Filed on behalf of Unified Patents Inc.
By:

ERISE IP, P.A.
Eric A. Buresh, Reg. No. 50,394
eric.buresh@eriseip.com
7015 College Blvd., Suite 700
Overland Park, Kansas 66211

Michelle Callaghan, Reg. No. 75,665
michelle.callaghan@eriseip.com
5600 Greenwood Plaza Blvd., Suite 200
Greenwood Village, Colorado 80111
Telephone: (913) 777-5600

UNIFIED PATENTS INC.
Ashraf Fawzy (Reg. No. 67,914)
afawzy@unifiedpatents.com
1875 Connecticut Ave., NW, Floor 10
Washington, D.C. 20009

Roshan Mansinghani, Reg. No. 62,429
roshan@unifiedpatents.com
Unified Patents Inc.
13355 Noel Road, Suite 1100
Dallas, TX, 75240
Telephone: (214) 945-0200

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UNIFIED PATENTS INC.,
Petitioner

v.

VELOS MEDIA, LLC,
Patent Owner

Case No. IPR2019-00757
Patent No. 9,930,365

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,930,365

TABLE OF CONTENTS

I. INTRODUCTION

Petitioner Unified Patents Inc. ("Petitioner" or "Unified") respectfully requests *inter partes* review ("IPR") of Claims 1-20 (collectively, the "Challenged Claims") of U.S. Patent No. 9,930,365 ("the '365 Patent," Ex. 1001).

The '365 Patent describes systems and methods for encoding and decoding video data for "large" block sizes, i.e., block sizes larger than 16×16 pixels. However, the concept of employing video blocks larger than 16×16 pixels was already practiced years before the time of the '365 Patent. The claims of the '365 Patent were allowed during prosecution based on the requirement of using two distinct syntax elements representative of a minimum and maximum size for blocks in a sequence of pictures. However, there is no suggestion in the specification that the concept of using minimum and maximum syntax elements was a novel concept, or that the use of syntax elements indicating minimum and maximum block sizes was somehow necessary or even beneficial to enabling the use of larger block sizes in video coding. In any case, the use of syntax elements representative of the minimum and maximum block sizes in a coding unit was known long before the time of the '365 Patent, as shown by both *Kalker* and *Novotny*, discussed in more detail below. The Challenged Claims are therefore obvious over the prior art cited herein and should be found unpatentable.

II. SUMMARY OF THE PATENT

A. Technology Background

Digital video is formed from a sequence of video frames that include picture element (or pixel) data. *See Freedman Decl.* (Ex. 1009) at ¶34 (citing Richardson, Ex. 1011). During playback, the frames are successively displayed at a certain rate, rendering the video for display. *Id*. The rate at which successive frames are displayed should be high enough such that the transition from frame to frame is imperceptible to the human eye. *Id*. Each frame is an array of pixels organized in rows and columns to form the image represented by the frame, which reflect characteristics of objects represented in a scene of a video.

Video files can be large due to the large amounts of image data associated with each frame. *Id*. at ¶35. Therefore, video coding techniques are used to compress (i.e. encode) video files for efficient transmission for receipt and decompression (i.e., decoding) and output at an end-user display device. *Id*. Such compression is achieved by removing redundancy in and between frames. *Id*. at ¶36. Specifically, within a particular sequence of video images, individual frames can be correlated to benefit from redundant video information from within a given frame (spatial correlation) and from successive frames captured at around the same time (temporal correlation):

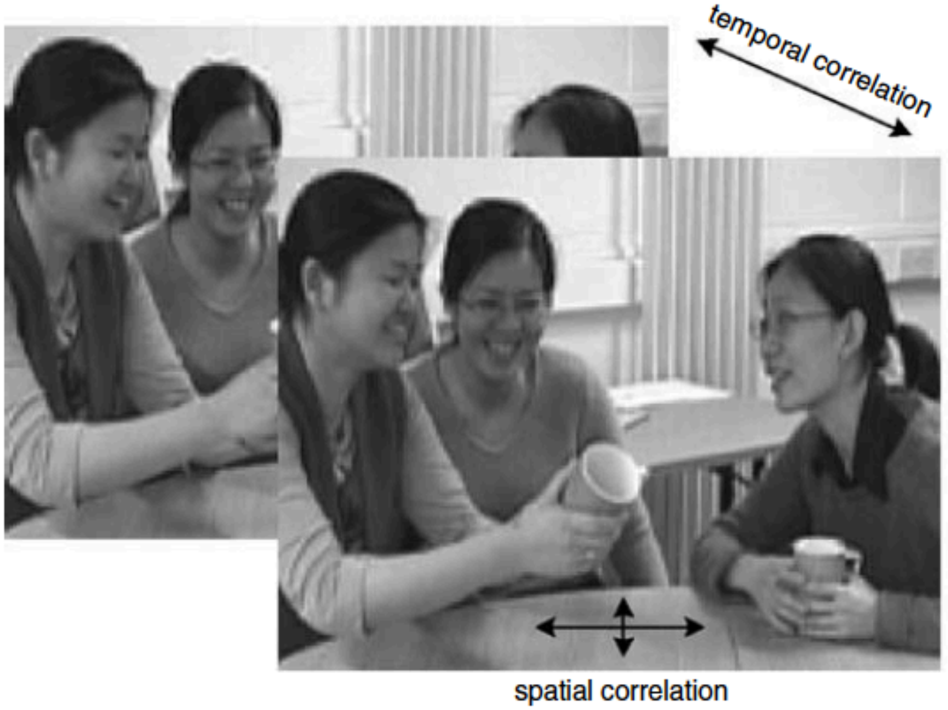


Figure 3.2 Spatial and temporal correlation in a video sequence

Richardson (Ex. 1011) at 53, Fig. 3.2.

Many aspects of video coding were well-known long before the ’365 Patent, including block-based video coding employing prediction techniques to remove spatial and temporal redundancy in coded video data. *See ’365 Patent* (Ex. 1001) at 1:39-49. To do so, video coders would use four processes (*inter alia*), discussed below: (1) partitioning frames into different sections, such as slices, macroblocks, and sub-blocks, (2) removing redundancy by identifying predicted blocks and their respective reference block or frame, (3) removing residual data that contains unimportant visual information using transform operations and quantization, and (4) encoding the reduced amount of data using various techniques, such as Huffman coding and/or variable-length coding (describing frequent events with shorter code-words than those used for less frequent events).

In block-based video coding, such as the H.264 standard mentioned in the background of the ’365 Patent, each video frame is partitioned into macroblocks containing a certain number of pixels, and each macroblock could be further partitioned into sub-blocks or blocks. *Id.*; *see also Freedman Decl.* (Ex. 1009) at ¶37. For original blocks of data, a prediction technique or mode generates a corresponding block elsewhere in the video frame or sequence. *Id.* As acknowledged in the background of the ’365 Patent (and illustrated in Fig. 3.2 above), it was well-known to generate predicted blocks using **intra-prediction** (spatial prediction) and **inter-prediction** (temporal prediction). *See ’365 Patent* (Ex. 1001) at 1:39-49. **Intra-prediction** involves generating a predicted block using similarities that exist between an original block and other blocks within the **same frame**. *Id.*; *see also Freedman Decl.* (Ex. 1009) at ¶¶37-38. **Inter-prediction** involves generating a predicted block using similarities existing between neighboring blocks in the same frame or temporal prediction with respect to corresponding blocks in **other frames** in the video sequence. *See Freedman Decl.* (Ex. 1009) at ¶¶37-38.

The block is next subject to mathematical transform operations, such as a discrete cosine transform, that converts frame pixel data from the spatial domain into a frequency domain. *Id.* at ¶39. This operation discards the less important visual information within a predicted frame. *Id*. These transformed pixel values are referred to as "transform coefficients." *Id*. The coefficients may be further compressed via an irreversible process called "quantization," whereby a matrix of transform coefficients are divided by a corresponding quantization value and the resulting coefficient is rounded. Quantized blocks containing all zero values are often referred to as "skipped" or "zero" blocks and encoded with very few bits to indicate that the predicted block is rendered as identical to the reference block.

Tables 3.9-11 of *Richardson* illustrate this concept. Table 3.9 shows residual data for an 8×8 block of pixels:

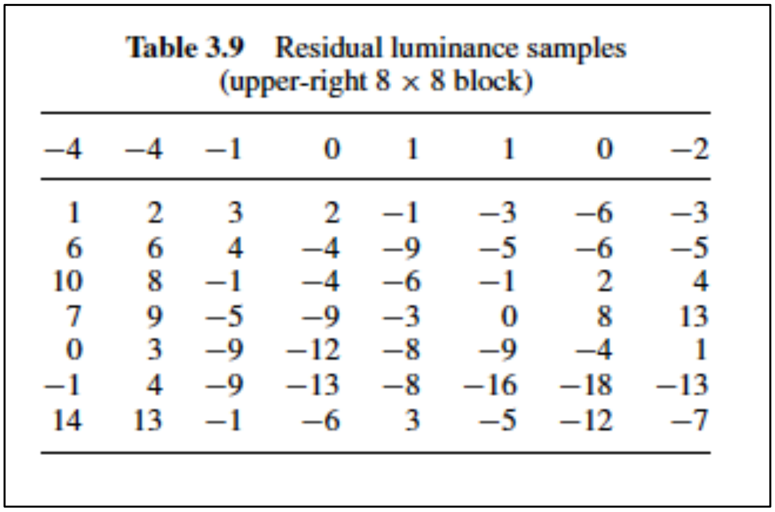


Table 3.9 Residual luminance samples (upper-right 8 × 8 block)

−4	−4	−1	0	1	1	0	−2
1	2	3	2	−1	−3	−6	−3
6	6	4	−4	−9	−5	−6	−5
10	8	−1	−4	−6	−1	2	4
7	9	−5	−9	−3	0	8	13
0	3	−9	−12	−8	−9	−4	1
−1	4	−9	−13	−8	−16	−18	−13
14	13	−1	−6	3	−5	−12	−7

Table 3.10 shows the same residual data with a DCT transform operation assigned to it to transform the pixels into the frequency domain:

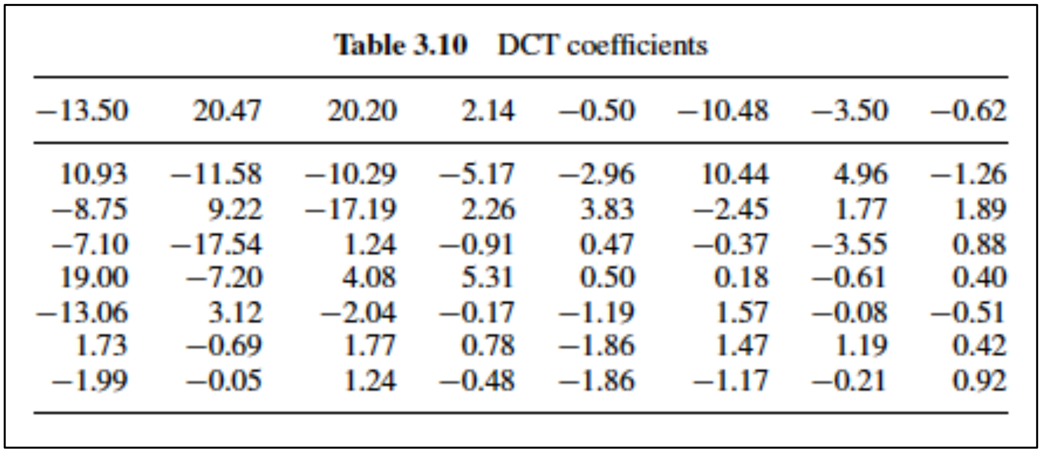


Table 3.10 DCT coefficients

−13.50	20.47	20.20	2.14	−0.50	−10.48	−3.50	−0.62
10.93	−11.58	−10.29	−5.17	−2.96	10.44	4.96	−1.26
−8.75	9.22	−17.19	2.26	3.83	−2.45	1.77	1.89
−7.10	−17.54	1.24	−0.91	0.47	−0.37	−3.55	0.88
19.00	−7.20	4.08	5.31	0.50	0.18	−0.61	0.40
−13.06	3.12	−2.04	−0.17	−1.19	1.57	−0.08	−0.51
1.73	−0.69	1.77	0.78	−1.86	1.47	1.19	0.42
−1.99	−0.05	1.24	−0.48	−1.86	−1.17	−0.21	0.92

And Table 3.11 shows this transformed data quantized by rounding the result of the coefficients divided by some quantization step size or parameter, which was 12 in this example:

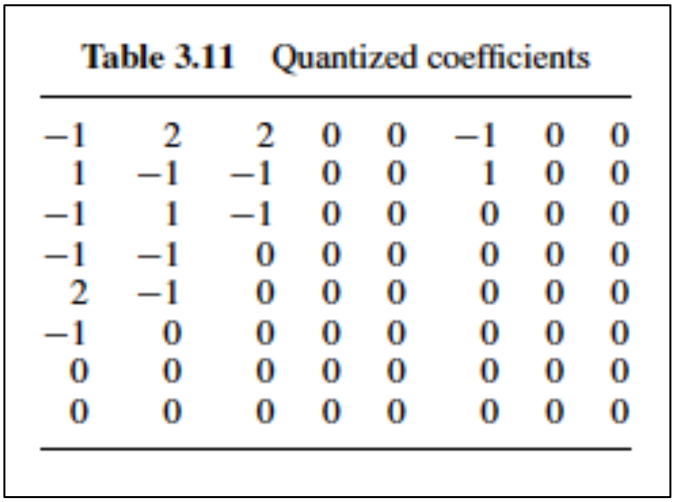


Table 3.11 Quantized coefficients

−1	2	2	0	0	−1	0	0
1	−1	−1	0	0	1	0	0
−1	1	−1	0	0	0	0	0
−1	−1	0	0	0	0	0	0
2	−1	0	0	0	0	0	0
−1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

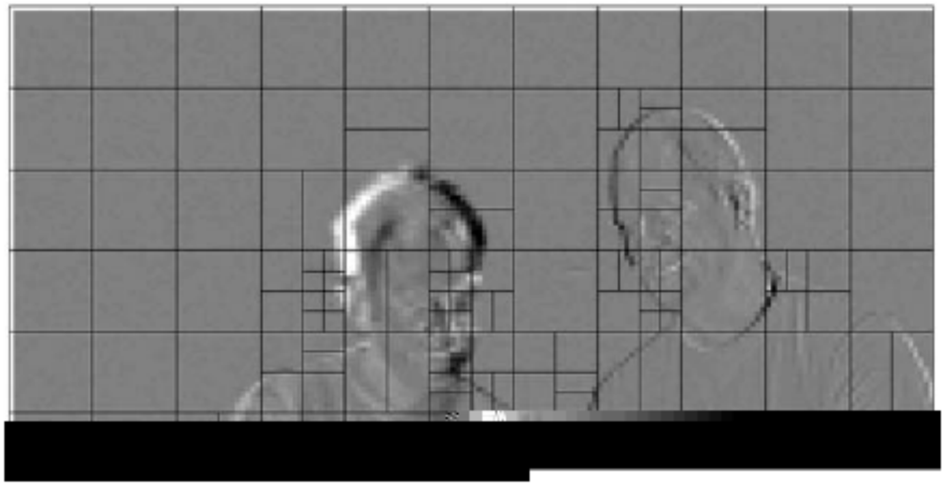
See Freedman Decl. (Ex. 1009) at ¶39 (citing *Richardson* (Ex. 1011)).

After the video data is encoded, it is stored or transmitted to a receiver for eventual decoding and display to a user. Decoders generally reverse the coding process performed by the corresponding encoder. *Id*. at ¶43. As acknowledged in the specification of the ’365 Patent, and argued by the applicant during prosecution of a parent patent, a PHOSITA would have recognized that the decoding side of the video codec simply performs a “decoding pass generally reciprocal to the encoding pass.” *See ’365 Patent* (Ex. 1001) at 18:6-8, 25:8-10. *See Pat. 9,788,015 File History* (Ex. 1005) at 3471 (“[O]ne of ordinary skill in the art would certainly appreciate that any data encoded by a video encoder must necessarily be decoded by a video decoder.”); *see also Freedman Decl.* (Ex. 1009) at ¶¶43-44. For example, decoders contain an inverse quantizer and transformer for reversing the transformation/quantization phase of the compression process. The inverse quantizer cannot perfectly reverse the quantization process performed by the encoder due to the rounding step; instead, it re-scales, or multiplies, the rounded coefficients by some value, such as the quantization parameter. *Freedman Decl.* (Ex. 1009) at ¶44. The re-scaled coefficients are then subject to inverse transformation operations to reverse the DCT process. *Id.*

Macroblock Partitioning

In the partitioning phase, the chosen partition size (e.g., 4×4 as compared to 32×32) involves a trade-off between the quality of the image and the quantity of data needed to represent the sequence of images. *See Freedman Decl.* (Ex. 1009) at ¶40. Frames made up of larger blocks require less data and, therefore the *rate* for transmission and decoding of such frames is higher; however, frames made up of smaller blocks, while more complex to encode, are more likely to account for anomalies and therefore contain less *distortion*. *Id*. To determine the best block size to use, many video encoders employ Lagrangian optimization functions that attempt to minimize distortion at a desired bit rate. *Id*.

Further, by 2008, video encoders were not limited to a single macroblock size per picture frame. Instead, it was conventional to partition frames into macroblocks and sub-blocks of varying size in a "tree" structure arranged in blocks of N×N, N×N/2, or N/2×N pixels, where *N* is an integer that is a power of two (e.g., 4, 8, 16, 32, 64, 128, 256). *See Freedman Decl.* (Ex. 1009) at ¶41. The example figure below shows a residual frame with different block sizes superimposed – the largest blocks in this example are 16×16 pixels, covering background areas where there is a significant amount of redundancy (i.e., relatively less color variation):



[redacted]

Figure 6.12 Residual (without MC) showing choice of block sizes

See id. (citing *Richardson* (Ex. 1011)). Although the H.264 and H.262 coding standards assumed a 16×16 macroblock size, it was conventional in various encoding systems to employ starting macroblocks of greater size. For example, *Chiang*, discussed in more detail below, assumes an initial block size of 256×256 pixels. *See Chiang* (Ex. 1008) at 5:43-60; *see also Freedman Decl.* (Ex. 1009) at ¶¶41-42.

B. Description of the Alleged Invention of the ’365 Patent

The ’365 Patent is directed to techniques for encoding digital video data using large macroblocks, i.e., macroblocks larger than a 16×16 array of pixels. *Id*. at 1:53-62. As mentioned, one benefit of encoding video frames using larger macroblocks is that a higher compression efficiency can be achieved, particularly in video data generated with higher spatial resolutions and frame rates (i.e., the number of frames displayed in a given unit of time). *See id*. at 7:11-19. The ’365 Patent notes that while a large macroblock generally refers to initial macroblocks of greater than 16×16 pixels, “large” macroblock includes a conventional 16×16 block, depending on the video resolution and frame rate. *Id*. at 7:43-57; *see also id*. at 7:8-10, 6:50-54.

The claims of the ’365 Patent are generally directed to the decoding of encoded video data employing three syntax elements. In the ‘365 Patent, a syntax element is used to describe information that is communicated to and used by the decoder to understand the characteristics or processing of encoded data so that the decoder can determine how to decode the encoded data. *See, e.g.*, ’365 Patent (Ex. 1001) at 11:13-18; *see also Freedman Decl.* (Ex. 1009) at ¶54. Particular claim limitations specify the information contained in particular syntax elements. For example, the Challenged Claims describe one syntax element representing a minimum size of blocks in a sequence of pictures, a second syntax element representing a maximum size of blocks in a sequence of pictures, where the maximum is greater than 16×16 pixels, and a third syntax element representing the encoding mode of the blocks (i.e., intra-prediction, inter-prediction). *See id*. at Claims 1, 7, 15. Accordingly, when processing encoded video data that has been partitioned into blocks, if a block of data is equal to the minimum size as indicated by the first syntax element, the decoder understands that the sub-block does not have further partitions. If the sub-block does not have further partitions, the decoder will decode the block according to the encoding mode as represented by the third syntax element.

While the claims require both larger-sized macroblocks and the use of minimum and maximum syntax elements, the specification fails to elaborate on how the use of such syntax elements enables the use of larger macroblocks. Nonetheless, both concepts are found in the prior art as detailed below.

C. Summary of the Prosecution History of the ’365 Patent

The ’365 Patent issued from an application filed on September 6, 2017 and claims priority to three provisional applications, the earliest of which was filed on October 3, 2008.[1] *See* ’365 Patent (Ex. 1001). The ’365 Patent is the fourth in a family of continuation patent applications, all of which were allowed based on a similar concept—the use of distinct syntax elements representative of the minimum and maximum block sizes in a coded unit. *See ’365 File History* (Ex. 1002) at 126; *see also Pat. 8,503,527 File History* (Ex. 1003) at 545 (statement of reasons for allowance); *Pat. 8,948,258 File History* (Ex. 1004) 1805, 64-70 (indicating that

[1] For the purposes of this Petition, Petitioner assumes, but does not concede, this to be the earliest priority date of the ’365 Patent.

dependent claims 2-3, 9-10,15-16, 19-20, 23-24, 30-31, 36-37, and 40-41, reciting limitations referring to first and second syntax elements, would be allowed if re-written in independent form); *Pat. 9,788,015 File History* (Ex. 1005) at 3645-46, 3474-76 (arguing that the prior art did not disclose a syntax element representing a minimum size of blocks of a sequence of pictures).

D. Level of Ordinary Skill in the Art

A person having ordinary skill in the art ("PHOSITA") would have been a person having, as of October 3, 2008: (1) at least an undergraduate degree in electrical engineering or closely related scientific field, such as physics, computer engineering, or computer science, or similar advanced post-graduate education in this area; and (2) two or more years of experience with video or image processing systems. *See Freedman Decl.* (Ex. 1009) at ¶¶30-32.

III. REQUIREMENTS FOR *INTER PARTES* REVIEW UNDER 37 C.F.R. § 42.104

A. Grounds for Standing

Petitioner certifies that the '365 Patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the Challenged Claims. 37 C.F.R. § 42.104(a).

B. Identification of Challenged Claims and Relief Requested

In view of the prior art, evidence, and analysis discussed in this Petition, IPR should be instituted and Claims 1-20 of the '365 Patent should be found unpatentable and cancelled based on the following proposed grounds of unpatentability. 37 C.F.R. § 42.104(b)(2).

Proposed Grounds of Unpatentability	**Exhibit No.**
Claims 1-4,6-10, 12-18, and 20 are obvious over U.S. Patent 5,999,655 to Kalker *et al.* ("*Kalker*") in view of U.S. Pub. 2005/0123282 to Novotny *et al.* ("*Novotny*")	**1006** **1007**
Claims 5, 11, and 19 are obvious over *Kalker* in view of *Novotny* in further view of U.S. Patent 6,084,908 to Chiang *et al.* ("*Chiang*")	**1006** **1007** **1008**

In view of the prior art, evidence, and arguments herein, the Challenged Claims are unpatentable and should be cancelled. 37 C.F.R. § 42.104(b)(1). Based on the prior art references identified below in light of the knowledge of a PHOSITA, IPR of these claims should be instituted. 37 C.F.R. § 42.104(b)(2). This review is governed by pre-AIA 35 U.S.C. §§ 102 and 103.

Section IV, *infra*, identifies where each element of the Challenged Claims is found in the prior art. 37 C.F.R. § 42.104(b)(4). The exhibit numbers of the evidence relied upon to support the challenges are provided above and the relevance of the evidence to the challenges raised is provided in Section IV. 37 C.F.R. § 42.104(b)(5). **Exhibits 1001-1021** are also attached.

C. Claim Construction

In IPR proceedings, claims are interpreted under the same standard applied by Article III courts (i.e. the *Phillips* standard) in post-grant proceedings. *See* 37 C.F.R. § 42.100(b); *see also* 83 Fed. Reg. 197 (Oct. 11, 2018); *Phillips v. AWH Corp.*, 415

F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). Under this standard, words in a claim are given their plain meaning, which is the meaning understood by a person of ordinary skill in the art after reading the entire patent. *Phillips*, 415 F.3d 1303, 1312-13.

At this time, Petitioner proposes that the claims be construed pursuant to their plain and ordinary meaning in light of the specification of the ’365 Patent. Petitioner reserves the right to rebut any issues related to claim construction that may be raised by Patent Owner in an attempt to circumvent the prior art cited herein.

IV. THE CHALLENGED CLAIMS ARE UNPATENTABLE

The below grounds demonstrate how the cited prior art teaches and/or renders obvious each and every limitation of the Challenged Claims. To avoid unnecessary repetition, the claims are grouped based on their similar limitations, as many of the limitations are nearly identical across corresponding Challenged Claims.

A. Ground 1: Claims 1-4, 6-10, 12-18, and 20 are obvious over *Kalker* in view of *Novotny*

U.S. Patent 5,999,655 to Kalker et al. (“*Kalker*”) issued on December 7, 1999, and, therefore, is prior art to the ’365 Patent at least under 35 U.S.C. § 102(b). *See Kalker* (Ex. 1006). *Kalker* is directed to an advanced video compression coding system employing variable block size transforms to improve video compression efficiency. *Id*. at Abstract. An encoding-side transmitting station receives an input signal, X_{in}, such as a video signal. *Id*. at 2:41-46. The signal is transformed and quantized. *Id*. at 2:48-54. The input signal is also applied to a segmentation circuit, which considers rate-distortion costs for determining the best block size for different partitions of the input signal. *Id.* at 2:60-63. The plurality of block-sizes that make up a picture, referred to as a segmentation map, are applied to a transform circuit that identifies a current block size for each block. *Id.*at 2:66-3:7. The block sizes are encoded for transmission to a receiving station or storage. *Id*. at 2:66-2:7. Because block sizes smaller than the largest size will always appear at least in pairs, runlength-encoding is used to indicate adjacent blocks of the same size. *Id*. at 1:55-58.

Kalker teaches a grid-based encoding and decoding system. Specifically, for each picture or frame, the encoder scans a picture and generates a block segmentation map containing macroblocks and sub-blocks of varying sizes and assigns block-size codes, denoted by the letter "S," that represent each block size (e.g., S=1 to S=3) contained within the picture. *Id*. at Abstract; *see also id*. at 1:26-53. Each block-size code is assigned a value by the encoder, with the smallest block represented by S=1, and the largest block represented by S=3. *See id*. The value corresponding to the block size codes may vary from one coding unit (picture or frame) to another, as only block sizes that actually exist within a picture or frame are assigned a block-size code by the encoder. *See id*. at 1:50-52; *see also id*. at 3:54-58. Because the block-size codes may vary from picture to picture, the encoder must not only communicate that a given partition in a grid is assigned a particular block size code (e.g., 1 or 3), but it must also communicate what value the block-size code represents for a given picture (e.g., 4×4, 8×8, 16×16). *See id.*; *see also id.* at 5:15-21 (describing alternatives for block-size codes, such as S=3 corresponding to an 8×8 block); *see also Freedman Decl.* (Ex. 1009) at ¶49. Thus, the actual block size value to which a block size code corresponds may vary from picture to picture. *Id.*

In one embodiment in *Kalker*, the encoder identifies the *largest* block size in the frame, establishes that as the highest S value, and builds the map (and the decoder reconstructs the map) on the basis of a grid corresponding to the *largest* block size, such as a 16×16 grid corresponding to code S=3. *Id.* at 5:29-65, Fig. 9. In the example of this embodiment, the largest block size, represented by code S=3, is a 16×16 starting block size for the coding/decoding grid. *Id.* A grid block with starting size of 16×16 may be subdivided into smaller blocks down to a minimum block size, represented by the block-size code S=1 (e.g., 4×4). Because the grid size in this embodiment corresponds to the largest assigned "S" code, which in turn corresponds to the largest block size in the frame, this S value (e.g., S=3=16×16) has two functions: (1) on the frame-wide basis, it indicates the grid block starting size (e.g., 16×16), which corresponds to the maximum S value, which in turn corresponds to the largest block size, and (2) on a specific block basis, where a given block in the frame is assigned a block-size code of 3, it specifies to the decoder that the particular block being examined is equal to the maximum block size and contains no partitions. *See id.* at 1:50-52; *see also id.* at 3:54-58, 5:31-57; *see also Freedman Decl.* (Ex. 1009) at ¶50.

Figure 9 illustrates the largest-block-size-based grid method of the second embodiment:

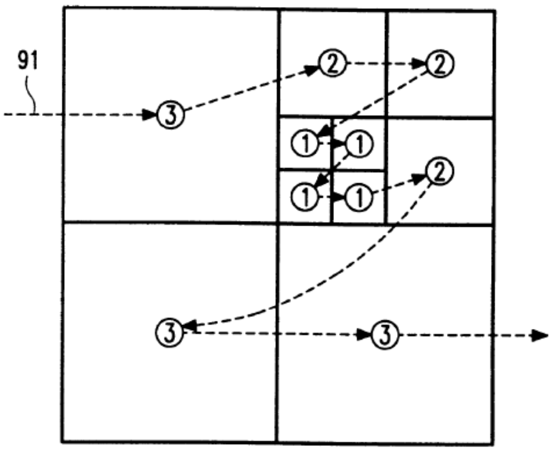


FIG. 9

Kalker (Ex. 1006) at Fig. 9 (representing an alternative grid based on a largest-block scanning pattern). The receiving station includes, among other components, a segmentation map decoder circuit, which stores the relevant grid corresponding to the largest block size in memory. *See id.* at 3:8-25; *see also id.* at 4:36-50, Figs. 2, 5, and 9. This segmentation map decoder circuit reconstructs the image based on the relevant grid size (represented by the value assigned to the maximum block size code), and the extracted block-size code elements identifying the individual blocks or sub-blocks within the frame:

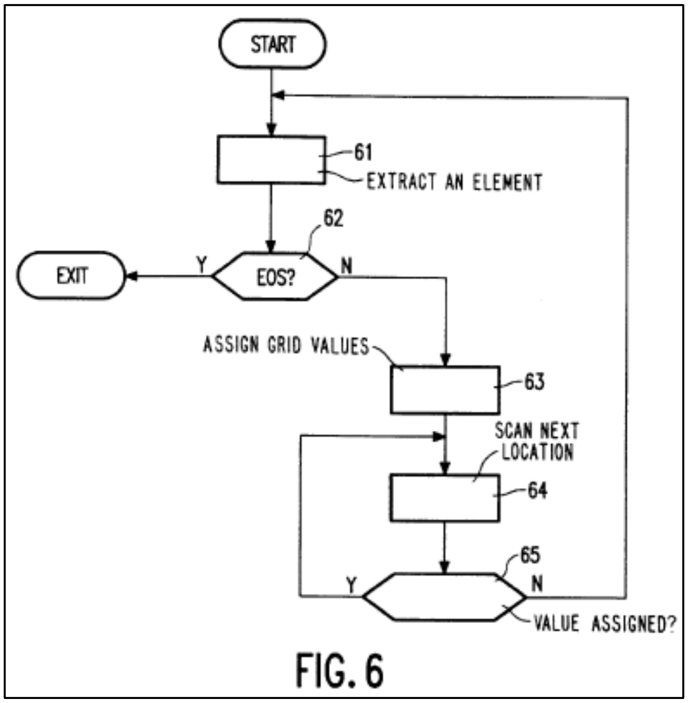


FIG. 6

See id. at Fig. 6; *see also id.* at 5:31-57 (describing scanning process for the cited embodiment).[2] In this embodiment, the encoder (and decoder) recognizes while

[2] *Kalker* describes the components and operation of the decoding receiving station primarily in its discussion of a first embodiment that performs scanning using a grid based on the smallest block size. A PHOSITA would have recognized that the same components discussed with respect to the first embodiment are applicable to the embodiment mapped below, as the decoder simply reverses the encoding process in both embodiments. *See Freedman Decl.* (Ex. 1009) at ¶52. This is at least because in order to reconstruct an encoded image frame, a decoder generally would be programmed to follow instructions provided by an encoder to inverse the encoding

scanning the grid that when it encounters one block assigned a block-size code S=1, that block and neighboring blocks in a given sub-block are equal to the minimum block size and require no further partitioning. *See Kalker* (Ex. 1006) at 5:45-52; *see also Freedman Decl.* (Ex. 1009) at ¶57.

Kalker is both within the field of endeavor and reasonably pertinent to the ’365 Patent. *See Freedman Decl.* (Ex. 1009) at ¶53. The ’365 Patent relates to the field of block-based video encoding/decoding methods and systems, and *Kalker’s* disclosure is directed toward this field also. *See Kalker* (Ex. 1006) at Abstract, 1:7-15; *compare with* ’365 Patent (Ex. 1001) at 1:22-23. Further, both *Kalker* and the ’365 Patent are concerned with at least one problem in this field—improving compression efficiency in video encoding. *Kalker* (Ex. 1006) at 1:43-44, 3:55-58, 5:61-65; *see also* ’365 Patent (Ex. 1001) at 7:10-18. Therefore, *Kalker* is analogous art to the ’365 Patent.

process. *See id*. Indeed, as much was argued during prosecution of one of the parent patents of the ’365 Patent to overcome a ¶112 rejection. *See Pat. 9,788,015 File History* (Ex. 1005) at 3471 (“[O]ne of ordinary skill in the art would certainly appreciate that any data encoded by a video encoder must necessarily be decoded by a video decoder.”).

U.S. Pub. 2005/0123282 to Novotny *et al.* ("*Novotny*") was published on June 9, 2005, and, therefore, is prior art at least under 35 U.S.C. 102(b). *See Novotny* (Ex. 1007). *Novotny* eventually issued on April 22, 2008 as U.S. Pat. 7,362,804. *Novotny* relates to an apparatus that employs standard H.264 video encoding methods, but further displays graphical symbols to depict syntax elements of an encoded video bitstream to an end user on top of decoded video data. *See id.* at Abstract; *see also id.* at [0037]. Like the '365 Patent and *Kalker*, *Novotny* describes a video coding and decoding system in which picture elements are partitioned into different block sizes:

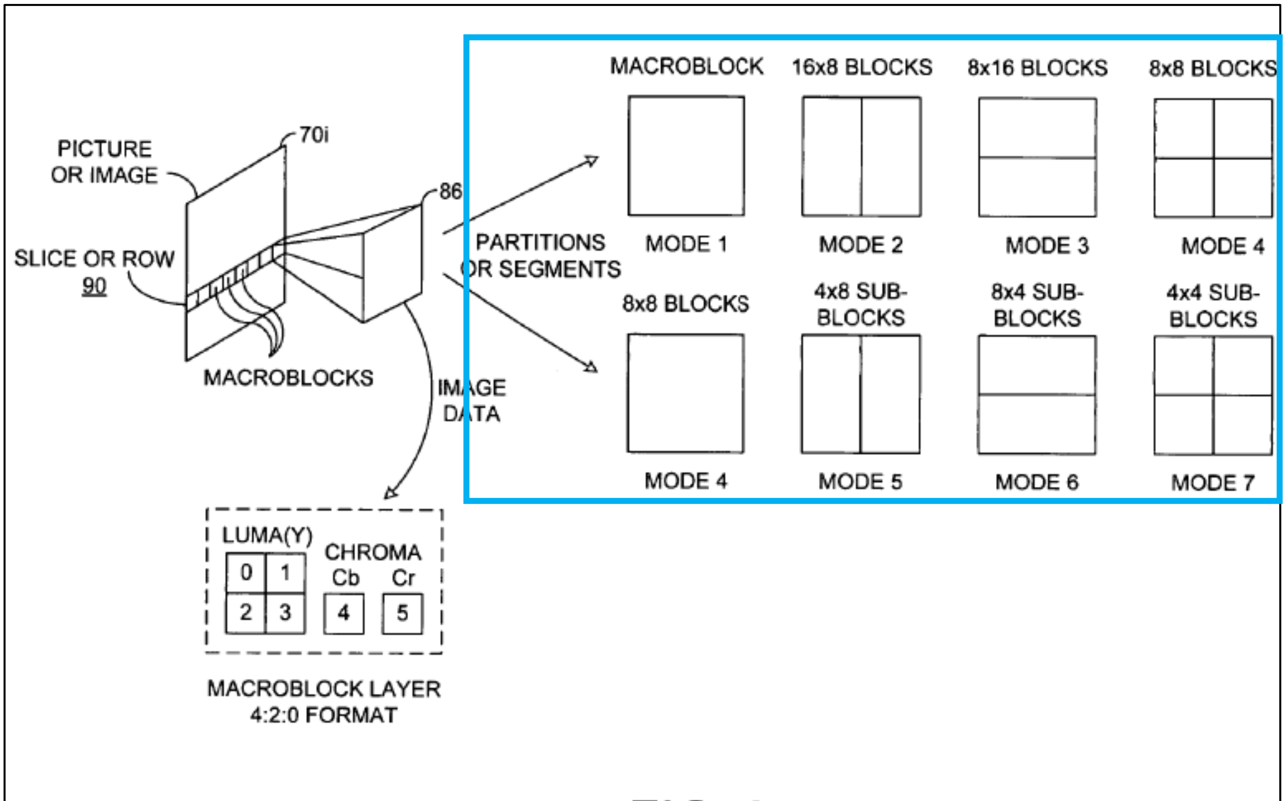


FIG. 3

Id. at Fig. 3 (annotated to highlight various block sizes); *see also id.* at [0031] ("[A] picture … may be divided (e.g., segmented, partitioned, etc.) into a number of macroblocks 86. The macroblocks generally comprise an array of pixels having vertical and horizontal dimensions of equal size (e.g., 32×32, 16×16, etc.).”). Consistent with MPEG/H.264 systems, a bit stream of video images are compressed and output from an encoder via a transmission medium, such as a “broadcast, cable, satellite, network, DVD, hard drive, or any other medium implemented to carry, transfer, and/or store a compressed bit stream.” *Id*. at [0035]. The encoded bitstream also includes syntax elements with the video data. *See id*. at [0037]-[0039]; *see also id*. at Abstract, [0029]. *Novotny* teaches a system in which syntax information may be graphically displayed over corresponding decoded video content. *Id*. at [0037]; *see also id*. at Abstract, [0042], [0045], [0003].

Novotny describes different GUIs for displaying the various syntax elements that are transmitted with encoded video and decoded by the decoder for display to an end user. For example, as discussed in more detail below, syntax elements such as macroblock and sub-block types and prediction directions (inter-, intra-) may be encoded with the bitstream displayed to an end-user. *See id.* at [0050]-[0065]. *Novotny* is cited below both to complement the teachings of *Kalker* with respect to limitations taught by *Kalker* and for its specific teachings regarding the architecture of an encoding/coding system, the types of syntax elements known and used in the art, and for its teaching of using starting macroblock sizes of greater than 16×16 pixels.

Novotny is both within the field of endeavor and reasonably pertinent to the ’365 Patent. *See Freedman Decl.* (Ex. 1009) at ¶62. Like the ’365 Patent, *Novotny* is directed toward the field of block-based video encoding/decoding methods and systems. *See Novotny* (Ex. 1007) at [0022]; *see also id*. at [0001], [0023], [0030]. Further, *Novotny* is reasonably pertinent to at least two problems concerning the ’365 Patent. For example, *Novotny* teaches the use of large macroblocks (i.e., larger than 16×16 pixels), and explains that macroblocks of sizes other than 16×16 pixels may be used to meet the “design criteria of a particular application.” *See id*. at [0037]; *see also id*. at [0031] (providing 32×32 as an example starting size for macroblocks). Additionally, *Novotny* is reasonably pertinent to the problem of efficiently communicating encoded syntax information to the decoder. Like the ’365 Patent, *Novotny* solves this problem by using various encoded syntax elements to inform a decoder regarding how to decode an encoded video, including those syntax elements set forth in the claims of the ’365 Patent. *See id*. at [0071] (setting a pair of variables, such as min_mb_size and max_mb_size to the minimum and maximum macroblock size within a picture); *see also id*. at [0050]-[0066] (describing syntax elements including the macroblock type, sub-macroblock type, and prediction directions). Therefore, *Novotny* is analogous art to the ’365 Patent.

i. Independent Claims 1, 7, and 15[3]

1[P]. A method of decoding video data, the method comprising:

7[P]. A device for decoding video data, the device comprising: [a] a memory configured to store decoded video blocks of the video data; and [b] a processor, in communication with the memory, configured to:

15[P]. A non-transitory computer-readable storage medium having stored thereon instructions that, when executed, cause a processor to:

To the extent the preambles are limiting, *Kalker* teaches, or at least renders obvious the preambles. *See Kalker* at 4:36-42; *see also id*. at 5:31-57, 3:8-18, Figs. 1, 5, Claim 8. *Kalker* teaches a video-receiving station (i.e., a *non-transitory computer-readable storage medium*) that includes a video decoder (i.e., *a device for decoding video data* that performs a *method of decoding video data*):

[3] The mapping of the prior art is grouped by claims containing identical substantive language. *Italicized* font in quoted claim language indicates where claim language differs, generally based on the different claim types.

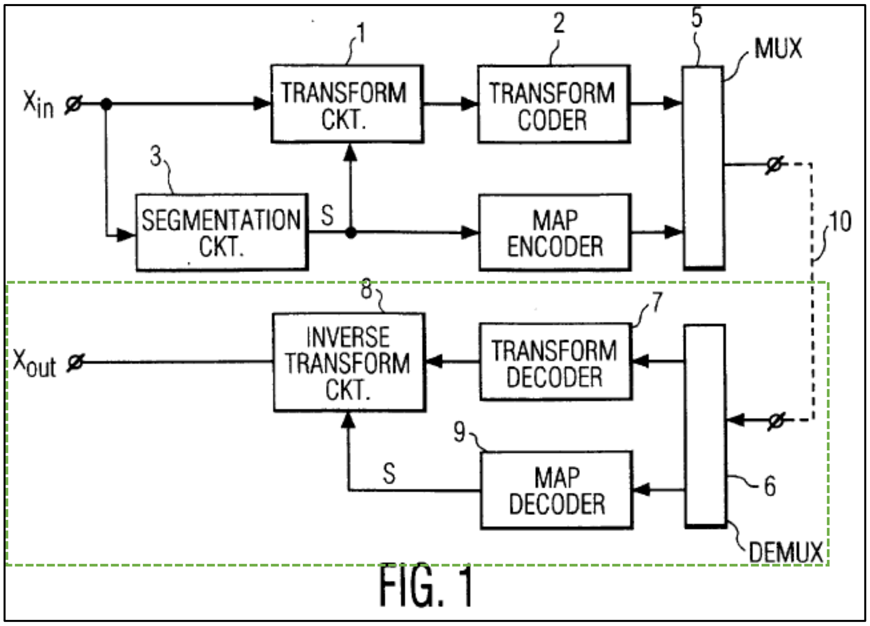


FIG. 1

Kalker (Ex. 1006) at Fig. 1 (annotated); *see also id*. at Fig. 5; 3:8-18 (describing components of the receiving station), 4:36-47, 5:36-42. The receiving station includes a map decoder circuit 9 (i.e., *a processor* containing *instructions* on a *computer-readable storage medium*) that processes video in accordance with the scanning process of the encoder. *See id*. at 3:8-18; *see also id*. at 4:36-50; *see also id*. at 3:4-7, Fig. 1 (showing storage medium 10 at both the encoder and decoder sides). The map decoder circuit reconstructs encoded video frames using a segmentation map reconstruction circuit, which includes memory for storing video data as it is processed (i.e., *memory configured to store decoded video blocks* that is *in communication with the processor*). *See id*. Further, as a PHOSITA would have known, a decoding device, such as the receiving station in *Kalker* includes some memory, even if only to temporarily store blocks and frames that have been decoded before being displayed. *See Freedman Decl.* (Ex. 1009) at ¶51. *Kalker* teaches that the receiving station includes a decoder and segmentation map reconstruction circuit that stores and scans segmented blocks of video data to perform a "reconstruction process" of the compressed video data based on elements describing the scanned blocks. *See id.* at 4:43-67; *see also id.* at 5:31-57 (describing processing blocks on the basis of the largest block size); *see also Freedman Decl.* (Ex. 1009) at ¶51.

Further, like the '365 Patent, *Kalker* teaches both encoding and decoding video data. Its most detailed discussion is provided from the perspective of the encoding process, while much of the decoding process is generally described with respect to the information and data received from the encoder. *See id.* at 3:8-18 (receiving station includes demultiplexer and decoding components to perform the inverse operations of the coder), 4:36-42 (decoder circuits need not be described in detail given extensive description of coding counterparts); *compare with* '365 Patent (Ex. 1001) at 18:6-8 (decoder performs decoding pass "generally reciprocal" to the encoding pass); *see also id.* at 2:43-3:5, 3:36-67 (decoder decodes based on block-type syntax information), Fig. 17. However, as a PHOSITA would have recognized, *Kalker*'s teachings of its encoding steps would be reversed by a corresponding decoder device. *See Freedman Decl.* (Ex. 1009) at ¶52; *see also Kalker* (Ex. 1006) at 4:48-50 (scanning order performed by decoder corresponds to scanning order in the encoder).

Therefore, to the extent limiting, these preambles are taught, or at least rendered obvious by, *Kalker*.

***1[a] / 7[b][i] / 15[a]. decoding / decode / decode* a first syntax element associated with a sequence of pictures of the video data, the first syntax element representing a minimum size of blocks of the sequence of pictures;**

This limitation requires that the decoder decode a first syntax element. As noted above, a PHOSITA would understand that a syntax element in the general sense is used to describe information that is communicated to and used by the decoder to understand the characteristics or processing of encoded data so that the decoder can determine how to decode the encoded data. '365 Patent (Ex. 1001) at 11:13-18. *See also Freedman Decl.* (Ex. 1009) at ¶¶54-55. In this claim limitation, the syntax element must represent the minimum size of blocks in the sequence of pictures. This limitation is taught by, or at least obvious over, *Kalker*. *Kalker* teaches an encoding-side transmitting station that *assigns* particular size values to multiple block-size codes (i.e., *syntax elements*) for an entire coded unit (e.g., a picture or frame *in Kalker*) and uses these codes to communicate the different block sizes contained in the coded unit; and these block-size codes for the coded unit are then used by the decoding-side receiving station to decode the data and reconstruct the image. *See id.* at 3:8-18; *see also id.* at claim 1 ("[T]he step of encoding said segmentation map comprises **assigning a block-size code to each block size**.”)[4]; 1:20-24 (“Each picture is segmented into picture blocks, the size of which is adapted to local picture contents.”); 3:31-35, 4:55-56 (“If an **element** is not the EOS code, it represents a block size S”), claim 8.

For example, the encoder sets a block-size code of “3” to represent the largest blocks in the coded unit, such as 16×16 blocks, a block-size code of “2” can represent intermediary blocks, such as 8×8 blocks, and a block-size code of “1” represents the smallest, or minimum block size in the coded unit, e.g., 4×4 blocks. *See, e.g.*, *id*. at 3:25-34, 5:36-57; *see also id*. at 4:43-67 (in describing the reconstruction process at the decoding unit, referring to block-sized codes as “elements” extracted by the decoding unit). Note that the actual size of the minimum size block in *Kalker* is variable picture by picture. *See id.* at 1:50-52 (“[O]nly block-size codes are transmitted for blocks which are not divided into smaller blocks,” i.e., that exist in the picture); *see also id*. at 5:15-21 (providing alternative example of block sizes for a picture); 1:20-21 (“Each picture is segmented into picture blocks, the size of which is adapted to local picture contents.”). Therefore, the encoder must scan the entire coded unit (e.g., a picture or frame) and assign the value of S=1 for that coded unit, depending on the actual smallest block sizes in the given coded unit, with a 4×8

[4] Unless otherwise indicated, all emphasis has been added by Petitioner.

block provided as an alternative example in *Kalker*. *See id.* at 5:10-21, Fig. 8; *see also id.* at 1:50-52, 1:20-21. Because the block-size codes vary from one coded unit to the next, the code S=1 does not represent a constant size, but rather must be set in a given coded unit to represent the actual smallest block size in that coded unit. *See Freedman Decl.* (Ex. 1009) at ¶¶49, 55. Otherwise, the decoder would not be able to understand the block sizes that correspond to the S variable as it relates to each coded unit (e.g., a picture). Put another way, a PHOSITA would have appreciated that *Kalker*'s encoder communicates to the decoder that S=1=4×4 and S=3=16×16 for each coded unit for which this is true, as these values for S are not a given for every coded unit. *Id.* at ¶49. Therefore, a PHOSITA would have reasonably understood that for each coded unit in *Kalker*, the encoder instructs the decoder as to the particular smallest block value that is then characterized by the code S=1 for that particular coded unit. *See id.* at ¶¶49, 54-55, 57. Therefore, the block-size code "S=1" is a *first syntax element representing a minimum size of blocks* in a coded unit.

In one example in *Kalker*, the minimum size of blocks in the coded unit being scanned is a 4×4 block, so the encoder sets the "S=1" code to represent a block size of 4×4 for the entire coded unit, or picture. *See id.* at 3:27-32; *see also id.* at 4:43-47, 5:49-52. Then, while scanning a segmentation map, the encoder may assign a block-size code of S=1 to any 4×4 block in the coded unit, or picture, and communicate those block sizes via the S=1 code to the decoder.[5] In this example, upon receiving the bitstream for a given coded unit, the decoder will understand that the minimum block size in the coded unit is 4×4 (i.e., because S=1 corresponds to a 4×4 block size). *Kalker*'s decoder then uses the knowledge that S=1 represents the smallest block size to enhance efficiency. *See Freedman Decl.* (Ex. 1009) at ¶57. Specifically, when an S=1 block is encountered, it is "accordingly" known that any adjacent sub-blocks will also all have a size of "1" and will not contain any further partitions:

> First, the top left 16*16 block is analyzed. As this block is not further divided into smaller blocks, the block size code S=3 is generated. Then, the next (top right) 16*16 block is analyzed. This block is segmented into smaller blocks and **will now completely be scanned** before proceeding to the next 16*16 block. More particularly, the top left 8*8 block is now analyzed. As it is not further divided, the block size code S=2 is generated. Similarly, the block size code S=2 is generated for the next (top right) 8*8 block. **Then the bottom left 8*8 block is analyzed. It is segmented into smaller blocks and will thus be scanned before proceeding to the next 8*8 block. Accordingly, an**

[5] To be clear, it is not the fact that an S-code is attached to a particular block to indicate the partitioning size for that block that satisfies this limitation; rather, it is the fact that the value corresponding to the block-size code (S=1) is set based on the actual minimum size of blocks in the coded unit, e.g., a picture.

> **S=1 block size code is generated for the top left 4*4 block, the top right 4*4 block, the bottom left 4*4 block and the bottom right 4*4 block, successively**. The scanning then proceeds to the next (bottom right) 8*8 block for which, in this example, the block size code S=2 is produced. Now, the top right 16*16 block has completely been processed and the scanning proceeds to the left bottom 16*16 block (S=3) and the right bottom 16*16 block (S=3).

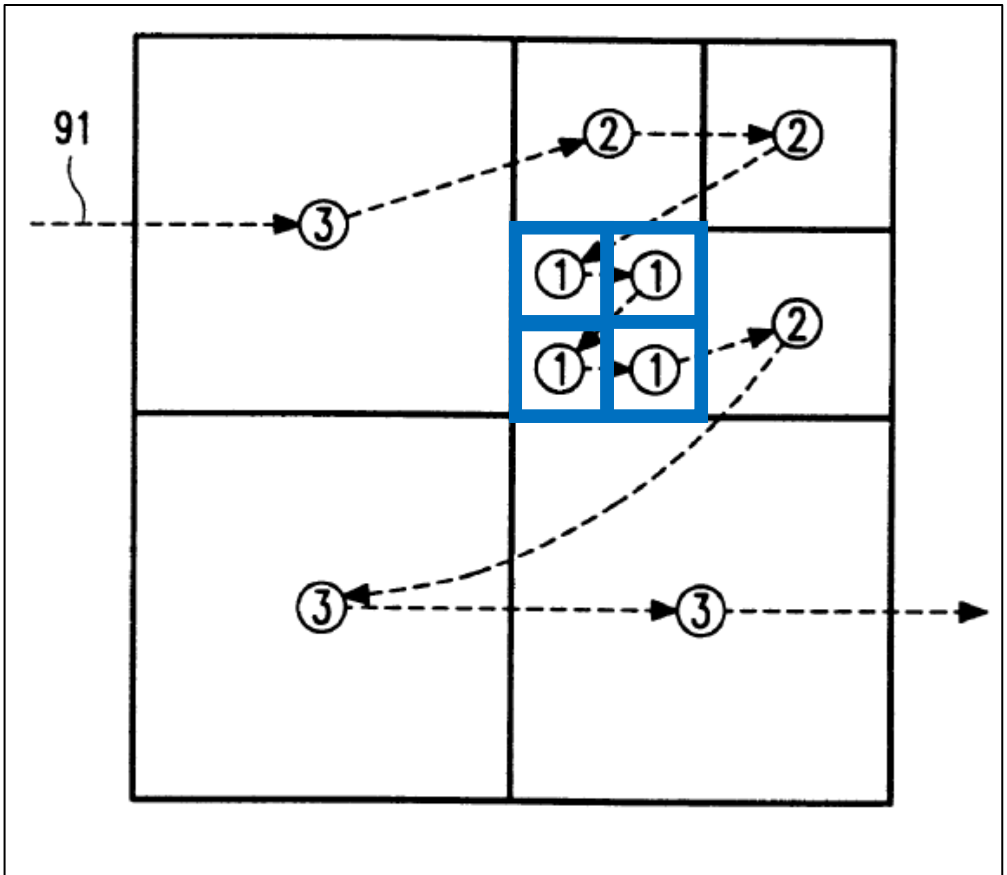


FIG. 9

Kalker (Ex. 1006) at 5:38-57, Fig. 9 (annotated to highlight the smallest blocks, represented by block-size code S=1); *see also id.* at 4:43-67, 5:1-3, Fig. 7.

Further, it would have been obvious to a PHOSITA that *Kalker*'s use of the S=1 syntax element would be applicable to any conventional coded unit, including a group of pictures, also referred to as a *sequence of pictures* or frames. *See Freedman Decl.* (Ex. 1009) at ¶58. Although *Kalker* describes its embodiments in terms of an individual frame or picture (*see, e.g.*, claim 1, 1:20-21), the value of the code S=1 is set for an entire frame containing multiple blocks. It would have required no great exercise of creativity to set the same block-size value indicated by the code S=1 for several consecutive video frames (i.e., *a sequence of pictures*). *See Freedman Decl.* (Ex. 1009) at ¶¶46, 58. In other words, because several consecutive video frames are likely to involve similarly sized blocks, (e.g., smallest blocks of 4×4 pixels), it would have been obvious to set the code S=1 as representing a 4×4 value for those consecutive frames. A PHOSITA would have recognized that by applying the same syntax information to a sequence of pictures rather than an individual picture, the system would reduce the amount of overhead data necessary to communicate the encoded syntax elements such as, for example, the value assigned to a given block-size code. *See id.* A PHOSITA would have been motivated to do so because reducing overhead data is a desired goal in video coding generally, including as disclosed in *Kalker*. *See id.* Furthermore, reduced overhead is a predictable result of using a larger coded unit to which syntax data is applied. This is particularly true for a series of consecutive frames in the same scene, which are likely to have highly redundant data and block sizes. *Id*. Therefore, although *Kalker* provides examples of its teachings of assigning block-size codes on a picture-by-picture basis, a PHOSITA would have found it obvious and would have been motivated to apply *Kalker*'s teaching to coding units of larger sizes such as a "sequence of pictures." *See id.* A PHOSITA would have had a reasonable expectation of success in making such a modification because it would have required nothing more than a minor modification in software (i.e., where to designate the syntax element) to adjust the size of the coded unit using only one of a few available coding-unit sizes for which syntax information were already identified in existing coding standards. *See id.*

Furthermore, *Kalker* is a *video* decoding system that was already operating on a series of pictures that form a video. *See Kalker* (Ex. 1006) at Abstract ("An advanced video compression coding system which employs variable block size transforms to improve compression efficiency for transmission of *video pictures*."); *see also id.* at 1:7-15. Similarly, the '365 Patent acknowledges that the size of the coding unit is not of particular import, noting that "[a] coded unit may comprise a video frame, a slice, or a group of pictures (also referred to as a "sequence")." *'365 Patent* (Ex. 1001) at 38:45-47; *see also id.* at 13:4-9 (listing a group, or sequence, of pictures among multiple "independently decodable unit[s] defined according to applicable coding techniques"); 8:34-37, 13:30-32, 38:52-54. Indeed, in describing the use of the minimum syntax element in the specification, the ’365 Patent refers to a non-specific “coded unit,” not to a sequence of pictures. *Id.* at 39:37-41. There is no difference between an individual video frame (picture) or a sequence of pictures as a coded unit in the context of the ’365 Patent, and there is no technical difficulty associated with transitioning the system of *Kalker* from an individual frame-based system to one encoding and decoding a sequence of such frames based on the same syntax information. *See Freedman Decl.* (Ex. 1009) at ¶58.

Therefore, this limitation is taught, or at least rendered obvious, by *Kalker*.

***1[b] / 7[b][ii] / 15[b]. decoding / decode / decode* a second syntax element, separate from the first syntax element, associated with the sequence of pictures, the second syntax element representing a maximum size of the blocks of the sequence of pictures, wherein the maximum size is greater than 16×16 pixels;**

This limitation is obvious over *Kalker* in view of *Novotny*. *Kalker* teaches providing a block-size code of “S=3,” that is set to represent the actual maximum size of the blocks in the coded unit (i.e., *second syntax element…associated with the sequence of pictures*). This S=3 code is *separate from* the block size code “S=1” representing the smallest block size (i.e., *the first syntax element*). The maximum block-size code (S=3) can be set to represent, for example, 16×16 blocks in a given coded unit, which is then scanned on the basis of a 16×16 grid size:

> In a further embodiment of the scanning circuit 41 (See FIG. 2), the segmentation map is scanned **<u>on the basis of the largest block size</u>**. If

> a block comprises smaller blocks, it is scanned on the basis of the next smaller block size. This is an iterative process.
>
> FIG. 9 shows a segmentation map illustrating this embodiment. The scanning pattern is denoted 91 in this Figure. First, the top left 16*16 block is analyzed. **<u>As this block is not further divided into smaller blocks, the block size code S=3 is generated. Then, the next (top right) 16*16 block is analyzed</u>**.

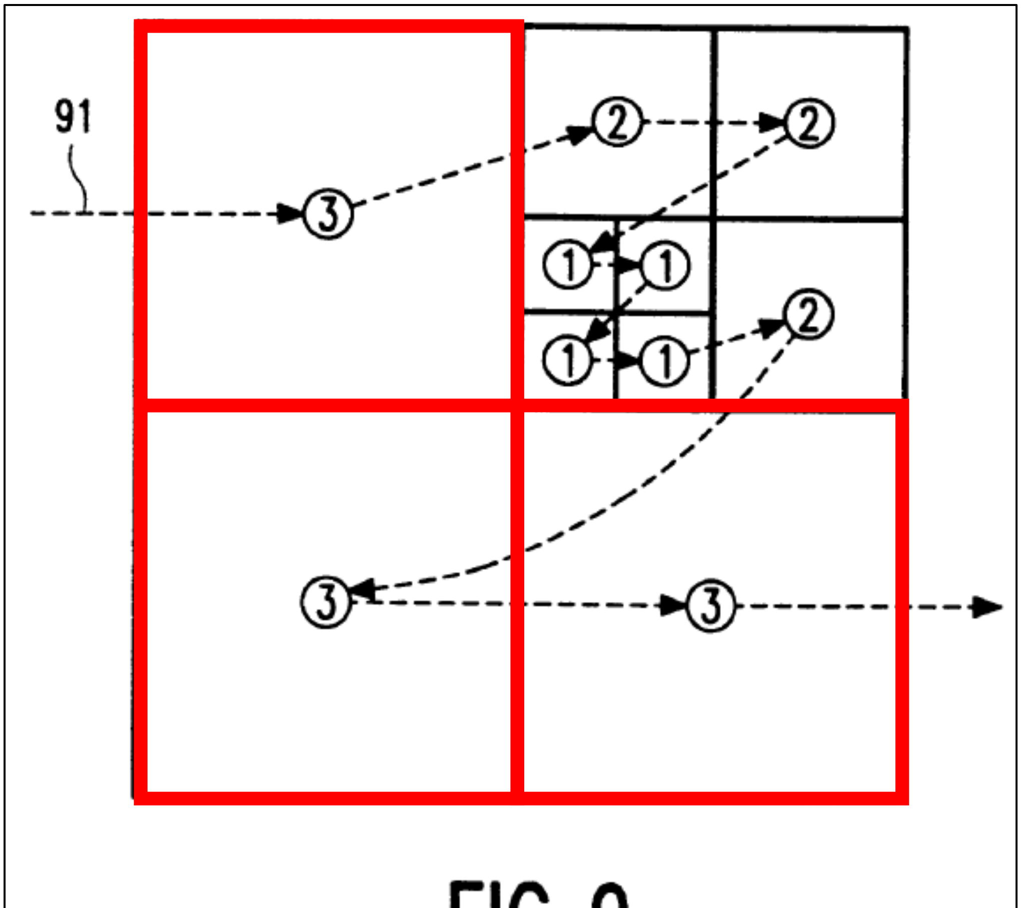


FIG. 9

See Kalker (Ex. 1006) at 5:31-41, Fig. 9 (annotated to highlight the largest blocks, represented by S=3). As with the preceding limitation, *Kalker* teaches that the largest block size represented by S=3 can vary in size from one coded unit to another, with another provided example being a maximum block size of 8×8. *See id.* at 5:15-20; *see also id*. at 1:50-54. Thus, the encoder defines the value represented by the block-size code S=3 in terms of an actual maximum size of blocks for a given coded unit. *See id*. at Abstract, 1:7-13, 1:20-21, claim 1. *Freedman Decl.* (Ex. 1009) at ¶56. Once the maximum block size is established, according to the second embodiment of Fig. 9, the scanning grid used in encoding and decoding corresponds to that maximum block size, and that value is represented by the syntax element S=3. *See id*.; *see also id*. at 4:48-50.

Kalker is able to use the knowledge that block size code S=3 represents the largest block size to enhance coding and decoding efficiency. *See Freedman Decl.* (Ex. 1009) at ¶56. Upon learning that a block is equal to the maximum block size (which corresponds to the grid size), the encoder (and therefore, the decoder) may jump immediately to the next grid location (or the largest block partition) without further scanning. Further, *Kalker* explains that block-size codes are only generated for a given block size that is *not* divided into smaller blocks at least once. *See id*. at 1:50-52 ("[O]nly block-size codes are transmitted for blocks which are not divided into smaller blocks."). Therefore, once the largest block size is determined, the grid size is set to correspond to that block size, which in turn minimizes the required scanning to the extent possible. *See Freedman Decl.* (Ex. 1009) at ¶56. The highest "S" value (S=3 in the examples provided) is thus directly representative of the largest, or maximum block size for the entire coded unit and is the basis of the grid for the entire coded unit and ultimately the decoder's image reconstruction process for that entire coded unit. *Id*.

To be clear, the encoder communicates two things to a decoder related to a maximum block size. First, it communicates the largest block size of a given picture, represented by S=3 and equal to a 16×16 block size, so that the decoder can inverse the encoding process on the basis of a grid corresponding to the largest block size and represented by this syntax element. *See id*. at ¶56. The encoder also communicates when a given block in the picture is an S=3 block, in which case the decoder knows that the given block has no partitions and can jump to the next 16×16 block in the picture, as that is the scanning grid size. *See id*.

It would have been obvious to a PHOSITA that the teachings in *Kalker* would have been applicable to systems employing block sizes *wherein the maximum size is greater than 16×16 pixels*. For example, *Novotny*, which discloses an encoding system based on MPEG/H.264, teaches that the maximum size block stored in memory and output to a decoder may be larger than a 16×16 block. *See Novotny* (Ex. 1007) at [0031] (providing an example of 32×32 pixel starting block); *see also id*. at [0030], [0037] (other size macroblocks may be implemented to meet the design criteria of an application). Although 16×16 is an exemplary maximum in *Kalker*, a PHOSITA would have recognized that encoding/decoding systems were not limited to 16×16 macroblock sizes. It would have been obvious to incorporate *Novotny*'s teachings of using block sizes larger than 16×16 pixels into the similar system of *Kalker*. Like *Novotny*, *Kalker* contemplates the use of MPEG-like coding methods. *See, e.g.*, *Kalker* (Ex. 1006) at Abstract, 2:4:35-42, Figs. 2-5; *see also Novotny* (Ex. 1007) at [0002]-[0003], [0025]. Further, a PHOSITA would have recognized (a) that the use of 16×16 in *Kalker* is merely exemplary and (b) that the use of larger block sizes would be desirable to efficiently encode sequences of images where little variance occurs across pictures. *See Freedman Decl.* (Ex. 1009) at ¶¶40-42, 63-64; *see also Kalker* at 3:30-34 (providing the 16*16, 8*8, and 4*4 block sizes in "the present *example*"); 4:43-47 (describing an embodiment whereby the decoder performs reconstruction starting from the "smallest" size, where 4*4 blocks are used "in the present *example*"); 1:52-54 (noting that a "few" different block sizes are used).

A PHOSITA would have been motivated to use larger blocks, as taught in *Novotny*, in *Kalker*'s system because the use of larger blocks as a starting block size would have provided for higher compression efficiency, while small blocks require many bits. *See Freedman Decl.* (Ex. 1009) at ¶¶40-42, 64; *see also Kalker* (Ex. 1006) at 1:31-33 ("Small blocks have a plurality of higher level nodes and thus require many bits."). By using a larger initial block size, a programmer could allow a video codec system to use very little data for areas of simple images with large redundancy, such as background elements of a scene, while not compromising the encoder's ability to partition the block further for scenes with greater variance. *See id.* at ¶64. Further, a PHOSITA would have had a high expectation of success in implementing these concepts into the system of *Kalker*, as it would have involved (1) increasing the minimum block size if only three block sizes were to be used, or (2) increasing the number of available block sizes (or "levels" or "modes" of partitioning). *See id.* A PHOSITA would have recognized that such modifications would still accomplish *Kalker*'s desired bit savings because a system designer would only use a larger block size if such resulted in less encoded data being sent due to the use of the larger block size. *See id.* Put another way, the availability of larger block sizes beyond 16×16 in a coded unit would have enhanced the flexibility of *Kalker*'s system to maximize any available efficiency gains that may be had where a particular sequence of images has a high degree of redundancy in a predictable way. *See id.* A PHOSITA would have been motivated to gain this flexibility by providing for larger available block sizes than 16×16 and would have had a reasonable expectation of success in doing so in *Kalker*. *Id.*

As discussed above and for limitation 1[a], *Kalker* teaches the value for each block-size code may vary from one coded unit to the next. *See id.* at 1:50-52, 1:20-21, 5:15-21. Because of this variability, a PHOSITA would have reasonably understood that the encoder would communicate information to the decoder providing the value of each block-size code (including the maximum block-size code, S=3 in the example provided) as such varied for coded unit. *See Freedman Decl.* (Ex. 1009) at ¶¶49, 55-57. Although *Kalker* employs a frame or picture as the coded unit size for its teachings, for the same reasons discussed in limitation 1[a], *Kalker's* teachings would apply to any conventional coded unit size, including a *sequence of pictures* or frames, as in claim 1[b]. *See id.* at ¶58. Thus, *Kalker* renders obvious the concept of decoding a second syntax element, separate from the first syntax element, associated with the sequence of pictures, where the second syntax element represents the maximum block size of the sequence of pictures.

Therefore, this limitation is obvious over *Kalker* in view of *Novotny*.

***1[c] / 7[b][iii] / 15[c]. determining / determine / determine* that a current block of a plurality of blocks of the sequence of pictures has a starting size equal to the maximum size using the second syntax element;**

Kalker teaches this limitation, or at least renders it obvious. In the second embodiment, *Kalker* teaches that the decoder decodes based on a grid where the block being scanned (i.e., the *current block of a plurality of blocks*) corresponds in size to the maximum block size of any block in the coding unit represented by the *second syntax element*, S=3. *See id. at* 5:31-35 ("In a further embodiment of the Scanning circuit 41 (See FIG. 2), <u>the segmentation map is scanned on the basis of the largest block size</u>. If a block comprises smaller blocks, it is scanned on the basis of the next smaller block size. This is an iterative process."); *see also id.* at 5:36-57, Fig. 9, claims 8, 14. Thus, the partitioning process in *Kalker* scans a *current block* in the grid with a *starting size equal to the maximum size* as indicated by (i.e., *determined using*) the *second syntax element*, and then partitions down through an iterative partitioning process potentially to the smallest block size. Because the initial block of a grid may be sub-partitioned, the size of the grid is a *starting size* and the sub-blocks that make up the grid block may be indicated by further scanning. As discussed previously, while this embodiment is describing the development of the segmentation map at the encoder, a PHOSITA would have recognized that the decoder is performing the inverse of these steps. *See Freedman Decl.* (Ex. 1009) at ¶¶43, 52; *see also id.* at 4:43-50 (explaining that the segmentation map reconstruction circuit in the decoder performs a scanning order that "corresponds to the scanning order in the encoder."). When the decoder encounters a grid block that is also represented by the highest S-value (S=3), the decoder recognizes that the given grid block is a maximum size block, and the decoder proceeds to the next grid block, again starting at the maximum block size:

> FIG. 9 shows a segmentation map illustrating this embodiment. The scanning pattern is denoted 91 in this Figure. First, the top left 16*16 block is analyzed. **<u>As this block is not further divided into smaller blocks, the block size code S=3 is generated. Then, the next (top right) 16*16 block is analyzed</u>**.

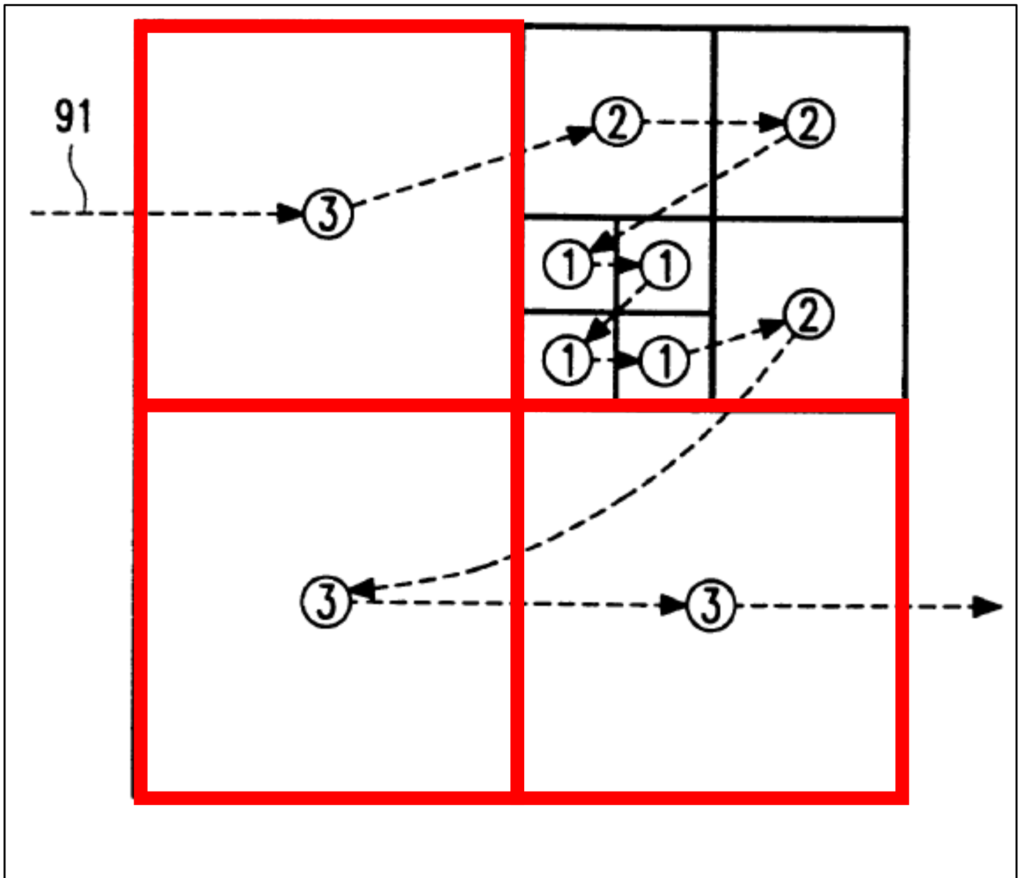


FIG. 9

Id. at 5:36-41, Fig. 9 (annotated). Because the encoder scans (and, therefore, the decoder reconstructs) the grid block-by-block on the basis of the largest block size, corresponding to the second syntax element S=3, *Kalker* teaches, or at least renders obvious, *determining that a current block has a starting size equal to the maximum size using the second syntax element.*

Further, as discussed regarding Claim 1[b] (Claims 7[b][ii] and 15[b]), *supra*, it would have been obvious to a PHOSITA that *Kalker*'s use of 16×16 as the maximum block size was merely exemplary. A PHOSITA would have recognized that it would have been possible and desirable to employ larger block sizes than 16×16 in *Kalker*'s system, as taught for example in *Novotny*, to enable using a fewer number of large blocks for highly redundant images, thereby increasing compression efficiency. *See Freedman Decl.* (Ex. 1009) at ¶64.

Further, as discussed regarding limitation 1[a], it would have been obvious to a PHOSITA that *Kalker*'s use of its syntax elements would be applicable to any coding unit size, such as a video frame or a *sequence of pictures* or frames. *See Freedman Decl.* (Ex. 1009) at ¶58.

Therefore, this limitation is obvious over *Kalker* in view of *Novotny*.

1[d] / 7[b][iv] / 15[d]. partitioning / partition / partition **the current block to obtain a plurality of sub-blocks for the current block, wherein partitioning comprises determining that a sub-block of the sub-blocks of the current block does not include further separately encoded sub-partitions when the size of the sub-block is equal to the minimum size indicated by the first syntax element;**

The '365 Patent describes that, in a given coded unit, if the decoder sees that a block is represented by a syntax element indicating the smallest block size, it may determine that the block does not have further separately encoded sub-partitions. *'365 Patent* at 39:37-41. *Kalker* teaches, or at least renders obvious, this limitation. For example, *Kalker* discloses *partitioning a current block* of the grid with multiple levels of partitions (i.e., a *plurality of sub-blocks for the current block* and *a sub-block of sub-blocks of the current block*), including sub-blocks of the minimum size corresponding to the code S=1 (i.e., the *minimum size indicated by the first syntax* *element*), when the system encodes and decodes based on a grid corresponding to the largest block size (i.e., the *current block*):

> Then, the next (top right) 16*16 block is analyzed. This block is segmented into smaller blocks and will now completely be scanned before proceeding to the next 16*16 block. More particularly, the top left 8*8 block is now analyzed. As it is not further divided, the block size code S=2 is generated. Similarly, the block size code S=2 is generated for the next (top right) 8*8 block. Then **the bottom left 8*8 block is analyzed. It is segmented into smaller blocks and will thus be scanned before proceeding to the next 8*8 block. Accordingly, an S=1 block size code is generated for the top left 4*4 block, the top right 4*4 block, the bottom left 4*4 block and the bottom right 4*4 block, successively.**

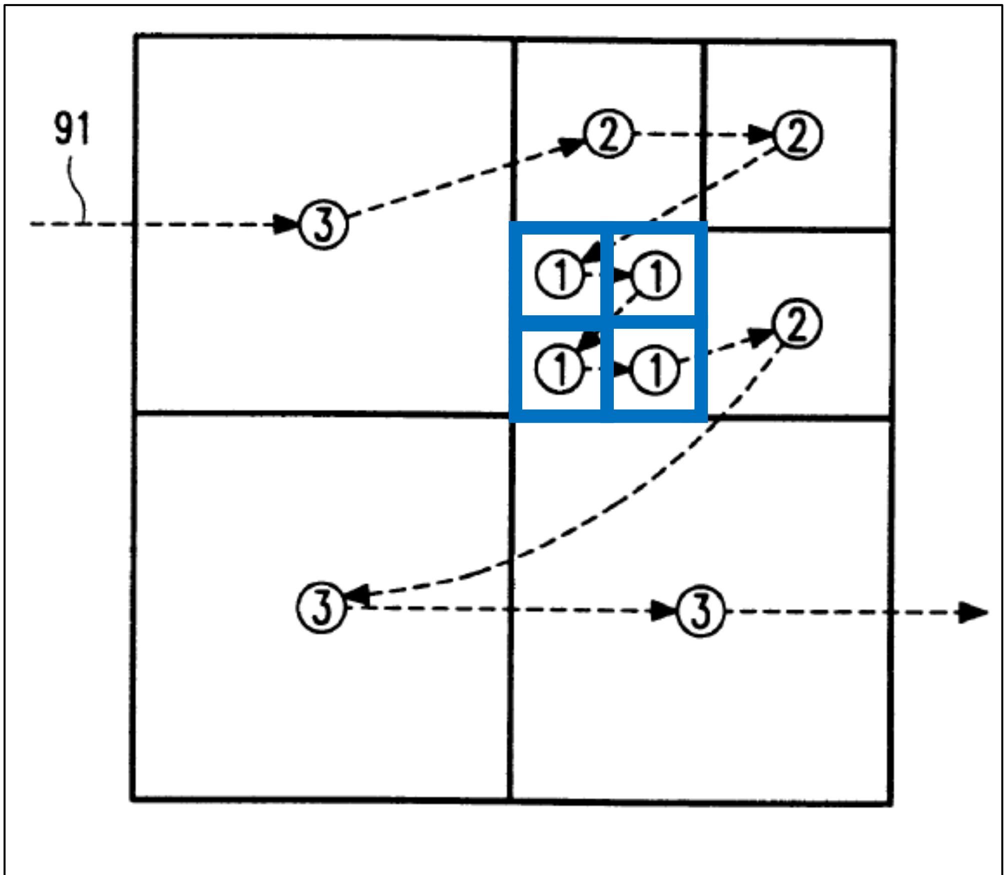


FIG. 9

Id. at 5:40-52, Fig. 9 (annotated). As is shown above, when, during the process of *partitioning*, an encoder or decoder in *Kalker* encounters a sub-block in a partitioned layer below the grid size of the current block in the grid (i.e., *a sub-block of the sub-blocks of the current block*) that is of the size assigned a block-size code S=1 (i.e., *when the size of the sub-block is equal to the minimum size indicated by the first syntax element*), it recognizes (i.e., *determines*) that this represents the smallest possible block size (i.e., *does not include further separately encoded sub-partitions*), and does not need to scan further. *See id*. at 5:47-52 (noting that when an S=2 block is partitioned, the sub-blocks are each assigned S=1 without the need for performing further scanning of the S=1 block for sub-partitions).

The ’365 Patent provides a similar disclosure. When the size of a sub-block is equal to the minimum size, it is recognized that the block does not have separately encoded sub-partitions. *’365 Patent* (Ex. 1001) at 39:37-41; *see also id*. at 39:5-12. Similarly, in *Kalker*, the decoder partitions the current block (i.e., a block of the scanning grid that has a starting size equal to the maximum size, represented by S=3 and equal to 16×16 pixels), to obtain a plurality of sub-blocks (8×8 and 4×4 blocks in the exemplary embodiment) for the current block, wherein partitioning comprises determining that a sub-block (4×4 block) of the sub-blocks (8×8 blocks) of the current block (16×16 block) does not include further separately encoded sub-partitions when the size of the sub-block is equal to the minimum size (4×4 pixels) indicated by the first syntax element (S=1). *See Kalker* (Ex. 1006) at 5:31-57, Fig. 9.

Therefore, *Kalker* teaches, or at least renders obvious, this limitation.

***1[e] / 7[b][v] / 15[e]. decoding / decode / decode* a third syntax element, separate from the first syntax element and the second syntax element, the third syntax element representing an encoding mode used to encode the sub-block, wherein the encoding mode comprises one of an intra-prediction mode and an inter-prediction mode;**

Kalker in view of *Novotny* teaches, or at least renders obvious, this limitation. *Kalker* describes encoding a video signal that may be a video picture or a motion-compensated prediction thereof and using "MPEG2-like coding" based on DCT. *See id*. at 2:41-59. As was well-known in the art, when coding video signals, such as in MPEG2, the encoding mode was used to indicate to a decoder what type of mode, inter- or intra-prediction, would be used for decoding a given block. *See Freedman Decl.* (Ex. 1009) at ¶45. *Kalker* expressly discloses that the decoder reverses the process done by the encoder. *See Kalker* (Ex. 1006) at 4:48-50. And a PHOSITA would have found it obvious that the encoding mode for a system employing motion-compensated prediction would need to be supplied to the decoder so that it could reverse the encoding process. *See Freedman Decl.* (Ex. 1009) at ¶¶45, 65. However, *Kalker* does not necessarily say that the encoding mode is sent and decoded as a third syntax element.

However, this limitation is obvious over *Kalker* in view of *Novotny*, which, like *Kalker*, teaches video coding compatible with the MPEG standards, and further expressly teaches transmitting the encoding mode (either intra- or inter- prediction) to the decoder via a syntax element (i.e., *the third syntax element representing an encoding mode used to encode the sub-block, wherein the encoding mode comprises one of an intra-prediction mode and an inter-prediction mode*). As explained by *Novotny* (and in the Technology Background above), intra-prediction refers to a coding unit referring to locations within the same frame to account for spatial redundancy, while inter-prediction refers to corresponding bits in other frames in a video sequence to account for temporal redundancy and to provide motion compensation. *See Novotny* (Ex. 1007) at [0027]-[0028]. *Novotny* teaches that the coding mode used for a particular block is indicated in a syntax element that identifies the coding mode for each block (inter, intra – referred to as the macroblock type or MB type):

> For example, **the macroblock (MB) type** generally specifies how a macroblock (e.g., a 16×16 block of video frame pixels) is partitioned (or segmented) and/or encoded. The MB types generally **include, but are not limited to, Intra16×16, Intra4×4**, Skip, Direct, **Inter** and PCM.

Novotny (Ex. 1007) at [0050]; *see also id.* at [0051]-[0065] (providing specific examples of modes and syntax elements used to represent the same); *see also id.* at [0025]-[0028] (describing the use of intra- and inter-coding in existing standards); *see also Freedman Decl.* (Ex. 1009) at ¶45 ("macroblock type" was a common syntax element in video coding). Thus, *Novotny* describes the use of a *syntax element representing an encoding mode used to encode the sub-block, wherein the encoding mode comprises one of an intra-prediction mode and an inter-prediction mode*.

A PHOSITA would have recognized that, like disclosed in *Novotny* where the syntax element representing the encoding mode (e.g. "Intra") is associated with a syntax element representing block size (e.g., 16×16), that the encoding mode would similarly be associated with *Kalker*'s block-size codes, S. *See id.*; *see also Freedman*

Decl. (Ex. 1009) at ¶65. A PHOSITA also would have recognized that this would require nothing more than including bits representing "Intra" or "Inter" into *Kalker*'s run length encoded bit stream. *Id*. The result of the inclusion of these bits representing "Intra" or "Inter" would predictably be the transmission of a *third syntax element, separate from the first syntax element* (S=1) *and the second syntax element* (S=3). As should also be obvious, *Novotny* expressly discloses that the syntax elements, including data indicating "intra" or "inter" encoding mode, are in fact *decoded*. *See e.g., Novotny* at Claim 1 (referencing decoded syntax elements).

A PHOSITA would have been motivated to incorporate the teachings of *Novotny* of assigning a coding mode for a given block (i.e., inter- or intra-prediction mode) and transmitting the encoding mode to the decoder. First, such information must be conveyed to a decoder to enable it to perform motion compensation prediction and improve video quality. *See Freedman Decl.* (Ex. 1009) at ¶¶45, 65. Second, *Kalker,* like *Novotny*, is already transmitting block size syntax elements in an encoded bit steam. It would have been obvious to place an encoding mode syntax element in the encoded bit stream transmitting the S values already present in *Kalker* because this would be the most natural place to communicate an encoding-mode related syntax element. *See Freedman Decl.* (Ex. 1009) at ¶65. A PHOSITA would have had a reasonable expectation of success in implementing the teachings of *Novotny* with respect to transmitting a syntax element representing an encoding mode in the system of *Kalker*, as such would have required minor software modifications and yielded predictable results with no need for experimentation because sending an encoding mode was already a well-known and practiced element used to instruct a decoder on how to reconstruct encoded images. *Id.* Thus, the combination of *Kalker* with the encoding-mode syntax element disclosed in *Novotny* renders this limitation obvious.

1[f] / 7[b][vi] / 15[f] . decoding / decode / decode* the sub-block according to the encoding mode, without further partitioning the sub-block, based on the determination that the block does not include further separately encoded *subpartitions / sub-partitions / sub partitions.

This limitation is obvious in view of the combination of *Kalker* and *Novotny. Kalker* teaches that a sub-block can be processed *without further partitioning the sub-block, based on the determination that the block does not include further separately encoded subpartitions.* For example, when the partitioning grid corresponds to the exemplary largest block size of 16×16, the decoder will encounter a grid block and scan for block-size codes. If the block-size code is S=3 (a 16×16 block), the decoder knows that block has no further separately encoded sub-partitions, and it moves to the next grid, starting with a block of the largest size. *See Kalker* (Ex. 1006) at 5:38-41. The indication of the partitioning is also true for sub-blocks in the current block. If the decoder moves to the next grid block, it may first encounter an S=2 (an 8×8 block). The decoder may begin decoding that 8×8 sub-block based on the determination that the 8×8 sub-block does not include further separately encoded sub-partitions. *See Kalker* (Ex. 1006) at 5:36-57, claim 8; *see also id*. at 4:48-50. The decoder may then proceed to the next 8×8 sub-block, where it may encounter an S=1. The decoder may similarly begin decoding that 4×4 sub-block of the sub-block based on the determination that the 4×4 sub-block does not include further separately encoded sub-partitions. *Id.*

Further, as discussed above, it would have been obvious to decode the blocks according to one of the intra- or inter-prediction coding modes indicated by the encoding mode syntax elements, as taught by *Novotny*, for the reasons discussed with respect to Claim 1[e]. As the decoder determines that a given sub-block has been assigned a given partition or has no further partitions, the receiving side decodes the picture data according to the assigned encoding mode based on a prediction list representing the reference frame used to predict a block. *See Novotny* (Ex. 1007) at [0066]; *see also id*. at Figs. 8, 9; *see also Kalker* (Ex. 1006) at 2:46-48 (system applies to inputs with motion-compensated video streams). As repeatedly noted, *Kalker* discloses that its decoder operates reciprocally to the encoder, and this would include with respect to the coding mode selected. *See Freedman Decl.* (Ex. 1009) at ¶65; *see also Kalker* (Ex. 1006) at 4:48-50.

ii. Claims 2, 8, and 16

***2 / 8 / 16. The method of claim 1 / the device of claim 7 / the non-transitory computer-readable medium of claim 15,* wherein at least one other block of the plurality of blocks has a size that is less than the maximum size and greater than the minimum size.**

Kalker teaches this limitation, or at least renders it obvious. For example, *Kalker* teaches in one embodiment, a mid-sized sub-block, such as an 8×8 block represented by S=2 *has a size that is less than the maximum size* (S=3) *and greater than the minimum size* (S=1):

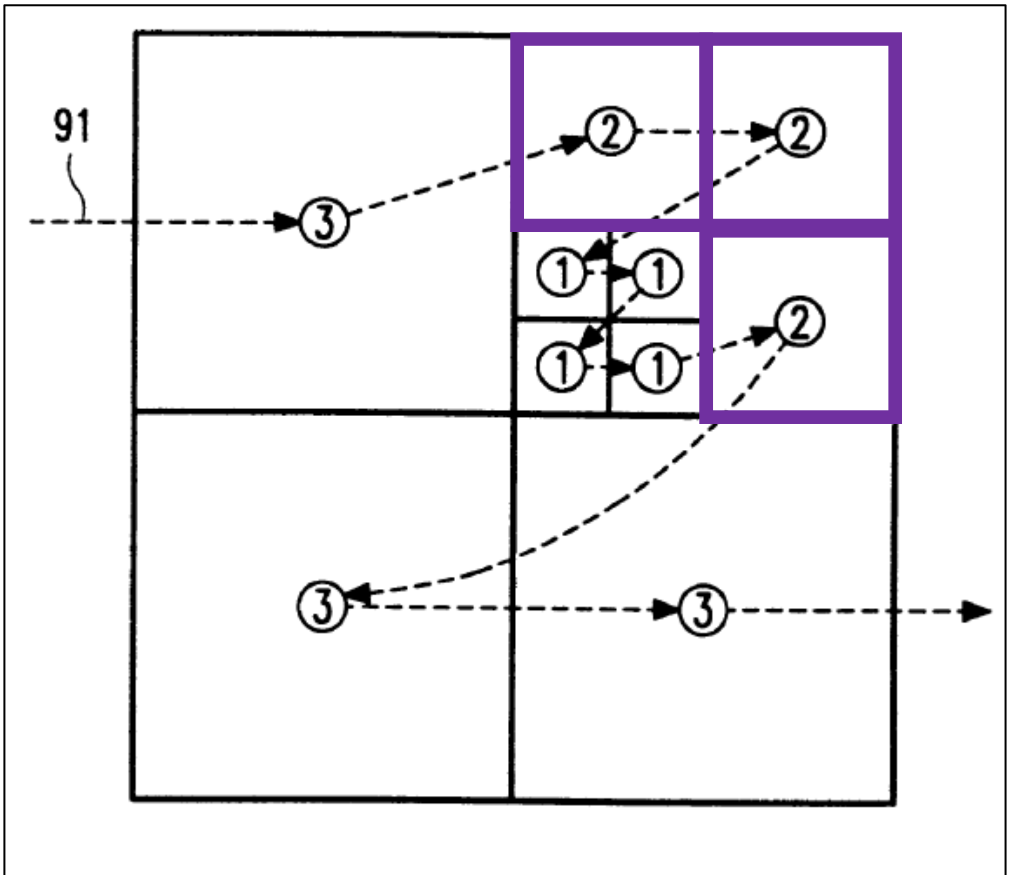


FIG. 9

Kalker (Ex. 1006) at Fig. 9 (annotated to highlight partitioned blocks smaller than the maximum size and larger than the minimum size); *see also id.* at 5:41-46-54; *see also id.* at 3:8-18 (the receiving station applies the relevant block size S to the inverse transform circuit for decoding encoded data). Once the decoder sees that this is an S=2 block, it reconstructs the block accordingly and goes to the next sub-block in the sequence or, if it has reached the end of the grid size corresponding to the maximum block size, to the next grid block. As discussed regarding Claim 1[b] (Claims 7[b][ii] and 15[b]), a PHOSITA would have found it obvious to apply *Kalker*'s teachings to video codec systems employing block sizes larger than 16×16 pixels as the maximum size, particularly in light of *Novotny*. *See Freedman Decl.* (Ex. 1009) at ¶¶40-42, 64

Therefore, Claims 2, 8, and 16 are obvious over *Kalker* in view of *Novotny*.

iii. Claims 3, 9, 17

***3 / 9 / 17. The method of claim 1, further comprising decoding / the device of claim 7, wherein the processor is further configured to decode / the non-transitory computer-readable medium of claim 15, further comprising instructions that cause the processor to decode* one or more syntax elements representative of partitioning for the current block.**

Kalker teaches this limitation, or at least renders it obvious. *Kalker* teaches assigning block-size codes (i.e., *syntax elements representative of partitioning for the current block*) that at the individual block level are representative of the partitioning for each current block being scanned by a decoder. As mentioned regarding Claim 1[c], *Kalker* teaches an embodiment in which a current block in the partitioning grid has a starting size equal to the largest block size being decoded in the coding unit. For example, in embodiment cited for claims 1, 7, and 15,, the largest block size is 16×16 and the corresponding grid is thus made up of 16×16 blocks (any one of which could be a *current block*). Within each grid, block-size codes (*syntax elements*) represent the partitioning of the current block. A block-size code of "3" represents a 16×16 block such that the grid block is not further partitioned, a "2" represents an 8×8 block, and a "1" represents a 4×4 block. *See Kalker* (Ex. 1006) at 5:36-57; *see also id.* at 3:31-34 ("Each block size is represented by a block-size code S. In the present example, S=1 for 4*4 blocks, S=2 for 8*8 blocks, and S=3 for 16*16 blocks."); 1:45-52, 2:8-12, 3:8-18; 3:50-67, Figs. 3, 6-9; claims 1, 8; *see also Freedman Decl.* (Ex. 1009) at ¶¶54-55. These block-size codes are ultimately assigned bit codes of 1's and 0's that are representative of the block-size code syntax elements and transmitted to the receiving station for decoding. *See Kalker* (Ex. 1006) at 4:11-42. Where the receiving station's segmentation map reconstruction circuit receives the segmentation map encoded by the transmitting station, it decodes block-size codes (i.e., *syntax elements*) and partitions the grid blocks (corresponding to the maximum size block in the coding unit) accordingly. *See id.* at 4:36-47; *see also id.* at 4:55-56 ("If an **element** is not the EOS code, it represents a block size S."); *see* *also id.* at 5:31-57 (describing embodiment based on scanning on the basis of the largest block size).

Therefore, Claims 3, 9, and 17 are obvious over *Kalker* in view of *Novotny*.

iv. Claims 4, 10, and 18

4 / 10 / 18. The method of claim 1 / the device of claim 7 / the non-transitory computer-readable medium of claim 15,* wherein the encoding mode is *the intra-predicting mode / the intra-predicting mode / intra-prediction,* and decoding the sub-block according to the encoding mode comprises predicting the sub-block from one or more neighboring pixels according to *the intra-predicting mode / intra-predicting mode / intra-prediction.

Claim 4 (and Claims 10 and 18, *infra*) describes what intra-prediction entails—reducing spatial redundancy by *predicting a sub-block from one or more neighboring pixels* in a corresponding reference sub-block. *See Freedman Decl.* (Ex. 1009) at ¶¶37-38, 60. As discussed regarding Claim 1[e], it would have been obvious to employ the encoding/decoding modes taught in *Novotny*, which includes the intra-prediction mode. *See id.* Further, as discussed in the Technology Background, a decoder's purpose is to decode encoded data according to the instructions given by an encoder, which includes applying an encoding mode set by the encoder. *See id.*; *see also supra*, p. 4-5; *see also Kalker* (Ex. 1006) at 3:8-18 (receiving station includes demultiplexer and decoding components to perform the inverse operations of the coder), 4:36-42. A PHOSITA would have understood and appreciated that when the encoding mode is intra-prediction as is disclosed in *Novotny*, the encoding mode by definition comprises *predicting the sub-block from one or more* *neighboring pixels*. *See Freedman Decl.* (Ex. 1009) at ¶60; *see also Novotny* (Ex. 1007) at [0025].

Therefore, Claims 4, 10, and 18 are obvious over *Kalker* in view of *Novotny*.

v. Claims 6, 12, and 20

***6 / 12 / 20. The method of claim 1, further comprising receiving / the device of claim 7, wherein the processor is configured to receive / the non-transitory computer-readable medium of claim 15, further comprising instructions that cause the processor to receive* a quantization parameter modification value for the sub-block, wherein decoding the sub-block comprises dequantizing the sub-block according to the quantization parameter modification value.**

Kalker teaches this limitation, or at least renders it obvious alone or in combination with *Novotny*. The '365 Patent describes the quantization parameter modification value for a given sub-block as syntax information used by the decoder for performing inverse quantizing for a given block. *See* '365 Patent (Ex. 1001) at 16:64-17:2. *Kalker* provides the same disclosure. *See Kalker* (1006) at 3:10-13 ("The transform coefficients are applied to an entropy decoder and inverse quantizer 7 <u>which performs the inverse operations of quantizer </u>and entropy coder 2."); *see also id.* at 2:48-54 ("The input signal is applied to a transform circuit 1 which subjects <u>picture blocks having a variable block size S</u> to a picture transform. … The transform coefficients are quantized and the quantized coefficients are lossless coded by a quantizer and entropy coder 2."); *see also id.* at 2:54-59 (providing examples of transformation and quantization "well-known" in the art); *see also Freedman Decl.* (Ex. 1009) at ¶39.

To the extent it is argued that *Kalker* does not teach individual quantization parameter modification values for a given sub-block, this limitation is obvious over the combination of *Kalker* and *Novotny*. *Novotny* teaches providing a syntax element indicating a "macroblock quantization parameter" (i.e., a quantization parameter modification value). *See Novotny* (Ex. 1007) at [0037]; *see also id.* at [0069]-[0070] (describing the macroblock quantization parameter), Fig. 10. A PHOSITA would have been motivated to incorporate *Novotny*'s teachings for using a quantization parameter for each block into the system of *Kalker* because optimizing quantization parameters in video coding systems was known to reduce distortion. *See Freedman Decl.* (Ex. 1009) at ¶¶39, 66. Further, *Kalker* leaves open the specific methods of quantization that may be used in its system which would prompt a PHOSITA to consider other known methods of quantization. *See id.* A PHOSITA would have had a reasonable expectation of success in incorporating *Novotny*'s teachings into the system of *Kalker* because *Novotny* merely discloses conventional means of transformation and quantization of coded units, and incorporating these known techniques would have required at most minor software modifications in the video codec system taught in *Kalker*. *See id.*

Therefore, Claims 6, 12, and 20 are obvious over *Kalker* in view of *Novotny*

vi. Claim 13

13. The device of claim 7, wherein the device further comprises a display configured to display the sequence of pictures.

Claim 7 is obvious over the combination of *Kalker* and *Novotny*. *See supra*, Sec. IV.A.i. Further, this limitation is obvious over *Kalker* in view of *Novotny*. *Kalker* teaches that the video-receiving station generates an output signal X_{out} upon decoding a video sequence. *See id*. at 3:13-15, Fig. 1. A PHOSITA would recognize, and the '365 Patent acknowledges, that a common purpose of video coding and decoding is to display video to a user; therefore, it would have been obvious to a PHOSITA that the X_{out} signal is meant for a display configured to display the decoded sequence of video pictures. *See Freedman Decl.* (Ex. 1009) at ¶¶43, 67; *see also* '365 Patent (Ex. 1001) at 1:27-33, (describing the "wide range" of existing display devices used to implement known video compression techniques, such as digital televisions, *inter alia*); *see also id*. at 11:22-27.

However, to the extent the concept of *Kalker*'s receiving station outputting a decoded video signal X_{out} does not satisfy this limitation, this limitation is obvious over the combination of *Kalker* and *Novotny*. *Novotny* teaches that the purpose of its invention is to display decoded video along with decoded bitstream syntax elements. *See id*. at [0005]; *see also id*. at Figs. 6, 9-13. *Novotny* provides examples of display devices at the receiving end, including televisions, monitors, a computer, "or any other medium implemented to … display … the uncompressed bitstream (decoded video signal) and/or information regarding the bitstream syntax elements.”). *See id.* at [0036]; *see also id.* at [0037], [0041]-[0042]; *see also id.* at [0046] (may display decoded video only). Given *Kalker*’s similar teachings using MPEG2-like coding and its teaching of “transmitting” video images to a receiver, it would have been obvious to a PHOSITA that the receiving station would have a display device, such as a television screen, as taught in *Novotny*. *See id.*; *see also Kalker* (Ex. 1006) at 2:54-59 (MPEG2-like coding used), 1:7-15 (invention relates to a method of “transmitting” encoded video pictures); 3:8-15 and Fig. 1 (diagram of video transmitting and receiving stations, illustrating an output signal from the receiving station); *see also Freedman Decl.* (Ex. 1009) at ¶67. A PHOSITA would have been motivated to incorporate *Novotny*’s teaching of having a display at a decoding device in *Kalker*, and would have had a reasonable success of doing so, because displaying video has been the primary purpose of decoding video long before 2008 and, therefore, would have simply required the implementation of known techniques in similar prior art systems. *See Freedman Decl.* (Ex. 1009) at ¶67.

Therefore, Claim 13 is obvious over the combination of *Kalker* and *Novotny*.

vii. Claim 14

14. *The device of claim 7, wherein the device is one or more of a camera, a computer, a mobile device, a broadcast receiver device, or a set-top box.*

Claim 7 is obvious over the combination of *Kalker* and *Novotny*. *See supra*, Sec. IV.A.i. Further, this limitation is obvious over *Kalker* in view of *Novotny*. Given

Kalker's teachings of a receiving station receiving a transmission from a "video transmitting station," and in light of *Novotny*'s teachings of the application of using a computer or television, *inter alia*, discussed regarding Claim 13 immediately above, it would have been obvious to a PHOSITA to employ a broadcast receiver device, set-top box, or computer as *Kalker*'s receiving station. *See Kalker* (Ex. 1006) at 1:41-48; Fig. 1; *see also Freedman Decl.* (Ex. 1009) at ¶¶43, 67.

Therefore, Claim 14 is obvious over the combination of *Kalker* and *Novotny*.

B. Ground 2: Claims 5, 11, and 19 are obvious over *Kalker* in view *Novotny* in further view of *Chiang*

U.S. Patent 6,084,908 to Chiang *et al.* ("*Chiang*") issued on July 4, 2000, and, therefore, is prior art to the '365 Patent at least under 35 U.S.C. § 102(b). *See Chiang* (Ex. 1008). *Chiang* is directed to systems and methods for determining an optimal quadtree structure for variable block-sized systems by calculating the rate-distortion costs for the various types and sizes of blocks. *See id.* at Abstract. Like *Kalker*, *Chiang* is directed to a variable-block-based system for encoding video data:

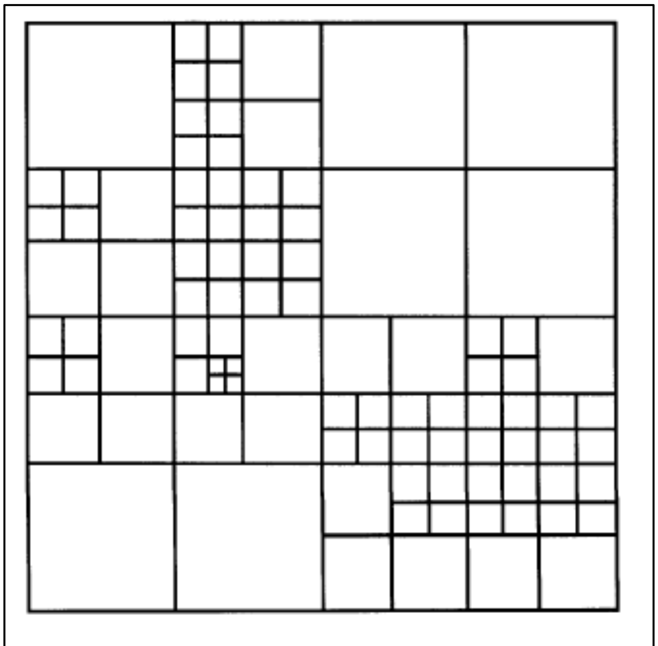


FIG. 9

See id. at Fig. 9 (depicting an optimum quad-tree structure for a frame of images); *see also id*. at 12:10-13. Notably, *Chiang* teaches that an initial partitioning size for the above frame is a 256×256-pixel macroblock. *See id*. at 11:62-12:4; *see also id*. at 5:43-54 (explaining that an initial block may be a 256×256 block).

Chiang is both within the field of endeavor of and reasonably pertinent to the ’365 Patent. *See Freedman Decl.* (Ex. 1009) at ¶¶68-69. Like the ’365 Patent, *Chiang* is directed toward the field of block-based video encoding and decoding. *See Chiang* (Ex. 1008) at 1:8-13. Further, *Chiang* is reasonably pertinent with at least one problem with which the inventor of the ’365 Patent was concerned. For example, as mentioned, *Chiang* teaches the use of blocks starting at sizes larger conventional 16×16 macroblock in video coding. *See id*. at 5:43-54; *see also id*. at 11:62-12:4.

Further, *Chiang* is also directed to the problem of improving the efficiency of video coding by reducing picture distortion at a target bit rate. *See id*. at 2:60-66; *see also* '365 Patent (Ex. 1001) at 9:20-30.

i. Claims 5, 11, and 19

***5 / 11 / 19. The method of claim 1 / the device of claim 7 / the non-transitory computer-readable medium of claim 15*, wherein the current block has a size of at least 64×64 pixels.**

As discussed in the first ground above, the combination of *Kalker* and *Novotny* renders Claim 1, 7, and 15 obvious. *See supra*, Sec. IV.A.i. *Novotny* teaches that an initial block size may be "other" sizes than a 16×16 block and provides 32×32 as a specific example, but does not explicitly provide that block sizes may be at least 64×64 pixels. *Chiang*, however, does. *Chiang* teaches employing a video coding/decoding system wherein an initial block size may be as large as 256×256 pixels.

A PHOSITA would have been motivated to incorporate *Chiang*'s teachings of employing a larger block size into the system of *Kalker*, as modified by *Novotny*. As discussed with respect to claims 1[b], 7[b][ii], and 15[b], a PHOSITA would have recognized *Kalker*'s teachings of using a 16×16 block as the maximum block size simply as exemplary for its teachings, not mandatory, and that larger block sizes would be desirable to capture sequences of images where little variance occurs across pictures. *Freedman Decl.* (Ex. 1009) at ¶70. And *Chiang* expressly explains the benefits of using larger blocks—enabling lower rates for highly redundant prediction images without the penalty of introducing greater distortion. *See Chiang* (Ex. 1008) at 5:61-6:4; *see also id*. at 2:14-23 (explaining trade-offs between small and large blocks). By using a larger initial block size, a programmer could allow a video codec system to use very little data for areas of large redundancy, such as background elements of a scene, while not compromising the encoder's ability to partition the block further for scenes with greater variance. *See Freedman Decl.* (Ex. 1009) at ¶70. Further, as discussed regarding Claim 1[c] a PHOSITA would have had a high expectation of success in implementing larger starting macroblocks into the system of *Kalker*, as it at most would have involved (1) increasing the minimum block size if only three block sizes were to be used, or (2) increasing the number of available block sizes (or "levels" or "modes" of partitioning). *See Freedman Decl.* (Ex. 1009) at ¶70; *see also* Sec IV.A.i, *supra*. Such modifications would still accomplish *Kalker*'s desired bit savings because a system designer would only use a larger block size if such resulted in less encoded data being sent due to the use of the larger block size. *See Freedman Decl.* (Ex. 1009) at ¶70. In this way, these modifications would have had the predictable and desirable result of increasing the flexibility and compression of video data. *See Freedman Decl.* (Ex. 1009) at ¶70.

Therefore, Claims 5, 11, and 19 are obvious over *Kalker* in view of *Novotny* in further view of *Chiang*.

V. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of Claims 1-20 of U.S. Patent No. 9,930,365.

Respectfully submitted,
ERISE IP, P.A.

BY: */s/ Eric A. Buresh*
Eric A. Buresh, Reg. No. 50,394
Roshan Mansinghani, Reg. No. 62,429
Ashraf Fawzy, Reg. No. 67,914
Michelle Callaghan, Reg. 75,665

ATTORNEYS FOR PETITIONER

VI. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Parties-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that Unified is the real party-in-interest, and further certifies that no other party exercised control or could exercise control over Unified's participation in this proceeding, the filing of this petition, or the conduct of any ensuing trial. In view of *Worlds Inc. v. Bungie, Inc.*, 903 F.3d 1237, 1242-44 (Fed. Cir. 2018), Unified has submitted voluntary discovery in support of its certification. *See* Petitioner's Voluntary Interrogatory Responses (Ex. 1021).

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), Unified is unaware of any law suits in which the '365 Patent is asserted or challenged.

C. Lead and Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4). Eric Buresh will serve as lead counsel. Ashraf Fawzy will serve as first back-up counsel. Roshan Mansinghani and Michelle Callaghan will serve as additional back-up counsel. Please direct all correspondence regarding this proceeding to lead and back-up counsel at their respective email addresses listed below. 37 C.F.R. § 42.8(b)(4).

Lead Counsel	**Back-Up Counsel**
Eric A. Buresh (Reg. No. 50,394) eric.buresh@eriseip.com ptab@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600	Ashraf Fawzy (Reg. No. 67,914) afawzy@unifiedpatents.com Postal and Hand-Delivery Address: Unified Patents Inc. 1875 Connecticut Ave., NW, Floor 10 Washington, D.C. 20009
	Roshan Mansinghani, Reg. No. 62,429 roshan@unifiedpatents.com Unified Patents Inc. 13355 Noel Road, Suite 1100 Dallas, TX, 75240 Telephone: (214) 945-0200
	Michelle Callaghan (Reg. No. 75,665) michelle.callaghan@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 5600 Greenwood Plaza Blvd., Suite 200 Greenwood Village, Colorado 80111 Telephone: (913) 777-5600

APPENDIX OF EXHIBITS

Exhibit 1001	U.S. Patent 9,930,365 ("'365 Patent")
Exhibit 1002	File History for '365 Patent ("*'365 File History*")
Exhibit 1003	File History for U.S. Patent 8,503,527 ("*Pat. 8,503,527 File History*")
Exhibit 1004	File History for U.S. Patent 8,948,258 ("*Pat. 8,948,258 File History*")
Exhibit 1005	File History for U.S. Patent 9,788,015 ("*Pat. 9,788,015 File History*")
Exhibit 1006	U.S. Patent 5,999,655 to Kalker *et al*. ("*Kalker*")
Exhibit 1007	U.S. Pub. 2005/0123282 to Novotny *et al.* ("*Novotny*")
Exhibit 1008	U.S. Patent 6,084,908 to Chiang *et al.* ("*Chiang*")
Exhibit 1009	Declaration of Dr. Immanuel Freedman ("*Freedman Decl.*")
Exhibit 1010	Curriculum Vitae of Immanuel Freedman, Ph.D.
Exhibit 1011	Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd. (2003) ("*Richardson*")
Exhibit 1012	Peter Symes, *Video Compression Demystified*, McGraw-Hill (2001) ("*Symes*")
Exhibit 1013	ITU-T Recommendation H.264, International Telecommunication Union (Nov. 2007) ("*ITU H.264*")
Exhibit 1014	U.S. Pub. 2006/0002464 to Au *et al.* ("*Au*")
Exhibit 1015	Int'l Pub. No. WO 2005/038603 to Woods *et al.* (published Apr. 28, 2005) ("*Woods*")
Exhibit 1016	U.S. Pat. 6,233,017 to Chadda (filed Jun. 30, 1997) ("*Chadda*")
Exhibit 1017	U.S. Pat. 6,778,709 to Taubman *et al.* (field Mar. 12, 1999) ("*Taubman*")
Exhibit 1018	Matthew Drake *et al.*, *MPEG-2 Decoding in a Stream Programming Language*, Proceedings 20th IEEE International Parallel & Distributed Processing Symposium (Apr. 2006)
Exhibit 1019	*ITU-T Recommendation H.262*, International Telecommunication Union (Feb. 2000) ("*ITU H.262*")
Exhibit 1020	U.S. Pub. 2007/0074265 to Bennett *et al*. (published Mar. 29, 2007) ("*Bennett*")
Exhibit 1021	Petitioner's Voluntary Interrogatory Responses

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. § 42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. §42.8, certificates of service or word count, or appendix of exhibits, contains 13,401 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: February 28, 2019

BY: */s/ Eric A. Buresh*
Eric A. Buresh, Reg. No. 50,394

ATTORNEY FOR PETITIONER

CERTIFICATE OF SERVICE ON PATENT OWNER UNDER 37 C.F.R. § 42.105

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105, the undersigned certifies that on February 28, 2019, a complete and entire copy of this Petition for *Inter Partes* Review, including all exhibits listed in the Appendix of Exhibits, as well as the accompanying Power of Attorney, was provided via Federal Express to the Patent Owner by serving the counsel of record for the '365 Patent as listed on PAIR:

Nixon & Vanderhye P.C./OPTIS
RE: U.S. Patent App. No. 15/696,263
901 North Glebe Road, 11th Floor
Arlington VA 22203

Dated: February 28, 2019

BY: *<u>/s/ Eric A. Buresh</u>*
Eric A. Buresh, Reg. No. 50,394

ATTORNEY FOR PETITIONER