag with a value of "0" or "1" has no certainty about what frames follow in the bitstream. Thus, a POSITA would

understand that closed_gop flag is not "an indication of at least one image frame, which is the first image frame." EX2004, ¶104.

Dr. Richardson provides an example of an indication in video coding that a POSITA would understand. The H.264 Standard defines a "NAL unit" as "A syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes." EX2006, ITU-T Rec. H.264 (03/2005) at 8 (as paginated in the document). In H.264, the NAL unit syntax includes a parameter called nal_unit_type that serves as an indication of the type of data contained in a NAL unit. Table 7-1 (excerpted below) provides possible values of nal_unit_type. EX2006, TU-T Rec. H.264 (03/2005) at 57 (as paginated in the document):

Table 7-1 – NAL unit type codes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C
0	Unspecified	
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3
6	Supplemental enhancement information (SEI) sei_rbsp( )	5
7	Sequence parameter set seq_parameter_set_rbsp( )	0
8	Picture parameter set pic_parameter_set_rbsp( )	1

EX2004, ¶¶105-107

As Dr. Richardson explain, A POSITA would understand that, for example, nal_unit_type = 7 serves as an indication of a sequence parameter set. In other words, if nal_unit_type = 7, then the NAL unit is necessarily a sequence parameter set. Consider the opposite: if this were not so, then a decoder would not know with certainty based on nal_unit_type what type of data was contained in the NAL unit. And in that scenario, nal_unit_type would not serve as an indication of the type of data contained in a NAL unit. The decoder would need other information from the

bitstream to resolve what type of data was contained in the NAL unit (if the decoder were ever to resolve the uncertainty). EX2004, ¶107

So, while the closed_gop flag can indicate whether the GOP is closed, it cannot and does not indicate a first image, in decoding order, of that group of pictures. EX2004, ¶110.

This distinction is not mere semantics. The "idea of the ['321 Patent's] invention" "is to determine an initiation picture in an independently decodable group of pictures, whereby in the decoding phase any picture prior to said initiation picture is defined as unusable for reference." '321 Patent, 4:31-38. And in failing to do so, the random access benefits described by the '321 Patent are not be achieved. EX2004, ¶111.

In granting institution, the Board relied on Petitioner's assertion that five factors were all true, and therefore the "indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence" is met. Paper No. 10, at 26-27. But at least three of these factors are premised on incorrect interpretations of MPEG-1.

First, in item (1), the Board is incorrect that it is always the case that "a GOP header precedes the first image frame in the GOP in decoding order." As Dr. Havlicek agreed, if playback begins from a random point in the bitstream, an MPEG-1 decoder will "discard all bits until it finds a sequence start code, a group of pictures

start code, or a picture start code which introduces an I-picture." EX2005, 101:17-22; EX1003, ¶100. In deposition, Dr. Havlicek admitted that if that random playback begins after the first I-frame in a closed GOP, the decoder would not read a GOP header and therefore would not decode the closed_GOP flag (which is what Petitioner alleges to be the claimed indication). EX2005, 101:17-103:8.

Thus, the alleged closed_gop flag of MPEG-1 does not disclose the claimed indication of the first image frame in decoding order because the flag only signals to the decoder whether the *entire* GOP is open or closed and does not indicate afirst image frame in decoding order, particularly when a user selects for playback a frame in the middle of a video sequence and the closed_gop flag is not decoded. Moreover, as discussed previously, the fact that a first image frame might follow a closed_gop flag does not mean that the closed_gop flag serves to indicate that first image frame to a decoder. In MPEG-1, such an indication is accomplished through picture layer syntax. By this logic, any syntax that exists prior to syntax for a first image frame (e.g., sequence layer and group of pictures layer syntax exists before picture layer syntax in an MPEG-1-compliant bitstream) would serve as an indication of that first image frame. This is inconsistent with how a POSITA would understand an "indication" in a video sequence. EX2004, ¶¶102-108, 116-120102-108.]

For <u>item (3)</u>, "the closed_gop flag in the GOP header is decoded before the first image frame in the closed GOP is decoded," the Board and Petitioner assume

that the closed_gop flag is always decoded before the first image frame in a closed GOP is decoded. Paper No. 10, at 27. That is incorrect. The closed_gop flag in the GOP header is not always decoded before a first image frame is decoded. For example, when a user selects a frame in the middle of a video sequence, MPEG-1 teaches that all bits are discarded until a start code is reached, as discussed above. In Example (c) of Figure D.19 of MPEG-1 (EX1005, 87), when the user selects, for example, frame B4, the first frame that is decoded is frame I8. Neither the closed_gop flag nor any of the GOP header were decoded or read in the bitstream. Rather, frame I8 was decoded first. Thus, the Board and Petitioner are incorrect that item (3) is true. Although item (3) may be true in some circumstances, that is not the teaching of MPEG-1. Thus, MPEG-1 does not teach a decoder that decodes an indication of an image frame, which is the first image frame, in decoding order, of an independent sequence. Moreover, as explained, the fact that the closed_gop flag in the GOP header is sometimes decoded before a first image frame is decoded does not mean that the closed_gop flag serves as an indication of that first image frame. By this logic, any syntax decoded before a first image frame would serve as an indication of that first image frame. This is inconsistent with how a POSITA would understand an "indication" in a video sequence. EX2004, ¶¶102-108, 121

For item (5), "the closed_gop flag also indicates that 'the next image frame decoded is the first image frame in decoding order' of the 'independent sequence,' i.e., of the closed GOP," the Board and Petitioner make two mistakes. First, the Board and Petitioner conflate correlation with the confirmation required by a POSITA's understanding of "indication." That is, the Board and Petitioner treat the fact that any frame—which has yet to be received or confirmed to a decoder—following a closed_gop flag has been indicated by that closed_gop flag. But this is incorrect. As Dr. Richardson explains, a decoder that receives a closed_gop flag has no confirmation or indication of any possible subsequent image frame. Indeed, the lack of any confirmation on the part of the decoder is the reason for MPEG-1's picture layer syntax that indicates image frames. MPEG-1's picture layer syntax is not superfluous, as a POSITA would know. Second, the Board and Petitioner again assume that the closed_gop flag is always decoded before the first image frame when a closed GOP is decoded. Paper No. 10, at 27. This is incorrect. As discussed above, the first image frame decoded is not necessarily the image frame following the closed_gop flag. Instead, MPEG-1 teaches that the first decoded image frame is the frame following one of the indicated start codes and therefore the closed_gop flag is not always decoded and does not indicate the first image frame in decoding order, as discussed above. EX2004, ¶122. Thus, for similar reasons discussed for items (1) and (3), item (5) is not taught by MPEG-1

b) MPEG-1's Temporal Reference Is Not the Claimed "Indication"

Alternatively, Petitioner points to MPEG-1's Temporal Reference of zero as teaching "decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence." Pet. 23-24. The Board and Petitioner are incorrect that the Temporal References of zero teaches this claim element. MPEG-1 does not teach that a Temporal Reference of zero is an indication of the first image frame in ***decoding order*** of ***an independent sequence***. EX2004, ¶123. Instead, MPEG-1 teaches that "[t]he first picture, in ***display order***, in each group must have Temporal Reference equal to zero." EX1005, 87; EX2004, ¶123. MPEG-1 goes on, saying "[t]he Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed." *Id.* EX2004, ¶124. As Dr. Richardson explains, the Temporal Reference is not used only for *closed GOPs* (*independent sequences*) but also *open GOPs* (*dependent sequences*). Thus, the Temporal Reference cannot indicate the first image frame in *decoding* order of an *independent* sequence; it is merely the first picture in display order as taught by MPEG-1. EX2004, ¶124.

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in display order	I	B	P	B	P							
	0	1	2	3	4							
Example (a) in decoding order	I	P	B	P	B							
	0	2	1	4	3							
Example (b) in display order	B	B	I	B	B	P	B	B	P	B	B	P
	0	1	2	3	4	5	6	7	8	9	10	11
Example (b) in coded order	I	B	B	P	B	B	P	B	B	P	B	B
	2	0	1	5	3	4	8	6	7	11	9	10
Example (c) in display order	B	I	B	B	B	B	P	B	I	B	B	I	I
	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in coded order	I	B	P	B	B	B	B	I	B	I	B	B	I
	1	0	6	2	3	4	5	8	7	11	9	10	12

Figure D.19 -- Examples of groups of pictures and temporal references

EX1005, 87. EX2004, ¶124.

During deposition, Dr. Havlicek confirmed that he did not consider a decoder for MPEG-1 for everything it teaches, but instead Dr. Havlicek considered a MPEG-1 decoder that was designed only for closed GOPs, with no B-frames in the beginning, and no more than two to the tenth (or 1024) frames in a GOP:

> Q. And there – and that also requires – your opinion also requires the encoder and decoder designed in the way you previously mentioned, only closed GOPs and no B-frames in the beginning; is that correct?
>
> A. And one more.
>
> Q. Right.

> A. You can't have more than two to the tenth frames in GOP because then the number would roll over to a zero again.
>
> Q. Right. Because you can only index to 1024 under MPEG-1; is that correct?
>
> A. That's exactly it.

EX2005, 97:21-98:11; EX2004, ¶125.

MPEG-1 specifies a decoding process. MPEG-1 decoders cannot pick and choose parts of MPEG-1 to follow, particularly because MPEG-1 decoders do not have foreknowledge of the bitstreams they will decode. Under MPEG-1's full teaching, it is not true that any GOP (or even the closed GOP that Dr. Havlicek limits himself to) will always have a first picture in decoding order that is an "I-frame with a Temporal Reference of zero" as the Petition suggests. EX2004, ¶126. Neither closed_GOP flag nor Temporal Reference indicate any such arrangement. EX2004, ¶126. For example, Figure D.19 shows many instances where an I-picture is not the first image in display order and an I-picture does not have a Temporal Reference of zero:

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in	I	B	P	B	P							
display order	0	1	2	3	4							
Example (a) in	I	P	B	P	B							
decoding order	0	2	1	4	3							
Example (b) in	B	B	I	B	B	P	B	B	P	B	B	P
display order	0	1	2	3	4	5	6	7	8	9	10	11
Example (b) in	I	B	B	P	B	B	P	B	B	P	B	B
coded order	2	0	1	5	3	4	8	6	7	11	9	10
Example (c) in	B	I	B	B	B	B	P	B	I	B	B	I	I
display order	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in	I	B	P	B	B	B	B	I	B	I	B	B	I
coded order	1	0	6	2	3	4	5	8	7	11	9	10	12

Figure D.19 -- Examples of groups of pictures and temporal references

EX1005, 87; EX2004, ¶126.

Dr. Havlicek acknowledges that under Patent Owner's understanding of a fully complient MPEG-1 decoder—which must be designed to handle all possible bitstreams, including the ones Dr. Havlicek has excluded (such as those with open GOPs or B-frames preceding I-frames in closed GOPs)—the temporal_reference syntax element would not be an indication of a first image frame:

Q. You understand that patent's owner's position is that would not be an indication of the first image frame because the temporal reference of a B-frame in a closed GOP could take the zero and, therefore, it would not indicate a first image frame in display order, is that right?

A. Well, it would – it would indicate the first frame in display order, but it wouldn't be the I-frame, that's the point. And yes, I agree that it's allowable to operate that way within the standard. It is. Absolutely.

EX2005. 98:20-99:22; EX2004, ¶127.

Thus, Petitioner has failed to show that its partially constructed MPEG-1 decoder discloses an indication of an image frame, which is the first image frame, in decoding order, of an independent sequence as required by the [a] limitations. EX2004, ¶128.

2. Petitioner fails to demonstrate that limitations 8[b], 10[b], and 11[b] are obvious over MPEG-1

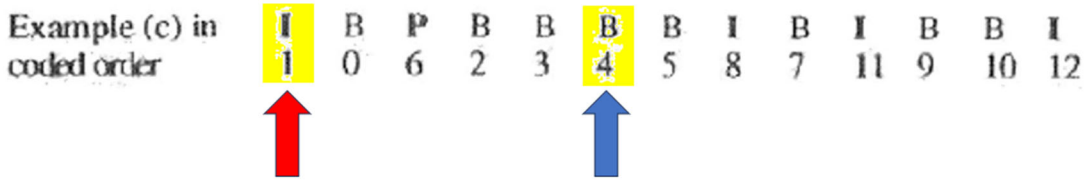
Each of limitations 8[b], 10[b], and 11[b] are written with similar language and require an "starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame" where "from said first image frame of the independent sequence" is the first image frame identified from the "indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence" from 8[a], 10[a], and 11[a].

Dr. Havlicek acknowledged that MPEG-1, when considered in its full teachings, starts decoding during random access at either a sequence start code,

group of pictures start code, or picture start code of an I-frame, and discards all preceding bits. EX1003, ¶100; EX2005, 101:17-104:8; EX2004, ¶131.

MPEG-1 discloses that after a user specifies a random-access point, the decoder looks for the next sequence start code, group of pictures (GOP) start code, or picture start code which introduces an I-picture (collectively, "start codes") to start decoding. EX1005, 43, 87, 120. As part of that, all prior frames are discarded, and decoding and playback starts from that next start code (not the first image frame of an independent sequence). EX2004, ¶131.

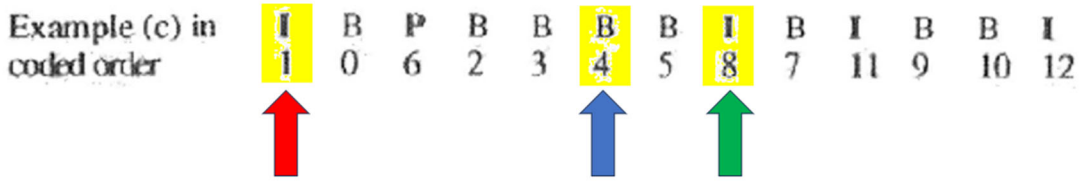
Dr. Richardson identifies an illustration below using Example (c) of Figure D.19 below (which applies equally to all of MPEG-1), the first frame of the GOP in decoding order is I1 (red arrow) and the user selects, for example, frame B4 (blue arrow) as the playback point.



EX2004, ¶132.

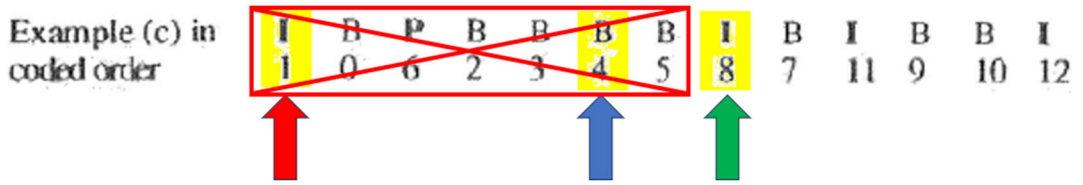
Going on, MPEG-1 starts decoding from frame I8 (green arrow), which is not the first frame I1 (red arrow) of the independent sequence. MPEG-1 starts decoding from I8 (green arrow) because MPEG-1 teaches that after a user specifies a random-access point, the decoder looks for the next sequence start code, group of pictures

(GOP) start code, or picture start code which introduces an I-picture (collectively, "start codes") to start decoding. EX1005, 43, 87, 120. In Example (c), the next start code is the picture start code which introduces an I-picture, I8. *See id.*



EX2004, ¶133.

As Dr. Richardson further illustrates, the decoder discards all frames until frame I8 and starts decoding and playback from I8 because MPEG-1 teaches that all prior frames are discarded and decoding and playback starts from that next start code (rather than from the first image frame of an independent sequence, as claimed in the '321 Patent). EX1005, 43, 87, 120.



EX2004, ¶134.

And further, Dr. Richardson points out that none of the start codes disclosed in MPEG-1 are first image frames of independent sequences. Instead, the start codes identify "some of the layers in the coding syntax." EX1005, 26, 31, 41-44. Moreover, additional information follows the start codes before decoding the first image frame, as shown below (from EX1005, 36-38):

2.4.2.3 Sequence header

Syntax	No.of bits	Mnemonic
sequence_header() {		
sequence_header_code	32	bslbf
horizontal_size	12	uimsbf
vertical_size	12	uimsbf
pel_aspect_ratio	4	uimsbf
picture_rate	4	uimsbf
bit_rate	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantizer_matrix	1	
if (load_intra_quantizer_matrix)		
intra_quantizer_matrix []	8*64	uimsbf
load_non_intra_quantizer_matrix	1	
if (load_non_intra_quantizer_matrix)		
non_intra_quantizer_matrix []	8*64	uimsbf
next_start_code()		
if (nextbits() == extension_start_code) {		
extension_start_code	32	bslbf
while (nextbits () != '0000 0000 0000 0000 0000 0001') {		
sequence_extension_data	8	
}		
next_start_code()		
}		
if (nextbits() == user_data_start_code) {		
user_data_start_code	32	bslbf
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
user_data	8	
}		
next_start_code()		
}		
}		

EX1005, 36.

2.4.2.4 Group of pictures layer

Syntax	No. of bits	Mnemonic
group_of_pictures() {		
group_start_code	32	bslbf
time_code	25	
closed_gop	1	
broken_link	1	
next_start_code()		
if (nextbits() == extension_start_code) {		
extension_start_code	32	bslbf
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
group_extension_data	8	
}		
next_start_code()		
}		
if (nextbits() == user_data_start_code) {		
user_data_start_code	32	bslbf
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
user_data	8	
}		
next_start_code()		
}		
do {		
picture()		
} while (nextbits() == picture_start_code)		
}		

EX1005, 37.

2.4.2.5 Picture layer

Syntax	No. of bits	Mnemonic
picture() {		
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if ((picture_coding_type == 2) || (picture_coding_type == 3)) {		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if (picture_coding_type == 3) {		
full_pel_backward_vector	1	
backward_f_code	3	uimsbf
}		
while (nextbits() == '1') {		
extra_bit_picture	1	"1"
extra_information_picture	8	
}		
extra_bit_picture	1	"0"
next_start_code()		
if (nextbits() == extension_start_code) {		
extension_start_code	32	bslbf
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
picture_extension_data	8	
}		
next_start_code()		
}		
if (nextbits() == user_data_start_code) {		
user_data_start_code	32	bslbf
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
user_data	8	
}		
next_start_code()		
}		
do {		
slice()		
} while (nextbits() == slice_start_code)		
}		

EX1005, 38; EX2004, ¶135

Thus, Petitioner has failed to show that its partially constructed MPEG-1 decoder discloses starting the decoding of the video sequence of the first image frame of the independent sequence. EX2004, ¶¶137.

3. Petitioner fails to demonstrate that limitations 8[d], 10[d], and 11[d] are obvious over MPEG-1 because MPEG-1 does not teach or suggest resetting the identifier value as required

Each of limitations 8[d], 10[d], and 11[d] are written with similar language and require "[resetting/reset] the identifier value for the indicated first image frame of the independent sequence." Petitioner has failed to show that MPEG-1 teaches or suggests these [d] limitations. EX2004, ¶138.

Petitioner claims that the Temporal Reference discloses "resetting the identifier value for the indicated first image frame of the independent sequence," but Petitioner fails to establish that the *decoder* resets the identifier value as the claim requires. EX2004, ¶139. Dr. Havlicek acknowledged that the challenged claims involve actions by decoders rather than encoders. EX2005, 38:19-39:1 ("They [the challenged claims] are all saying a decoding method and implementations thereof.").

The claims require the ***decoder*** (not the encoder) to reset the identifier value for the indicated first image frame of the independent sequence. The Board, in the Institution Decision, acknowledged that Temporal Reference is "assigned (in sequence) to the . . . pictures in display order," beginning with 0 for the first picture in display order. Paper No. 10, at 35 (citing EX1005, 44, 87). Temporal Reference is assigned by the encoder with no further action by a decoder. EX2004, ¶140. In sum, MPEG-1's temporal reference is only set by the ***encoder*** for the first image frame in display order with no action by a decoder corresponding to the challenged

claims. EX1005, 44, 87. Thus, MPEG-1 fails to teach this claim element. EX2004, ¶140.

The Temporal Reference is a syntax element whose value is determined by the encoder that informs the decoder which picture is the first in display order "since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them." EX1005, 87; EX2004, ¶141.

It is clear that MPEG-1 does not teach that the ***decoder resets*** the Temporal Reference to 0. As Dr. Richardson explains, at most, it teaches that the ***encoder*** assigns the first picture in display order to zero, and the decoder reads that zero from the bitstream. EX1005, 87; EX2004, ¶142. The closest Petitioner comes to showing a "reset" is in arguing that after decoding an exemplary closed GOP, "when an MPEG-1 decoder decodes the next exemplary closed GOP, the Temporal Reference for the first picture in decoding order of that exemplary closed GOP would be reset to zero, as that picture would be the first picture in display order for that next GOP." Pet. 28.

The Board relied on thesame passage. Paper No. 10, at 35-36 (citing to Ex. 1005, 44; *see* Ex. 1003, ¶¶ 109-110). This disclosure and argument fails for two reasons. First, the encoder, not the decoder assigns the Temporal Reference value, and therefore there is no system or method for decoding in MPEG-1 that resets the Temporal Reference. EX1005, 87; EX2004, ¶143.

Second, the claim language states "resetting the identifier value for ***the*** first image frame of ***the*** independent sequence." *See* '321 Patent, Claim 8 (and similar for other challenged claims). That is, the resetting of the identifier value is reset for the independent sequence of the prior limitations. EX2004, ¶144. This is the antecedent basis used in the claims.

But Petitioner (and the Board) rely on the Temporal Reference being set to zero for the ***subsequent first image frame of the subsequent*** independent sequence, not the first image frame of the independent sequence being decoded: "when an MPEG-1 decoder decodes ***the next*** exemplary closed GOP, the Temporal Reference for the first picture in decoding order of ***that exemplary closed GOP*** would be reset to zero, as that picture would be the first picture in display order for ***that next GOP***." Pet. 28 (emphasis added). Thus, even putting aside the fact that an MPEG-1 decoder does not reset Temporal Reference, at best Petitioner has pointed to a resetting of "the next independent sequence" not "the independent sequence" referred to earlier in the same claim, though Petitioner has not actually shown the decoder resetting any identifier value. EX2004, ¶145.

Further, for Petitioner to succeed on its argument, it has to show that the Temporal Reference is reset to zero for the indicated first image frame of the independent sequence. But MPEG-1 does not require that an I-frame (which is what Petitioner asserts to be the indicated first image frame in decoding order) always

have a temporal reference of zero, and as a result there can be no "resetting" of an identifier value. This is confirmed by Fig. D.18(b), which shows B-frames with Temporal Reference values of 0 and 1, while the first I-frame has a Temporal Reference of 2. Ex 1005 at 86; EX2004, ¶147.

Thus, Petitioner has failed to show that MPEG-1 decoder discloses resetting the identifier value for the indicated first image frame of the independent sequence. EX2004, ¶¶148.

B. Petitioner Has Not Shown That The Challenged Claims Are Unpatentable In Light Of Ground 2—Kim.

Unlike the '321 Patent, which is directed to streaming video, Kim is directed to time lapse recording/reproduction of individual image frames. EX2004, ¶79; EX1006, 1:1-19. Time lapse captures intermittent images and records them. Examples include security systems where continuous video is not necessary. Kim's time lapse recording/reproduction is not intended for continuous playback like the claimed video sequence. EX2004, ¶¶79-80.

Kim's decoder, shown in Fig 4, decodes and displays a single picture, based on a user request for one image. If the requested image is an I-picture, the corresponding coded image data is simply read from the optical disk, decoded, and displayed. As shown in the annotated figure below (Figure 4 alongside column 5 of Kim), the first image data decoded is the image data for the frame the user selected.

Kim is unaware at that point whether the frame is an I-frame or a P-frame. EX2004, ¶93.

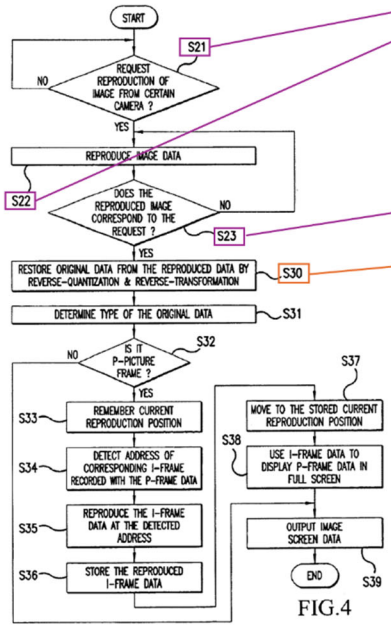


FIG.4

As shown in Steps S21 and S22 of FIG. 4, if the user requests reproduction of an image data from a particular camera or channel among the intermittent image data recorded on the optical disc 1, the controller 150 controls the recording/reproducing unit 140 to reproduce the corresponding image data from the optical disc 1. The image data reproduced from the optical disc 1 is multiplexed data, which is demultiplexed by the de-MUX 131. By demultiplexing, the camera (channel) ID code CHID is separated from the reproduced image data and output to the controller 150. Based on the camera ID code CHID, the controller 150 determines whether or not the reproduced image data corresponds to the selected camera or channel, in Step S23. This ensures that the correct image data is processed to comply with the user's request. If incorrect image data has been reproduced, the controller 150 controls the recording/reproduction unit 140 to reproduce the correct image data from the optical disc 1. In Step S30, the image data corresponding to the user's request is transmitted and decoded in the decoder 111. The decoding process of the decoder 111 involves reverse-quantization and reverse-transformation to restore the original image data prior to data compression carried out during recording. Then the property (type) of the currently restored image data is determined in Step S31 by the reproducing DSP 121, and the determination result is transmitted to the controller 150. More specifically, 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. Then in Step S34, the controller 150 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) from the reproduced image data output from the de-MUX 131, and reproduces based on the detected address the corresponding (or previous) I-picture frame data recorded on the optical disc 1 by controlling the recording/reproduction unit 140.

In Step S36, the reproduced corresponding I-picture frame data is processed by the reproducing DSP 121, and decoded by the decoder 111 to be stored in the frame memory 101. The controller 150, after Step S36, controls the recording/reproducing unit 140 to move the video heads to the original recording position stored in Step S33.

Annotated Kim, Figure 4 and 5:1-41; EX2004, ¶94.

As Dr. Richardson explains, ff the requested image is a P-picture, as Petitioner contends, the P-picture data is first read and decoded. Then the corresponding I-picture is located, read. and decoded. Then the P-picture is reconstructed (by combining the decoded P-picture and I-picture data) then displayed. EX1006, Fig 4, 5:8-6:8. Thus, in the scenario where a user requests a P-picture, the 0000 IAD value (which correlates to a different picture from the one the user requested) cannot be the indication of the ***first image frame in decoding order***. EX2004, ¶95.

If the requested image is an I-picture, no indications are used. Rather, the corresponding coded image data is simply read from the optical disk, decoded, and displayed as shown in annotated figure above (Figure 4 alongside column 5 of Kim).

Thus, Kim does not identify decoding any indication of a first image frame of an independent sequence, starting the decoding from the indicated first image frame, and resetting an identifier value of the indicated first image frame. EX2004, ¶95. When the user selects an image for output, Kim teaches to start decoding at the user-selected image (not an indicated first image frame of an independent sequence). EX2004, ¶96. Nor does Kim teach that the decoder resets the IAD value of the user-selected image—or any image whatsoever. Kim's output of a single image frame does not solve the random-access problems discussed above. EX2004, ¶96. Kim fails to disclose several limitations of the asserted claims that avoid errors and enable random access from the user-specified image.

1. Petitioner fails to demonstrate that limitations 8[a], 10[a], and 11[a] are obvious over Kim

Each of limitations 8[a], 10[a], and 11[a] are written with similar language and require "[decoding/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." Petitioner has failed to show that Kim teaches or suggests these limitations. EX2004, ¶154.

As Dr. Richardson explains, Petitioner identified the 0000 IAD value of the I-frames as satisfying an indication of the first image frame in decoding order. However, as explained, when a user selects a P frame, the user-selected frame (not the I frame of a sequence) is decoded first, and thus the 0000 IAD value cannot indicate the first image frame in decoding order. EX1006, 5:12-42, Fig. 4; EX2004, ¶155. Moreover, as also explained, when the user selects an I frame, the 0000 IAD value is not used as any indication at all. EX1006, at Figure 4.

Petitioner looksEX2004, ¶ Figure 5A of Kim as "show[ing] the recited 'indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.'" Pet. 44. Specifically, Petitioner states that "Kim also teaches the recited 'indication'" because "in Kim's Figure 5A, an IAD of 0000 or zero is 'in indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence.'" EX2004, ¶156.But this is incorrect, as discussed below.

First, during deposition, Dr. Havlicek confirmed that decoding starts at S23, which is the user-selected frame. EX2005, 28:6-20. Specifically, "under the assumption that a P-picture was requested, then some elements of the data associated with the P-picture will be decoded again in diamond S23." EX2005, 27:3-28:5. Thus,

even if Kim decodes the IAD when it decodes the user-selected frame (as a P-frame), it still starts decoding that user selected frame, not the frame referenced by the IAD that has an associated IAD of 0000. EX2004, ¶154.

Thus, even accepting all of Petitioner's arguments about the IAD being an indication of an image frame, that IAD is not of the first image frame in decoding order of the independent sequence because Dr. Havlicek confirmed that Kim starts decoding at the user selected frame regardless of whether it is an I-frame or a P-frame. EX2005, 28:6-20. EX2004, ¶157.

Further, Petitioner misses that Kim confirms that these IAD values are not decoded by the decoder at all and are instead only used by the controller after the decoder steps of Figure 4 occur. EX2004, ¶159; EX1006, 5:29-31, Fig. 4. In Petitioner's argument, it suggests that a POSITA would combine the bitstream of Figure 5B with the decoding pathways of Figure 4. Pet. 41. EX2004, ¶159. This is crucial, because Figure 4, then, provides the only source of potential evidence for Petitioner on decoding. But the specification makes clear that the controller of Kim does not look at IAD information until after it decodes the requested frame. EX2004, ¶159. This is true regardless of whether a requested frame is an I-frame or a P-frame.

As explained in column 5 of Kim, the decoder first goes about reproducing the correct, requested image data (lines 11-25, also shown as steps S21, S22, and S23 in Figure 4). Only after that decoding does the Kim decoder look to determine

what "type" the original data is. EX2004, ¶160; Fig. 4 (step S31/S32); 5:38-45. If the Kim decoder started decoding at the claimed first image frame in decoding order, the Kim decoder would not need to determine what type (I- or P-) the original data is to begin decoding. EX2004, ¶160. Even the Board acknowledged that "Kim discloses that 'the image data corresponding to the user's request is transmitted and decoded' by a decoder" and then "[a]fter decoding the image data, Kim's controller uses the IAD in the image data, e.g. to 'detect[] the address of the corresponding I-picture frame' if the image data is a P-picture." Paper No. 10, at 58. Thus, the IAD of Kim is not an indication of a first image frame in decoding order of an independent sequence because it does not indicate which image frame for the decoder to begin decoding. Kim begins decoding at the user-selected frame. EX2004, ¶161.

Dr. Havlicek suggested that the IAD or CAD may be read before step 23 or outside of Figure 4. EX2005, 51:1-16. But, as Dr. Richardson explains, this does not help Petitioner, and, to the extent Petitioner attempts to argue from it, such an argument is not disclosed in the Petition or in Dr. Havlicek's declaration and therefore would be an improper new argument. Further, decoding IAD and CAD values when finding the user-selected frame still does not disclose decoding an identification of a first image frame in decoding order, because (as Dr. Havlicek admits) the first image frame to be decoded is the one the user selected, whether it is an I-frame or a P-frame. EX2004, ¶162. Moreover, the IAD and CAD values that

Petitioner pointed to are those in Figure 5A (the logical addresses). But when searching on optical medium, Kim would utilize the physical address:

> Q. Kim discloses recording onto an optical disk, which is a physical media, right?
>
> A. Correct.
>
> Q. And the way one would seek into a physical media would be, for example, going into the physical address of where that information is stored, correct?
>
> A. At the low level of the device, yes, I agree.

EX2005, 55:16-56:4. And Kim does not describe the physical address of the requested frame or the corresponding I-frame as the IAD or CAD but treats those as separate values. *See* EX1006, 4:59-5:7 (describing the physical addresses); EX2004, ¶163. Moreover, these could not be IAD or CAD values because each value would need to be unique as a physical address when stored on the optical medium. EX2004, ¶163. In contrast, the IAD or CAD values are logical addresses (as disclosed in EX1006, 4:46-58). These do not need to be unique because the IAD points to the "corresponding or previous I-frame" and is a relative pointer to the requested user frame, not an absolute address. *See* EX1006, 4:46-58; EX2004, ¶¶163-164.

Thus, Petitioner has failed to show that Kim discloses an indication of an image frame, which is the first image frame, in decoding order, of an independent sequence as required by the [a] limitations. EX2004, ¶165

2. Petitioner fails to demonstrate that limitations 8[b], 10[b], and 11[b] are obvious over Kim

Each of limitations 8[b], 10[b], and 11[b] are written with similar language and require "starting the decoding of the video sequence from said first image frame of the independent sequence." Kim fails to disclose these limitations because Kim teaches starting decoding from the user-selected frame, not the first image frame of the independent sequence. EX1006, 5:12-38, Fig. 4. Instead, Kim's decoding begins at step S30 of Figure 4, where the user-selected image is decoded. The user-selected image is not the first image frame of the independent sequence. EX2004, ¶166.

As Dr. Richardson explained, Petitioner argues that the indicated first image frame in decoding order is the I-frame of Figure 5A that precedes the P-frames that refer to it through the IAD. Pet. 47. As stated above, however, instead of starting the decoding of the video sequence from said first image frame (the one for which the indication identified), Dr. Havlicek admitted that Kim teaches starting the decoding of the video sequence from the user selected frame, which can be a P-picture frame, rather than starting the decoding of the video sequence from "said first image frame", which Petitioner identifies as the corresponding I-picture frame:

> Q. And where will that decoding process start in either of those two cases [whether the user requests an I-picture or a P-picture]?
>
> A. As you said correctly, in the diamond S23."

Ex. 2005, 28:18-20

> "Q. And so the first step that Kim takes in [servicing the user request] is it's going to start the decoding of whatever frame or image that the user requested by whichever manner that that happens, like you just talked about; is that right?
>
> A. I agree.

Ex. 2005, 48:8-13.. EX2005, 28:6-20, 48:8-13; EX2004, ¶XX.

As Dr. Havlicek admits, Kim does not start decoding at the first image frame of the independent sequence that Petitioner identified (the I-frame). Petitioner may argue, however, that before the user selected frame is displayed as an output, Kim discloses decoding the I-frame. *See, e.g.*, Pet. 51-52 ("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."); EX2004, ¶169. But the claim language refers to "starting the decoding of

the video sequence." Petitioner attempts to add an additional requirement that starting the decoding must mean producing the image data for display. This is incorrect. EX2004, ¶169.

Further, Petitioner suggests this limitation would be disclosed by only the I-frame path (i.e. if Kim did not have any P-pictures). *See* Pet. 51. But the Kim disclosure must be understood for all its teachings, which is a time lapse device for recording and reproducing intermittent data using P-pictures, e.g. compressed image data such as I-picture frame data and P-picture frame data. *See* Paper No. 10, at 43-44 (citing EX1006, 3:60-61, 4:19, 4:28-33, 6:4-8, 6:55-58 (claim 3)." As Dr. Richardson explains, a POSITA would not reasonably view Kim's disclosure of how to process I-frames as teaching a decoder that starts decoding an independent sequence at said first image frame. EX2004, ¶170. Further, the said first image frame is what Petitioner has identified is the frame that corresponds to the IAD, which Petitioner alleges to be the claimed "indication." But if Kim only handled I-frames, the IAD (that is zero or absent) would not identify any image frame and would not be needed since all frames would be I-frames. EX2004, ¶171. This contorted view is improper.

Thus, Petitioner has failed to show that Kim discloses "starting the decoding of the video sequence from said first image frame of the independent sequence." EX2004, ¶172.

3. Petitioner fails to demonstrate that limitations 8[c], 10[c], and 11[c] are obvious over Kim

Each of limitations 8[c], 10[c], and 11[c] are written with similar language and require "decoding identifier values for image frames according to a numbering scheme." Petitioner relies on (and the Board adopted) that the logical addresses in Figure 5A were the identifier values for image frames according to a numbering scheme that are being decoded. As shown below, the decoder does not decode these logical addresses during decoding but only physical addresses (and there only sometimes).

Petitioner uses, and the Board adopts Petitioner's position, that the logical addresses in Figure 5A as the identifier values for image frames according to a numbering scheme that are decoded. But this is incorrect. Kim does not disclose decoding these logical values at all. EX2004, ¶174.

When Kim begins decoding, as stated above, Kim starts with the user-selected frame. Dr. Havlicek acknowledges that recording and reproducing on an optical disk, which is physical media, involves utilizing the physical address of what that information is stored. EX2005, 55:16-56:4 (Q. Kim discloses recording onto an optical disk, which is a physical media, right? A. Correct. Q. And the way one would seek into a physical media would be, for example, going into the physical address of

where that information is stored, correct? A. At the low level of the device, yes, I agree."). EX2004, ¶175.

When beginning decoding of the user-selected frame, Kim does not disclose using the logical address. Instead, if anything, the physical address is used. For an I-frame, no other address is required (and Step 34 is never encountered). For a P-frame, Kim discloses using the physical address for an I-frame, not the logical address that Petitioner points to. EX2004, ¶176.

More specifically, in Step S34, Kim's controller "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) from the reproduced image data output from the de-MUX **131**." EX1006, 5:41-44. Area **(3)** of the bit stream B2 shown in Figure 6 is "the address of the corresponding I-picture frame," *id.* at 4:59-65, which Figure 6 confirms is the physical address of the I-frame (not the logical address). For an I-frame, no other address is required (and Step 34 is never encountered). For a P-frame, Kim does discloses using the physical address for the I-frame, not the logical address that Petitioner points to. EX2004, ¶177.

That is, only physical addresses are decoded. These physical addresses are not according to a numbering scheme. Figure 6 further illustrates why the CAD is an address that is not according to a numbering scheme. One knows this because the CAD is dependent on an image size, which is variable and not predictable. For

example, Figure 6 shows that the address for the second P picture is 100000 + N, where N is the Byte length of the I frame (B1). Specifically, N is not determined according to a numbering scheme but rather according to the encoded size of the picture (which depends on the coding type, content, and size of the picture data) that is necessarily variable and unpredictable. EX2004, ¶¶178-179.

Thus, the logical addresses are not decoded by Kim and cannot meet this limitation. Likewise, the physical addresses are not according to a numbering scheme. Thus, this limitation is not met. EX2004, ¶¶179.

4. Petitioner fails to demonstrate that limitations 8[d], 10[d], and 11[d] are obvious over Kim because Kim does not teach or suggest resetting the identifier value as required

Each of limitations 8[d], 10[d], and 11[d] are written with similar language and require a decoder to "[reset] the identifier value for the indicated first image frame of the independent sequence." Petitioner has failed to show that Kim teaches or suggests these limitations because the claims require that the ***decoder*** reset the identifier value, but Kim teaches that the ***encoder*** sets any identifier value. Kim never teaches a decoder resetting any identifier values. EX2004, ¶¶180.

First, Petitioner asserts that the IAD component of each logical address is reset to zero for each I-picture, but Petitioner fails to establish that the *decoder* resets the identifier value as required by the claims. Pet., 55. In fact, neither Petitioner nor Dr. Havlicek even allege that Kim teaches a *decoder* resetting the identifier value. *See*,

e.g., Pet., 55-58; EX1003, ¶¶ 169-174. Contrary to the claims of the '321 Patent, Kim teaches that "the address of the I-picture data received from the encoder 110." EX1006, 4:38-42; EX2001, ¶¶ 119-120. The Board's Institution Decision does not address this fundamental flaw. In stating that a prior-art reference must be considered for everything it teaches, the Board did not address that Kim only teaches the alleged resetting of an identifier value at the *encoder* and not the *decoder*. Paper No. 10, 61-63; EX2004, ¶181.

Second, and contrary to Petitioner's argument, Kim does not teach that its reproducing method resets the alleged identifier value, IAD. Petitioner argues, at page 55 of the Petition, that Figure 5A "shows that the IAD component of each logical address is reset to zero for each I-picture" and provides an annotated Figure 5 (reproduced below) where the I-picture logical address is in red and the "whole logical address" is outlined in purple:

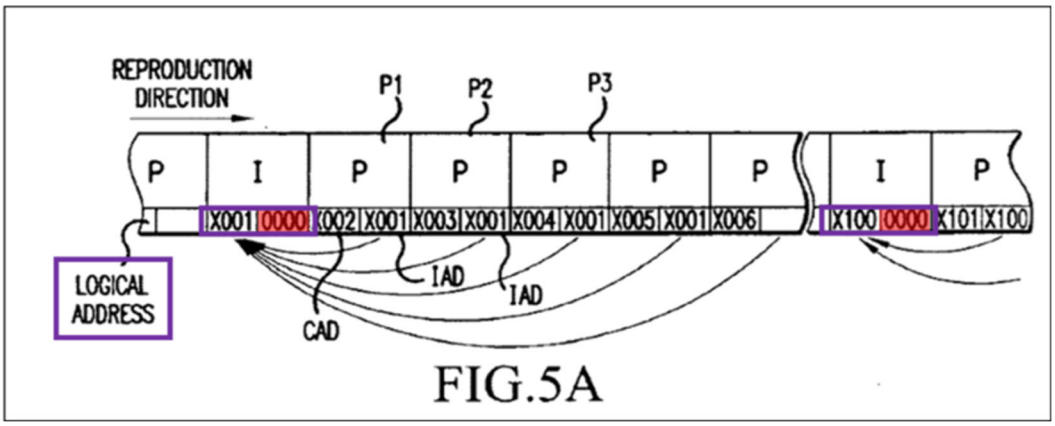


FIG.5A

Pet. 56. EX2004, ¶181. Likewise, the Board, in its Institution Decision pointed solely to Figure 5A as the disclosure of the indicated value being reset. Paper No. 10, at 63. This is incorrect. EX2004, ¶182. Specifically, the Board, stated that "regardless whether the other parts of Kim's specification use words to describe resetting the IAD component of each logical address to zero for each I-picture, Figure 5A still discloses that feature." Paper No. 10, at 63.

But Kim describes that Figure 5A "illustrate[s] examples of logical data structure of a digital data stream ***compressed and recorded on the optical disk***." EX1006, 2:48-51 (emphasis added); EX2004, ¶183. This (that is, compression and recording) involves encoding of the data, not the decoding (or reproducing) of the data. In contrast, Kim discusses the act of reproducing (and as shown in Figure 4) as the decoding of the image data, and Dr. Havlicek agrees (with the caveat that he believes there is some part of decoding that occurs which may not be shown in Figure 4 for the IAD). EX2005, 37:3-38:15. Petitioner likewise points to Figure 4 (which Kim calls reproducing) for where Kim allegedly discloses two paths for decoding an image in response to a user requesting playback. *See* Pet. 48. Thus, an IAD that is recorded in Figure 5A is done so during the encoding process, not the decoding process, and therefore Figure 5A cannot disclose a decoder resetting the identifier value. EX2004, ¶¶183-184.

Kim does not teach or suggest a decoder resetting the IAD when decoding or reproducing the requested image data. Thus, Petitioner has failed to show that Kim teaches "[resetting/reset] the identifier value for the indicated first image frame of the independent sequence" of the [d] limitations. EX2004, ¶185.

C. Petitioner Has Not Established A Reasonable Likelihood That The Challenged Claims Are Unpatentable In Light Of Ground 3.

1. Petitioner fails to demonstrate that Claim 9 is obvious over MPEG-1 in combination with Yagasaki

Petitioner cannot show it met its burden for Ground 3. Claim 9 depends on Claim 8 and Petitioner relies on its Ground 1 to show the predicate independent claim is taught or suggested. But, as shown above, MPEG-1 fails to teach or suggest at least limitation 8[a], 8[b], and 8[d]. EX2004, ¶XX. For that reason, Ground 3 fails.

IV. CONCLUSION

For the reasons set forth above, Patent Owner respectfully requests that the Board find that Petitioner has not shown any of the challenged claims invalid under any asserted ground.

Dated: December 18, 2024

Respectfully submitted,

/s/Scott W. Hejny
Scott W. Hejny
Reg. No. 65,320
shejny@mckoolsmith.com
Eric S. Hansen
Reg. No. 63095
ehansen@mckoolsmith.com
McKool Smith, P.C.
300 Crescent Court, Suite 1500
Dallas, Texas 75201
Telephone: (214) 978-4000
Facsimile: (214) 978-4044

Counsel for Patent Owner, Nokia Technologies Oy

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PATENT OWNER'S PRELIMINARY RESPONSE** specified in 37 C.F.R. §42.24 include 13,999 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. §42.24(b)(1). This word count was prepared using Microsoft Word 2010.

Dated: December 18, 2024

Respectfully submitted,

/s/ *Scott W. Hejny*
Scott W. Hejny
Reg. No. 45,882

CERTIFICATION OF SERVICE

The undersigned hereby certifies that a copy of this **PATENT OWNER'S PRELIMINARY RESPONSE PURSUANT TO 37 C.F.R. § 42.10(b)** has been served on Petitioner via email on the following counsel of record for Petitioner:

Jessica Kaiser
kaiser-ptab@perkinscoie.com

Robin Brewer
Brewer-ptab@perkinscoie.com

R. Tyler Kendrick
kendrick-ptab@perkinscoie.com

Ji Wang
Wang-ji-ptab@perkinscoie.com

Jon R. Carter
Carter-ptab@perkinscoie.com

Amazon-NokiaIPR@perkinscoie.com

Dated: December 18, 2024

Respectfully submitted,

/s/ *Scott W. Hejny*
Scott W. Hejny
Reg. No. 45,882