UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SNAP INC. and HISENSE USA CORPORATION,
Petitioners

v.

NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2025-01375
U.S. Patent No. 11,805,267

DECLARATION OF IMMANUEL FREEDMAN IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,805,267

TABLE OF CONTENTS

I, Immanuel Freedman, declare as follows:

1. My name is Immanuel Freedman. I am a Senior Member of the Institute of Electrical and Electronic Engineers (IEEE) and Volunteer Researcher in areas related to computer estimation and modeling in the State University of New York at Buffalo. I have prepared this report as an expert witness retained by Snap Inc. and Hisense USA Corporation. In this report I give my opinions as to whether certain claims of U.S. Patent No. 11,805,267 ("the '267 patent") are invalid. I provide technical bases for these opinions as appropriate.

2. This declaration is based in part on prior declarations I prepared in 2024 for Amazon.com, Inc. in IPR Case Nos. IPR2024-00626 and IPR2024-00627 involving the same Nokia '267 patent. The prior Amazon IPRs were instituted but I understand the proceedings were terminated prior to reaching a final decision based on a settlement between Nokia and Amazon.

3. This report contains statements of my opinions formed to date and the bases and reasons for those opinions. I may offer additional opinions based on further review of materials in this case, including opinions and/or testimony of other expert witnesses. I make this declaration based upon my own personal knowledge and, if called upon to testify, would testify competently to the matters contained herein.

I. OVERVIEW OF THE TECHNOLOGY

A. Video Compression Basics

4. Video encoding, also referred to as video compression, exploited redundancies in video data to reduce the size of video. Since the early 1990s, major video coding standards such as MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263, and H.264 have applied the same model, where video encoders have a motion estimation and compensation front end, a transform stage such as Discrete Cosine Transform ("DCT"), and an entropy encoder at the back end for generating the coded bitstream. At the decoder, the inverse process was used to decode the video. In 2003, the H.264 standard, also known as Advanced Video Coding ("AVC"), was introduced. This standard quickly became a prevalent and widely adopted video format. The model of a typical general video encoder is illustrated below. Ex. 1005, Fig. 2. This fundamental model has been used by major video encoding standards since the 1990s.

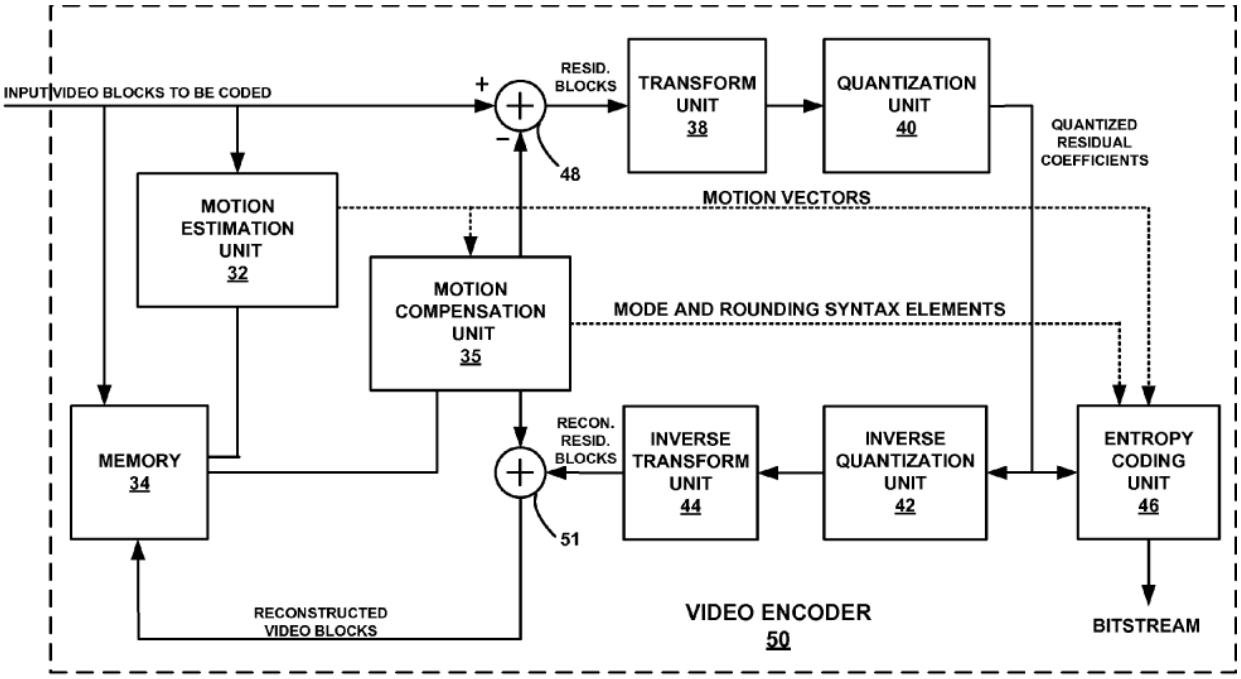


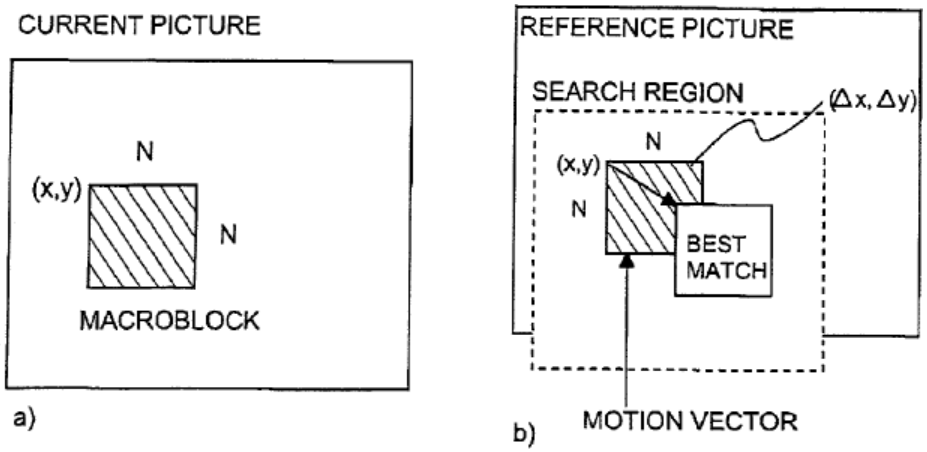
FIG. 2

5. Video was made up of a series of pictures known as frames. Each frame was segmented into blocks of pixels (e.g., referred to as video blocks, macroblocks, sub-macroblocks, etc.). Each block contained a group of pixels, such as 16x16, 8x8, or 4x4 pixels.

B. Motion Estimation and Compensation

6. Video blocks were encoded with reference to each other. This was known as predictive coding. Two main types of predictive coding were intra-frame and inter-frame encoding. Intra-frame encoding used predictive coding within the same frame, where a block was encoded with reference to another block within the same frame. This took advantage of similarities within the same frame. Inter-frame coding allowed a block to be encoded with reference to blocks in other frames.

This type of temporal prediction, called motion estimation and compensation, took advantage of similarities between different frames. For example, when an object appeared in successive frames, inter-frame prediction encoded and transmitted information for a first frame and encoded subsequent frames by reference to reference blocks in the first frame. A motion vector indicated the displacement of a current block with respect to a reference block, for example indicating that a block moved to the right 5 pixels and moved down 3 pixels between frames. Ex. 1010, ¶18, Fig. 4:



Where the same subject moved—such as a ball rolling, or the horizon moving slightly while a car traveled across the screen—an encoder could transmit information for that subject once and use motion vectors after that.

7. In many video standards, including H.264, blocks encoded with only intra-frame encoding were known as "I" blocks. Conversely, there were two types of inter-coded blocks: "P" and "B" blocks. "P" blocks allowed unidirectional

prediction to other frames, while "B" blocks allowed bidirectional prediction, meaning blocks within the frame could be predicted in the forward *and* backwards directions. Ex. 1011, 000002, 000007; Ex. 1012, 000198-200.

8. Motion estimation and compensation involved identifying the movement of objects or regions between successive frames in a video sequence. In bidirectional prediction for a target block, the encoder searched for similar blocks in two reference frames, such as a past/previous reference frame and a future/subsequent reference frame, that best match the target block. This process resulted in two motion vectors, each pointing to a different block in a different reference frame. Ex. 1012, 000194-195, 000200, 000062, 000090.

9. The encoder combined the two matching blocks to create a bidirectional prediction of the target block. For example, the bidirectional prediction was commonly calculated as an average (or weighted average) of the two reference blocks, with each pixel in the bidirectionally-predicted block being an average of the corresponding pixels in the blocks obtained from the reference frames. Ex. 1011, 000011; Ex. 1012, 000195 (averaging 16x16 reference blocks from List0 and List1 into bi-prediction block):

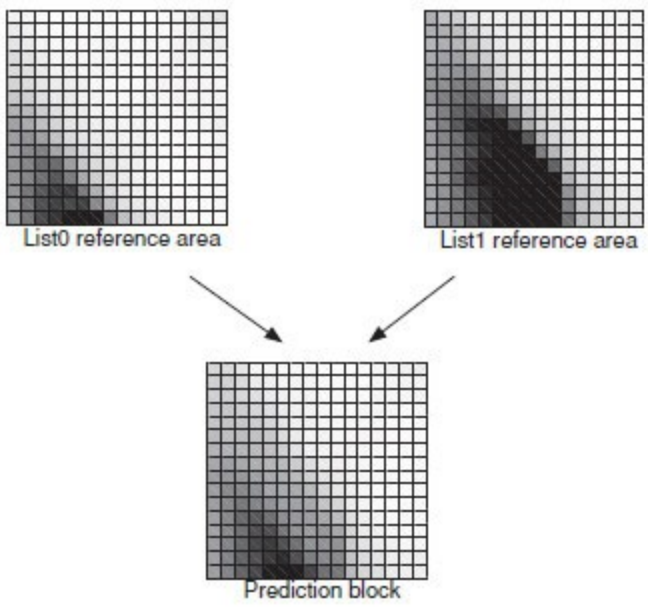


Figure 6.35 Bipredictionexample

For example, if one frame depicted the moment a ball starts rolling and another frame depicted where the ball stops, it is easy to deduce that the ball traveled between those two points—that's bidirectional prediction.

10. The difference between the target block and the bidirectional prediction was calculated to obtain the residual block. Ex. 1012, 000062-63, 000117. This residual block included, for each pixel of the target block, a difference between the pixel and its corresponding predicted pixel value. The motion vectors and the residual block were encoded and transmitted to the decoder. *See* Ex. 1011, 000007; Ex. 1012, 000062-63, 000102. Since the motion vectors and residuals typically required fewer bits than the original pixel values, bidirectional motion prediction contributed to significant data compression.

11. The decoder performed the inverse process to reconstruct the target block based on data received from the encoder. The decoder first extracted and decoded the motion vectors transmitted from the encoder. These vectors indicated the displacement between the target block in the current frame and the matching blocks in the past and future reference frames. Using the decoded motion vectors, the decoder located the corresponding matching blocks in the past and future reference frames. Ex. 1012, 000062, 000084.

12. The decoder then combined these blocks in the same manner as the encoder to reconstruct the bidirectional prediction of the target block, which included a prediction value for each pixel of the target block. Ex. 1012, 000062, 000074. For example, when a weighted bidirectional prediction method was used at the encoder, the decoder used the same weights to combine the two reference blocks. The decoder also extracted and decoded the residual block that was transmitted from the encoder. The original target block was reconstructed by adding the decoded residual block to the reconstructed bidirectional prediction. Ex. 1012, 000059-60, 000062, 000074, 000123.

C. Subpixels and Interpolation

13. When a motion vector pointed to an integer pixel position, the values of the block at that position were used to generate a predicted block. But the H.264 video compression standard, along with multiple other standards, allowed motion

vectors to have more granular subpixel resolution by pointing to fractional pixel positions (e.g., half-pixel or quarter-pixel positions), resulting in more accurate motion estimation and compensation. Ex. 1011, 000002; Ex. 1012, 000184. This situation arose when the best match for a target block in a reference frame was not located at an exact integer pixel position, for example where an object moved exactly one pixel between frames, but rather at a fractional (subpixel) position, e.g., where an object moved a half or quarter pixel between frames.

14. When a motion vector points to an integer pixel position, the values of the reference block are used to generate the predicted block. When a motion vector pointed to a subpixel position, the encoder/decoder used interpolation to generate the predicted block. Interpolation involved creating new pixel values at the subpixel positions based on the surrounding integer pixel values. For example, in half-pixel interpolation, the value at a half-pixel position could be calculated as the average of six adjacent integer pixel values. Ex. 1011, 000010; Ex. 1012, 000187. In summary, bi-prediction used two motion vectors, which could point to integer or sub-pixel positions. In bidirectional prediction, a B macroblock in the current picture is predicted from reference blocks in two different reference pictures. Thus, bidirectional prediction is a type of "interpolation" because the two reference blocks are used to interpolate the B macroblock. Therefore, bi-prediction often involved motion vectors that could point to integer or sub-pixel positions, with for

example both vectors pointing to interpolated blocks, or one vector pointing to an interpolated block and the other pointing to an integer pixel position. *See* Ex. 1011, 000010; Ex. 1012, 000184-189:

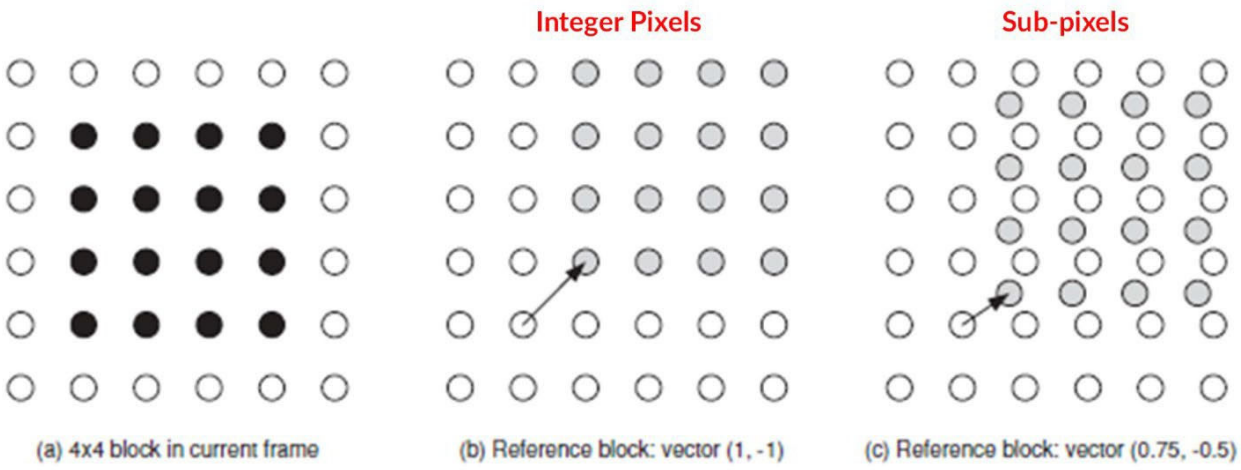


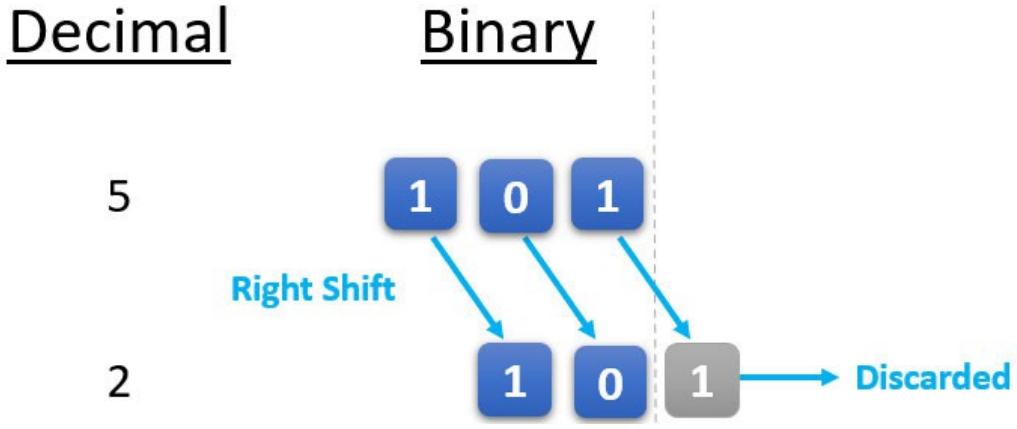
Figure 6.18 Example of integer and sub-pixel prediction

D. Precision and Bit Shifting

15. In computers, numeric values, such as pixel values (predicted or otherwise), were represented in registers or memories as binary numbers. An uncompressed binary number includes a series of positive or negative powers of 2. In a binary number, each digit is a 0 or 1. The precision of the binary number is the number of terms in that series, which is also the number of bits needed to represent its value in this form. When a calculation may result in multiple possible values, the precision of the result of the calculation corresponds to the number of bits needed to represent the possible values in an uncompressed binary form, which could be changed by mathematical operations. For example, an uncompressed variable having values that range from 0 to 3 can be represented by a binary

number of 2 bits. When multiplied by two, the result could range from 0 to 6 and thus required 3 bits of precision to represent the largest possible value 6 (110 in binary). Since multiplication and addition made numbers larger, they may have required more bits (higher precision) to represent the results in uncompressed binary form. Conversely, since division made numbers smaller, it may have reduced the number of bits needed to represent the result in uncompressed binary form. Higher precision allowed for a more accurate representation of a value but required more bits, while lower precision reduced the number of bits needed but might lead to a loss of accuracy. Ex. 1008, 000015; Ex. 1009, 7:4-19, 16:22-17:36, 20:42-47; Ex. 1010, ¶¶61-62. In video, pixel values were often represented using 8 bits, which represented 2^8=256 possible values (from 0-255). A pixel on a two-color (black and white) display can be represented with a single bit having two possible values: 0 or 1. A pixel with 2 bits of precision would represent values from 0-3. A pixel with 3 bits of precision would represent values from 0-7, and so on. Motion prediction and sub-pixel interpolation involved mathematical calculations, including multiplication and addition, that increased the number of bits needed to represent the intermediate results of those calculations. For example, when two 8-bit integer pixel values were averaged together, they were first added together, resulting in a sum that may require 9 bits to represent. The sum is next divided by two, resulting in an 8-bit number. Division can cause a rounding error

because, in integer arithmetic, any remainder from the division is discarded. I explain this in more detail below.



16. To satisfy precision constraints and ensure consistency in computations, adjustment of precision was often performed in computer logic. A widely used way to adjust the precision of binary numbers included shifting the bits of the number. A binary number could be truncated by shifting its bits to the right by a desired number of positions. Ex. 1004, ¶46; Ex. 1005, ¶57. A right shift reduced the number of bits, effectively truncating the desired number of bits from the right side of the number (the least significant bits). Right-shifting was mathematically equivalent to division by 2 for each position shifted, and the least significant bits were discarded. Each right shift effectively halved the value and decreased the precision of the numerical value by one bit, as the rightmost bit was discarded. This is similar to dividing a decimal number by 10, which moves the whole number to the right one digit.

17. The inverse of truncation can be applied to a binary number by shifting its bits to the left by the desired number of positions. Ex. 1009, 20:42-47; Ex. 1013, ¶91, ¶116, ¶131. Left shifting effectively performs multiplication by 2 for each position shifted. Ex. 1004, ¶46; Ex. 1009, 20:42-47; Ex. 1013, ¶91, ¶116, ¶131. The newly added least significant bits were filled with zeros. Each left shift effectively doubled the value and also increased the precision by one bit, as a new zero bit was added on the right.

18. Truncating a binary number is sometimes referred to as rounding the binary number. This rounding operation might cause a rounding error due to the difference between the original number and the rounded number. For example, truncating the binary number 101 (5 in decimal) by one bit results in 10 (2 in decimal). This is equivalent to dividing 5 by 2, which equals 2.5. Since the remainder is discarded, there is a rounding error of 0.5. This could occur when averaging 2 and 3, which normally would result in 2.5, but with binary numbers results in $\frac{2+3}{2} = \frac{5}{2} = 2$. Rounding errors could accumulate in calculations, especially when many operations were performed sequentially. This could lead to significant discrepancies between the computed result and the true value. In this example, if the above average is calculated twice and then added together, the expected result would be 2.5 + 2.5 = 5. However, because two right-shift (division) operations occur during the average, the result is 2 + 2 = 4, meaning the rounding

error is 1. To prevent this error, it was known in the art to maintain higher precision for the calculations, e.g., by delaying rounding (division/right-shift) operations. *E.g.*, Ex. 1009, 7:4-19. Applying that concept to this example, using basic arithmetic, the division in the averaging operation would be delayed until the end, thereby reaching the expected result of 5 and preventing rounding error:

$$\frac{(2+3)}{2}+\frac{(2+3)}{2}=\frac{(2+3)+(2+3)}{2}=\frac{5+5}{2}=\frac{10}{5}=5.$$

19. Rounding offsets were often used to adjust the result of rounding operations, particularly in binary computational systems such as video coding and 0digital signal processing. Rounding offsets were added to a binary value before a rounding operation, such as a right shift, to reduce the systematic bias that can occur in the rounding process. Ex. 1011, 000010; Ex. 1005, ¶55.

20. One well-known and commonly used approach to address accumulation of rounding errors was to maintain higher precision in intermediate steps of calculations. Ex. 1009, 7:4-19 ("An advantage of embodiments according to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."), 16:22-17:36 ("[A] higher bit-depth prediction[] and a higher bit-depth reconstruction residual information[] may lead to higher precise reconstructed samples[] of the block[], and therefore to a smaller needed residual

information, as in systems, wherein a rounding of prediction samples and of reconstructed residual samples occurs before the prediction and residual reconstruction process."); Ex. 1008, 000015 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding and clamping. It is possible to achieve additional accuracy by retaining a higher precision result after the first stage of filtering. … The two shifts are chosen so as to (a) add up to the required shift for normalizing the filters and (b) to allow for a 16 bit implementation - where the intermediate values in the second filtering operation are within 16 bits."); Ex. 1010, ¶¶61-62 ("[T]runcation of the ¼ resolution sub-pixel values has a deleterious effect on the precision of some of the ¼ resolution sub-pixel values. Specifically, the ¼ resolution sub-pixel values are less precise than they would be if calculated from values that had not been truncated and clipped. … . In the encoder the interpolation method according to TML6 works like the previously described TML5 interpolation method, except that maximum precision is retained throughout. This is achieved by using intermediate values which are neither rounded nor clipped.").

21. Since each rounding step had the potential to discard information and thereby introduce rounding errors, this practice delayed rounding and used more

bits to represent numbers during the computation process than were used in the final output. By doing so, the accumulation of rounding errors was minimized, as intermediate operations had a finer granularity and could represent values more accurately. *Id*.

II. THE '267 PATENT

A. Overview

22. The '267 patent is directed to "[a]pparatuses, methods and computer programs … for utilizing motion prediction in video coding." Ex. 1001, Abstract. The '267 patent discusses a process for generating a bi-directional prediction for a current block, including "us[ing] motion vector information to determine which block is used as a first reference block for the current block and which block is used as a second reference block for the current block," "us[ing] some pixel values of the first reference block to obtain first prediction values and some pixel values of the second reference block to obtain second prediction values," and combining "the two prediction values." *See, e.g.*, Ex. 1001, 12:41-55, 13:43-55, Fig. 10:

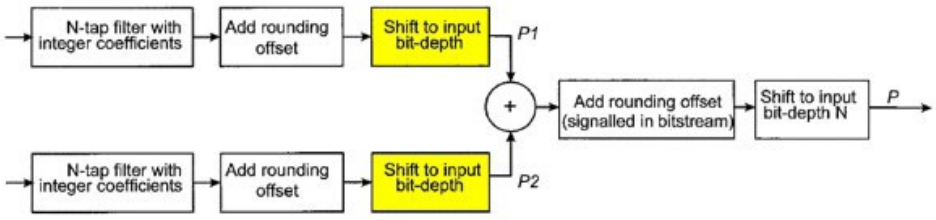


Fig. 10

On the encoder side, residual data (i.e., prediction errors) are determined based on differences between the current block and the prediction values, encoded, and sent to the decoder. *See, e.g.*, Ex. 1001, 11:47-12:3. On the decoder side, the received residual data are decoded and added to the prediction values to reconstruct the current block. *See, e.g.*, Ex. 1001, 12:4-20.

23. The '267 patent does not purport to invent this conventional process of bi-directional prediction, admitting that video coding processes according to standards such as MPEG-2, H.263, and H.264 were known in the art. Ex. 1001, 1:34-46 ("Background Information" section). The '267 patent states that "Background" art includes motion compensated prediction and specifically "bi-directional prediction" (e.g., Ex. 1001, 2:35-59), where reference blocks are determined based on motion vectors (e.g., Ex. 1001, 2:20-34, 3:12-18), predictions are determined based on reference blocks (e.g., Ex. 1001, 1:34-46), a bi-directional prediction is obtained by combining two predictions based on two reference blocks (e.g., Ex. 1001, 3:49-55, 3:66-4:20), and residual data are calculated as differences between the prediction values and the current block, encoded, and later used to reconstruct the current block (e.g., Ex. 1001, 1:52-59, 3:25-30, 2:1-12).

24. The '267 patent discusses that motion vectors may point to subpixels and that prediction values for a reference block may be subpixel prediction values determined based on interpolation using pixel values of reference blocks. *See, e.g.*,

Ex. 1001, 12:41-13:42. The interpolation is carried out using "a P-tap filter such as a six-tap filter." *Id*. These features were known in the prior art. The '267 patent admits that conventional standards, such as H.264, allow motion vectors to point to subpixels (e.g., half-pixel or quarter-pixel) and provide interpolation methods for determining subpixel predictions using P-tap filters. Ex. 1001, 2:60-3:11 ("The motion vectors are not limited to having full-pixel accuracy, but could have fractional-pixel accuracy as well. … The H.264/AVC video coding standard supports motion vectors with up to quarter-pixel accuracy. Furthermore, in the H.264/AVC video coding standard, half-pixel samples are obtained through the use of symmetric and separable 6-tap filters, while quarter-pixel samples are obtained by averaging the nearest half or full-pixel samples.").

25. The purportedly inventive concept of the '267 patent is to maintain prediction signals "in a higher precision during the prediction calculation" and reduce the precision "after the two or more prediction signals have been combined with each other." Ex. 1001, 4:29-43, 6:51-57, 12:41-13:55. By doing so, the '267 patent claims to "enable[] reducing the effect of rounding errors in bi-directional and multi-directional prediction." Ex. 1001, 4:29-35, 6:51-57, Fig. 11:

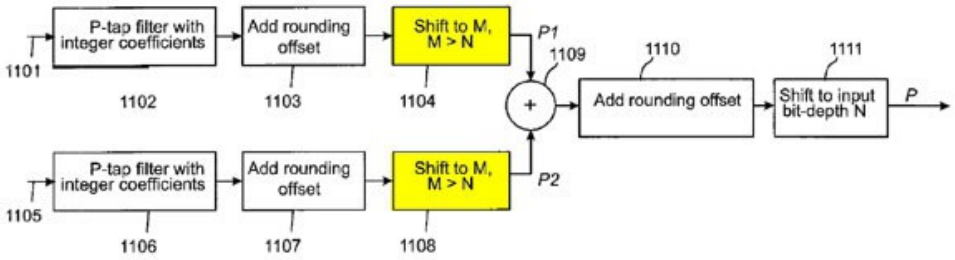


Fig. 11

However, the idea of reducing rounding error by maintaining higher precision in intermediate steps of calculations was known and applied in video coding art well before the timeframe of the '267 patent. *Supra* §I.D.

B. Priority Date

26. The '267 patent was filed May 24, 2021. The '267 patent was issued as a member of a chain of continuation applications, claiming priority to U.S. Patent No. 9,432,693, filed January 6, 2012, and U.S. Provisional Application No. 61/430,694, filed January 7, 2011. For purposes of this Declaration, I have analyzed obviousness as of January 7, 2011. I do not offer an opinion as to whether the '267 patent is entitled to a certain priority date. My invalidity opinions would not change if a later date (e.g., January 6, 2012) was determined to be the correct priority date because the prior art relied upon in this declaration would still be prior art.

C. Challenged Claims

27. I understand that Petitioner is challenging the validity of all claims (1-36) of the ’267 patent in the Petition for *Inter Partes* Review to which this declaration will be attached. The claims are reproduced in Appendix 3.

III. LEVEL OF ORDINARY SKILL IN THE ART

28. I have analyzed the ’267 patent and determined that the field of the patent is video encoding/decoding. *See, e.g.*, Ex. 1001, Abstract (“Apparatuses, methods and computer programs are provided for utilizing motion prediction in video coding.”). The ’267 patent characterizes its technical field as “an apparatus, a method and a computer program for producing and utilizing motion prediction information in video encoding and decoding.” Ex. 1001, 1:20-22.

29. In determining the characteristics of a hypothetical person of ordinary skill in the art (“POSITA”) of the ’267 patent at the time of the claimed invention, I considered several things, including various prior art techniques relating to video encoding/decoding, the type of problems that such techniques gave rise to, and the rapidity with which innovations were made.

30. I also considered the sophistication of the technologies involved, and the educational background and experience of those actively working in the field at the time. I also considered the level of education that would be necessary to understand the ’267 patent. Finally, I placed myself back in the relevant period of

time and considered the engineers and programmers that I have worked with and led in the field of video encoding/decoding.

31. I came to the conclusion that a POSITA at the time of the alleged invention of the '267 patent would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa.

IV. CLAIM CONSTRUCTION

32. For purposes of this inter partes review, I have considered the claim language, specification, and portions of the prosecution history, to determine the meaning of the claim language as it would have been understood by a person of ordinary skill in the art at the time of the invention. The "plain and ordinary meaning" or *Phillips* standard has traditionally been applied in district court litigation, where a claim term is given its plain and ordinary meaning in view of the specification from the viewpoint of a person of ordinary skill in the art.

33. I have applied the *Phillips* standard in my analysis. Unless otherwise stated, I have applied the plain and ordinary meaning to claim terms.

A. "precision"

34. In IPR2024-00626/627, the Board noted in its institution decision there was "no dispute … that … precision can be the number of bits needed to represent possible values." Ex. 1017, 20; Ex. 1018. I agree that, in the context of the '267 patent, the term precision covers the number of bits needed to represent possible values.

35. Claims 1, 7, and 13 recite that "the pixels of the current block, the first reference block, and the second reference block have values with a first precision," "said first prediction having a second precision," "said second prediction having the second precision," and "precision of said combined prediction" Ex. 1001, cls. 1, 7, 13. Dependent claims 6, 12, and 18 further recite "wherein the first *precision indicates a number of bits needed to represent the values* of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction." Ex. 1001, cls. 6, 12, 18. Therefore, these claim limitations can be satisfied when precision indicates a number of bits needed to represent the possible values of binary data, including uncompressed representations of binary data.

36. This interpretation is confirmed by the specification of the '267 patent, which uses the term "precision" to refer to a number of bits representing possible values, with examples of the number of bits used to represent the possible

pixel and prediction values. *See, e.g.*, Ex. 1001, 12:41-13:18 ("The precision M is higher than the precision of the expected prediction value. For example, *pixel values and the prediction values may be represented by N bits* wherein M>N. In some example implementations N is 8 bits and M is 16 bits but it is obvious that also other bit lengths can be used with the present invention."), 14:4-10 ("For example, if a motion vector of one of the prediction directions point to an integer sample, the bit-depth of prediction samples with integer accuracy may be increased by shifting the samples to the left so that the filtering can be performed with *values having the same precision*."), 13:19-55. These passages of the specification demonstrate that the precision describes a number of bits needed to represent possible values.

V. INVALIDITY GROUNDS

37. There are a number of patents and publications that constitute prior art to the '267 patent. I have reviewed and considered the prior art discussed in this section, along with the materials listed in Appendix 2.

38. Based on my review and analysis of the materials cited herein, my opinions regarding the understanding of a POSITA in the relevant timeframe (*supra* §§II.B, III), and my training and experience, it is my opinion that the challenged claims of the '267 patent are invalid in view of the following grounds:

Grounds	Claims	Statutory Basis	Prior Art
1	1-36	§ 103	Wada in view of Karczewicz-I
1A	3, 9, 15, 21, 27, 33	§ 103	Wada in view of Karczewicz-I and Srinivasan
2	1-36	§ 103	Karczewicz-I in view of Karczewicz-II
2A	3, 9, 15, 21, 27, 33	§ 103	Karczewicz-I in view of Karczewicz-II and Srinivasan

A. Ground 1: Claims 1-36 Are Rendered Obvious by Wada and Karczewicz-I

39. It is my opinion that a POSITA would have found Claims 1-36 obvious based on the teachings of Wada and Karczewicz-I.

1. U.S. Patent Application Publication No. 2010/0322303 ("Wada") (Ex. 1004)

40. I have reviewed the Wada reference. I understand that Wada was not cited or considered during prosecution of the '267 Patent based primarily on the fact that Wada is not cited on the face of the patent, nor have I seen the reference discussed in the prosecution history.

41. Wada was published December 23, 2010, and was filed as U.S. Application No. 12/853,956 on Aug. 10, 2010 as a continuation of an earlier-filed international application. Therefore, I understand Wada is prior art under at least pre-AIA §§102(a) and (e) because it was filed and published before the earliest

possible filing date for the '267 patent, its provisional application's filing date of January 7, 2011.

42. Wada discloses coding techniques that optionally apply a restoration filter to improve the quality of a reference image used for prediction. Ex. 1004, Abstract; ¶¶17-19, 91-98, 127. Wada's "Seventh Embodiment" is particularly relevant. *Id.*, ¶¶256-272. As part of that embodiment, Figure 46 (below) shows encoding apparatus 7000 which includes prediction image generator 804 (blue).

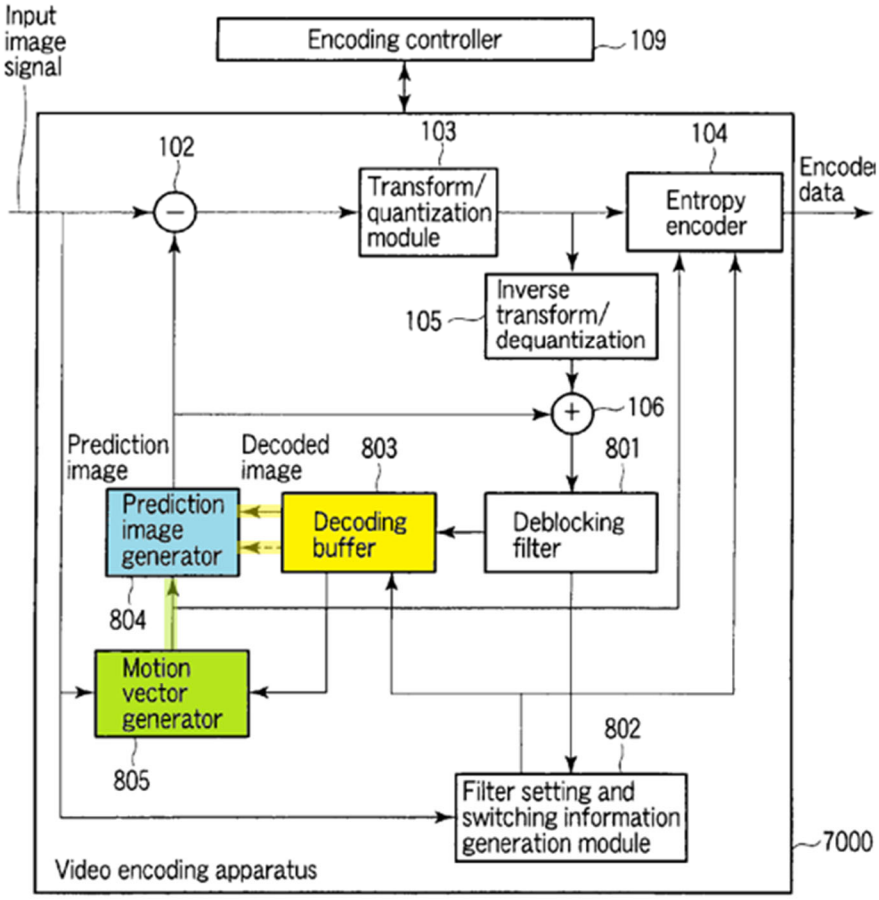


FIG. 46

Ex. 1004, Fig. 46. "[P]rediction image generator 804 uses … motion vector information generated with the motion vector generator 805, and generates the motion-compensated prediction image." Ex. 1004, ¶[0262]. Figure 47 of Wada shows a corresponding *decoding* apparatus 8000 with corresponding prediction image generator 814. Ex. 1004, ¶266; Fig. 47.

43. Wada's Figure 49 shows a "concrete implementation of the prediction image generators" from Figures 46 and 47. Ex. 1004, ¶269.

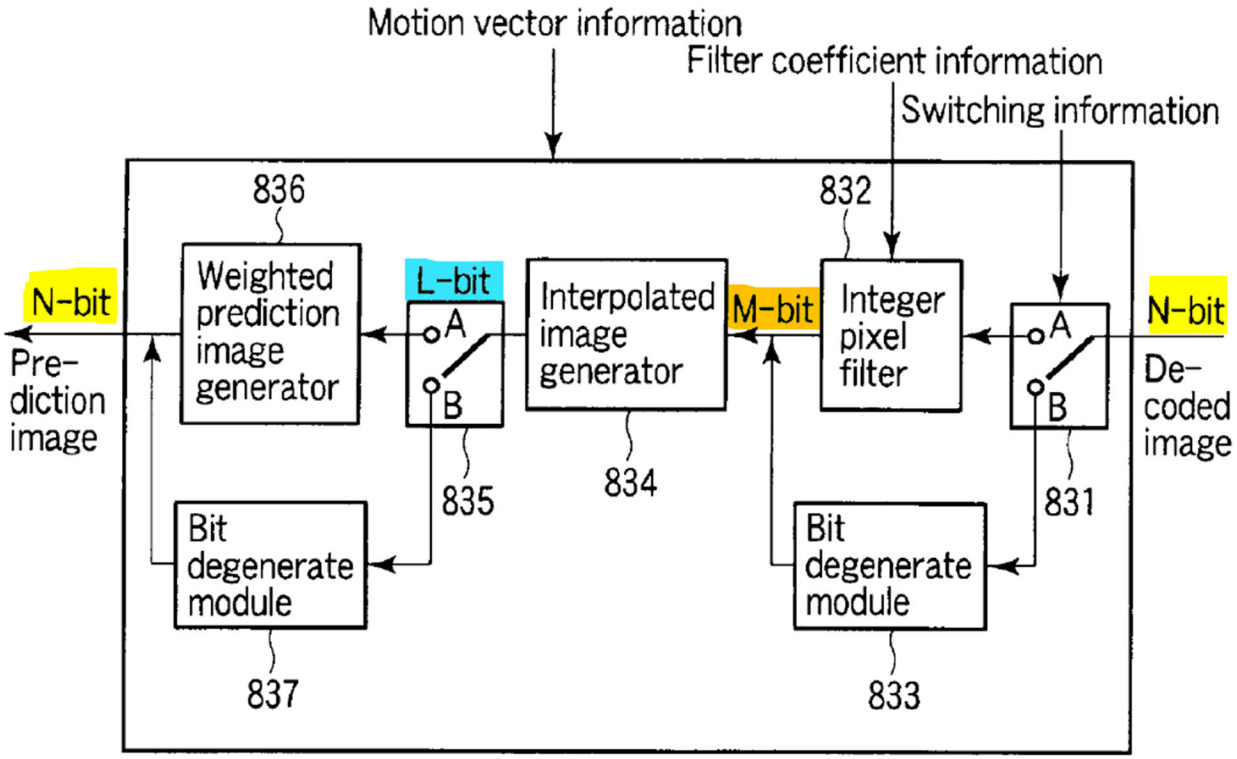


FIG. 49

Id., Fig. 49. The figure shows the pixel bit length of the decoded image input at the right is N bits but intermediate image data has increased precision of M-and L-bits before precision is ultimately decreased back to N bits for the outputted prediction image at the left. The increase from N to M bits can be performed by bit extension module 833 which applies a left-shift. Ex. 1004, ¶270. "[T]he weighted prediction image generator 836 generates a prediction image based on a weighted prediction equation given in … AVC …." *Id.*, ¶271. A POSITA would have understood AVC to refer to Advanced Video Coding which is another name used for the H.264 standard. *See, e.g.,* Ex. 1011 (titled "Overview of the H.264/AVC Video Coding Standard").

44. In my opinion, Wada is in the same field of endeavor as the '267 patent (video encoding/decoding). *See supra* §III. Similar to the '267 patent, Wada is directed to motion prediction techniques. *See, e.g.*, Ex. 1004, Abstract ("A video encoding method wherein an encoded image is used as a reference image for prediction of an image to be encoded next …."); ¶262 (discussing "prediction image generator 804 which "generates the motion-compensated prediction image"); Fig. 49 (showing prediction image generator). Wada references the same H.264/AVC standard as the '267 patent for video encoding and decoding. *Compare* Ex. 1004, ¶271 *with* Ex. 1001, 2:60-3:11, 9:26-41.

2. U.S. Patent Application Publication No. 2011/0007799 ("Karczewicz-I") (Ex. 1005 & Ex. 1006)

45. I have reviewed the Karczewicz-I reference. I understand that Karczewicz-I was not cited or considered during prosecution of the '267 Patent, based primarily on the fact that Karczewicz-I is not cited on the face of the patent, nor have I seen the reference discussed in the prosecution history.

46. Karczewicz-I was filed July 9, 2009 and published January 13, 2011. Therefore, I understand Karczewicz-I is prior art under at least pre-AIA §102(e) because it is a published patent application filed before January 7, 2011.

47. Karczewicz-I teaches block-based techniques for motion prediction. *E.g.*, Ex. 1005, ¶35, ¶44, ¶¶55-60. Karczewicz-I teaches using techniques under known standards, such as H.264, for video encoding and decoding. Ex. 1005, ¶35. Karczewicz-I teaches calculating bi-directional predictions by averaging predictions based on two reference blocks, where the prediction based on each reference block may be calculated using interpolation. *E.g.*, Ex. 1005, ¶41, ¶60.

48. Karczewicz-I incorporates by reference "a submission numbered VCEG-AI33, titled 'High Precision Interpolation and Prediction,' by Yan Ye, Peisong Chen, and Marta Karczewicz and presented at the 35th Meeting in Berlin, Germany, occurring on Jul. 16-18, 2008." Ex. 1005 ("AI33 submission"), ¶37. I understand from counsel that Ex. 1006 is an exact copy obtained from the Patent

Office of the AI33 submission that was actually filed with the patent application that issued as Karczewicz-I when that application was filed on July 9, 2009.

49. In my opinion, Karczewicz-I is in the same field of endeavor as the '267 patent (video encoding/decoding). *See supra* §III. Karczewicz-I is directed to "video encoding techniques that use bi-directional prediction." Ex. 1005, ¶2. Similar to the '267 patent, Karczewicz-I is directed to motion prediction techniques. *See, e.g.*, Ex. 1005, ¶53. Karczewicz-I teaches using the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex. 1005, ¶¶35-37 *with* Ex. 1001, 2:60-3:11, 9:26-41.

3. Motivation to Combine and Reasonable Expectation of Success

50. In my opinion, a POSITA would have been motivated to combine the teachings of Wada and Karczewicz-I. First, as I discussed, Wada discloses a prediction image generator that includes a weighted prediction image generator that applies a weighted prediction equation from the H.264/AVC standard to generate weighted prediction images. Karczewicz-I includes and discusses weighted prediction equations for the H.264/AVC standard. Ex. 1005, ¶¶57, 60, 61, 62, 64, 69. Second, beyond the equations, Karczewicz-I describes the different weighted prediction modes in H.264/AVC and how to select between them. *Id.*, ¶¶27, 42-46, 71. In my opinion, these explanations would be relevant and helpful to a POSITA considering Wada's prediction image generator.

51. Third, Karczewicz-I describes improvements to conventional H.264/AVC weighted prediction. For example, Karczewicz-I discloses eliminating rounding adjustments where they reduce prediction accuracy. Ex. 1005, ¶¶22, 26, 47. Karczewicz-I also discloses techniques that permit the use of offsets to adjust the predictive data even where the weight factors are those associated with default or implicit weighted prediction rather than explicit weighted prediction. *Id.*, ¶¶8, 28, 49, 69. Based on Karczewicz-I's teachings, a POSITA would have understood these techniques would improve the quality of the weighted prediction images generated by Wada's prediction image generator.

52. Fourth, Wada discloses, via the AI33 submission, techniques for high-precision interpolation of subpixel values. Ex. 1005, ¶37; Ex. 1006. A POSITA would have appreciated that the use of these interpolation techniques would generate more accurate interpolated subpixels which in turn would reduce the size of the residual data and improve coding efficiency. *See supra* §I.C.

53. Additionally, in my opinion a POSITA would have had a reasonable expectation of success in combining the teachings of Wada and Karczewicz-I in the way I have discussed. Wada suggests using weighted prediction equations from the AVC/H.264 standard and Karczewicz-I describes such equations. The use of Karczewicz-I's weighted prediction equations is consistent with Wada's teachings and would not have changed the principle of operation of either reference.

54. I discuss the motivation for additionally combining the Srinivasan reference below in addressing dependent claims 3, 9, and 15.

4. Encoder Claims

a. Independent Claims 1, 7, and 13

[1a]	**A method for encoding a block of pixels, the method comprising:**

55. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [1a].

56. Wada teaches a method for encoding blocks of pixels, for example, in connection with encoding apparatus 7000 in Fig. 46. Ex. 1004, ¶¶256-262. Wada's encoding method uses a prediction image generator to generate motion-compensated prediction images. *Id.*, ¶262. Wada describes encoding and decoding macroblocks which are "block[s] of pixels." For example, Wada discloses a syntax structure including "a macroblock level syntax (1907) [that] recites transform coefficient data, prediction mode information and a motion vector which are needed for every macroblock." *Id.*, ¶102, Fig. 31.

57. Similar to Wada, Karczewicz-I describes encoding methods that use inter-prediction to encode blocks of pixels within video frames. Ex. 1004, ¶8 ("In bi-directional prediction, a video block may be predictively encoded and decoded

based on two different lists of predictive reference data."); *see also id.*, ¶52 ("As shown in FIG. 2, video encoder 50 receives a current video block within a video frame or slice to be encoded.").

[7a]	**An apparatus for encoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:**

58. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [7a].

59. Wada discloses video encoding apparatus 1000 which "can be realized using general-purpose computer equipment." Ex. 1004, ¶132. Wada further explains that the various encoder blocks "can be realized by making a processor mounted on the computer equipment execute a program" after, for example, "installing the program in the computer equipment beforehand." *Id.* By the same token, a POSITA would have understood from Wada that video encoding apparatus 7000 of Fig. 46 could likewise be implemented with conventional computer technology. By the time of the alleged invention, a POSITA would have been well-aware that computers with processors, memory, and program code may be used to encode video. At least by 2005, there was reference encoder and

decoder software for the H.264 standard. *See* Ex. 1014. Similar to Wada, Karczewicz-I teaches that the described coding techniques can be implemented using processors and software stored in memory. Ex. 1004, ¶¶98-101.

[13a]	**A computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:**

60. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [13a].

61. As I discussed for preamble [7a], both Wada and Karczewicz-I teach computer-based encoders that execute installed software. Karczewicz-I specifically describes "a computer-readable medium comprising instructions that, when executed in a processor, performs" the described encoding methods. Ex. 1005, ¶99.

[1b], [7b], [13b]	**[determining/determine], for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;**

62. In my opinion, Ground 1 teaches or suggests [1b], [7b], and [13b].

63. Wada discloses an encoder 7000 with a decoded image buffer 803 for storing local decoded images and a motion vector generator 805. Ex. 1004, ¶¶261-262. Prediction image generator 804 uses the local decoded images and the motion vector information from motion vector generator 805 to generate motion-compensated prediction images. Ex. 1004, ¶262.

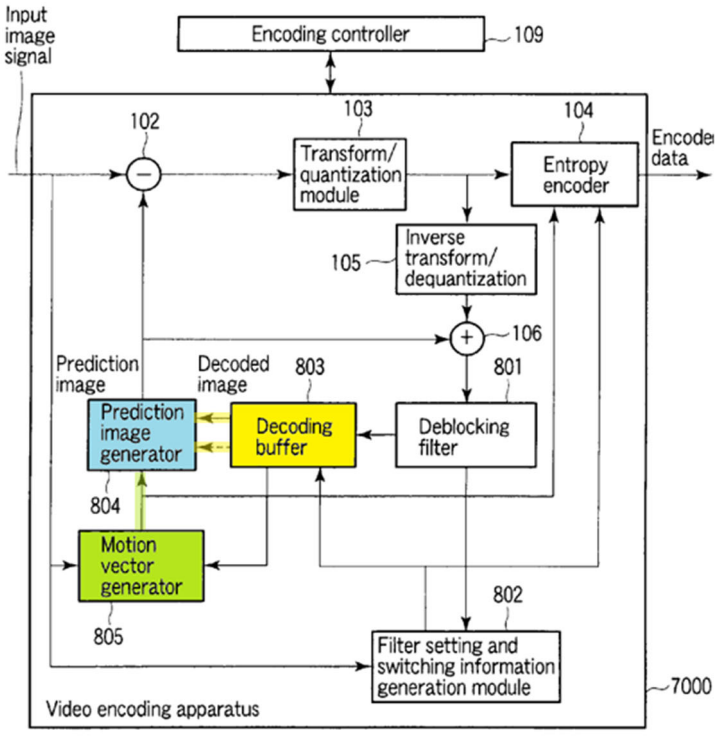


FIG. 46

Id., Fig. 46.

64. Wada discloses that the pixel bit depth of the samples from the decoded image data is N bits. Ex. 1004, ¶270 ("decoded image of N-bit").

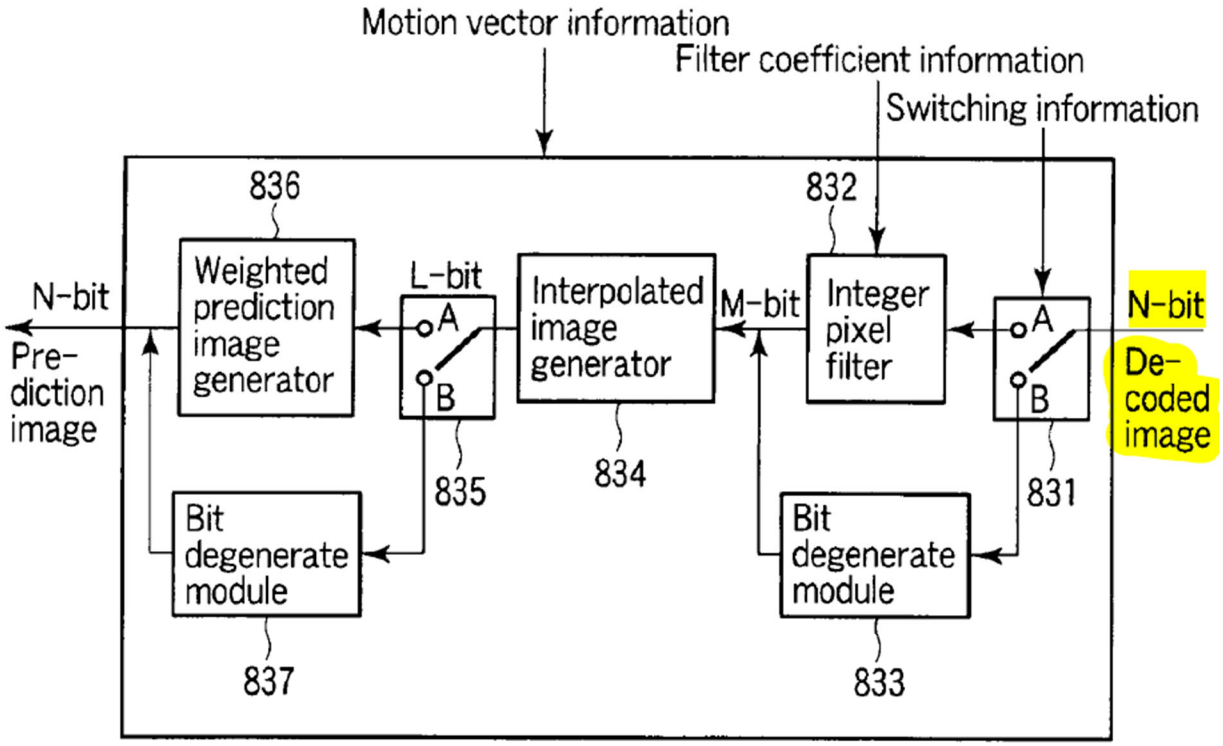


FIG. 49

Id., Fig. 49. Wada's pixel bit length of N bits indicates the number of bits needed to represent the possible values in the decoded image and is a "first precision."

65. Although Wada does not include a detailed discussion of bipredication,[1] Karczewicz-I does. Karczewicz-I explains that "blocks in a B picture may be predicted" using "motion compensated prediction from the combination of [two] reference pictures" one from "list 0" and one from "list 1." Ex. 1005, ¶58.

[1] As discussed, bipredication was well-known in the art and Wada does reference B pictures (Ex. 1004, ¶235) which a POSITA would know use bipredication. *Supra* §1.B.

Karczewicz-I also explains that motion compensation involves "generating the predictive block based on the motion vector determined by motion estimation." *Id.*, ¶53. A POSITA would have known that in bidirectional motion compensated prediction the blocks of reference pixels used to generate the combined prediction would either be integer pixels from the reference picture or subpixels interpolated based on integer pixels in the reference picture. *Supra* §I.C.

66. Accordingly, Wada and Karczewicz-I together teach or suggest determining first and second reference blocks based on motion vectors for use in generating a combined prediction of a current block and that the reference blocks have a first precision of N bits.

[1c], [7c], [13c]	**[using/use] said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;**
[1d], [7d], [13d]	**[using/use] said second reference block to obtain a second prediction, said second prediction having the second precision;**

67. In my opinion, Ground 1 teaches or suggests [1c], [7c], and [13c] as well as similar limitations [1d], [7d], and [13d].

68. Wada discloses increasing the pixel bit length of the samples from the decoded image that are used to generate the prediction image. Ex. 1004, ¶270. The pixel bit length is M bits and L bits after the first and second stages of processing, respectively.

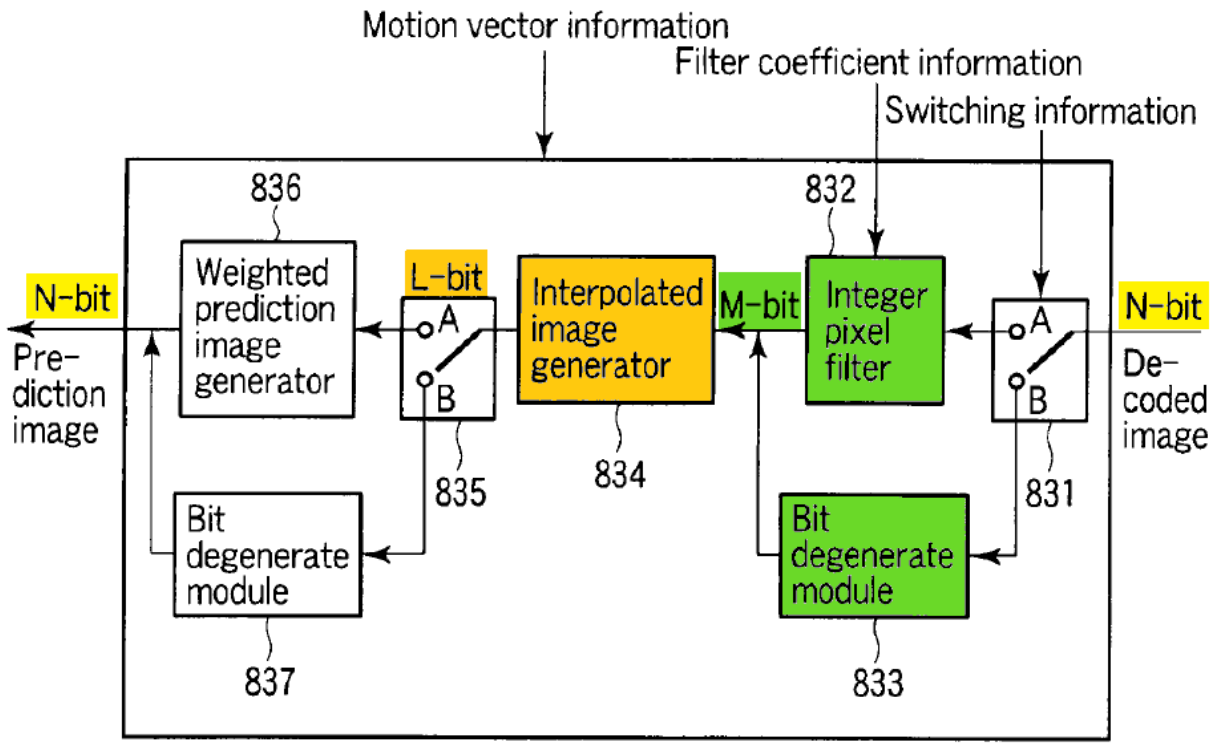


FIG. 49

Ex. 1004, Fig. 49. In the first stage (*i.e.*, the green blocks) The pixel bit length is increased from N bits to M bits using either bit extension module 833 or integer pixel filter 832. Ex. 1004, ¶270. The bit extension module uses a left shift to expand the bit length of the pixel values from the decoded image from N to M bits. *Id.* Regardless of whether integer pixel filter 832 or bit extension module 833 is used, N-bit blocks from a reference picture are used to obtain M-bit prediction values having increased precision.

69. In the second stage (*i.e.*, the orange block), interpolated image generator 834 generates L-bit prediction data. *Id.*, ¶271 ("the output is the pixel bit

length of L-bit wherein L≧N"). A POSITA would have understood that the decoded image data is used to obtain this higher-precision prediction data output by the interpolated image generator. Relatedly, Karczewicz-I describes specific processes for interpolating higher-precision subpixels from integer pixels of a reference image. Ex. 1006.

70. The claimed "first prediction" and "second prediction" could be either the M-bit prediction data (green) output by the first stage of prediction image generation or the L-bit prediction data (orange) output by the second stage of prediction image generation.

71. As I discussed for [1b]/[7b]/[13b], Karczewicz-I discloses predicting B blocks using pixels from two reference pictures. Thus, it would have been obvious to a POSITA for Wada's prediction image generator to generate predictions based on *two reference blocks* as claimed.

[1e], [7e], [13e]	**[obtaining/obtain] a combined prediction based at least partly upon said first prediction and said second prediction;**

72. In my opinion, Ground 1 teaches or suggests [1e], [7e], and [13e].

73. Wada's prediction image generator includes weighted prediction image generator 836 that "generates a prediction image based on a weighted prediction equation given in H.262/AVC, etc." Ex. 1004, ¶271.

74. Karczewicz-I discloses H.264/AVC weighted prediction formulas for predicting B video blocks. For example, Karczewicz-I discloses the following equation for bidirectional default weighted prediction:

pred(i,j) = (pred0(i,j) + pred1(i,j) + 1) >> 1

Ex. 1005, ¶60. Karczewicz-I notes that in the above formula "pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1." *Id.* Accordingly, pred(i,j) is a combined prediction based on two predictions—pred0(i,j) and pred1(i,j).

75. As I discussed, it would have been obvious to a POSITA for Wada's prediction image generator to predict B video blocks by combining predictions obtained using reference blocks from two reference pictures consistent with Karczewicz-I's AVC-based weighted prediction equations and how biprediction was conventionally performed prior to the alleged invention. *Supra* §V.A.3.

76. As I discussed for the preceding limitations, the claimed "first prediction" and "second prediction" could be the L-bit prediction data output from Wada's interpolated image generator 834. Wada's Figure 49 shows that data feeding directly into weighted prediction image generator 836. As I also discussed, the claimed "first prediction" and "second prediction" could alternatively be the M-bit output of integer pixel filter 832 and/or bit extension module 833. Figure 49 shows that the L-bit precision data that flows into weighted prediction generator 836 is based on the M-bit precision data output from integer pixel filter 832 and bit

extension module 833. Accordingly, it would have been obvious to a POSITA that the combined prediction would be based "at least partly" on the M-bit precision data as well.

[1f], [7f], [13f]	**[decreasing/decrease] a precision of said combined prediction by shifting bits of the combined prediction to the right; and**

77. In my opinion, Ground 1 teaches or suggests [1f], [7f], and [13f].

78. Wada discloses decreasing the precision of the generated prediction image. Ex. 1004, ¶271. Specifically, the weighted prediction image generator 836 outputs a prediction image having a pixel bit depth of N-bits (which matches the original pixel bit depth of the decoded image) even though the input to the weighted prediction image generator has a higher precision of L-bits.

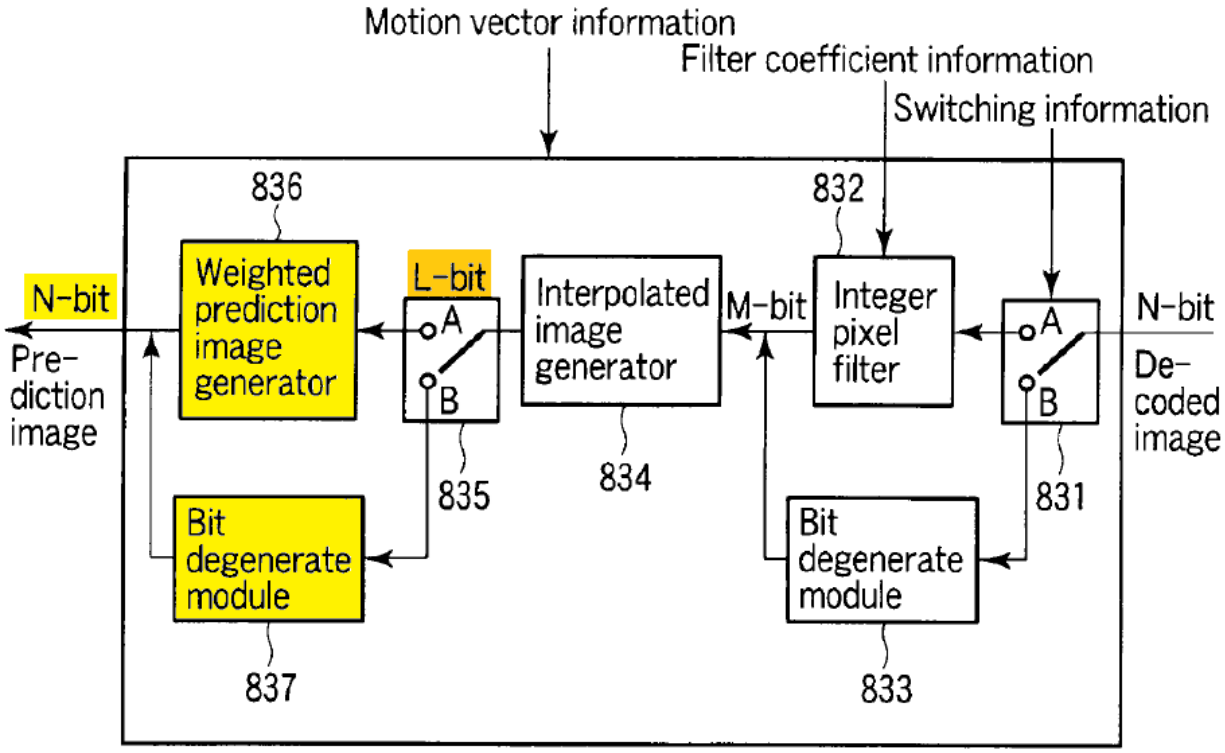


FIG. 49

Ex. 1004, Fig. 49; *see also id.*, ¶271("[T]he processing is done so that the output of the pixel bit length of N-bit is obtained with respect to the input of the pixel bit length of L-bit.").

79. As I have discussed, it would have been obvious to a POSITA for Wada's prediction image generator to generate predictions for B video blocks in accordance with Karczewicz-I's AVC-based equations. *Supra* §V.A.3. Where, as Wada discloses, the predictions input to the weighted prediction image generator have increased precision of L-bits as opposed to N-bits, the combined prediction

obtained by averaging those values using Karczewicz-I's default weighted prediction equation would have the same increased precision of L bits. To obtain an N-bit prediction, the precision of the L-bit prediction would have to be decreased.

80. For example, if N were 8 bits and L were 16 bits, the default weighted prediction obtained by averaging two 16-bit prediction values using Karczewicz-I's equation would be a 16-bit value. It would have been obvious to a POSITA to reduce the precision of that 16-bit prediction to obtain an 8-bit (*i.e.*, N-bit) prediction as Wada discloses. Notably, the other branch in the third stage of Figure 49 simply rounds the L-bit value down to N-bits using bit degenerate module 837. Ex. 1004, ¶271 ("the bit degenerate module 847 performs rounding processing so that the input of L-bit is N-bit wherein L≧N"). This also would have suggested to a POSITA that weighted prediction image generator 836 would similarly decrease the precision of the L-bit prediction data to N-bits.

81. It also would have been obvious to a POSITA in my opinion to use a right-shift to decrease the precision of the combined prediction from L-to N-bits. First, a POSITA would have been familiar with using left-and right-shifts to increase and decrease precision, respectively. *Supra* §I.D. Second, Karczewicz-1, though the AI33 submission, uses right-shifts to reduce the precision of higher-precision interpolated subpixels to standard 8-bit precision. *See* Ex. 1006, 1

(quarter-pixel interpolation equations use right-shifts of 6 bits to get to 8-bit precision). Third, in describing the prediction image generator in Fig. 49, Wada discloses using a left-shift to increase pixel bit length. Ex. 1004, ¶270 (describing bit extension module 833). Using a right-shift to decrease precision would have been consistent with Wada's approach for increasing precision in the same prediction image generator. Fourth, Karczewicz's default sample weighted prediction formula discussed above already includes a right-shift. Ex. 1005, ¶60. It would have been obvious to a POSITA to reduce the precision of the prediction image back to N-bits (as Wada discloses) by simply increasing the extent of the right shift (by L – N bits).

[1g], [7g], [13g]	**[encoding/encode] residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.**

82. In my opinion, Ground 1 teaches or suggests [1g], [7g], and [13g].

83. Wada's encoding apparatus 7000 includes subtracter 102 which "generates a prediction error signal between the generated prediction image and the input image." Ex. 1004, ¶262. That "prediction error signal" reflects a difference between the actual block and the predicted block and is thus residual data. *Id.*, ¶79 ("The subtracter 102 calculates a difference between an acquired input image signal 10 and the prediction image signal 11 to output a prediction error image signal 12.").

84. Fig. 46 shows that the residual output from subtracter 102 is transformed and quantized by block 103, entropy coded by block 104, and then output as encoded data:

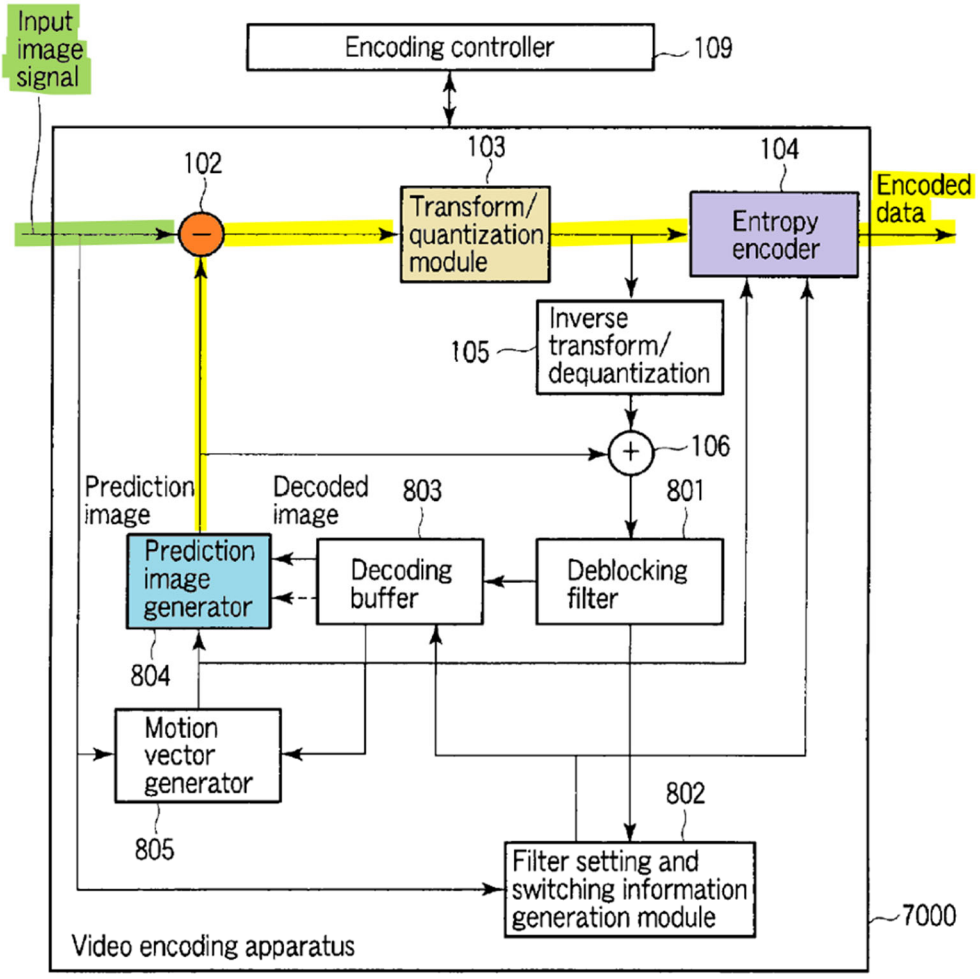


FIG. 46

Id., Fig. 46; *see also id.*, ¶¶79, 89 (describing 103), ¶82 (describing 104), ¶100 (quantized transform coefficient 13 output from module 103 is multiplexed into bit stream transmitted to decoder), Fig. 1 (similar encoder with same components).

b. Dependent Claims 2, 8, and 14

2. The method according to claim 1, **8. The apparatus according to claim 7,** **14. The computer program product according to claim 13,** **wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.**

85. In my opinion, Ground 1 renders obvious claims 2, 8, and 14.

86. Wada's prediction image generator 804 includes "interpolation picture generator 834 [which] generates an interpolation picture of a fractional pixel position as a prediction image, based on the motion vector." Ex. 1004, ¶¶268, 271.

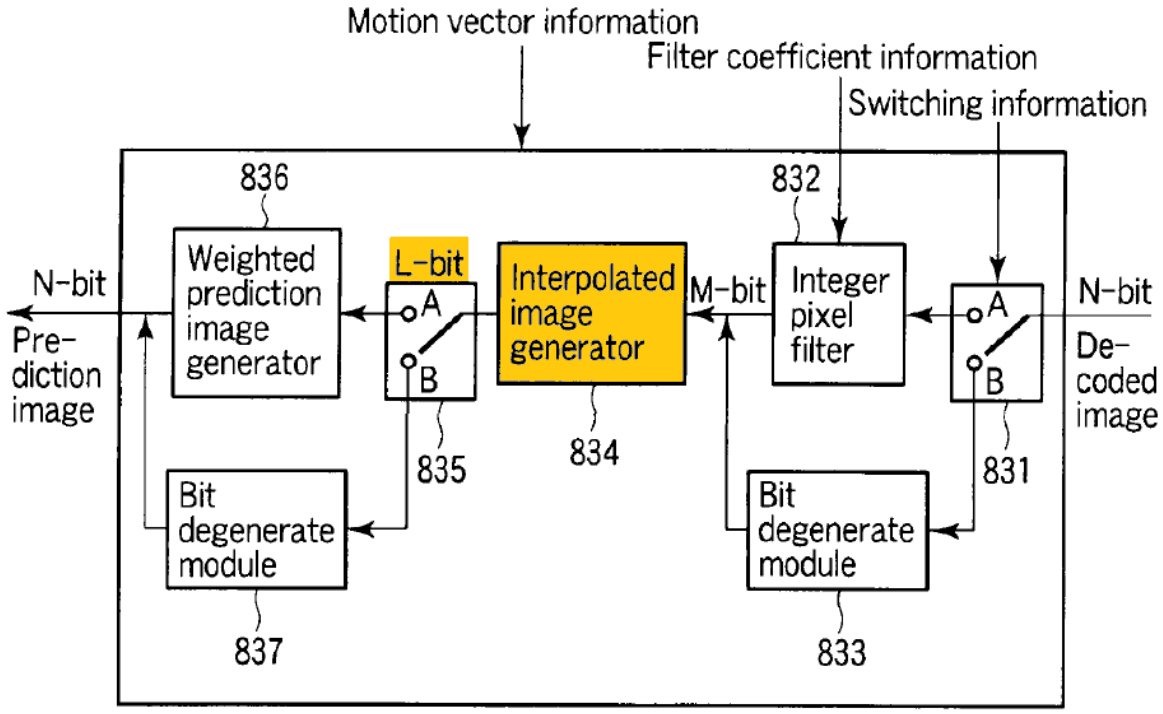


FIG. 49

Id., Fig. 49. As shown, the output of interpolated image generator 834 feeds into weighted prediction image generator 836. A POSITA would have understood that the reference blocks used to generate the weighted predictions could be interpolated where the motion vector points to a subpixel or fractional pixel location. As I noted, subpixel interpolation was well-understood by the time of the alleged invention. *Supra* §I.C.

87. Additionally, Karczewicz-I, through the AI33 submission, discloses high precision techniques for interpolating subpixels at half-and quarter-pixel positions. Ex. 1006. The interpolation calculations use integer pixel values which are shaded and labeled in Fig. 1 (below) with upper-case letters:

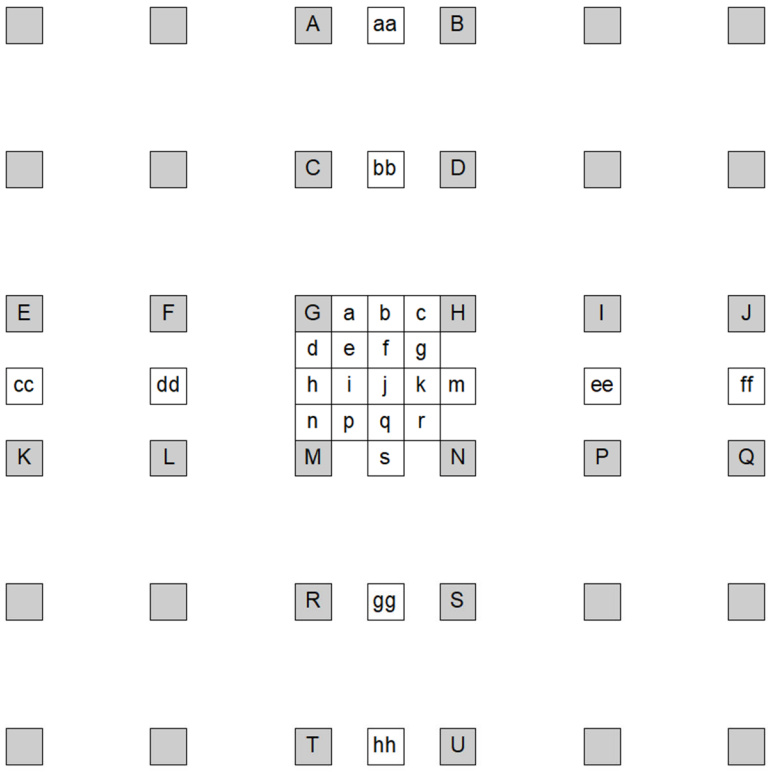


Figure 1: Integer pixels (shaded blocks with upper-case letters) and fractional pixel positions (white blocks with lower-case letters).

Id., Fig. 1; *see also id.*, 1 (interpolation equations).

88. Accordingly, in my opinion it would have been obvious to a POSITA that where a motion vector points to a subpixel (as was well-known), the prediction would be obtained by interpolation using the block of integer pixel values from the reference image (*i.e.,* "pixel values of said first reference block") indicated by the motion vector.

c. Dependent Claims 3, 9, and 15

3. The method according to claim 2, **9. The apparatus according to claim 8,** **15. The computer program product according to claim 14,** **wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.**

(i) Ground 1

89. In my opinion, Ground 1 renders obvious claims 3, 9, and 15 for two reasons.

90. First, Wada discloses that the input to interpolated image generator 834 is M-bits which is greater than N-bits. Ex. 1004, ¶270; Fig. 49. Karczewicz-I's AI33 submission discloses that applying a 6-tap filter to 8-bit samples results in intermediate values for half pixels that require "15 bits plus sign (15s)." Ex. 1006, 3. In my opinion, it would have been obvious to a POSITA that applying a 6-tap filter to higher-precision (*e.g.,* M = 10 or 12 bits) samples would yield intermediate

values for half-pixels requiring more bits and that this would have posed a problem for averaging with other samples using 16-bit arithmetic. Karczewicz-I addresses a similar situation involving the half-pixel in the j position where the intermediate value j1 generated from the P-tap filter requires 20 bits. Ex. 1006, 3. Specifically, Karczewicz-I discloses right-shifting the sum of the P-tap filter by 4 bits to obtain a 16-bit representation of j1 to enable that value to be averaged with other intermediate values using 16-bit arithmetic:

r1 = r1 + r3	r1 is ((y0+y5)-5*(y1+y4)+20*(y2+y3))>>4	-9977	26232	16s
j1 = r1	**store result**			

Ex. 1006, 4. From these teachings, it would have been obvious to a POSITA to apply the same straightforward right-shift step to intermediate values obtained from applying a P-tap filter to higher-resolution (M-bit) samples where they exceed 16 bits.

91. Second, Karczewicz-I's disclosures regarding quarter-pixel interpolation also render these claims obvious in my opinion. Karczewicz-I's quarter-pixel interpolation calculations right-shift a sum of a P-tap filter. For example, Karczewicz-I discloses interpolating quarter-pixel 'a' by summing 'b1' with integer pixel 'G' (left-shifted by 5 bits) together with a constant rounding adjustment of 32, which a POSITA would have recognized as a P-tap filter.

$$a = (G << 5 + b1 + 32) >> 6$$

Ex. 1006 (AI33 submission), 1. A POSITA would have known that the H.264 equations for interpolating quarter-pixels use a 2-tap filter. *See, e.g.*, Ex. 1019 (Wang), ¶36 (describing H.264: "All quarter-pixel locations undergo 2-tap average filtering using integer and half-pixels."); Ex. 1020 (Cordeiro), 246 ("All quarter-pixel values are computed by a 2-tap average filtering using both integer and half-pixels."), 247 (equation [2]).

92. The integer pixels (*e.g.,* 'G') used to interpolate quarter-pixels are values of the first reference block as claimed. Additionally, the intermediate values (*e.g.,* 'b1') are themselves "derived by applying [a] 6-tap filter … horizontally" using integer pixel values E, F, G, H, I, and J (which, again, are "values of [the] first reference block"). *Id.*, Fig. 1. The sum of b1 and two other addends (G<<5 and rounding adjustment 32) is right-shifted by six bits.

(ii) Ground 1A

93. In my opinion, Ground 1A also renders obvious claims 3, 9, and 15.

94. Srinivasan was published and publicly available no later than September 13, 2005 and therefore qualifies as prior art under §102(b). Ex. 1026 (Hall-Ellis declaration); Ex. 1008.

95. Srinivasan describes how VC-1 performs sub-pixel interpolation in two dimensions. Ex. 1008, 000015. In my opinion, a POSITA would have known that sub-pixel interpolation in two dimensions is used to interpolate a subpixel in ½

pixel horizontal and ½ pixel vertical location (*e.g.,* the j location in Karczewicz-I's AI33 submission). Srinivasan discloses that "[f]irst, filtering is performed in the vertical direction" using a four-tap filter. Ex. 1008, 000015. Next, "a rounding factor is added to the result and some bits are shifted out." *Id.* In other words, a right-shift is performed so that "the intermediate values … are within 16 bits." *Id.* Srinivasan explains that "[i]t is possible to achieve additional accuracy by retaining a higher precision result" (*i.e.,* higher than 8 bits) "after the first stage of filtering." *Id.* This first stage of Srinivasan's two-stage interpolation process right-shifts a sum of a P-tap filter using integer pixel values from the reference picture, *i.e.,* "using values of said first reference block" as claimed. Then, the partially-rounded intermediate results from the vertical filtering are filtered in the horizontal direction to obtain a result which is rounded and clipped to obtain the final interpolated value. *Id.* Srinivasan is analogous art in the same field as the '267 patent (*supra* §III) because it is generally directed to video encoding/decoding and discusses motion compensation and subpixel interpolation. *See* Ex. 1008, 000014-15.

96. It would have been obvious to a POSITA in my opinion to combine Srinivasan's teachings regarding high-precision sub-pixel interpolation in two dimensions with the teachings of Wada and Karczewicz-I. As I discussed for Ground 1, a POSITA would have understood that applying a P-tap filter to

higher-precision (M-bit) pixel values as Wada discloses could result in intermediate values for interpolated half-pixels that exceed 16 bits. Srinivasan describes partially rounding such intermediate values to allow for a 16-bit implementation while still obtaining the benefits of more accurate sub-pixel interpolation flowing from the increased precision of the intermediate values. Although Karczewicz-I's AI33 submission discloses a similar technique that involves partially rounding intermediate values for interpolated subpixels, a POSITA might have opted for Srinivasan's approach in a given implementation, for example, because it uses a 4-tap filter rather than a 6-tap filter which would have reduced the number of calculations during interpolation.

97. A POSITA also would have had a reasonable expectation of success in combining the teachings of Srinivasan with those of Wada and Karczewicz-I in my opinion. Srinivasan's technique of right-shifting intermediate values to reduce them to 16-bit precision was straightforward and it would have been well-within the level of ordinary skill to apply similar right-shifts to intermediate values obtained from the sum of a P-tap filter applied to Wada's higher-resolution (M-bit) samples. Although Srinivasan describes VC-1, the Petition does not propose to incorporate any aspect of VC-1 beyond the high-precision sub-pixel interpolation technique described by Srinivasan.

d. Dependent Claims 4, 10, and 16

4. The method according to claim 2,
10. The apparatus according to claim 8,
16. The computer program product according to claim 14,

wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

98. In my opinion, Ground 1 renders obvious claims 4, 10, and 16.

99. First, Wada discloses that prediction image generator 804 includes bit extension module 833 which left-shifts values from the decoded image to increase their pixel bit depth from N to M bits. Ex. 1005, ¶270, Fig. 49. The M-bit samples are fed into interpolated image generator 834 and then to the weighted prediction image generator 836. *Id.*, Fig. 49. It would have been obvious to a POSITA that where the motion vector points to reference pixels at integer positions, the prediction would not rely on interpolated subpixels but simply use the integer pixel values from the decoded image having increased precision owing to the left-shift applied by bit extension module 833.

100. Second, Karczewicz-I, via the AI33 submission, discloses left-shifting integer pixel values (*e.g.,* G, H, M) before averaging them with 15-bit interpolated half-pixel values (*e.g.,* b1, h1).

```
a = (G<<5+b1+32)>>6
c = (H<<5+b1+32)>>6
d = (G<<5+h1+32)>>6
n = (M<<5+h1+32)>>6
```

Ex. 1006, 1; *id.*, 3 (noting b1 and h1 require 15 bits). Because, as I discussed, biprediction involves similar averaging of pixel values that can be represented with different precisions, it would have been obvious to a POSITA to left shift the integer pixel values in order to average them with higher-precision interpolated subpixel values in the same way Karczewicz-I describes.

101. In my opinion, these two teachings, whether taken separately or together, render obvious claims 4, 10, and 16.

e. Dependent Claims 5, 11, and 17

5. The method according to claim 1, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises,

11. The apparatus according to claim 7, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by,

17. The computer program product according to claim 13, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to,

[inserting/insert] a rounding offset to the combined prediction before said decreasing.

102. Karczewicz-I discloses adding a rounding offset to the combined prediction before the right shift in computing a bidirectional default weighted prediction:

pred(i,j) = (pred0(i,j) + pred1(i,j) + 1) >> 1

Ex. 1005, ¶60 (highlighting of rounding offset added for emphasis); *see also id.*, ¶55 (explaining that weighted prediction in AVC involves adding a "rounding adjustment" of 1 "to the addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit"). These rounding adjustments are "rounding offsets" as claimed.

103. As I discussed for limitations [1f], [7f], and [13f], it would have been obvious to a POSITA to increase the right shift by M – N bits in Karczewicz-I's weighted prediction calculations where higher precision sample values are used as Wada discloses. Increasing the right-shift would impact the rounding adjustment used. *See* Ex. 1005, ¶63 ("Generally, a rounding adjustment of 2^{r-1} is commonly used prior to a right shift by r, where r represents a positive integer.").

f. Dependent Claims 6, 12, and 18

6. The method according to claim 1, **12. The apparatus according to claim 7,** **18. The computer program product according to claim 13,** **wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.**

104. As I discussed for [1b]/[7b]/[13b], the "first precision" in Wada is N bits. Wada discloses that N bits are needed to represent the values of the pixels in the decoded images. Ex. 1004, ¶270 ("the pixel bit length of the decoded image is N bits"); Fig. 49.

105. As I discussed for [1c]/[7c]/[13c] and [1d]/[7d]/[13d], the claimed "first prediction" and "second prediction" could be the L-bit prediction data output from Wada's interpolated image generator 834 in which case the "second precision" is L bits. A POSITA would have understood from Wada that L bits is the number of bits needed to represent the output of interpolated image generator 834. *Id.*, ¶271 (the input to interpolation picture generator 834 "is the pixel bit length of M-bit whereas the output is the pixel bit length of L-bit wherein L≧N"). Further, as I have discussed, Karczewicz-I shows how subpixel interpolation can generate higher-precision values for interpolated subpixels. Ex. 1006.

106. Alternatively, as I also discussed for [1c]/[7c]/[13c] and [1d]/[7d]/[13d], the claimed "first prediction" and "second prediction" could be the M-bit output of integer pixel filter 832 and/or bit extension module 833 in which case the "second precision" is M bits. Wada suggests M bits are needed to represent these values. Ex. 1004, ¶270 ("the pixel bit length of the output of the integer pixel filter 832 [is] M-bit wherein M≧N"); *id.* ("[T]he decoded image is input to the bit extension module 833, so that the decoded image of N-bit is expanded to be M-bit wherein M≧N.").

5. Decoder Claims

107. The decoder-side claims of the '267 patent recite almost identical limitations to the encoder-side claims I discussed above. Moreover, Wada

discloses that the same prediction image generator depicted in Fig. 49—which was the focus of the discussion above for the encoder-side claims—is used both for encoding and decoding. Ex. 1004, ¶269. Similarly, Karczewicz-I discloses that the decoder performs "the reciprocal decoding techniques to the encoding techniques described" for the encoder including support for weighted prediction. Ex. 1005, ¶83, *see also id.*, ¶¶84-86, 88-89 (discussing decoding including block based predictive decoding using weighted prediction data); ¶44 ("For different types of weighted bi-directional prediction in accordance with ITU-T H.264, … video *decoder* 28 may generally support three different types of prediction modes.").

108. Given the high degree of similarity in the claims, my discussion below for the decoder-side claims refers back to my detailed discussion above of similar limitations of the encoder-side claims.

a. Independent Claims 19, 25, and 31

[19a]	**A method for decoding a block of pixels, the method comprising:**

109. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [19a].

110. Wada teaches methods for decoding blocks of pixels, for example, in connection with decoding apparatus 8000 in Figure 47. Ex. 1004, ¶¶263-266.

These decoding methods use a prediction image generator that generates motion-compensated prediction images. *Id.*, ¶266. A POSITA would have understood Wada to disclose decoding methods that process blocks of pixels. *See id.*, ¶102 (describing "a macroblock level syntax (1907) [that] recites transform coefficient data, prediction mode information and a motion vector which are needed for every macroblock"); Fig. 31.

111. Similar to Wada, Karczewicz-I describes methods for decoding blocks of pixels. Ex. 1005, ¶39 ("Video encoder 22 and video decoder 28 may operate on video blocks within individual video frames in order to encode and decode the video data."); ¶8 ("In bi-directional prediction, a video block may be predictively encoded and decoded based on two different lists of predictive reference data.").

[25a]	**An apparatus for decoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:**

112. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [25a].

113. Wada discloses video decoding apparatus 2000 which "can be realized using general-purpose computer equipment." Ex. 1004, ¶132. Wada

further explains that the various decoder blocks "can be realized by making a processor mounted on the computer equipment execute a program" after, for example, "installing the program in the computer equipment beforehand." *Id.* Similarly, a POSITA would have understood from Wada that video decoding apparatus 8000 of Figure 47 could likewise be implemented with conventional computer technology. By the time of the alleged invention, a POSITA would have been well-aware that computers with processors, memory, and program code may be used to decode video. For example, as I mentioned above, reference decoder software was available for H.264 by 2005. *See* Ex. 1014. Similar to Wada, Karczewicz-I discloses that its described decoding techniques can be implemented using processors and software stored in memory. Ex. 1005, ¶¶98-101.

[31a]	**A computer program product for decoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:**

114. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Wada and Karczewicz-I teaches the preamble and any additional limitations of element [31a].

115. As I just discussed for preamble [25a], both Wada and Karczewicz-I teach computer-based decoders that execute installed software. Karczewicz-I specifically describes "a computer-readable medium comprising instructions that, when executed in a processor, performs" the described decoding methods. Ex. 1005, ¶99.

[19b], [25b], [31b]	**[determining/determine], for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;**

116. Limitations [19b], [25b], and [31b] are identical to limitations [1b], [7b], and [13b]. Ground 1 teaches or suggests these limitations as discussed. *Supra* §V.A.4.a.

[19c], [25c], [31c]	**[using/use] said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;**
[19d], [25d], [31d]	**[using/use] said second reference block to obtain a second prediction, said second prediction having the second precision;**

117. Limitations [19c], [25c], and [31c] are identical to limitations [1c], [7c], and [13c] and limitations [19d], [25d], and [31d] are identical to limitations [1d], [7d], and [13d]. Ground 1 teaches or suggests all these limitations as I discussed. *Supra* §V.A.4.a.

[19e], [25e], [31e]	**[obtaining/obtain] a combined prediction based at least partly upon said first prediction and said second prediction;**

118. Limitations [19e], [25e], and [31e] are identical to limitations [1e], [7e], and [13e]. Ground 1 teaches or suggests these limitations as I discussed. *Supra* §V.A.4.a.

[19f], [25f], [31f]	**a precision of said combined prediction by shifting bits of the combined prediction to the right; and**

119. Limitations [19f], [25f], and [31f] are identical to limitations [1f], [7f], and [13f]. Ground 1 teaches or suggests these limitations as I discussed. *Supra* §V.A.4.a.

[19g], [25g], [31g]	**[reconstructing/reconstruct] the block of pixels based on the combined prediction.**

120. Wada's decoding apparatus 8000 includes subtracter 102 which "generates a prediction error signal between the generated prediction image and the input image." Ex. 1004, ¶262; Fig. 47. That "prediction error signal" reflects a difference between the actual block and the predicted block and is thus residual data. *Id.*, ¶79 ("The subtracter 102 calculates a difference between an acquired input image signal 10 and the prediction image signal 11 to output a prediction error image signal 12.").

121. Figure 47 shows that the output of prediction image generator—the combined prediction where the current picture is a B-picture—is combined with the decoded residual, deblocked, and then output:

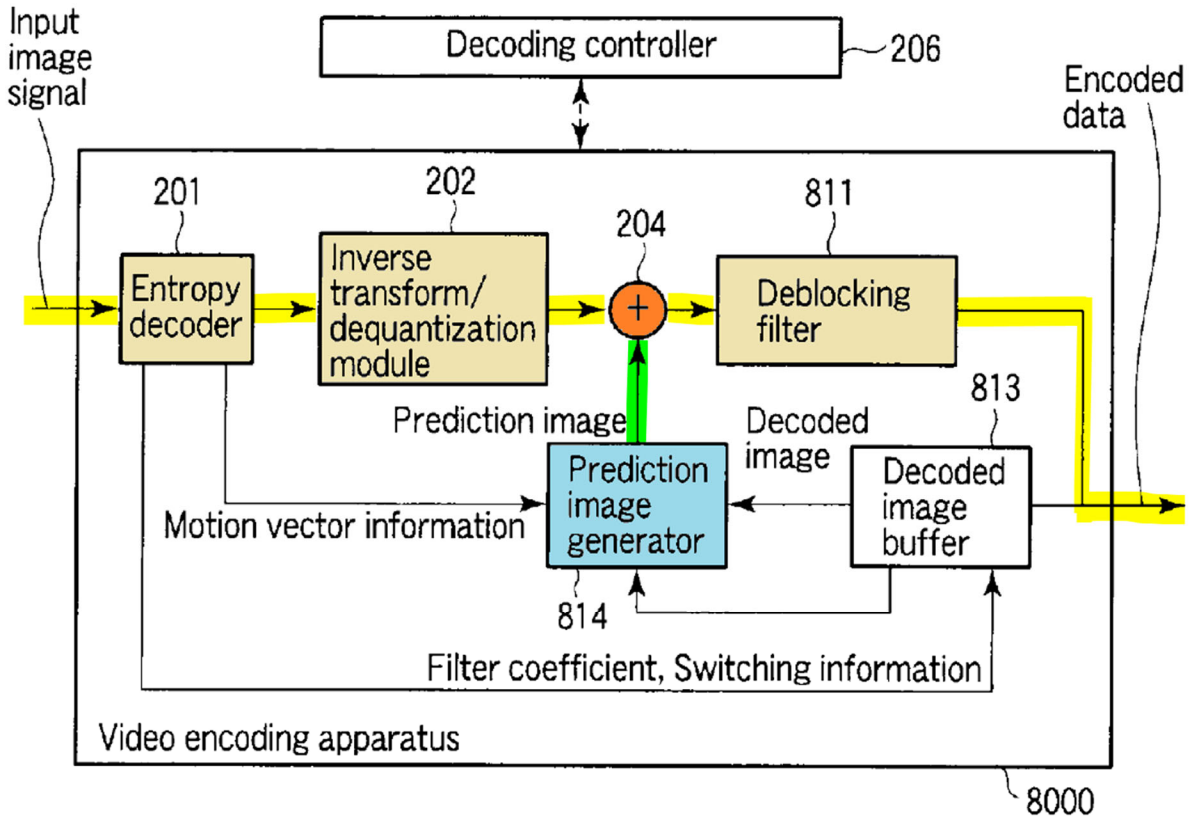


FIG. 47

Id., Fig. 47; *see also id.*, ¶¶263-265 (describing Fig. 47); *see also id.*, ¶¶110 (describing adder 204), 120 ("The prediction error image signal 15 is added to the prediction image signal 11 output from the prediction signal generator 203 with the adder 204, whereby the decoded image signal 21 is generated."), Fig. 5 (similar decoder with same components). Although Figure 47 includes some obvious typos (e.g., "video *en*coding apparatus"), a POSITA would have understood that—

consistent with decoder Figure 5—the input to decoding apparatus 8000 would be encoded data and the output would be an output image signal (i.e., the reconstructed image).

b. Dependent Claims 20, 26, and 32

20. The method according to claim 19,
26. The apparatus according to claim 25,
32. The computer program product according to claim 31,

wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

122. These claims add the same limitations as encoder-side dependent claims 2, 8, and 14 and are obvious for the same reasons in my opinion. *Supra* §V.A.4.b.

c. Dependent Claims 21, 27, and 33

21. The method according to claim 20,
27. The apparatus according to claim 26,
33. The computer program product according to claim 32,

wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.

123. These claims add the same limitations as encoder-side dependent claims 3, 9, and 15 and Grounds 1 and 1A render them obvious for the same reasons in my opinion. *Supra* §V.A.4.c.

d. Dependent Claims 22, 28, and 34

22. The method according to claim 20,

28. The apparatus according to claim 26,

34. The computer program product according to claim 32,

wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

124. These claims add the same limitations as encoder-side dependent claims 4, 10, and 16 and are obvious for the same reasons in my opinion. *Supra* §V.A.4.d.

e. Dependent Claims 23, 29, and 35

23. The method according to claim 19, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises:

29. The apparatus according to claim 25, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by:

35. The computer program product according to claim 31, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:

[inserting/insert] a rounding offset to the combined prediction before said decreasing.

125. These claims add the same limitations as encoder-side dependent claims 5, 11, and 17 and are obvious for the same reasons in my opinion. *Supra* §V.A.4.e.

f. Dependent Claims 24, 30, and 36

24. The method according to claim 19,
30. The apparatus according to claim 25,
36. The computer program product according to claim 31,

wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

126. These claims add the same limitations as encoder-side dependent claims 6, 12, and 18 and are obvious for the same reasons in my opinion. *Supra* §V.A.4.f.

B. Ground 2: Claims 1-36 Are Rendered Obvious by Karczewicz-I in View of Karczewicz-II

127. It is my opinion that a POSITA would have found Claims 1-36 obvious based on the combination of Karczewicz-I and Karczewicz-II.

1. U.S. Patent Application Publication No. 2011/0007799 ("Karczewicz-I")

128. I have reviewed the Karczewicz-I reference. I understand that Karczewicz-I was not cited or considered during prosecution of the '267 Patent, based primarily on the fact that Karczewicz-I is not cited on the face of the patent, nor have I seen the reference discussed in the prosecution history.

129. Karczewicz-I was filed July 9, 2009 and published January 13, 2011. Therefore, I understand Karczewicz-I is prior art under at least pre-AIA §102(e) because it is a published patent application filed before January 7, 2011.

130. Karczewicz-I teaches block-based techniques for motion prediction. *E.g.*, Ex. 1005, ¶35, ¶44, ¶¶55-60. Karczewicz-I teaches using techniques under known standards, such as H.264, for video encoding and decoding. Ex. 1005, ¶35. Karczewicz-I teaches calculating bi-directional predictions by averaging predictions based on two reference blocks, where the prediction based on each reference block may be calculated using interpolation. *E.g.*, Ex. 1005, ¶41, ¶60.

131. Karczewicz-I is in the same field of endeavor as the '267 patent (video encoding/decoding). *See supra* §III. Karczewicz-I is directed to "video encoding techniques that use bi-directional prediction." Ex. 1005, ¶2. Similar to the '267 patent, Karczewicz-I is directed to motion prediction techniques. *See, e.g.*, Ex. 1005, ¶53. Karczewicz-I teaches using the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex. 1005, ¶¶35-37 *with* Ex. 1001, 2:60-3:11, 9:26-41.

2. U.S. Patent Application Publication No. 2009/0257499 Karczewicz ("Karczewicz-II") (Ex. 1007)

132. I have reviewed the Karczewicz-II reference. I understand that Karczewicz-II is cited on the face of the '267 patent. Karczewicz-II was cited in an Information Disclosure Statement filed June 7, 2021 along with 76 other references. Other than acknowledging the Information Disclosure Statement, the Examiner did not discuss Karczewicz-II during the prosecution history or use Karczewicz-II in any rejections.

133. Karczewicz-II was filed April 8, 2009 and published October 15, 2009. Therefore, I understand Karczewicz-II is prior art under at least pre-AIA §§102(a), 102(b) and 102(e) because it was filed and published more than one year before January 7, 2011.

134. Karczewicz-II teaches block-based techniques for motion prediction. *E.g.*, Ex. 1007, ¶8, ¶¶35-36, ¶46, ¶54. Karczewicz-II teaches improved calculations for predictions involving interpolated fractional pixel positions. Ex. 1007, ¶2, ¶10, ¶¶93-106. Karczewicz-II teaches that prediction values for quarter-pixel positions can be calculated as an average of two adjacent integer or half-pixel positions. Ex. 1007, ¶¶96-102. Karczewicz-II teaches such calculation for multiple different scenarios. For example, Karczewicz-II teaches a quarter-pixel position between an integer pixel and a half-pixel position is calculated as an average of the predictions of the integer pixel and the half-pixel position. Ex. 1007, ¶96, ¶99, ¶103, Tables 1, 3, 5. In addition, Karczewicz-II teaches a quarter-pixel position between a center-pixel position (i.e., the position that is at the center of four integer pixels) and a half-pixel position is calculated as an average of predictions for the two positions. Ex. 1007, ¶96, ¶99, ¶108, Tables 1, 3, 8. Moreover, Karczewicz-II teaches a quarter-pixel position between two half-pixel positions is calculated as an average of predictions for the two positions. Ex. 1007, ¶97, ¶101, ¶103, Tables 2, 4, 6.

135. Karczewicz-II teaches an improved method for averaging interpolated pixel values. Specifically, Karczewicz-II discloses maintaining higher precision for intermediate values (e.g., integer or half-pixel predictions) during calculation and delaying rounding until later in the process in order to reduce rounding inaccuracies. *See, e.g.*, Ex. 1007, ¶10, ¶39, ¶53, ¶59, ¶¶99-106.

136. Karczewicz-II is in the same field of endeavor as the '267 patent (video encoding/decoding). *See supra* §III. Karczewicz-II is directed to "digital video coding" and "interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex. 1007, ¶2, Abstract. Karczewicz-II uses and improves the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex. 1007, ¶¶46-47, ¶82, ¶88, ¶¶93-0106 *with* Ex. 1001, 2:60-3:11, 9:26-41.

3. Motivation to Combine and Reasonable Expectation of Success

137. Karczewicz-I and Karczewicz-II are Qualcomm patent applications by the same inventors Marta Karczewicz, Peisong Chen, and Yan Ye. Both are directed to video coding and apply their teachings to similar architectures.

Ex. 1005, ¶2, Fig. 1; Ex. 1007, ¶2, Fig. 1

Karczewicz-I **Karczewicz-II**

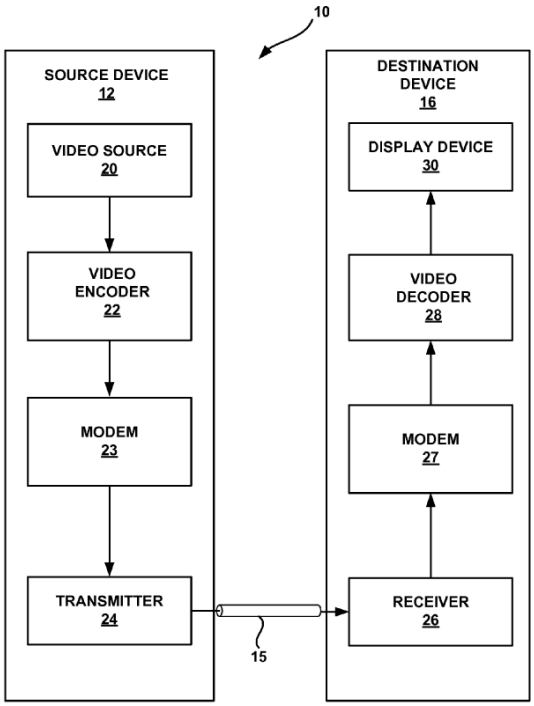


FIG. 1

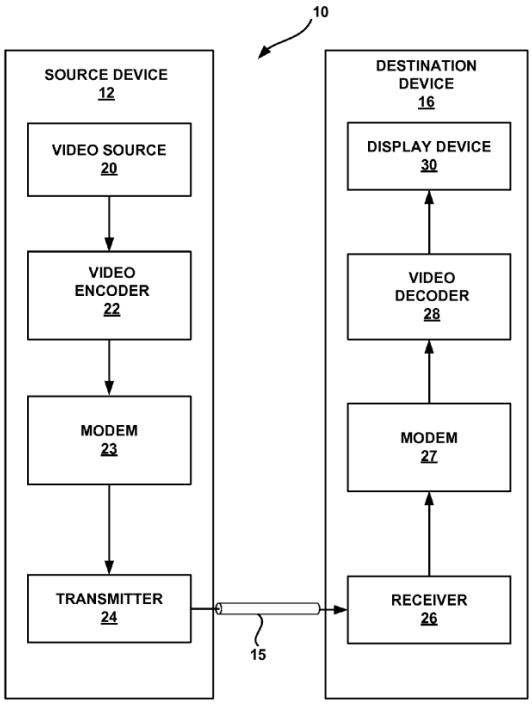


FIG. 1

Ex. 1007, Fig. 1, ¶¶40-53. Ex. 1005, Fig. 1, ¶¶29-50.

138. Karczewicz-I and Karczewicz-II are both directed to block-based, e.g., H.264, techniques for motion prediction. Ex. 1005, ¶35, ¶44, ¶¶55-60; Ex. 1007, ¶8, ¶¶35-36, ¶46, ¶54. Both include teachings for a video encoder that performs motion estimation and compensation for inter-predictive coding:

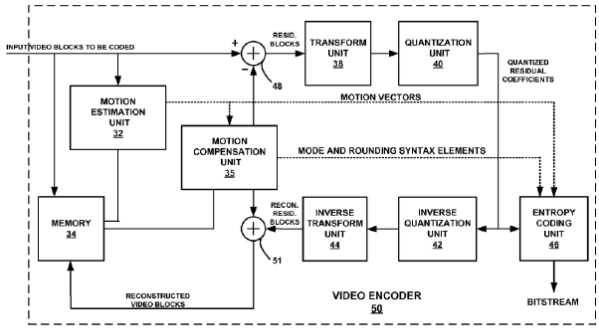


FIG. 2

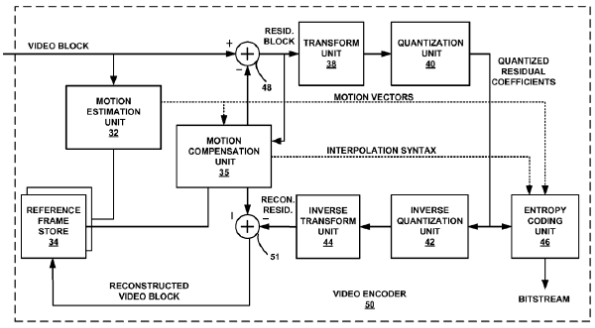


FIG. 2

Ex. 1005, Fig. 2, ¶53 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding.").

Ex. 1007, Fig. 2, ¶56 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding.").

Both teach decoders having similar architectures that perform motion compensation for block-based decoding:

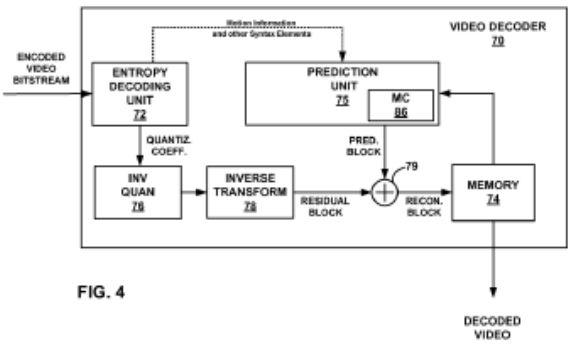


FIG. 4

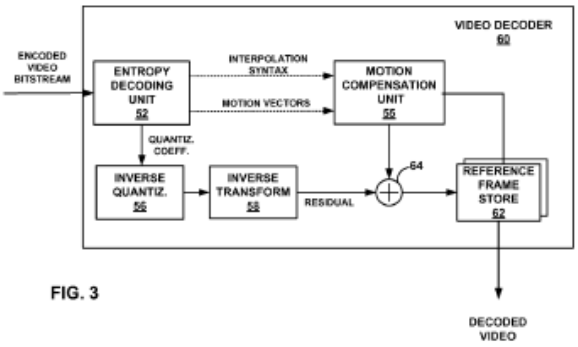


FIG. 3

Ex. 1005, Fig. 4 (above left); ¶89; Ex. 1007, Fig. 3 (above right), ¶63.

139. Since Karczewicz-I and Karczewicz-II are from the same inventors of the same company and are directed to performing inter-predictive coding using video encoding and decoding systems of the same architecture, a POSITA would have found it obvious the combine the teachings of Karczewicz-I and Karczewicz-II, including those teachings described above. *Supra* §V.B.1 (Karczewicz-I teachings), §V.B.2 (Karczewicz-II teachings). The similarities of

Karczewicz-I and Karczewicz-II's architecture would have suggested to a POSITA to implement techniques taught by Karczewicz-I and techniques taught by Karczewicz-II using that common architecture.

140. This would have been a combination of prior art elements according to known methods. Karczewicz-I teaches bi-prediction techniques for the H.264 standard with two motion vectors pointing to two blocks of pixels values that are averaged together. Ex. 1005, ¶53, ¶35, ¶44, ¶60. For interpolated sub-pixels, this calculation averages the interpolated values together. Beyond integer pixels, Karczewicz-II teaches that, for H.264, motion vectors can also point to fractional sub-pixels (Ex. 1007, ¶¶56-58, ¶¶93-102),[2] and Karczewicz-II teaches an improved calculation for averaging interpolated pixel values, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate calculations (E.g., Ex. 1007, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-106). Therefore, Karczewicz-II provides complementary teachings that improve Karczewicz-I's teachings for averaging predicted pixel values.

141. As Karczewicz-II explains, its use of higher precision for intermediate steps eliminates the propagation of rounding inaccuracies and improves the

[2] This was known in the art, as admitted in the '267 patent's "Background Information" section. *Supra* §I.C; Ex. 1001, 2:60-3:11 ("The motion vectors are not limited to having full-pixel accuracy, but could have fractional-pixel accuracy as well. That is, motion vectors can point to fractional-pixel positions/locations of the reference frame, where the fractional-pixel locations can refer to, for example, locations 'in between' image pixels. … The H.264/AVC video coding standard supports motion vectors with up to quarter-pixel accuracy.").

accuracy of the average. *See, e.g.*, Ex. 1007, ¶102 ("By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate."), ¶10 ("In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values."), ¶39, ¶53, ¶59. Therefore, a POSITA would have been motivated to apply Karczewicz-II's improved calculations to Karczewicz-I's teachings.

142. As further explained below, the calculations taught by Karczewicz-II correspond to the calculations used for Karczewicz-I's bi-predicted pixel values when applied to integer and sub-pixel values. *Infra* Subsections Scenarios 1-3. Karczewicz-I and Karczewicz-II both teach motion prediction and both encompass calculations that interpolate pixel values and then average the interpolated pixel values together. *Id.* While Karczewicz-II teaches its improved calculations in the context of quarter-pixel interpolation, Karczewicz-I and Karczewicz-II teach the same calculations for interpolating and averaging pixel values when motion vectors point to interpolated sub-pixels. Therefore, since Karczewicz-II teaches an

improved implementation of the mathematical calculations taught by Karczewicz-I, a POSITA would have been motivated to use the known technique of Karczewicz-II to improve similar devices/methods, as taught by Karczewicz-I, in the same way, using higher precision when combining interpolated pixel values, as Karczewicz-II teaches. This would have achieved the same benefits taught by Karczewicz-II, e.g., eliminating the propagation of rounding inaccuracies and improving prediction accuracy. *E.g.*, Ex. 1007, ¶102, ¶10. And it would have had predictable results because it applies teachings from Karczewicz-II to corresponding mathematical calculations used for Karczewicz-I, using similar video encoding architectures as explained above. A POSITA would have recognized the applicability of Karczewicz-II to the corresponding calculations in Karczewicz-I without undue experimentation.

143. Karczewicz-I and Karczewicz-II provide complementary teachings. Karczewicz-I is directed to weighted bi-directional prediction that averages two prediction values together. *See, e.g.*, Ex. 1005, ¶8. A weighted bi-directional prediction is determined based on two reference blocks pointed to by two motion vectors. Ex. 1005, ¶53. Karczewicz-I teaches weighted bi-directional prediction that averages two predictions based on two reference blocks (e.g., pred0(i,j), pred1(i,j)) with an offset (e.g., +1), including a default mode (Ex. 1005, ¶55) with equal weights that performs a simple average. Ex. 1005, ¶60:

> Default weighted prediction may be defined by the following equations for unidirectional prediction and bidirectional prediction, respectively.
>
> …
>
> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

144. Karczewicz-I's discussion focuses on methods for combining two predictions, relying on known techniques, such as the techniques under the H.264 standard, for determining the two predictions to be combined. *See, e.g.*, Ex. 1005, ¶35 ("Video encoder 22 and video decoder 28 may operate according to a video compression standard, such as the ITU-T H.264 standard[.]"). The H.264 standard provides that, when a motion vector points to a fractional pixel/subpixel, interpolation is used to calculate a prediction. *Supra* §I.C; Ex. 1007, ¶¶68-69, ¶74, ¶93. Consistent with H.264, Karczewicz-I teaches that the inter-predictive coding process, where motion vectors point to different frames, includes interpolation, and therefore provides express teaching, suggestion, and motivation ("TSM") to combine with known teachings for interpolation. Ex. 1005, ¶41 ("Following inter-based predictive encoding (which includes interpolation and the techniques of this disclosure to efficiently select a prediction algorithm or mode by which to predict a coded unit) …").

145. Karczewicz-II teaches improved calculations for predictions involving interpolated fractional pixel positions in H.264, where rounding is delayed until

later in the process, thereby maintaining higher precision for intermediate values. *See, e.g.*, Ex. 1007, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."), ¶99, ¶10. Karczewicz-II explains that, according to the H.264 standard, predictions for half-pixel positions are calculated using 6-tap interpolation filters that interpolate the sub-pixel based on nearby pixels in the same row (e.g., half-pixel "b") or column (e.g., half-pixel "h"). Ex. 1007, ¶74, ¶¶93-94, Fig. 4A-4B:

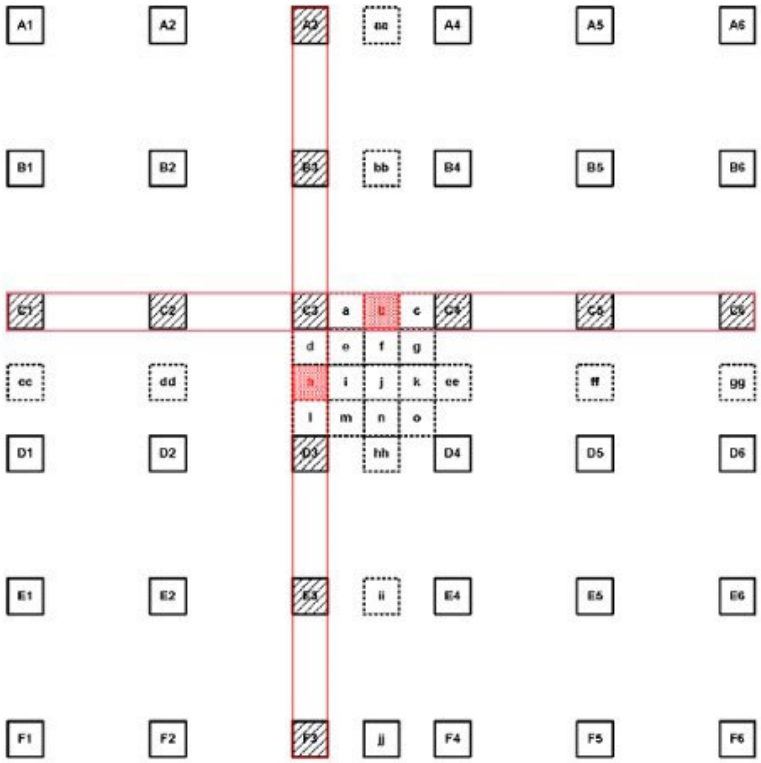


FIG. 4B

146. This process involves two steps. First, a 6-tap filter multiplies the six pixels in the row (or column) by the filter values (1,−5, 20, 20,−5, 1), adds the

products together to produce a non-rounded prediction (e.g., b1). Second, the result is rounded down, e.g., using a right-shift operation (">>") that decreases the number of bits needed to represent the prediction back down to the original precision.[3] *Supra* §I.D. Here is the process for interpolating sub-pixel "b":

b1=C1−5*C2+20*C3+20*C4−5*C5+C6

…

b=max(0, min(255, (b1+16)>>5))

Ex. 1007, ¶¶93-94, Fig. 4B. This produces a rounded prediction (e.g., b).

147. For half-pixel positions that are in the center of four integer pixels on two dimensions (e.g., j) (referred to as a "center-pixel position" hereinafter), the interpolation process is applied in two rounds: one to interpolate a middle row of half pixels, and another to interpolate that middle row into the center pixel. In other words, the 6-tap interpolation filter is applied on values of six half-pixel predictions. Ex. 1007, ¶95, Fig. 4C.

148. Karczewicz-II teaches calculating quarter-pixel predictions by averaging the predictions of the two nearest integer or half-pixel positions as explained above. Ex. 1007, ¶¶96-97. Karczewicz-II teaches an improvement to

[3] A POSITA would have understood that the right shifting operation is equivalent to dividing the weighted sum by the total weight. In binary computation, right shifting by a number of bits is equivalent to division by 2 to the power of the number of bits shifted. For example, right shifting by 5 bits is equivalent to dividing the weighted sum by 25, which equals 32; here, 32 is the sum of the six weights of the 6-tap filter (i.e., 1-5+20+20-5+1=32).

reduce coding inefficiencies and increase the accuracy of pixel calculations by "keep[ing] the highest possible precision through the intermediate steps" and avoiding "any shifting, rounding and clipping operations[] until the very last step of the interpolation process." Ex. 1007, ¶99. This prevents the propagation of rounding errors. Ex. 1007, ¶102 ("By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate."), ¶10 ("In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values."), ¶39, ¶53, ¶59.

149. Calculation Scenarios. Karczewicz-I and Karczewicz-II both calculate pixel values, including interpolated pixel values, with Karczewicz-I performing calculations for bi-directional prediction of pixel values and Karczewicz-II teaching corresponding calculations for interpolating pixel values, e.g., in quarter-pixel positions.

150. Karczewicz-I teaches bi-predicted blocks (e.g., in B frames) where motion prediction is based on motion vectors for two reference frames, and the default weighted prediction averages those two predicted values together:

$$pred(i,j)=(pred0(i,j)+pred1(i,j)+1)>>1$$

Ex. 1005, ¶¶58-60. As explained above, H.264 allows motion vectors to point to integer pixels or fractional pixels, such as half pixels, and uses interpolation to calculate predictions for fractional pixels. *See e.g.*, Ex. 1005, ¶41; Ex. 1007 ¶¶93-102 ("A sub-pixel motion vector refers to a sub-pixel position in a reference picture which needs to be interpolated."), ¶¶56-58; *supra* §I.C.[4] Therefore, the predictions pred0(i,j) and pred1(i,j) encompass scenarios that include integer pixel prediction, a half-pixel prediction, or a center-pixel prediction, and Karczewicz-I calculates an average of these pixel values.

151. Karczewicz-II teaches optimizations for averaging integer, half, and center pixels, which a POSITA would have been motivated to apply to at least three scenarios for Karczewicz-I's teachings, based on whether the two motion vectors in Karczewicz-I's bidirectional prediction point to integer or sub-pixel positions.

[4] This was known in the art, as admitted in the '267 patent's "Background Information" section. *Supra* §I.C; Ex. 1001, 2:60-3:11.

152. Scenario 1 (with the first motion vector pointing to a half-pixel position and the second motion vector pointing to an integer pixel position). Beyond integer pixels, Karczewicz-II explains how, for H.264, motion vectors can also point to half-pixel positions. *E.g.*, Ex. 1007, ¶¶93-102, ¶¶56-58. When the two motion vectors in Karczewicz-I's bi-directional prediction point to a half-pixel position and an integer pixel position, the default weighted prediction is calculated as an average of the half-pixel and the integer pixel. *See* Ex. 1005, ¶60, ¶55.

153. Karczewicz-II teaches an improved calculation for averaging a half-pixel value with an integer pixel value. While Karczewicz-II uses this calculation to interpolate a quarter-pixel position, it is the same calculation as Scenario 1 because it averages an integer and half pixel. *See* Ex. 1007, ¶96, Fig. 4D. Karczewicz-II explains the conventional calculation for averaging two numbers: adding the half-pixel value (e.g., "b") with the integer pixel value (e.g., C3) and a rounding offset 1, then dividing by 2 (using a right-shift ">>" operation that is mathematically equivalent to dividing by 2). Ex. 1007, Fig. 4D, Table 1:

TABLE 1

a = (C3 + b + 1) >> 1
c = (C4 + b + 1) >> 1
d = (C3 + h + 1) >> 1
l = (D3 + h + 1) >> 1
f = (j + b + 1) >> 1
i = (j + h + 1) >> 1
k = (j + ee + 1) >> 1
n = (j + hh + 1) >> 1

154. Karczewicz-II improves this conventional approach by "keep[ing] the highest possible precision through the intermediate steps[.]" Ex. 1007, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex. 1007, 99. The integer pixel value (e.g., C3) is multiplied by 32 (by bit-shifting to the left 5 bits, which is mathematically equivalent), taking its precision from 8 to 13 bits. Ex. 1007, Table 5. Instead of using a rounded 8-bit half-pixel (e.g., b), Karczewicz-II delays the rounding step and instead uses a *non-rounded* half-pixel prediction (e.g., b1) that is 15 bits. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (">>6"). *Id.*, Table 3:

TABLE 3

a = (C3 << 5 + b1 + 32) >> 6
c = (C4 << 5 + b1 + 32) >> 6
d = (C3 << 5 + h1 + 32) >> 6
l = (D3 << 5 + h1 + 32) >> 6
f = (j1 >> 5 + b1 + 32) >> 6
i = (j1 >> 5 + h1 + 32) >> 6
k = (j1 >> 5 + ee1 + 32) >> 6
n = (j1 >> 5 + hh1 + 32) >> 6

155. The operations for this improved approach are shown in Table 5 of Karczewicz-II. Ex. 1007, ¶103 ("The following Tables show the interpolation process for other sub-pixels in sixteen bit storage elements. In the Tables below,

the operations defined in each column are performed sequentially through the respective table."), Table 5:

TABLE 5

positions {a, c, d, l} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = x	r1 is integer pixel x	0	255	8u
r1 = r1 << 5	r1 is 32 * x	0	8160	13u
r2 = y0	r2 is y0 (y0 is a one-dimensional (1-D) half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is 32 * x + y0	−2550	18870	16s
r1 = r1 + 32	r1 is 32 * x + y0 + 32	−2518	18902	16s
r1 = r1 >> 6	r1 is (32 * x + y0 + 32) >> 6	−39	295	11s
r1 = max (0, r1)	clip r1 on the low side	0	295	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

156. Since Karczewicz-II teaches this optimization for averaging an interpolated half-pixel and an integer pixel, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of an integer pixel and an interpolated half-pixel in Scenario 1. For

example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex. 1005, ¶60):

> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

157. For Scenario 1, the two predictions would be a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)). *See* Ex. 1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel prediction (pred0(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied (Ex. 1007, ¶¶93-94). To match this higher precision, the integer pixel (pred1(i,j)) is left-shifted, as Karczewicz-II teaches. Ex. 1007, ¶99, Table 3. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

pred(*i,j*)=(non-rounded(pred0(*i,j*))+pred1(*i,j*)<<5+32)>>6

See Ex. 1007, ¶96, ¶99, Tables 1 and 3; Ex. 1005, ¶60.

158. Scenario 2 (with the first motion vector pointing to a center-pixel position and the second motion vector pointing to a half-pixel position). Karczewicz-II explains how, for H.264, motion vectors point to center-and half-pixel positions. *E.g.*, Ex. 1007, ¶¶93-95. When the two motion vectors in

Karczewicz-I's bi-directional prediction point to a center-and half-pixel position, the default weighted prediction is calculated as an average of the center-and half-pixels. *See* Ex. 1005, ¶60, ¶55.

159. Karczewicz-II teaches an improved calculation for averaging a center-and half-pixel. Ex. 1007, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 2 because it averages a center and half pixel. As discussed above, Karczewicz-II explains the conventional calculation for averaging two numbers: adding the center-pixel value (e.g., "j") with the half-pixel value (e.g., "b") and a rounding offset 1, then dividing by 2 (using a right-shift ">>" operation that is mathematically equivalent to dividing by 2). Ex. 1007, ¶96, Fig. 4D, Table 1:

TABLE 1

a = (C3 + b + 1) >> 1
c = (C4 + b + 1) >> 1
d = (C3 + h + 1) >> 1
l = (D3 + h + 1) >> 1
f = (j + b + 1) >> 1
i = (j + h + 1) >> 1
k = (j + ee + 1) >> 1
n = (j + hh + 1) >> 1

160. Karczewicz-II improves this conventional approach by "keep[ing] the highest possible precision through the intermediate steps[.]" Ex. 1007, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex. 1007, ¶99. Whereas Table 1

used a fully-rounded center-pixel (e.g., "j"), Table 3 uses a partially-rounded pixel (e.g., "j1>>5") that remains 5 bits longer than the rounded value in Table 1.[5] *Infra* §V.B.6. Likewise, Table 3 replaces the rounded half-pixel (e.g., "b") with a non-rounded half-pixel (e.g., "b1") that is 15 bits. These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (">>6"). Ex. 1007, Table 3:

TABLE 3

a = (C3 << 5 + b1 + 32) >> 6
c = (C4 << 5 + b1 + 32) >> 6
d = (C3 << 5 + h1 + 32) >> 6
l = (D3 << 5 + h1 + 32) >> 6
f = (j1 >> 5 + b1 + 32) >> 6
i = (j1 >> 5 + h1 + 32) >> 6
k = (j1 >> 5 + ee1 + 32) >> 6
n = (j1 >> 5 + hh1 + 32) >> 6

161. The operations for this improved approach are shown in Table 8 of Karczewicz-II. Ex. 1007, ¶105 ("Table 8, below demonstrates steps that can be taken for sixteen-bit implementation of interpolating {f,i,k,n}, which are the positions that use to interpolate the intermediate value 'j1.'"), Table 8:

162. Since Karczewicz-II teaches this optimization for averaging interpolated center-and half-pixels, a POSITA would have been motivated to apply

[5] Here, the partially-rounded prediction is obtained by shifting the non-rounded prediction j1 to the right by 5 bits. This is fewer bits than the 10 bits shifted for calculating the fully rounded prediction j. Ex. 1006, ¶95.

that improved calculation to Karczewicz-I, which likewise calculates the average of an interpolated center pixel and an interpolated half-pixel in Scenario 2. For example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex. 1005, ¶60):

> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

163. For Scenario 2, the two predictions would be a center prediction (e.g., pred0(i,j)) and half-pixel prediction (e.g., pred1(i,j)). *See* Ex. 1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by only partially rounding the center pixel by 5 bits rather than 10 bits to keep it at a higher precision, as Karczewicz-II teaches. *E.g.*, Ex. 1007, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-103. Karczewicz-II uses the non-rounded half-pixel, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex. 1007, ¶74, ¶93, ¶¶99-100. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

> pred(*i,j*)=(non-rounded(pred0(*i,j*))>>5+non-rounded(pred1(*i,j*)) +32)>>6

See Ex. 1007, ¶96, ¶99, Tables 1 and 3; Ex. 1005, ¶60.

164. Scenario 3 (with both motion vectors pointing to half-pixel positions). Karczewicz-II explains how, for H.264, motion vectors point to pixels, including

half-pixel positions. *E.g.*, Ex. 1007, ¶93. When the two motion vectors in Karczewicz-I's bi-directional prediction point to two half-pixel positions, the default weighted prediction is calculated as an average of the two half-pixels. *See* Ex. 1005, ¶60, ¶55.

165. Karczewicz-II teaches an improved calculation for averaging two half-pixel values. Ex. 1007, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 3 because it averages two half pixels. As discussed above, Karczewicz-II explains the conventional calculation for averaging two numbers: adding the half-pixel values (e.g., "b" and "h") and a rounding offset 1, then dividing by 2 (using a right-shift ">>" operation that is mathematically equivalent to dividing by 2). Ex. 1007, ¶97, Fig. 4D, Table 2:

TABLE 2

e = (b + h + 1) >> 1
g = (b + ee + 1) >> 1
m = (h + hh + 1) >> 1
o = (ee + hh + 1) >> 1

166. Karczewicz-II improves this conventional approach by "keep[ing] the highest possible precision through the intermediate steps[.]" Ex. 1007, ¶99, ¶101. Karczewicz-II replaces the equations in Table 2 with those in Table 4, where the pixel values are combined at a higher precision. Ex. 1007, ¶101. Instead of using a rounded 8-bit half-pixel (e.g., "b" or "h"), Karczewicz-II delays the rounding step

and instead uses *non-rounded* half-pixels (e.g., "b1" and "h1") that are 15 bits each. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (">>6"). *Id.*, Table 4:

167. The operations for this improved approach are shown in Table 6 of Karczewicz-II. Ex. 1007, ¶103, Table 6:

TABLE 6

positions {e, g, m, o} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (y0 is a 1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = y1	r2 is y1 (y1 is a 1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is y0 + y1	−5100	21420	16s
r1 = r1 + 32	r1 is y0 + y1 + 32	−5068	21452	16s
r1 = r1 >> 6	r1 is (y0 + y1 + 32) >> 6	−79	335	11s
r1 = max (0, r1)	clip r1 on the low side	0	335	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

168. Since Karczewicz-II teaches this optimization for averaging two interpolated half-pixels, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of two interpolated half-pixels in Scenario 3. For example, Karczewicz-I calculates

the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex. 1005, ¶60):

> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

169. For Scenario 3, the two predictions would be half-pixel predictions. *See* Ex. 1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel predictions (pred0(i,j) and pred1(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex. 1007, ¶¶97-101. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

> pred(*i,j*)=(non-rounded(pred0(*i,j*))+non-rounded(pred1(*i,j*)) +32)>>6

See Ex. 1007, ¶97, ¶101, Tables 2 and 4; Ex. 1005, ¶60.

170. Express Teaching Suggestion or Motivation (TSM) in the Art. Preserving higher-precision intermediate values in computations related to video coding was well known in the art as a method for improving the accuracy of computation; the prior art therefore provides express TSM to apply such teachings from Karczewicz-II to related techniques in Karczewicz-I to achieve a higher accuracy for Karczewicz-I's bi-directional prediction. *See, e.g.*, Ex. 1015, Abstract

("Image quality from MPEG-style video coding may be improved by preserving a higher number of bits during intermediate encoding and decoding processing steps.").

171. Karczewicz-I is directed to motion compensation. Preserving higher-precision intermediate values was known to benefit motion compensation. For example, U.S. Patent No. 9,344,744 ("Kirchhoffer-744") (Ex. 1009) teaches performing "a prediction and a reconstruction for a block of a picture to be predicted … in a higher precision." Ex. 1009, 7:4-19, 16:22-17:36. This approach was known to result in a more precise prediction and a smaller residual information. *Id*. ("An advantage of embodiments according to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."). Therefore, the prior art provides express TSM to apply Karczewicz-II's teachings to Karczewicz-I and thereby perform motion compensation at a higher precision. *See id.*

172. It was also known that preserving higher-precision intermediate values improves the accuracy of interpolation. *See, e.g.*, Ex. 1008, 000015 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding

and clamping. It is possible to achieve additional accuracy by retaining a higher precision result after the first stage of filtering."); Ex. 1010, ¶¶61-62. This provides motivation to apply Karczewicz-II's teachings to other known video encoding methods that involves interpolation.

173. Compatible Teachings. The combination would not have changed the principle of operation of either reference, but merely includes the use of a known technique (e.g., Karczewicz-II's technique of calculating the average of two predictions with higher-precision intermediate values) to improve similar devices or methods (e.g., Karczewicz-I's system and method for encoding video using bi-directional prediction) in the same way. As explained above, Karczewicz-I and Karczewicz-II are filed by the same inventors from the same company and teach similar video encoding and decoding methods implemented on similar video encoding and decoding systems, both for averaging predicted pixel values in H.264. Ex. 1005, Figs. 1-2, ¶¶29-50, ¶53; Ex. 1007, Figs. 1-2, ¶¶40-53, ¶56. Given the similarities between Karczewicz-I and Karczewicz-II, a POSITA would have understood that Karczewicz-II's techniques are readily applicable to Karczewicz-I.

174. Moreover, the combination merely changes when rounding occurs for calculations that already included rounding. *See* Ex. 1005, ¶60, ¶55; Ex. 1007, ¶¶96-106, Tables 1-8. This minor implementation detail would not have changed the principle of operation of Karczewicz-I. Nor would it have changed the

principle of operation for Karczewicz-II since the underlying mathematics remains the same. In other words, Karczewicz-II rearranges the order of operations without changing the nature of the calculations, from an algebraic perspective, from the conventional approach for averaging interpolated pixels. Therefore, applying Karczewicz-II's teachings to Karczewicz-I would not have altered the principle of operation of either reference.

175. Reasonable Expectation of Success. A POSITA would have had a reasonable expectation of success when combining the teachings of Karczewicz-I and Karczewicz-II. As explained above, the combination simply uses Karczewicz-II's technique of using higher-precision versions of intermediate values and delaying the rounding in calculating an average of two predictions to the end, in the manner taught by Karczewicz-II, to improve corresponding calculations taught by Karczewicz-I. In other words, Karczewicz-II already teaches the mathematical principles of its improved calculations; those calculations can be applied to Karczewicz-I's scenarios without further modification.

176. Furthermore, a POSITA would have been more than capable of applying Karczewicz-II's teachings because Karczewicz-II's calculations involve elementary mathematics and logical operations (e.g., addition and bit-shifting) and fundamental video codec operations that were a core part of industry work in video codecs. The calculation of averages and the reordering of operations for such

calculations was a matter of simple algebra. Furthermore, the concept of using higher-precision intermediate values to improve accuracy was well-known and conventional techniques for many years before 2011. *Supra* §I.D. Therefore, given the level of skill in the art, a POSITA would have been more than capable of combining their teachings.

4. Encoder Claims

a. Independent Claims 1, 7, and 13

[1a]. A method for encoding a block of pixels, the method comprising:

177. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

178. Karczewicz-I "relates to video encoding and, more particularly, video encoding techniques that use bi-directional prediction." Ex. 1005, ¶2, ¶8 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction."), ¶22 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction."). Karczewicz-I teaches "a method of encoding video data." Ex. 1005, ¶9. Karczewicz-I further teaches a "video encoding and decoding system" that includes a "video encoder 22" that encodes video data. Ex. 1005, ¶¶29-30, ¶32 ("In each case, the captured,

pre-captured or computer-generated video may be encoded by video encoder 22."), Fig. 1. The video encoder "perform[s] … inter-coding of blocks within video frames" that includes motion estimation and compensation operations. Ex. 1005, ¶¶51-76, Fig. 2.

179. Karczewicz-I encodes a block of pixels. *See, e.g.*, Ex. 1005, ¶39 ("Video encoder 22 and video decoder 28 may operate on video blocks within individual video frames in order to encode and decode the video data. … Video blocks may comprise blocks of pixel data …"), ¶40 ("In general, macroblocks and the various sub-blocks may be considered to be video blocks."). Thus, Karczewicz-I teaches a method for encoding a block of pixels.

180. Karczewicz-II "describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex. 1007, Abstract, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."). Karczewicz-II teaches "method[s] of encoding video data." Ex. 1007, ¶12. Karczewicz-II's methods of encoding video data include block-based inter-coding that includes motion estimation and compensation operations. Ex. 1007, ¶4 ("Block based inter-coding is a very useful coding technique that relies on temporal prediction to reduce or remove temporal redundancy between video blocks of successive coded units of a video sequence. … For inter-coding, the

video encoder performs motion estimation and motion compensation to track the movement of corresponding video blocks of two or more adjacent coded units."). Similar to Karczewicz-I, Karczewicz-II teaches a "video encoding and decoding system" that includes a "video encoder 22" that encodes video data. Ex. 1007, ¶¶40-41, ¶43 ("In each case, the captured, pre-captured or computer-generated video may be encoded by video encoder 22."), Fig. 1. The video encoder "perform[s] … inter-coding of blocks within video frames" that includes motion estimation and compensation operations. Ex. 1007, ¶¶54-62, Fig. 2.

181. Similar to Karczewicz-I, Karczewicz-II encodes a block of pixels. *See, e.g.*, Ex. 1007, ¶49 ("Video encoder 22 operates on video blocks within individual video frames in order to encode the video data. … Video blocks may comprise blocks of pixel data…"), ¶50 ("In general, macroblocks (MBs) and the various sub-blocks may be considered to be video blocks.").

182. Karczewicz-II teaches interpolation methods for use in video encoding. *See, e.g.*, Ex. 1007, ¶14 ("In another example, this disclosure provides a method of interpolating predictive video data for video coding."), ¶35 ("This disclosure describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding."). As explained above, a POSITA would have found it obvious to modify Karczewicz-I's technique for obtaining a combined prediction, which is part of

Karczewicz-I's video encoding method, based on Karczewicz-II's interpolation techniques. *Supra* §V.B.3 (explaining the motivation to combine; that analysis is incorporated here). Thus, the combination of Karczewicz-I and Karczewicz-II teaches a method for encoding a block of pixels.

183. As explained below, Karczewicz-I's bi-directional prediction operations, as modified based on Karczewicz-II's interpolation techniques, teach the limitations of claim 1. *Infra* §§V.B.4.a[1b-1g]. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches a method for encoding a block of pixels, comprising the operations explained below for limitations [1b]-[1g].

184. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble and any additional limitations of element [7a].

185. As explained above for [1a], the combination of Karczewicz-I and Karczewicz-II teaches a method for encoding a block of pixels, including operations as described for limitations [1b]-[1g]. *Supra* §V.B.4.a[1a]. As explained below, the combination of Karczewicz-I and Karczewicz-II teaches a video encoder for encoding a block of pixels that performs operations as described for limitations [7b]-[7g]. *Infra* §§V.B.4[7b-7g].

186. Karczewicz-I and Karczewicz-II teach a video encoder performing video encoding operations. *Supra* §V.B.4.a[1a]; Ex. 1005, ¶¶29-30, ¶32, Fig. 1, ¶¶51-76, Fig. 2:

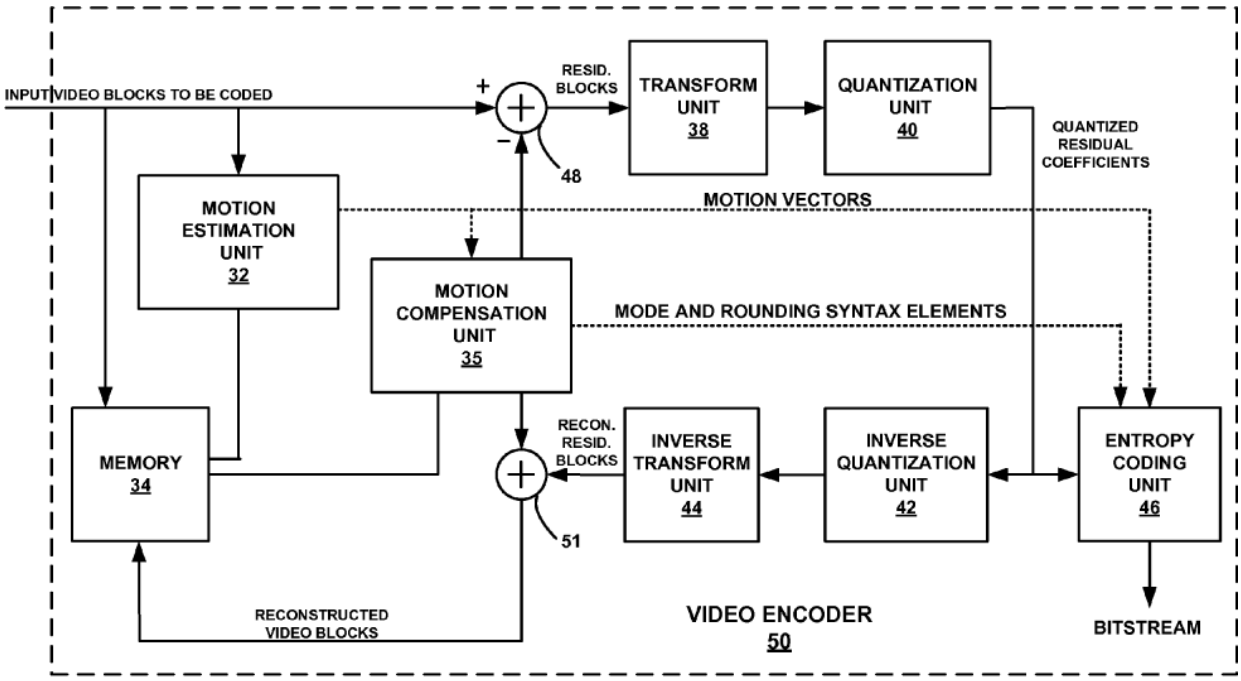


FIG. 2

Ex. 1007, ¶¶40-41, ¶43, Fig. 1, ¶¶54-62, Fig. 2.

187. Karczewicz-I and Karczewicz-II teach that the video encoder is an apparatus for encoding video. Ex. 1005, ¶10 ("[T]his disclosure describes a video encoder apparatus that encodes video data[.]"); Ex. 1007, ¶15 ("[T]his disclosure provides an apparatus that encodes video data, the apparatus comprising a video encoder …"), ¶17, ¶21, ¶23. Karczewicz-I and Karczewicz-II teach the video encoder encodes a block of pixels. *See, e.g.*, Ex. 1005, ¶¶39-40; Ex. 1007, ¶¶49-50. Karczewicz-I and Karczewicz-II teach implementing the video encoder in various

types of devices. Ex. 1005, ¶3 ("Digital multimedia capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless communication devices, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, video gaming devices, video game consoles, cellular or satellite radio telephones, digital media players, and the like."); Ex. 1007, ¶3.

188. Karczewicz-I and Karczewicz-II teach implementing the video encoder and performing operations for encoding a block of pixels using a processor and memory (e.g., computer readable medium) that includes computer program code (e.g., software). Ex. 1005, ¶12 ("The techniques described in this disclosure may be implemented in hardware, software, firmware, or any combination thereof[.] If implemented in software, the software may be executed in one or more processors, such as a microprocessor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), or digital signal processor (DSP). The software that executes the techniques may be initially stored in a computer-readable medium and loaded and executed in the processor."), ¶38 ("Video encoder 22 and video decoder 28 each may be implemented as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete

logic, software, hardware, firmware or any combinations thereof."), ¶98; Ex. 1007, ¶25.

189. Karczewicz-I and Karczewicz-II explain that the computer readable medium comprises well-known types of memories. Ex. 1005, ¶99 ("The computer-readable storage medium may comprise random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like."); Ex. 1007, ¶119. Karczewicz-I and Karczewicz-II further explain that its software includes computer program code. Ex. 1005, ¶100 ("The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry."); Ex. 1007, ¶26, ¶120.

190. Karczewicz-I and Karczewicz-II further teaches the processor executing computer program code in memory to cause the apparatus to carry out its functionalities. Ex. 1005, ¶99 ("If implemented in software, the techniques may be realized at least in part by a computer-readable medium comprising instructions that, when executed in a processor, performs one or more of the methods described

above."), ¶100 ("The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry."); Ex. 1007, ¶48, ¶¶119-120.

191. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches an apparatus for encoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to perform operations as described for limitations [7b]-[7g].

[13a].	**A computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:**

192. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

193. As explained above, Karczewicz-I and Karczewicz-II teach implementing the video encoder and performing operations for encoding a block of pixels using software stored in computer readable storage media and executed by a processor. *Supra* §V.B.4[7a]; Ex. 1005, ¶12, ¶38, ¶¶98-100; Ex. 1007, ¶¶25-26, ¶48, ¶¶119-120. A POSITA would have understood that the types of computer readable storage media disclosed by Karczewicz-I and Karczewicz-II, including random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, are forms of non-transitory computer readable storage medium, as opposed to transitory signals. Ex. 1005, ¶99; Ex. 1007, ¶119. Karczewicz-I and Karczewicz-II teach that the software, which is stored in computer readable storage medium and executable by a processor, includes computer executable program code portions that comprise program code instructions. Ex. 1005, ¶100; Ex. 1007, ¶26, ¶120.

194. The operations taught by the combination of Karczewicz-I and Karczewicz-II, which are performed by the software residing on the non-transitory computer-readable storage medium, teach the limitations of claim 13. *Infra* §V.B.4.a[13b-13g]. Karczewicz-I and Karczewicz-II further teaches forming a computer program product using the non-transitory computer readable storage media that stores software. Ex. 1005, ¶99 ("The computer-readable medium may

comprise a computer-readable storage medium and may form part of a computer program product …"); Ex. 1007, ¶119. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches a computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to perform the operations recited in claim 13.

[1b]/[7b]/[13b]	**[determining/determine], for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;**

195. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1b], [7b], and [13b]. Karczewicz-I and Karczewicz-II teach using inter-coding to encode a block. Ex. 1005, ¶4 ("Inter-coding relies on temporal prediction and transform coding to reduce or remove temporal redundancy between video blocks of successive video frames of a video sequence."), ¶51 ("Video encoder 50 may perform intra-and inter-coding of blocks within video frames … Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames of a video

sequence."), Ex. 1007, ¶4 ("Block based inter-coding is a very useful coding technique that relies on temporal prediction to reduce or remove temporal redundancy between video blocks of successive coded units of a video sequence."), ¶54 ("Video encoder 50 may perform intra-and inter-coding of blocks within video frames … Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames of a video sequence.").

196. Inter-coding includes motion estimation and motion compensation. Ex. 1005, ¶53 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding."), Fig. 2:

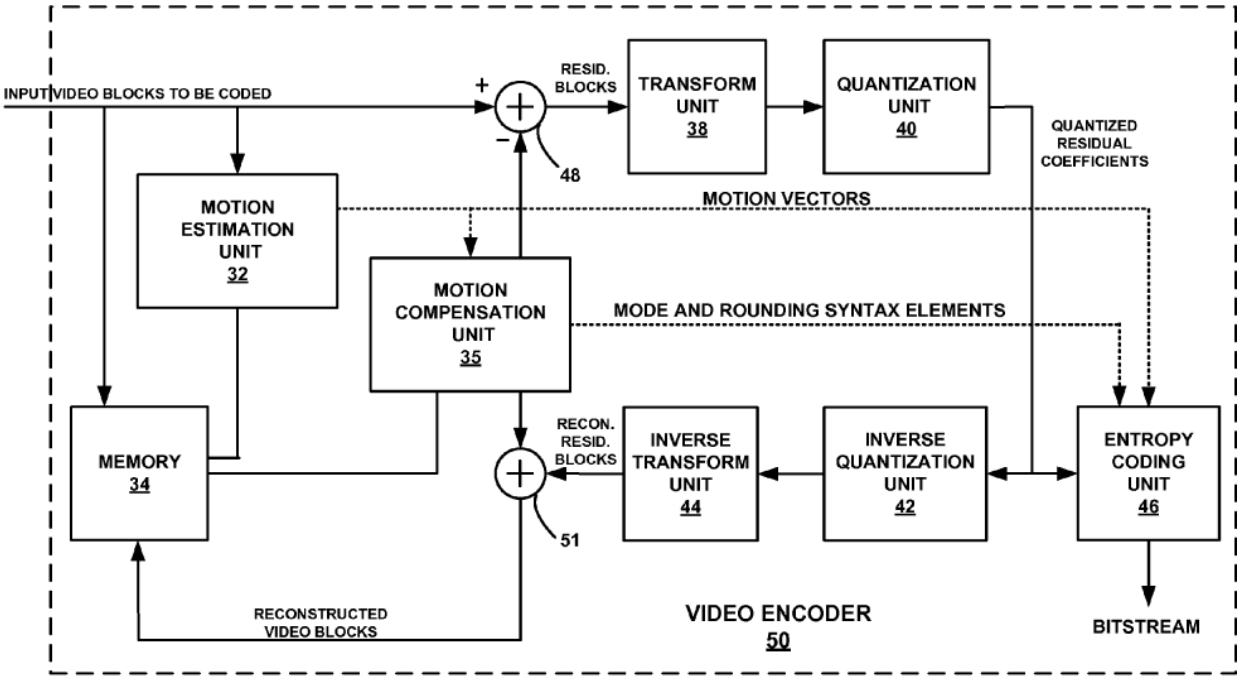


FIG. 2

Ex. 1007, ¶4 ("For inter-coding, the video encoder performs motion estimation and motion compensation to track the movement of corresponding video blocks of two or more adjacent coded units."), ¶6 ("Inter-coding based on motion estimation and motion compensation can achieve very good compression because successive video frames or other types of coded units are often very similar."), ¶56 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding."), Fig. 2.

197. Motion estimation generates motion vectors that point to reference blocks (e.g., predictive/prediction video blocks) **for a current block** and indicate the displacement between the reference blocks and the current block. Ex. 1005, ¶7 ("For P-and B-video blocks, motion estimation generates motion vectors, which indicate the displacement of the video blocks relative to corresponding prediction video blocks in predictive reference frame(s) or other coded units."), ¶53 ("A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit)."), ¶54 ("The selected motion vector for any given list may point to a predictive video block that is most similar to the video block being coded, e.g., as defined by a metric such as sum of absolute difference (SAD) or sum of squared difference (SSD) of pixel values of the

predictive block relative to pixel values of the block being coded."); Ex. 1007, ¶4 ("Motion estimation generates motion vectors, which indicate the displacement of video blocks relative to corresponding prediction video blocks in one or more reference frames or other coded units."), ¶56 ("Motion estimation is typically considered the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit).").

198. Motion compensation includes **determining reference blocks** (e.g., predictive video blocks) **based on motion vectors**. Ex. 1005, ¶7 ("Motion compensation uses the motion vectors to generate prediction video blocks from the predictive reference frame (s) or other coded units."), ¶53 ("Motion compensation is typically considered the process of fetching or generating the predictive block based on the motion vector determined by motion estimation."); Ex. 1007, ¶4 ("Motion compensation uses the motion vectors to generate prediction video blocks from the reference frame or other coded unit."), ¶56 ("Motion compensation is typically considered the process of fetching or generating the predictive block based on the motion vector determined by motion estimation."), ¶58 ("Once motion estimation unit 32 has selected the motion vector for the video block to be coded, motion compensation unit 35 generates the predictive video block

associated with that motion vector."), ¶73 ("Then, the prediction video block is formed during motion compensation using the best motion vector."). By generating the predictive video block associated with a motion vector, Karczewicz-II ascertains the values of the reference block that will be used for the current block, based on the associated motion vector. Alternatively, when Karczewicz-II generates a prediction video block from a reference frame, it determines a reference block from the reference frame that will be used for further calculations, e.g., interpolation. *See id.*

199. Karczewicz-I teaches that, for B blocks, inter-coding is bi-directional, where two lists of reference data are used to predict a current block. Ex. 1005, ¶5 ("[T]he term 'bi-directional' now refers to prediction based on two or more lists of reference data regardless of the temporal relationship of such reference data relative to the data being coded."), ¶6 ("Consistent with newer video standards such as ITU H.264, for example, bi-directional prediction may be based on two different lists which do not necessarily need to have data that resides temporally before and after the current video block. In other words, B-video blocks may be predicted from two lists of data, *which may correspond to data from two previous frames, two subsequent frames, or one previous frame and one subsequent frame.*"), ¶22 ("In bi-directional prediction, a video block is predictively encoded and decoded based on two different lists of predictive reference data."), ¶42 ("The

techniques of this disclosure are specifically applicable to weighted bi-directional prediction. As mentioned above, bi-directional prediction is prediction of so-called 'B-video blocks' based on two different lists of data. B-video blocks may be predicted from two lists of data from two previous frames, two lists of data from subsequent frames, or one list of data from a previous frame and one from a subsequent frame."), ¶54. Since Karczewicz-I teaches bi-prediction from two reference frames, it determines a first reference block (based on a first motion vector for one reference frame) and second reference block (based on a second motion vector for the other reference frame). *See id.*

200. Here, the reference data from the two lists include data in two reference blocks because Karczewicz-I teaches that the inter-prediction process for a current block includes determining reference blocks based on motion vectors (Ex. 1005, ¶7, ¶¶53-54), and when the current block is a B block having two motion vectors, two reference blocks are determined based on those two motion vectors. Therefore, the combination teaches **determining, for a current block, a** first reference block based on a first motion vector and second reference block based on a second motion vector.

201. Karczewicz-II further teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision (e.g., 8 bits). As explained above, the term "precision" is at least satisfied

by "a number of bits needed to represent possible values." *Supra* §IV.A. Karczewicz-II's use of the term "precision" is consistent with this interpretation. Ex. 1007, ¶89 ("The average filter may also be quantized to a certain fixed-point precision (e.g., 13-bit precision).").

202. Karczewicz-II teaches that 8 bits are needed to represent possible pixel values for the two reference blocks. For example, Table 5 teaches operations for calculating pixel values using interpolation. Ex. 1007, ¶103, Table 5:

TABLE 5

positions {a, c, d, l} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = x	r1 is integer pixel x	0	255	8u
r1 = r1 << 5	r1 is 32 * x	0	8160	13u
r2 = y0	r2 is y0 (y0 is a one-dimensional (1-D) half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is 32 * x + y0	−2550	18870	16s
r1 = r1 + 32	r1 is 32 * x + y0 + 32	−2518	18902	16s
r1 = r1 >> 6	r1 is (32 * x + y0 + 32) >> 6	−39	295	11s
r1 = max (0, r1)	clip r1 on the low side	0	295	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

Table 5 shows that integer pixels (e.g., integer pixel x) may take values between 0 and 255 and that 8-bit unsigned numbers (i.e., "8u") are needed to represent these possible values. *Id*.

203. Karczewicz-I teaches that bi-directional inter-prediction is performed based on reference blocks that are I-or P-blocks for a current B-block. *See, e.g.*, Ex. 1005, ¶7 ("I-and P-units are commonly used to define reference blocks for the inter-coding of P-and B-units."). Given Karczewicz-II's teaching that 8 bits are needed to represent possible integer pixel values, the combination teaches or suggests that the pixels of the current block, the first reference block, and the second reference block have values with a first precision (e.g., 8 bits).

[1c]/[7c]/[13c]	**[using/use] said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;**

204. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1c], [7c], and [13c].

205. First, Karczewicz-I teaches calculating a bi-directional prediction using a default weighted prediction mode, where equal weights are assigned to two reference blocks. Ex. 1005, ¶55:

> According to the ITU-T H.264/AVC standard, three motion-compensated bi-predictive algorithms or modes may be used to predict a B-frame or portions thereof, such as video blocks, macroblocks or any other discreet and/or contiguous portion of a B-frame. A first motion-compensated bi-predictive algorithm or mode, which is commonly referred to as default weighted prediction, may involve applying default weights to each identified video block of the first frame of list 0 and the second frame of list 1. The default weights may be programmed according to the standard, and are often selected to be equal for default weighted prediction. The weighted blocks of the first and second frames are then added together and divided by the total number of frames used to predict the B-frame, e.g., two in this instance. Often,

> this division is accomplished by adding 1 to the addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit. The addition of 1 is a rounding adjustment.

See also ¶24, ¶44, ¶48.

206. The default weighted prediction is calculated as an average of two predictions (e.g., pred0(i,j) and pred1(i,j), which are prediction data from list 0 and list 1). Ex. 1005, ¶60:

> Default weighted prediction may be defined by the following equations for unidirectional prediction and bidirectional prediction, respectively.
>
> …
>
> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

207. Here, pred0(i,j) is a prediction based on a "motion compensated reference area[] … obtained from list 0 … reference picture." Ex. 1005, ¶¶57-58. Similarly, pred1(i,j) is obtained from the list 1 reference picture. *Id.* Because Karczewicz-I teaches that the inter-prediction process for a current block includes determining reference blocks based on motion vectors (*supra* §V.B.4.a[1b]/[7b]/[13b]; Ex. 1005, ¶7, ¶¶53-54), the motion compensated reference area refers to a reference block.

208. As explained above, a POSITA would have found it obvious, based on Karczewicz-II's teachings of using higher-precision intermediate values, to modify Karczewicz-I's calculation of the bi-directional prediction to use

higher-precision predictions as intermediate values. *Supra* §V.B.3 (explaining how and why a POSITA would have combined teachings from Karczewicz-I and Karczewicz-II; that analysis is incorporated here). This modification is directed to calculation bit depth and order; it therefore does not change the reference blocks on which Karczewicz-I's predictions are based. Thus, the combination of Karczewicz-I and Karczewicz-II teaches obtaining said first prediction using said first reference block.

209. Karczewicz-I contemplates the following three scenarios (among others), where the bi-directional prediction is determined as: (1) an average of a half-pixel prediction and an integer pixel prediction; (2) an average of a center-pixel prediction and a half-pixel prediction; and (3) an average of two half-pixel predictions. *Supra* §V.B.3. In each scenario, the combination of Karczewicz-I and Karczewicz-II teaches that said first prediction having a second precision, which is higher than said first precision.

210. Scenario 1. When Karczewicz-I's bi-directional prediction is calculated as an average of a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)), it would have been obvious to replace the first prediction, which is a half-pixel prediction, with a non-rounded half-pixel prediction. *Supra* §V.B.3. Furthermore, it would have been obvious to replace the second prediction, which is an integer pixel prediction, with a left-shifted version

of the integer pixel prediction. *Id*. Karczewicz-I's equation for calculating the bi-directional prediction would have been modified as shown below:

$$pred(i,j)=(non\text{-}rounded(pred0(i,j))+pred1(i,j)<<5+32)>>6$$

Id.

211. Karczewicz-II teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Karczewicz-II further teaches that the left-shifted integer pixel prediction (e.g., r1<<5) has possible values from 0 to 8160 and that a 13-bit unsigned number (i.e., "13u") is needed to represent these possible values. Ex. 1007, ¶103, Table 5:

TABLE 5

positions {a, c, d, l} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = x	r1 is integer pixel x	0	255	8u
r1 = r1 << 5	r1 is 32 * x	0	8160	13u
r2 = y0	r2 is y0 (y0 is a one-dimensional (1-D) half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is 32 * x + y0	−2550	18870	16s
r1 = r1 + 32	r1 is 32 * x + y0 + 32	−2518	18902	16s
r1 = r1 >> 6	r1 is (32 * x + y0 + 32) >> 6	−39	295	11s
r1 = max (0, r1)	clip r1 on the low side	0	295	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

212. Therefore, 13 bits are needed to represent the possible values of each of the first prediction and the second prediction, since the original integer requires 8 bits and is left shifted by 5 bits in uncompressed binary form. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each need at precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4.a[1b]/[7b]/[13b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred0(i,j)))**, said first prediction having a second precision** (e.g., 13 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 1.

213. Scenario 2. When Karczewicz-I's bi-directional prediction is calculated as an average of a center-pixel prediction (e.g., pred0(i,j)) and a half-pixel prediction (e.g., pred1(i,j)), it would have been obvious to replace the first prediction, which is a center-pixel prediction, with a partially-rounded center-pixel prediction. *Supra* §V.B.3. Furthermore, it would have been obvious to replace the second prediction, which is a half pixel prediction, with non-rounded half-pixel prediction. *Id*. Karczewicz-I's equation for calculating the bi-directional prediction would have been modified as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))>>5+non-rounded(pred1(*i,j*)) +32)>>6

Id.

214. Karczewicz-II teaches that the partially-rounded center-pixel prediction (e.g., j1>>1)[6] has possible values from -4957 to 13116 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex. 1007, ¶105, Table 8. Karczewicz-II further teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex. 1007, ¶105, Table 8:

[6] The equations in Table 3 of Karczewicz-II show that j1 is shifted to the right by 5 bits. Table 8 accomplishes this in two steps. The first step (e.g., "r2 = j1") is to slightly round the value of j1 to fit in a 16-bit register, which shaves off 4 bits from the right side (the least significant bits). Karczewicz-II explains that, "in some cases, slight rounding may be applied to one particular half-pixel value that requires two levels of interpolation in order to ensure that fixed size storage elements (e.g., 16-bit registers) can be used to store any intermediate values." Ex. 1006, ¶59; see also ¶10, ¶39, ¶53. This applies to the center-pixel, which is calculated using two rounds of interpolation. *Supra* §V.B.3. The second step (e.g., "r2 = r2 >> 1") is to right shift by one further bit and bring the total right shift amount to 5 bits.

TABLE 8

positions {f, i, k, n} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = j1	r2 is j1 (2-D half-pixel j1 before shifting down)	−9914	26232	16s
r2 = r2 >> 1	r2 is j1 >> 1	−4957	13116	15s
r1 = r1 + r2	r1 is y0 + (j1 >> 1)	−7507	23826	16s
r1 = r1 + 32	r1 is y0 + (j1 >> 1) + 32	−7491	23842	16s
r1 = r1 >> 6	r1 is (y0 + (j1 >> 1) + 32) >> 6	−235	745	11s
r1 = max (0, r1)	clip r1 on the low side	0	745	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

215. Therefore, 15 bits are required to represent the possible values of each of the first prediction and the second prediction. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each has a precision of 15 bits, which is higher than the 8-bit precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4.a[1b]/[7b]/[13b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., partially-rounded center-pixel prediction, i.e., non-rounded(pred0(i,j))>>5)**, said first prediction having a second precision** (e.g., 15 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 2.

216. Scenario 3. When Karczewicz-I's bi-directional prediction is calculated as an average of two half-pixel predictions (e.g., pred0(i,j) and pred1(i,j)), it would have been obvious to replace the each prediction, which is a half-pixel prediction, with a non-rounded half-pixel prediction. *Supra* §V.B.3. Karczewicz-I's equation for calculating the bi-directional prediction would have been modified as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))+non-rounded(pred1(*i,j*)) +32)>>6

Id.

217. Karczewicz-II teaches that the non-rounded half-pixel predictions (e.g., b1, h1) each has values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex. 1007, ¶103, Table 6:

TABLE 8

positions {f, i, k, n} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = j1	r2 is j1 (2-D half-pixel j1 before shifting down)	−9914	26232	16s
r2 = r2 >> 1	r2 is j1 >> 1	−4957	13116	15s
r1 = r1 + r2	r1 is y0 + (j1 >> 1)	−7507	23826	16s
r1 = r1 + 32	r1 is y0 + (j1 >> 1) + 32	−7491	23842	16s
r1 = r1 >> 6	r1 is (y0 + (j1 >> 1) + 32) >> 6	−235	745	11s
r1 = max (0, r1)	clip r1 on the low side	0	745	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

218. Therefore, 15 bits are required to represent the possible values of each of the first prediction and the second prediction. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each has a precision of 15 bits, which is higher than the 8-bit precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4.a[1b]/[7b]/[13b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred0(i,j))), **said first prediction having a second precision** (e.g., 15 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 3.

[1d]/[7d]/[13d]	**[using/use] said second reference block to obtain a second prediction, said second prediction having the second precision**

219. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1d], [7d], and [13d].

220. As explained above, Karczewicz-I teaches calculating a bi-directional prediction as an average of two predictions (e.g., pred0(i,j) and pred1(i,j)). *Supra* §V.B.4.a[1c]/[7c]/[13c]. Here, pred1(i,j) is a prediction based on a "motion compensated reference area[] … obtained from … list 1 reference picture." Ex. 1005, ¶58. Because Karczewicz-I teaches that the inter-prediction process for a current block includes determining reference blocks based on motion vectors (*supra* §V.B.4.a[1b]/[7b]/[13b]; Ex. 1005, ¶7, ¶¶53-54), the motion compensated reference area refers to a reference block.

221. As explained above, a POSITA would have found it obvious to modify Karczewicz-I's calculation of the bi-directional prediction to use higher-precision versions of the predictions as intermediate values based on Karczewicz-II's teachings. *Supra* §V.B.3 (explaining the motivation to combine; that analysis is incorporated here). While the predictions of Karczewicz-I are modified to be higher-precision versions, the modification does not change the reference blocks on which Karczewicz-I's predictions are based. Thus, the

combination of Karczewicz-I and Karczewicz-II teaches obtaining said second prediction using said second reference block.

222. Karczewicz-I contemplates the following three scenarios (among others), where the bi-directional prediction is determined as: (1) an average of a half-pixel prediction and an integer pixel prediction; (2) an average of a center-pixel prediction and a half-pixel prediction; and (3) an average of two half-pixel predictions. *Supra* §V.B.3.

223. As explained with respect to limitations [1c], [7c], and [13c], in Scenario 1, the combination of Karczewicz-I and Karczewicz-II teach **using said second reference block** (e.g., block from list 1) **to obtain a second prediction** (e.g., left-shifted integer pixel prediction, i.e., pred1(i,j)<<5)**, said second prediction having the second precision** (e.g., 13 bits). *Supra* §V.B.4.a[1c]/[7c]/[13c].

224. In Scenario 2, the combination of Karczewicz-I and Karczewicz-II teach **using said second reference block** (e.g., block from list 1) **to obtain a second prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred1(i,j)))**, said second prediction having the second precision** (e.g., 15 bits). *Id*.

225. In Scenario 3, the combination of Karczewicz-I and Karczewicz-II teach **using said second reference block** (e.g., block from list 1) **to obtain a**

second prediction (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred1(i,j)))**, said second prediction having the second precision** (e.g., 15 bits). *Id*.

[1e]/[7e]/[13e]	**[obtaining/obtain] a combined prediction based at least partly upon said first prediction and said second prediction;**

226. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1e], [7e], and [13e].

227. As explained above, Karczewicz-I's pred0(i,j) and pred1(i,j) respectively satisfy said first prediction and said second prediction. *Supra* §§V.B.4.a[1c-1d]/[7c-7d]/[13c-13d]. Karczewicz-I teaches obtaining a bi-directional prediction by averaging the first and second predictions. Ex. 1005, ¶60:

> Default weighted prediction may be defined by the following equations for unidirectional prediction and bidirectional prediction, respectively.
>
> …
>
> Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1
>
> where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

228. A POSITA would have found it obvious, based on Karczewicz-II's teachings of using higher-precision intermediate values, to modify Karczewicz-I's calculation of averages such that higher-precision versions of predictions are used as intermediate values in calculating the average. *Supra* §V.B.3 (explaining how

and why Karczewicz-II's teachings would have been applied to Karczewicz-I; that analysis is incorporated here).

229. Karczewicz-I contemplates the following three scenarios (among others), where the bi-directional prediction is determined as: (1) an average of a half-pixel prediction and an integer pixel prediction; (2) an average of a center-pixel prediction and a half-pixel prediction; and (3) an average of two half-pixel predictions. *Supra* §V.B.3. Under each scenario, the combination of Karczewicz-I and Karczewicz-II teaches **obtaining a combined prediction** (e.g., a sum of the first prediction, the second prediction, and a rounding offset) **based at least partly upon said first prediction and said second prediction**.

230. Scenario 1. When Karczewicz-I's bi-directional prediction is calculated as an average of a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)), it would have been obvious to modify Karczewicz-I's equation based on Karczewicz-II's teachings, as shown below:

$$pred(i,j)=(non\text{-}rounded(pred0(i,j))+pred1(i,j)<<5+32)>>6$$

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))+pred1(i,j)<<5+32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said

second prediction (e.g., pred1(i,j)<<5), as well as the rounding offset (e.g., 32). *See* Ex. 1007, ¶103, Table 5. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

231. Scenario 2. When Karczewicz-I's bi-directional prediction is calculated as an average of a center-pixel prediction (e.g., pred0(i,j)) and a half-pixel prediction (e.g., pred1(i,j)), it would have been obvious to modify Karczewicz-I's equation based on Karczewicz-II's teachings, as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))>>5+non-rounded(pred1(*i,j*)) +32)>>6

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))>>5+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))>>5) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex. 1007, ¶105, Table 8. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

232. Scenario 3. When Karczewicz-I's bi-directional prediction is calculated as an average of two center-pixel prediction (e.g., pred0(i,j) and

pred1(i,j)), it would have been obvious to modify Karczewicz-I's equation based on Karczewicz-II's teachings, as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))+non-rounded(pred1(*i,j*)) +32)>>6

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex. 1007, ¶103, Table 6. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

[1f]/[7f]/[13f]	**[decreasing/decrease] a precision of said combined prediction by shifting bits of the combined prediction to the right; and**

233. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1f], [7f], and [13f].

234. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches the combined prediction as a sum of the first prediction, the second prediction, and a rounding offset. *Supra* §V.B.4.a[1e]/[7e]/[13e]. Both Karczewicz-I and Karczewicz-II teach the operation ">>" for shifting bits to the

right. Ex. 1005, ¶55 ("Often, this division is accomplished by adding 1 to the addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit."), ¶57 (">> is a right shift operation[.]"), ¶60; Ex. 1007, ¶94 ("In this disclosure, '>>' represents a right shift operation and '<<' represents a left shift operation."). The combination of Karczewicz-I and Karczewicz-II further teaches **decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right** for each of the three scenarios discussed above.

235. Scenario 1. When Karczewicz-I's bi-directional prediction is calculated as an average of a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)), it would have been obvious to modify Karczewicz-I based on Karczewicz-II's teachings such that the calculation is performed on a non-rounded half-pixel prediction and a left-shifted integer pixel prediction. *Supra* §V.B.3. Karczewicz-I's equation for determining the bi-directional prediction would have been modified as shown below:

$$pred(i,j)=(non\text{-}rounded(pred0(i,j))+pred1(i,j)<<5+32)>>6$$

Id. Bits of the combined prediction (e.g., non-rounded(pred0(i,j))+pred1(i,j)<<5+32) is shifted to the right (e.g., >>6).

236. The shifting decreases a precision of said combined prediction. Karczewicz-II's Table 5 teaches the number of bits needed to represent values associated with Scenario 1:

TABLE 5

positions {a, c, d, 1} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = x	r1 is integer pixel x	0	255	8u
r1 = r1 << 5	r1 is 32 * x	0	8160	13u
r2 = y0	r2 is y0 (y0 is a one-dimensional (1-D) half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is 32 * x + y0	−2550	18870	16s
r1 = r1 + 32	r1 is 32 * x + y0 + 32	−2518	18902	16s
r1 = r1 >> 6	r1 is (32 * x + y0 + 32) >> 6	−39	295	11s
r1 = max (0, r1)	clip r1 on the low side	0	295	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

Ex. 1007, ¶103, Table 5. In this table, the operation "r1=r1+32" calculates the sum of a non-rounded half-pixel prediction (e.g., y0 which can take the value of b1), a left-shifted integer pixel prediction (e.g., 32*x), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -2518 and 18902 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values between -39 and 295; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision"

is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

237. Scenario 2. When Karczewicz-I's bi-directional prediction is calculated as an average of a center-pixel prediction (e.g., pred0(i,j)) and a half-pixel prediction (e.g., pred1(i,j)), it would have been obvious to modify Karczewicz-I based on Karczewicz-II's teachings such that the calculation is performed on a partially-rounded center-pixel prediction and a non-rounded half-pixel prediction. *Supra* §V.B.3. Karczewicz-I's equation for determining the bi-directional prediction would have been modified as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))>>5+non-rounded(pred1(*i,j*)) +32)>>6

Id. Bits of the combined prediction (e.g., non-rounded(pred0(i,j))>>5+non-rounded(pred1(i,j)) +32) is shifted to the right (e.g., >>6).

238. The shifting decreases a precision of said combined prediction. Karczewicz-II's Table 8 teaches the number of bits needed to represent values associated with Scenario 2:

TABLE 8

positions {f, i, k, n} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = j1	r2 is j1 (2-D half-pixel j1 before shifting down)	−9914	26232	16s
r2 = r2 >> 1	r2 is j1 >> 1	−4957	13116	15s
r1 = r1 + r2	r1 is y0 + (j1 >> 1)	−7507	23826	16s
r1 = r1 + 32	r1 is y0 + (j1 >> 1) + 32	−7491	23842	16s
r1 = r1 >> 6	r1 is (y0 + (j1 >> 1) + 32) >> 6	−235	745	11s
r1 = max (0, r1)	clip r1 on the low side	0	745	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

Ex. 1007, ¶105, Table 8. In this table, the operation "r1=r1+32" calculates the sum of a partially-rounded center-pixel prediction (e.g., j1>>1), a non-rounded half-pixel prediction (e.g., y0, which may take the value of b1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -7491 and 23842 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values between -235 and 745; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent a value." *Supra* §IV.A. Because the number of bits needed to

represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

239. Scenario 3. When Karczewicz-I's bi-directional prediction is calculated as an average of two center-pixel prediction (e.g., pred0(i,j) and pred1(i,j)), it would have been obvious to modify Karczewicz-I based on Karczewicz-II's teachings such that the calculation is performed on two non-rounded half-pixel predictions. *Supra* §V.B.3. Karczewicz-I's equation for determining the bi-directional prediction would have been modified as shown below:

pred(*i,j*)=(non-rounded(pred0(*i,j*))+non-rounded(pred1(*i,j*)) +32)>>6

Id. Bits of the combined prediction (e.g., non-rounded(pred0(i,j))+non-rounded(pred1(i,j)) +32) is shifted to the right (e.g., >>6).

240. The shifting decreases a precision of said combined prediction. Karczewicz-II's Table 6 teaches the number of bits needed to represent values associated with Scenario 3:

TABLE 6

positions {e, g, m, o} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (y0 is a 1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = y1	r2 is y1 (y1 is a 1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is y0 + y1	−5100	21420	16s
r1 = r1 + 32	r1 is y0 + y1 + 32	−5068	21452	16s
r1 = r1 >> 6	r1 is (y0 + y1 + 32) >> 6	−79	335	11s
r1 = max (0, r1)	clip r1 on the low side	0	335	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

Ex. 1007, ¶103, Table 6. In this table, the operation "r1=r1+32" calculates the sum of two non-rounded half-pixel predictions (e.g., y0 and y1, which may take the values of b1 and h1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values from -5068 to 21452 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values from - 79 to 335; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is

decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

[1g]/[7g]/[13g]	**[encoding/encode] residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.**

241. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1g], [7g], and [13g].

242. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches determining the combined prediction in the motion compensation process. *Supra* §§V.B.4.a[1b-1f]/[7b-7f]/[13b-13f]. Karczewicz-I teaches **determining residual data** (e.g., a residual video block) **based upon a difference between the combined prediction** (e.g., prediction data) **and the block of pixels** (e.g., original video block). Ex. 1005, ¶73 ("Once the desired prediction data is identified by motion compensation unit 35, as described herein, video encoder 50 forms a residual video block by subtracting the prediction data from the original video block being coded."), ¶7 ("After motion compensation, a residual video block is formed by subtracting the prediction video block from the original video block to be coded."), Fig. 2. Here, the residual video block includes residual data. *See* Ex. 1005, ¶84 ("residual data (e.g., a residual block)"), ¶89. Karczewicz-II includes similar teachings. Ex. 1007, ¶4, ¶6, ¶35, ¶50, ¶58, ¶60, ¶73, Fig. 2.

243. Karczewicz-I further teaches **encoding** (e.g., transforming, quantizing, and entropy coding) **residual data in a bitstream**. Ex. 1005, ¶73 ("Transform unit 38 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform block coefficients. Transform unit 38, for example, may perform other transforms, such as those defined by the H.264 standard, which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform unit 38 applies the transform to the residual block, producing a block of residual transform coefficients."), ¶74 ("Quantization unit 40 quantizes the residual transform coefficients to further reduce bit rate."), ¶75 ("The coded bitstream may include entropy coded residual blocks, motion vectors for such blocks, and other syntax such as the syntax described herein."), Fig. 2. Karczewicz-II similarly teaches encoding residual data in a bitstream. Ex. 1007, ¶5, ¶35, ¶50, ¶¶60-61, ¶110, Fig. 2.

b. Dependent Claims 2, 8, and 14

2. The method according to claim 1,
8. The apparatus according to claim 7,
14. The computer program product according to claim 13,

wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

244. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §§V.B.3-4. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitations of claims 2, 8, and 14.

245. Karczewicz-II teaches obtaining predictions by interpolation using pixel values of reference blocks when motion vectors point to subpixels. Ex. 1007, ¶7 ("In this case, the predictive data generated during motion compensation, which is used to code a video block, may be interpolated from the pixels of video blocks of the video frame or other coded unit used in motion estimation. Interpolation is often performed to generate predictive half-pixel values (half-pel) and predictive quarter-pixel values (quarter-pel)."), ¶42 ("The interpolation techniques of this disclosure may be performed by any encoding device that supports motion compensated interpolation to sub-pixel resolution."), ¶66 ("Again, the techniques of this disclosure concern motion compensated interpolation in which pixel values of predictive video blocks are interpolated to sub-pixel resolution."), ¶¶68-72, Figs. 4A-4D.

246. As explained above, the combination of Karczewicz-I and Karczewicz-II contemplates at least three scenarios for determining a bi-directional

prediction. Karczewicz-I and Karczewicz-II's teachings with respect to each of the three scenarios satisfy the additional limitations of claims 2, 8, and 14.

247. Scenario 1. In this scenario, the first motion vector points to a half-pixel position. *Supra* §V.B.3. The half-pixel position refers to a subpixel. *See, e.g.*, Ex. 1007, ¶74 ("For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample 'C3' and labeled 'a' through 'o' in FIGS. 4A-4D."), Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

248. Karczewicz-II teaches obtaining the first prediction (e.g., the non-rounded half-pixel prediction) by interpolation using pixel values of said first reference block. Ex. 1007, ¶93:

> A sub-pixel motion vector refers to a sub-pixel position in a reference picture which needs to be interpolated. H.264 defines one interpolation process for sub-pixels in which sub-pixels b and h (see FIGS. 4A-4D) may be calculated by horizontal and vertical filtering with a 6-tap filer having tap values (1, −5, 20, 20, −5, 1) as follows:
>
> b1=C1−5*C2+20*C3+20*C4−5*C5+C6
>
> where "C1," "C2," "C3," "C4," "C5" and "C6" represent the six closest integer pixels that surround "b" in the horizontal direction, with pixels"C3" and "C4" being the closest, "C2" and "C5" being the next closest, and "C1" and "C6" being the next closest.

The interpolation is performed using pixel values of "six closest integer pixels that surround" the subpixel. Because the integer pixels are closest to the subpixel, they

are located in the reference block that the first motion vector points to. *See* Ex. 1007, Fig. 4B:

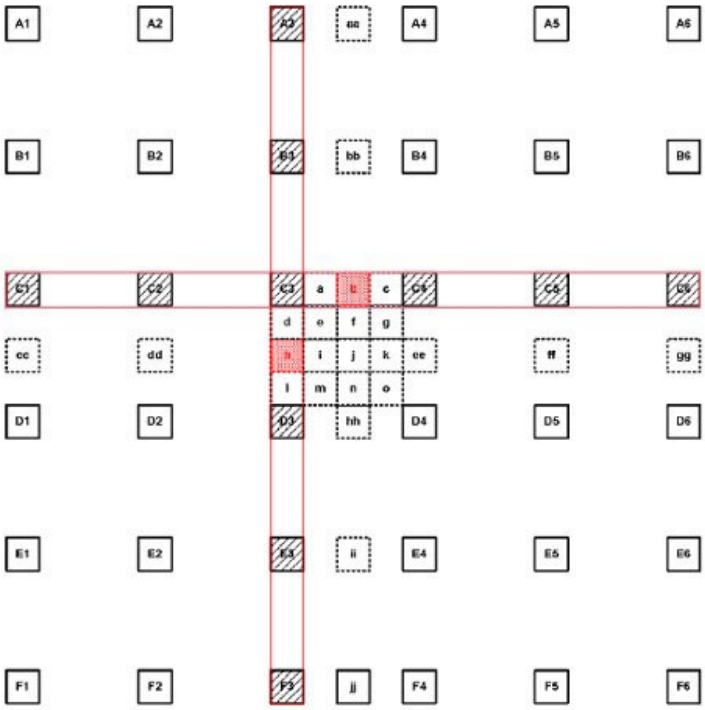


FIG. 4B

Thus, the interpolation is performed using pixel values of said first reference block.

249. Scenario 2. In this scenario, the first motion vector points to a center-pixel position. *Supra* §V.B.3. The center-pixel position refers to a subpixel. *See, e.g.*, Ex. 1007, ¶74, Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

250. Karczewicz-II teaches obtaining the first prediction (e.g., the partially-rounded center-pixel prediction) by interpolation using pixel values of said first reference block. Ex. 1007, ¶95:

> To interpolate sub-pixel "j," an intermediate value "j1" is first derived as:
>
> j1=aa1−5*bb1+20*b1+20*hh1−5*ii1+jj1,
>
> where the intermediate values denoted as "aa1," "bb1", "hh1," "ii1" and "jj1" are derived by applying the 6-tap filter horizontally in the same manner as the calculation of b1 at the positions of "aa," "bb," "hh," "ii" and "jj."

The interpolation is a two step process performed using non-rounded half-pixel predictions, which are in turn obtained by interpolation using integer pixel values. The half-pixel positions and their corresponding integer pixel positions are close to the center-pixel position and are located in the reference block that the first motion vector points to. *See* Ex. 1007, Fig. 4C:

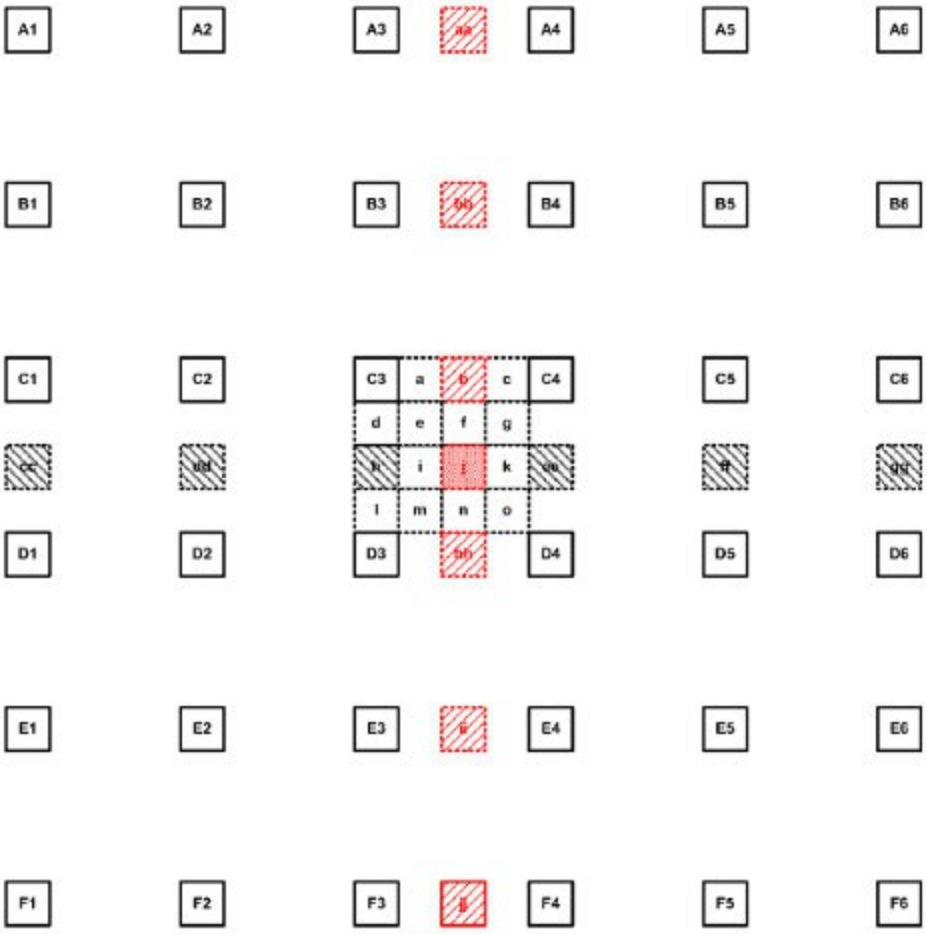


FIG. 4C

Thus, the interpolation is performed using pixel values of said first reference block.

251. Scenario 3. In this scenario, the first motion vector points to a half-pixel position. *Supra* §V.B.3. For the same reasons as explained for Scenario 1, the combination of Karczewicz-I and Karczewicz-II teaches that, in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

252. Moreover, the ’267 patent admits that the limitations of claims 2, 8, and 14 were known in the prior art by describing it in the “Background Information” section. Ex. 1001, 2:60-3:11.

c. Dependent Claims 3, 9, and 15

3. The method according to claim 2,
9. The apparatus according to claim 8,
15. The computer program product according to claim 14,

wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.

(i) Ground 2

253. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.B.4.b. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 3, 9, and 15 under Scenario 2.

254. In Scenario 2, the combination of Karczewicz-I and Karczewicz-II teaches using the following equation for calculating the bi-directional prediction:

pred(*i,j*)=(non-rounded(pred0(*i,j*))>>5+non-rounded(pred1(*i,j*)) +32)>>6

Supra §V.B.3. In this equation, the first prediction "non-rounded(pred0(i,j))>>5" is a partially-rounded center-pixel prediction. *Supra* §V.B.3, §V.B.4.a[1c]/[7c]/[13c]. As explained for claims 2, 8, and 14, the partially-rounded center-pixel prediction is determined by interpolation using values of said first reference block. *Supra* §V.B.4.b.

255. Karczewicz-II teaches calculating the partially-rounded center-pixel prediction (e.g., j1>>5) by first determining a sum of a 6-tap filter (e.g., j1). Ex. 1007, ¶95:

> To interpolate sub-pixel "j," an intermediate value "j1" is first derived as:
>
> j1=aa1−5*bb1+20*b1+20*hh1−5*ii1+jj1,
>
> where the intermediate values denoted as "aa1," "bb1", "hh1," "ii1" and "jj1" are derived by applying the 6-tap filter horizontally in the same manner as the calculation of b1 at the positions of "aa," "bb," "hh," "ii" and "jj."

Supra §V.B.4.b. This sum is a non-rounded version of the center-pixel prediction.

256. Next, the non-rounded center-pixel prediction is shifted to the right by 5 bits to obtain the partially-rounded center-pixel prediction (j1>>5). *Supra* §V.B.3; *see, e.g.*, Ex. 1007, 99, Table 3:

TABLE 3

a = (C3 << 5 + b1 + 32) >> 6
c = (C4 << 5 + b1 + 32) >> 6
d = (C3 << 5 + h1 + 32) >> 6
l = (D3 << 5 + h1 + 32) >> 6
f = (j1 >> 5 + b1 + 32) >> 6
i = (j1 >> 5 + h1 + 32) >> 6
k = (j1 >> 5 + ee1 + 32) >> 6
n = (j1 >> 5 + hh1 + 32) >> 6

Table 8 further teaches operations implementing the equations that include shifting bits to the right, consistent with Table 3. Ex. 1007, ¶105, Table 8.

257. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches that **said first prediction** (e.g., partially-rounded center-pixel prediction) **is obtained by interpolation using values of said first reference block by: right shifting** (e.g., >>5) **a sum of a P-tap filter** (e.g., non-rounded center-pixel prediction) **using values of said first reference block**.

258. Moreover, the ’267 patent admits that a P-tap filter that averages pixel values was known in the prior art by describing it in the “Background Information” section. Ex. 1001, 2:60-3:11.

(ii) Ground 2a

259. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.B.4.b. Srinivasan in addition to

the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 3, 9, and 15 and renders those dependent claims obvious.

260. Karczewicz-II discloses using a variety of different interpolation filters. Ex. 1007, ¶9 ("[D]ifferent numbers of filter coefficients (different number of taps) or different types of filters altogether may be pre-defined, and then selected and used during the encoding and decoding processes."). It would have been obvious to a POSITA for Karczewicz-II encoder and decoder to use the VC-1 interpolation filter described by Srinivasan as one of the available interpolation filters. Accordingly, the two-stage filtering process described by Srinivasan would be used during interpolation of the ½ pixel in location j, for example, which requires interpolation in two dimensions. As Srinivasan describes, the first stage would use a four-tap filter to calculate intermediate values from integer pixel values in the reference frame and then right-shift those intermediate values to limit them to 16 bits. Ex. 1008, 000015. That first stage discloses the added limitation in claims 3, 9, and 15.

261. A POSITA would have been motivated to combine the prior art in this manner. Both Karczewicz-II and Srinivasan teach the use of higher precision intermediate values during sub-pixel interpolation to improve accuracy. *Compare* Ex. 1007, ¶102 ("By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate.") *with* Ex. 1008, 000015 ("possible

to achieve additional accuracy by retaining a higher precision result after the first stage of filtering"). Moreover, both Karczewicz-II and Srinivasan recognize that some rounding may be necessary to limit intermediate values to 16 bits. *Compare* Ex. 1007, ¶59 ("slight rounding may be applied … to ensure that fixed size storage elements (*e.g.,* 16-bit registers) can be used to store any intermediate values") *with* Ex. 1008, 000015 ("shifts are chosen … to allow for a 16 bit implementation — where the intermediate values in the second filtering operation are within 16 bits").

262. Additionally, in my opinion a POSITA would have had a reasonable expectation of success in combining these teachings as proposed. Karczewicz-II discloses partial rounding of intermediate values to fit within 16-bit registers and for averaging with lower-precision intermediate values for other interpolated sub-pixels. Srinivasan discloses a similar approach for the same reasons during sub-pixel interpolation in two dimensions. As I noted, Karczewicz-II also discloses that a variety of different interpolation filters can be used. Although Srinivasan describes other aspects of VC-1, the Petition does not propose to incorporate any aspect of VC-1 beyond Srinivasan's teachings regarding high-precision sub-pixel interpolation that I discussed above.

d. Dependent Claims 4, 10, and 16

4. The method according to claim 2, **10. The apparatus according to claim 8,** **16. The computer program product according to claim 14,**

	wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

263. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.B.4.b. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 4, 10, and 16 under Scenario 1.

264. In Scenario 1, the second motion vector points to an integer pixel position. *Supra* §V.B.3. This is **an instance in which said second motion vector points to an integer sample**.

265. The combination of Karczewicz-I and Karczewicz-II teaches using the following equation for calculating the bi-directional prediction:

$$pred(i,j)=(non\text{-}rounded(pred0(i,j))+pred1(i,j)<<5+32)>>6$$

Supra §V.B.3. In this equation, the second prediction "pred1(i,j)<<5" is a left-shifted integer pixel prediction. *Supra* §V.B.3, §§V.B.4.a[1c-1d]/[7c-7d]/[13c-13d].

266. Karczewicz-II teaches calculating the left-shifted integer pixel prediction by shifting the pixel value to the left (e.g., C3<<5). *Supra* §V.B.3; *see, e.g.*, Ex. 1007, ¶99, Table 3:

TABLE 3

a = (C3 << 5 + b1 + 32) >> 6
c = (C4 << 5 + b1 + 32) >> 6
d = (C3 << 5 + h1 + 32) >> 6
l = (D3 << 5 + h1 + 32) >> 6
f = (j1 >> 5 + b1 + 32) >> 6
i = (j1 >> 5 + h1 + 32) >> 6
k = (j1 >> 5 + ee1 + 32) >> 6
n = (j1 >> 5 + hh1 + 32) >> 6

Here, the value that is shifted to the left is the value of an integer pixel sample (e.g., C3). Ex. 1007, ¶74 ("For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample 'C3' and labeled 'a' through 'o' in FIGS. 4A-4D."), ¶93 ("where 'C1,' 'C2,' 'C3,' 'C4,' 'C5' and 'C6' represent the six closest integer pixels that surround 'b' in the horizontal direction"). The left shift by 5 bits allows the integer pixel values to match the increased precision of the interpolated sub-pixels (e.g., b1, h1). Because the second motion vector points to this integer pixel sample, the integer pixel sample is part of the second reference block; its value is a value of said second reference block.

267. The teachings of Karczewicz-I and Karczewicz-II above explain calculations for one pixel in a block. Karczewicz-I and Karczewicz-II teach that motion compensation is performed on a block basis. *See supra* §§V.B.4.a[1b,

1g]/[7b, 7g]/[13b, 13g]; Ex. 1005, ¶7; Ex. 1007, ¶4, ¶73. Therefore, the references teach performing the predictions operations on all the pixels of a block. As explained above, in motion estimation and compensation, a motion vector indicates the displacement of between a reference block and a current block of pixels. *Supra* §V.B.4.a[1b]/[7b]/[13b]; Ex. 1005, ¶7, ¶¶53-54; Ex. 1007, ¶4, ¶56. Therefore, Karczewicz-I and Karczewicz-II teach performing the above operations, including the left-shift operation, for each pixel of the current block based on corresponding pixels of the reference block. In Scenario 1, the left shifting is performed for multiple pixels.

268. Karczewicz-I's calculations take an average of two pixels for the purpose of biprediction; it would have been obvious to apply Karczewicz-II's improvements to these calculations. *Supra* §V.B.3. It would have been obvious to apply a left shift to the integer pixel values when the second motion vector for the list 1 reference image points to an integer pixel and the first motion vector for the list 0 reference image points to an interpolated half-pixel (Scenario 1).

269. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches that in an instance in which said second motion vector points to an integer sample, said second prediction (e.g., left-shifted integer pixel prediction) is obtained by shifting values of said second reference block (e.g., values of integer pixels) to the left (e.g., <<5).

e. Dependent Claims 5, 11, and 17

5. The method according to claim 1, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises:

11. The apparatus according to claim 7, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by:

17. The computer program product according to claim 13, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:

[inserting/insert] a rounding offset to the combined prediction before said decreasing.

270. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §V.B.4. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitations of claims 5, 11, and 17.

271. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches obtaining a combined prediction via their disclosure of calculating a sum by adding up the first prediction, the second prediction, and the rounding offset (e.g., 32). *Supra* §V.B.4.a[1e]/[7e]/[13e]. The combination teaches decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right. *Supra* §V.B.4.a[1f]/[7f]/[13f]. The combination thus teaches **inserting a rounding offset** (e.g., 32) **to the combined prediction**, as explained

for limitations [1e], [7e], and [13e], **before said decreasing** of the precision, as explained for limitations [1f], [7f], and [13f].

272. The value added to the sum of the first and second predictions is a rounding offset. Karczewicz-I refers to the term that is added to the weighted sum of the first and second predictions as a rounding adjustment. Ex. 1005, ¶63 ("Generally, a rounding adjustment of 2^{r-1} is commonly used prior to a right shift by r, where r represents a positive integer."), ¶55.[7] As was well known to those skilled in the art, "rounding adjustment" was used interchangeably with "rounding offset." The value (e.g., 32) is inserted to the combined prediction, increasing the value of the combined prediction, right before the rounding operation. A POSITA would have understood that this value is a rounding offset according to the plain meaning of the term.

273. Karczewicz-I and Karczewicz-II teach this rounding offset as part of its rounding process, which decreases the precision as recited by the claims. Ex. 1005, ¶55, ¶63; Ex. 1007, ¶¶96-101, Tables 1-4. Additionally, it was obvious to include the rounding offset to the combined prediction (claim 5) prior to decreasing the precision because the insertion of the rounding offset is performed

[7] Karczewicz-I teaches a rounding adjustment of 2^{r-1} prior to a right shift by r. This is consistent with the modified equation for calculating the bi-directional prediction under each of the three scenarios as taught by the combination of Karczewicz-I and Karczewicz-II (*supra* §V.B.3), which teaches right shifting by 6 bits and a rounding offset of 32: 26-1=25=32.

immediately before the right-shifting to affect the direction of the rounding. The combination includes a rounding offset to control rounding error resulting from the right-shift operation that decreases precision. This was common in the art. *Supra* §I.D.

274. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the at least one memory and computer program code are configured to cause the apparatus to perform operations that render obvious limitation [7e] and [7f]. *Supra* §V.B.4[7a]. Because it would have been obvious for decreasing said precision to comprise inserting a rounding offset, it would have been obvious that **the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right by** inserting a rounding offset to the combined prediction before said decreasing (claim 11).

275. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the program code instructions are configured to perform the operations recited in claim 13, including limitations [13e] and [13f]. *Supra* §V.B.4.a[13a]. Because it would have been obvious for decreasing said precision to comprise inserting a rounding offset, it would have been obvious that **the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right**

further comprise program code instructions configured to inserting a rounding offset to the combined prediction before said decreasing (claim 17).

f. Dependent Claims 6, 12, and 18

6. The method according to claim 1,
12. The apparatus according to claim 7,
18. The computer program product according to claim 13,

> **wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.**

276. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §V.B.4.a. As explained below, it is my opinion that the combination further teaches the additional limitations of claims 6, 12, and 18.

277. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision because 8 bits are needed to represent the possible pixel values of these blocks. *Supra* §V.B.4.a[1b]/[7b]/[13b]. Here, **the first precision indicates a number of bits needed to represent the values of the pixels**.

278. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches said first prediction and second prediction having a second

precision that is higher than said first precision because more bits are needed to represent the possible values of the predictions under each of Scenarios 1, 2, and 3. *Supra* §§V.B.4.a[1c-1d]/[7c-7d]/[13c-13d]. In Scenario 1, 13 bits are needed to represent the possible values of the first prediction and the second prediction. *Id*. In Scenarios 2 and 3, 15 bits are needed to represent the possible values of the first prediction and the second prediction. *Id*. Here, **the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction**.

5. Decoder Claims

279. The decoder-side claims of the '267 patent recite almost identical limitations to the encoder-side claims I discussed above. Karczewicz-I discloses that the decoder performs "the reciprocal decoding techniques to the encoding techniques described" for the encoder including support for weighted prediction. Ex. 1005, ¶83, *see also id.*, ¶¶84-86, 88-89 (discussing decoding including block based predictive decoding using weighted prediction data); ¶44 ("For different types of weighted bi-directional prediction in accordance with ITU-T H.264, … video *decoder* 28 may generally support three different types of prediction modes."). Moreover, Karczewicz-II is directed to "digital video coding" and "interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex. 1007, ¶2

280. Given the high degree of similarity of the claims, my discussion below for the decoder-side claims refers back to my detailed discussion above of similar limitations of the encoder-side claims.

a. Independent Claims 19, 25, and 31

[19a]	**A method for decoding a block of pixels, the method comprising:**

281. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

282. Karczewicz-I "describes video encoding and decoding techniques applicable to bi-directional prediction." Ex. 1005, ¶8, ¶22 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction."). Karczewicz-I describes methods for decoding blocks of pixels. *Id.*, ¶39 ("Video encoder 22 and video decoder 28 may operate on video blocks within individual video frames in order to encode and decode the video data."); ¶8 ("In bi-directional prediction, a video block may be predictively encoded and decoded based on two different lists of predictive reference data.").

283. Similar to Karczewicz-I, Karczewicz-II decodes a block of pixels. *See, e.g.*, Ex. 1007, ¶64 ("Motion compensation unit 55 produces motion compensated blocks in the manner described herein, e.g., including interpolation

based on a set of interpolation filter coefficients identified by the syntax element (i.e., the interpolation syntax)."), ¶66 ("Again, the techniques of this disclosure concern motion compensated interpolation in which pixel values of predictive video blocks are interpolated to sub-pixel resolution."), ¶50 ("In general, macroblocks (MBs) and the various sub-blocks may be considered to be video blocks.").

284. Karczewicz-II teaches interpolation methods for use in video decoding. *See, e.g.*, Ex. 1007, ¶14 ("In another example, this disclosure provides a method of interpolating predictive video data for video coding."), ¶35 ("This disclosure describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding."). As explained above, a POSITA would have found it obvious to modify Karczewicz-I's technique for obtaining a combined prediction, which is part of Karczewicz-I's video decoding method, based on Karczewicz-II's interpolation techniques. *Supra* §V.B.3. Ground 2 teaches or suggests these limitations as discussed.

[25a]	**An apparatus for decoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:**

285. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

Karczewicz-I and Karczewicz-II teach a video decoder performing video decoding operations. Ex. 1005, ¶¶29-30, ¶33, Fig. 1, ¶¶83-86, Fig. 4:

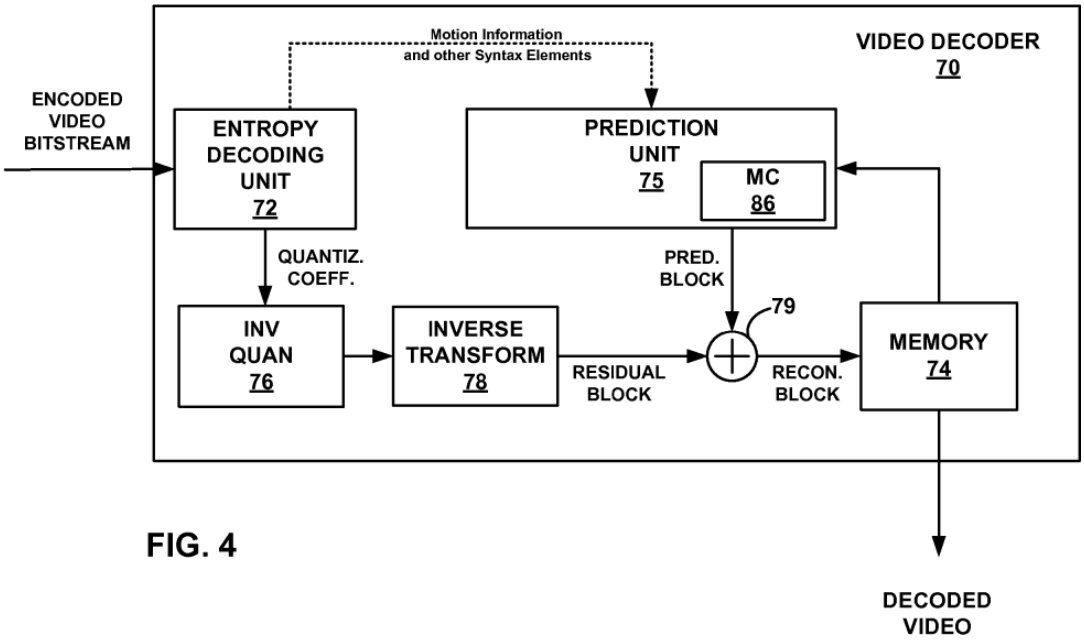


FIG. 4

Ex. 1007, ¶¶40-41, ¶44, Fig. 1, ¶¶63-66, Fig. 3:

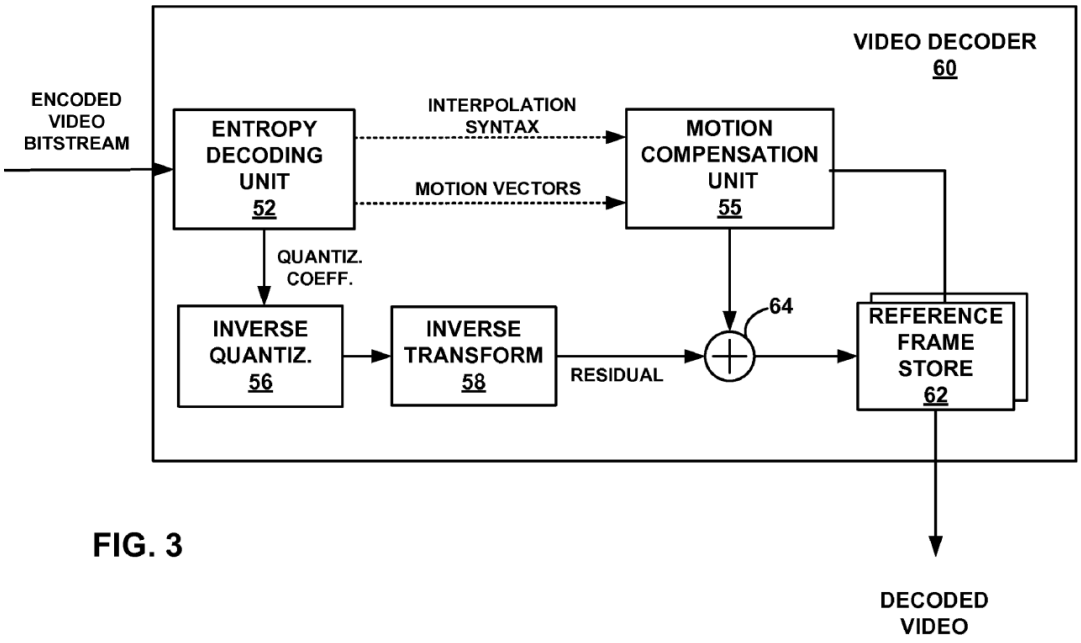


FIG. 3

286. Karczewicz-II, for example, teaches that the video decoder is an apparatus for decoding video. Ex. 1007, ¶15 ("[T]his disclosure provides an apparatus that decodes video data …"), ¶17, ¶22, ¶23. Karczewicz-I and Karczewicz-II teach the video decoder decodes a block of pixels. *See, e.g.*, Ex. 1005, ¶¶39-40; Ex. 1007, ¶50, ¶64, ¶66. Karczewicz-I and Karczewicz-II teach implementing the video decoder in various types of devices. Ex. 1005, ¶3 ("Digital multimedia capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless communication devices, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, video gaming devices, video game consoles, cellular or satellite radio telephones, digital media players, and the like."); Ex. 1007, ¶3.

287. Karczewicz-I and Karczewicz-II teach implementing the video decoder and performing operations for decoding a block of pixels using a processor and memory (e.g., a computer readable medium) that includes computer code (e.g., software). *Supra* §V.B.4.a; Ex. 1005 ¶12, ¶38, ¶98; Ex. 1007, ¶25, ¶118.

288. Karczewicz-I and Karczewicz-II explain that the computer readable medium comprises well-known types of memories. *Supra* §V.B.4.a; Ex. 1005, ¶99; Ex. 1007, ¶119. Karczewicz-I and Karczewicz-II further explain that its software includes computer program code. *Supra* §V.B.4.a; Ex. 1005, ¶100; Ex. 1007, ¶26, ¶120.

289. Karczewicz-I and Karczewicz-II further teaches the processor executing computer program code in memory to cause the apparatus to carry out its functionalities. *Supra* §V.B.4.a; Ex. 1005, ¶99, ¶100; Ex. 1007, ¶48, ¶¶119-120.

290. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches an apparatus for decoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to perform operations as described for limitations [25b]-[25g].

[31a]	**A computer program product for decoding a block of pixels, the computer program product comprising at least one**

	non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:

291. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

292. As I just discussed for preamble [25a], Karczewicz-I and Karczewicz-II teach computer-based decoders that execute installed software. Karczewicz-I specifically describes "a computer-readable medium comprising instructions that, when executed in a processor, performs" the described decoding methods. Ex. 1005, ¶99.

293. The remaining elements of this limitation are identical to limitation [13a]. Ground 2 teaches or suggests these limitations as discussed. *Supra* §V.B.4.a.

[19b], [25b], [31b]	**[determining/determine], for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;**

294. Limitations [19b], [25b], and [31b] are identical to limitations [1b], [7b], and [13b]. Ground 2 teaches or suggests these limitations as discussed. *Supra* §V.B.4.a.

[19c], [25c], [31c]	**[using/use] said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;**
[19d], [25d], [31d]	**[using/use] said second reference block to obtain a second prediction, said second prediction having the second precision;**

295. Limitations [19c], [25c], and [31c] are identical to limitations [1c], [7c], and [13c] and limitations [19d], [25d], and [31d] are identical to limitations [1d], [7d], and [13d]. Ground 2 teaches or suggests all these limitations as I discussed. *Supra* §V.B.4.a.

[19e], [25e], [31e]	**[obtaining/obtain] a combined prediction based at least partly upon said first prediction and said second prediction;**

296. Limitations [19e], [25e], and [31e] are identical to limitations [1e], [7e], and [13e]. Ground 2 teaches or suggests these limitations as I discussed. *Supra* §V.B.4.a.

[19f], [25f], [31f]	**a precision of said combined prediction by shifting bits of the combined prediction to the right; and**

297. Limitations [19f], [25f], and [31f] are identical to limitations [1f], [7f], and [13f]. Ground 2 teaches or suggests these limitations as I discussed. *Supra* §V.B.4.a.

[19g], [25g], [31g]	**[reconstructing/reconstruct] the block of pixels based on the combined prediction.**

298. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [19g], [25g], and [31g].

299. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches determining the combined prediction in the motion compensation process. *Supra* §V.B.4.a[19b-19f]/[25b-25f]/[31b-31f]. Karczewicz-I teaches reconstructing the block of pixels (e.g., reconstructed video block) based on the combined prediction (e.g., prediction block). Ex. 1005, ¶86 ("Adder 79 combines the prediction data (e.g., a prediction block) generated by prediction unit 75 with the residual block from inverse transform unit 78 to create a reconstructed video block, which may be stored in memory 74 and/or output from video decoder 70 as decoded video output."), ¶89 ("[V]ideo decoder 70 may invoke adder 79 to combine weighted prediction data (e.g., a prediction block) with residual video data (e.g., a residual block) in order to generate a reconstruction of the video data (e.g., a reconstructed video block)."); Figs. 4, 6. Karczewicz-II includes similar teachings. Ex. 1007, ¶6, ¶65, ¶111, Figs. 3, 6.

b. Dependent Claims 20, 26, and 32

20. The method according to claim 19, **26. The apparatus according to claim 25,** **32. The computer program product according to claim 31,**
wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

300. These claims add the same limitations as encoder-side dependent claims 2, 8, and 14 and are obvious for the same reasons in my opinion. *Supra* §V.B.4.b.

c. Dependent Claims 21, 27, and 33

21. The method according to claim 20,
27. The apparatus according to claim 26,
33. The computer program product according to claim 32,

wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.

301. These claims add the same limitations as encoder-side dependent claims 3, 9, and 15 and are obvious for the same reasons in my opinion. *Supra* §V.B.4.c.

d. Dependent Claims 22, 28, and 34

22. The method according to claim 20,
28. The apparatus according to claim 26,
34. The computer program product according to claim 32,

wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

302. These claims add the same limitations as encoder-side dependent claims 4, 10, and 16 and are obvious for the same reasons in my opinion. *Supra* §V.B.4.d.

e. Dependent Claims 23, 29, and 35

23. The method according to claim 19, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises:

29. The apparatus according to claim 25, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by:

35. The computer program product according to claim 31, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:

> **[inserting/insert] a rounding offset to the combined prediction before said decreasing.**

303. These claims add the same limitations as encoder-side dependent claims 5, 11, and 17 and are obvious for the same reasons in my opinion. *Supra* §V.B.4.e.

f. Dependent Claims 24, 30, and 36

24. The method according to claim 19,

30. The apparatus according to claim 25,

36. The computer program product according to claim 31,

> **wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.**

304. These claims add the same limitations as encoder-side dependent claims 6, 12, and 18 and are obvious for the same reasons in my opinion. *Supra* §V.B.4.f.

VI. BACKGROUND AND QUALIFICATIONS

305. This section contains a summary of my educational background, career history, publications, and other relevant qualifications. My full curriculum vitae is attached as Appendix 1 to this declaration.

306. I earned a Bachelor of Science degree in Physics from the University of Durham, England, in 1979. I obtained a Doctorate in Physics from the University of Durham, England in 1986. Between obtaining my undergraduate and doctoral degree, I developed a microcomputer system for detecting coalmine fires and heatings as a scientist for the National Coal Board and worked as a software engineer for Laser-Scan Ltd. in Cambridge, England.

307. After obtaining my Doctorate, I served as a Research Assistant at University College London from September 1986 to June 1987, where I developed digital image processing algorithms to improve image and stereo-matching quality for a digital terrain modeling system, including software and algorithms for affine transformation, edge filtering, kriging interpolation, and image stereo-matching with sub-pixel acuity. I continued my work with digital image processing as a Research Associate at the University of Maryland, from June 1987 to September 1988. During my time at the University of Maryland, I designed algorithms for filtering, segmenting, clustering, and path planning based on digital images organized by quad-tree data structures.

308. From September 1988 to June 1994, I worked as a Senior Systems Engineer for the Hughes STX Corporation. As part of my work, I developed methods for comparison of sky maps from the Cosmic Background Explorer (COBE) mission with sky maps from other missions based on scientific data stored in a spatially-referenced database using a quad-tree data structure. In this role, I led the Systems Engineering and end-to-end development of a novel system for compressing imaging and ancillary data that combined scientific modeling with statistical data compression. I was also charged with designing and developing evaluation tools to ensure user-transparent, system-wide compression of a 380-GB dynamic database at an image quality acceptable to end-user scientists. In public recognition of my work, I received National Aeronautics and Space Administration Group Achievement Awards in 1990 and 1992.

309. After June 1994, I began a six-month stint as a contract Software Engineer for the Federal National Mortgage Association in Washington D.C., for which I developed a graphical user interface to monitor and validate loan servicer input for a Loss Mitigation Project. I then served as an Independent Consultant to Optivision, Inc. for the next six months, where I researched and developed rate control algorithms and software based on the MPEG-2 Test Model 5 for the OPTIVideo^{TM} MPEG-2 video encoder, as well as adaptive quantization algorithms based on the then-JPEG-3 draft standard. In this role, I researched and developed

algorithms to improve the quality of gray scale image compression for the medical imaging DICOM Standard by providing a lossless hybrid algorithm encoding image residuals with a diagonal Golomb code based an Enhanced Universal Trellis Coded Quantization algorithm.

310. Between December 1995 and March 1996, I served as a Senior Staff Engineer/Firmware Engineer for General Instrument Inc., Comstream Inc., and Armor Safe Technologies Inc. At Comstream, I worked on integrating an MPEG-2 set top box with OpenTV interactive television middleware programmed in the Microtec C language ported to a Motorola 68340 processor under the pSOS operating system.

311. From January 1996-97, I was the sole proprietor of Anugraha, where I researched and developed algorithms and processes to compress fine art photography at an image quality acceptable to artists based on the JPEG imaging standard implemented with image pre-processing and adaptive quantization. For the next year or so, I worked as an engineering contractor or consultant for various companies, working primarily on image processing systems and digital interactive television set-top boxes.

312. In October 1998, I began a six-month engagement with Rockwell Collins Inc., where I worked as a Lead Systems Engineer tasked with harmonizing requirements for an MPEG-2 in-flight entertainment system. I then worked for Sun

Microsystems Inc. as a Software Engineer until November 1999. During my time at Sun Microsystems Inc., I developed a Distributed Component Object Model (DCOM) software interface between a TV control graphical user interface and a Microsoft broadcast application programming interface (API) with the goal of improving the visual quality of interactive TV displays derived from UDP/IP datagrams synchronized with MPEG-2 audio/video packet data.

313. For the next 22 months, from January 2000 to October 2002, I worked as the Chief Systems Engineer for Media Logic Systems Ltd. During my time at Media Logic Systems, I designed and developed a live interactive television system (iSeeTV) in which customers communicate with human sales agents in video-enabled call centers. To create this system, I researched and developed tools and encoder systems to improve image quality at prescribed latency and bit rate for distributing live video and audio streams encoded via low latency methods. To perform the above, I was required to understand and implement video codec systems employing the MPEG-2 Simple Profile at Main Level (CATV), MPEG-4 Visual Profile with background sprite coding, and the H.263+ Standard (now known as H.264).

314. Since November 2002, I have been an engineering contractor, and more recently an independent consultant in mathematical modeling, for several companies, such as Cyra Technologies Inc. and Amgen Inc. I also served as a

senior research fellow at Merck & Co., Inc., a manager at GlaxoSmithKline Inc., a director at Daiichi Sankyo, Inc., a senior director at Praxis Precision Medicines, and a director at Takeda Pharmaceuticals. During this time, I have developed mathematical models and simulations related to various systems, signals, and images. Specifically, I have focused on analyzing, processing, storing, and deriving information from biomedical imaging and other data. Using the information derived from these data, I have created a variety of models related to biology and the effects of drugs on the human body. In recognition of my work, I have received GlaxoSmithKline R&D Recognition Awards in 2012, 2013, and 2016, a Daiichi Sankyo recognition award in 2021, and Takeda Pharmaceuticals awards in 2022, 2023, and 2024.

315. In addition to my over thirty years of relevant industry experience, I have authored many publications relating to video and imaging coding. In 2003, I authored a chapter entitled "Video Compression" for the *Internet Encyclopedia*. In 2004 I authored the chapter entitled "Video" for the *Berkshire Encyclopedia of Human-Computer Interaction*. And in 2007 I authored a chapter titled "Video Compression" for the *Handbook of Computer Networks*.

316. I am also a Senior Member of the IEEE and serve as the current Philadelphia Chapter Chair of the Communications & Information Theory Societies as well as former Chair of the American Association of Pharmaceutical

Scientists Pharmaco-Imaging Community. I also served as the 2019 Vice Chair of the IEEE P2673 Intelligence Augmentation for Medical Imaging Standards Working Group. I also have been registered to practice as a patent agent for the United States Patent and Trademark Office since 2002 (Reg. No. 51,704).

317. From 2017, I have also volunteered as a Volunteer Researcher with the State University of New York at Buffalo. In this role, I am providing senior authorship and mentorship for a doctoral candidate in areas relating to computer modeling and estimation.

318. I would have met the requirements of a person of skill in the art in the 2011 timeframe, in light of the educational and work experience explained above. *Supra* §III. For example, my education in physics was comparable to a bachelor's in EE/CS because it included the types of applied mathematics that are relevant here, such as linear algebra and differential equations together with the analysis of images, which provide the basis for various transformations and operations in video coding. Additionally, I note that I have a higher level of education than the definition of a POSITA. I also had at least ten years of practical experience in video coding by the 2011 timeframe, including, for example, developing rate control algorithms for the OPTIVideo^{TM} MPEG-2 video encoder at Optivision, Inc., integrating an MPEG-2 set top box with OpenTV interactive television middleware at Comstream Inc., harmonizing requirements for an MPEG-2 in-flight

entertainment system at Rockwell Collins Inc., developing DCOM software interface at Sum Microsystems Inc., and designing the iSeeTV system at Media Logic Systems.

A. Compensation

319. For my efforts in connection with the preparation of this declaration I have been compensated at my usual rate for this type of consulting activity. My compensation is in no way contingent on the results of these or any other proceedings relating to the above-captioned patent.

B. Materials and Other Information Considered

320. I have considered information from various sources in forming my opinions. I have reviewed and considered each of the exhibits listed in the attached Appendix 2 (Materials Considered in the Preparation of This Declaration) in forming my opinions.

VII. UNDERSTANDING OF THE LAW

321. I am not an attorney. In forming my opinions in this Declaration, I applied the relevant legal principles provided to me by counsel, which are summarized in Appendix 4.

VIII. RESERVATION OF RIGHTS

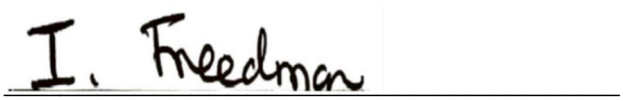
322. My opinions are based upon the information that I have considered to date. I am unaware of any evidence of secondary considerations with respect to the

’267 Patent that would render any of the challenged claims non-obvious. I reserve the right, however, to supplement my opinions in the future to respond to any arguments raised by the owner of the ’267 Patent and to take into account new information that becomes available to me.

323. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed on October 3rd, 2025.

By:



Immanuel Freedman

APPENDIX 1: CURRICULUM VITAE OF IMMANUEL FREEDMAN

IMMANUEL FREEDMAN, Ph. D, SMIEEE, MInstP, CPhys, FRSM

942 Clubhouse Drive
Harleysville, PA 19438
215-527-1779

SUMMARY OF EXPERIENCE

Systems, Signals and Algorithms Consultant with over 30 years experience of video, imaging, modeling, simulation, and systems analysis, design, development, and testing. He has served as an expert consultant providing technical analysis related to patent infringement, patent validity, and research tax credit.

EDUCATION

Ph. D., Physics, University of Durham, England, 1986
B.Sc. (Honors), Physics, University of Durham, England, 1979

LICENSES

Registered Patent Agent #51,704

EXPERIENCE

Freedman Patent **Harleysville, PA** **Oct '21-present**
Sole Proprietor
He provides consulting and expert witness services to industry and the legal profession.

Dr. Freedman was also Vice-Chair of IEEE P2673, a standard for digital biomedical data files with 3D anatomical mapping for big data and augmented intelligence systems and Secretary for IEEE 1900.8, a standard for the storage format of RFML datasets and the interfaces that connect stages of the RFML model training pipeline with use cases for RF signal detection, classification, and characterization, as well as identification of RF emitters.

He also serves as a member of IEEE P2795. This standard defines metadata and frameworks used to support interactions between nodes that are involved in a trusted multipart handshake, including discovery of data models and processing aspects available at a remote location, appropriate analytic configuration, and exchange of vetted critical analytic results.

As a member of the American Association of Pharmaceutical Scientists, he served as Former Chair, Pharmaco-Imaging Community, Former Chair, Predictive Modeling Community, and 2017 Member, Predictive Modeling Task Force.

State University of New York at Buffalo **Buffalo, NY** **Jun '17-present**
Volunteer Researcher
Dr. Freedman serves as a Senior Author and mentor to a doctoral candidate, supporting the analysis of chaotic and non-chaotic biological systems utilizing chaos synchronization methodology.

Takeda Pharmaceuticals **Cambridge, MA** **Mar ’22-Feb ‘25**
Clinical Pharmacology Director
Dr. Freedman provided mathematical modeling of systems, signals, and images.

Praxis Precision Medicines **Boston, MA** **Aug ’21-Sep ’21**
Senior Director, Pharmacometrics
Dr. Freedman provided mathematical modeling of systems, signals, and images.

Daiichi Sankyo, Inc. **Basking Ridge, NJ** **Nov ’20-Aug ’21**
Director, Modeling and Simulation
Dr. Freedman provided mathematical modeling of systems, signals, and images.

Freedman Patent **Harleysville, PA** **Jun ’16-Nov ’20**
Sole Proprietor
He provided consulting and expert witness services to industry and the legal profession. In particular, he performed requirements analysis and design for a precision dosing system, Graphical User Interface, and delivered key imaging analyses of 225Ac-mAb daughters and an anti-Her-3 mAb in oncology.

GlaxoSmithKline, Inc. **Upper Merion, PA** **Aug ’07-Jun ’16**
Manager, Clinical Pharmacology
Dr. Freedman provided mathematical modeling of systems, signals and images and participated in technical due diligence activities on demand.

Freedman Patent **Harleysville, PA** **Jul ’05-Aug ’07**
Sole Proprietor
He provided consulting and expert witness services to industry and the legal profession.

Merck & Co., Inc., **West Point, PA** **Nov ’04-Jul ’05**
Senior Research Fellow
Dr. Freedman provided mathematical models on demand.

Amgen, Inc. **Thousand Oaks, CA** **Apr ’03-Mar ’04**
Contractor
Dr. Freedman developed algorithms and software in MATLAB and FORTRAN for simulation and data modeling.

Cyra Technologies, Inc. **San Ramon, CA** **Nov ’02-Apr ’03**
Senior Hardware Engineer
Dr. Freedman designed, developed, and tested algorithms and software for calibrating a three-dimensional laser scanner. He calculated the statistical distribution of outcomes for an engineering tolerance stack by modeling and simulating the scanner response using a Jacobian sensitivity matrix to compare alternative placements of scanner calibration targets based on a D-matrix of scanner response.

Media Logic Systems Ltd. **Fleet, England UK** **Jan ’00-Oct ’02**
Chief Systems Engineer

Dr. Freedman designed and developed a novel live interactive television system (iSeeTV) in which served as a User Interface for customer communication with human sales agents in video-enabled call centers implemented via television and telephone, deployed to 50,000 subscribers of Telewest, UK.

He researched and developed tools and encoder systems to optimize image quality at prescribed latency and bit rate for distributing live video and audio streams encoded via low latency methods including MPEG-2 Simple Profile at Main Level (CATV), MPEG-4 Visual Profile with background sprite coding, and H.263+ (now known as H.264).

Dr. Freedman investigated the feasibility of wavelet–based software encoding schemes with motion compensation and perceptual quantization described by the MPEG Standards Committee Interframe Wavelet Ad Hoc Group. He interfaced video streams via ATM transport to Telewest, UK regional CATV head-ends switched via Harmonic Narrowcast Gateways for distribution via Video On Demand or Near Video On Demand systems to customer's homes.

Replay Networks, Inc. Mountain View, CA Dec'99-Jan '00
Contractor

Dr. Freedman researched and developed a method of porting an application developed for a Digital Video Recorder in the embedded C software language to standard set top box (STB) middleware to eliminate high development and maintenance costs associated with developing custom STBs. He optimized bit rate and encoder chip parameters to yield high-quality time-shifted MPEG-2 streams controlled by VCR-like consumer controls.

Sun Microsystems, Inc. Cupertino, CA Mar '99-Nov '99
Software Engineer (Contractor)

Dr. Freedman researched and developed a Distributed Component Object Model (DCOM) software interface between a TV Control Graphical User Interface and the Microsoft Broadcast Application Programming Interface (API) to improve the visual quality of interactive TV displays derived from UDP/IP datagrams synchronized with MPEG-2 audio/video packet data.

The software interface additionally resolved discontinuities in Presentation Timestamp according to a Normal Play Time defined by a Digital Storage Media –Command and Control standard.

He designed and implemented an API written in the pJava and Visual C++ software languages under the Windows CE operating system for the Motorola DCT 5000+ DTV Set Top Box based on the Advanced Television Systems Committee digital television standard.

Rockwell Collins, Inc. Pomona, CA Oct '98-Mar '99
Lead Systems Engineer (Contractor)

As Lead Systems Engineer with a two-engineer span of control, Dr. Freedman timely delivered harmonized requirements for an MPEG-2 in-flight entertainment system similar to a cable television system based on an advanced intranet implemented on an aircraft.

He trained his team to use a Rational Unified software development process based on a Spiral Development Model implemented in the Universal Modeling Language using the Rational/Rose 98i Computer Aided Software Engineering tool.

Stratagene, Inc. La Jolla, CA Aug '98-Oct '98
Engineer (Contractor to Permanent)

Dr. Freedman evaluated frame grabber hardware for resolution and quality of time-integrated imagery and specified algorithms including cluster analysis and trending, further

developing a user interface for a digital image processing system supporting gene-cloning science.

United Advanced Technologies, Inc. Long Beach, CA Feb ’98-Aug ’98
Firmware Engineer (Contractor)

Dr. Freedman analyzed and developed a nine-camera remote surveillance system with a Graphical User Interface developed in the Visual C++ software language under a Microsoft Windows operating system host and firmware developed in the embedded C software language implemented on Analog Devices' ADV601 wavelet video hardware.

He researched and developed Video for Windows parameters and on-chip settings for video quality control to deliver full-frame video over Plain Old Telephone Service telephone lines at quality acceptable to retail store security services.

KeyInfo Services, Inc. Spring Valley, CA Mar ’98-May ’98
Database Consultant

Dr. Freedman administered a database for providing web-based information developed in the Sybase SQL software language.

Mitek Systems, Inc. San Diego, CA Aug ’97-Jan ’98
Engineer (Contractor)

Dr. Freedman researched and developed an Intelligent Character Recognition digital image processing system based on neural nets implemented in the C software language to read handwritten checks and paper forms with about 80% accuracy on a real-time system deployed throughout the banking industry.

He researched and developed algorithms based on mathematical morphology implemented via neural nets to verify handwritten signatures on printed checks.

Symbionics Ltd. Cambridge, England UK Aug ’97
Principal Engineer (Temporary)

Dr. Freedman analyzed manpower estimates for design, development and testing of an MPEG-2 interactive television set-top box based on an OpenTV interactive television standard implemented for the "Open…." television commerce system deployed Spring 1999 in the United Kingdom.

VideoActive/HCR, Inc. Yorba Linda, CA Jan ’97-Dec ’97
Contractor

Dr. Freedman reviewed, analyzed and developed proprietary disk layout software coded in the Visual C++ software language for a Near Video on Demand system delivering movies over telephone systems such as Asymmetric Digital Subscriber Lines (ADSL).

Dr. Immanuel Freedman, Inc. Harleysville, PA Mar ’96-Jan ’16
President

He provided technical consulting services to industry.

Anugraha La Jolla, CA Jan ’96-Jan ’97
Sole Proprietor

Dr. Freedman researched and developed algorithms and processes to compress fine art photography at image quality acceptable to artists based on the JPEG imaging standard implemented with image pre-processing and adaptive quantization.

Armor Safe Technologies, Inc. Vista, CA Sep '95-Mar '96
Firmware Engineer (Contractor)

Dr. Freedman developed an ARINC RS422/RS485 serial link communications software component written in the embedded C software language for a major confidential client specialized in retail store security. His timely software delivery enabled the client to capture a firm order with additional future potential.

Comstream, Inc. San Diego, CA Jul '96-Aug '96
Firmware Engineer (Contractor)

Dr. Freedman integrated a MPEG-2 set top box with OpenTV interactive television middleware programmed in the Microtec C language ported to a Motorola 68340 processor under the pSOS operating system. He implemented native bindings of the middleware for the On-Screen Display (Graphical User Interface) and communications stack.

General Instrument, Inc. San Diego, CA Dec '95-May '96
Senior Staff Engineer

Dr. Freedman reviewed and evaluated methodologies, design and development and performance models for the DigiCipher 2 cable television conditional access system. He migrated a subscriber authorization system written in the C++ language from a DEC Alpha computing platform under the OpenVMS operating system to a Sun SPARCstation computing platform under the Solaris operating system.

Optivision, Inc. Davis, CA Mar '95-Sep '95
Consultant

Dr. Freedman researched and developed rate control algorithms and software based on MPEG -2 Test Model 5 for the OPTIVideo MPEG-2 video encoder written in the Visual C++ and C software languages.

He researched and developed adaptive quantization algorithms based on a JPEG-3 draft standard for possible inclusion in the draft National Imagery Transmission Format imaging standard.

He researched and developed algorithms to improve the quality of gray scale image compression for the medical imaging DICOM Standard by providing a lossless hybrid algorithm encoding image residuals with a diagonal Golomb code based on an Enhanced Universal Trellis Coded Quantization algorithm.

Federal National Mortgage Association Washington, DC Jul '94-Jan '95
Software Engineer (Contractor)

Dr. Freedman designed and developed a Graphical User Interface to monitor and validate loan servicer input for the Loss Mitigation Project. He developed the software in the C software language for a Sun SPARCstation 2 platform under a UNIX operating system.

Hughes STX Corporation Greenbelt, MD Sep '88-Jun '94
Senior Systems Engineer

As Spacecraft and Attitude Analyst for a mission to map the relict radiation from the Big Bang at near infrared, far infrared and microwave wavelengths, Dr. Freedman developed, simulated and calibrated the Cosmic Background Explorer (COBE) Attitude Determination System to yield a stable solution for the spacecraft orientation at a quality factor of 2 above customer's expectation. This solution included a quaternion estimator implemented via an Extended Kalman Filter.

He developed, calibrated and simulated the COBE spacecraft subsystem and provided graphical and statistical analysis of the spacecraft telemetry-word database. When a gyroscope failed during the Launch and Early orbit mission phase, he responded rapidly by plotting graphs of the thermal subsystem telemetry until he found a possible cause of failure.

Dr. Freedman developed a spatially referenced database based on a quad-tree data structure, which stored scientific data for comparison of sky maps from COBE with sky maps from other missions that served as a diagnostic user interface for the Diffuse Infrared Background Experiment.

For the COBE mission, he led the systems engineering and end-to-end development of a novel system for compressing data that combined scientific modeling with statistical data compression. He proposed the system concept and prepared the system level specification, design and project schedule. With a team of two engineers, Dr. Freedman tuned the compression system performance to yield a throughput greater than uncompressed data processing with a compression factor of 22-90%. He further designed and developed evaluation tools to ensure the user- transparent system-wide compression of a 380GB dynamic data base at image quality acceptable to scientists.

University of Maryland College Park, MD Jun '87-Sep '88
Research Associate

Dr. Freedman researched and developed digital image methods to process terrain models for a combat information processor sponsored by Battelle. It was developed in the C software language on Sun Microsystems workstation for porting to a supercomputer under a UNIX operating system.

He designed low-complexity algorithms for filtering, segmenting, clustering, and path planning based on digital images organized by quad-tree data structures.

University College London London, England UK Sep '86-Jun '87
Research Assistant

Dr. Freedman developed digital image processing algorithms to improve image and stereo-matching quality for a digital terrain modeling system based on satellite data.

As part of a UK Government Fifth Generation computing project (Alvey MMI-237) in collaboration with Thorn EMI, Royal Signals and Radar Establishment, and Laser Scan Ltd., he developed software and algorithms for affine transformation, edge filtering, kriging interpolation and image stereo matching with sub-pixel acuity.

Laser-Scan Ltd. Cambridge, England UK Sep '85-Sep '86
Software Engineer

Dr. Freedman researched and developed algorithms based on the mathematics of tessellation for efficient manipulation of spatially referenced data on serial computers and transputer arrays for a UK Government Fifth Generation computing project (Alvey MMI-237) in collaboration with Thorn EMI, Royal Signals and Radar Establishment, and Laser Scan Ltd.

National Coal Board Nuneaton, England UK Nov '82-Sep '84

Scientist (Management Grade 7)

For a Health and Safety project, Dr. Freedman developed and validated a microcomputer system to detect coalmine fires and heatings. Based on stochastic and temporal analysis of infrared data obtained via a tube bundle system, and telemetry data from underground thermocouples, the system detected growing trends of carbon monoxide concentration in the presence of noise from underground events such as blasting, diesel engine fumes, ventilation changes, and seismic activity.

PUBLICATIONS

Yuskaitis, J. C. et al. (2025), "Comparability of pharmacokinetics between HyQvia and TAK-881 in CIDP: a phase 3 study design," poster in Japanese Peripheral Nerve Society Meeting 2025, Fukuoka, JP

Yuskaitis, J. C. et al. (2025), "Comparability of pharmacokinetics between HyQvia and TAK-881 in CIDP: a phase 3 study design," poster in European Academy of Neurology 2025, Helsinki, FI

Yuskaitis, J. C. et al. (2025), "Comparability of Pharmacokinetics in Adults with Chronic Inflammatory Demyelinating Neuropathy Receiving Facilitated Subcutaneous Immunoglobulin: A Phase3 Study Design," poster in American Association of Neuromuscular & Electrodiagnostic Medicine 2025, San Francisco CA

Paul, A.K, Pillai, N., Freedman, I., & Bies R. (2025), "Estimating Controlling Parameters and Tracking Cortisol Secretion Profile using Chaos Synchronization", poster in AI & Data Science Symposium: IAD Days 2025, Institute for Artificial Intelligence and Data Science, University at Buffalo, Buffalo NY

Li, Z. & Freedman, I. (2025), "Similar Immunoglobulin 10% Pharmacokinetics in Patients with Chronic Inflammatory Demyelinating Polyradiculoneuropathy and Healthy Individuals," in Peripheral Nerve Society Annual Meeting 2025, 17-20 May 2025, Edinburgh UK

Freedman, I., Roepcke, S. & Li, Z. (2024), “Exposure-Response Relationships: Hyaluronidase-Facilitated Subcutaneous and Intravenous Immunoglobulin 10% in Chronic Inflammatory Demyelinating Polyradiculoneuropathy,” poster in American Association of Neuromuscular & Electrodiagnostic Medicine 2024, Savannah GA

Wassermann, R. L. et al. (2024), “Assessing hyaluronidase-facilitated subcutaneous immunoglobulin 20% (fSCIG 20%) pharmacokinetics, safety and tolerability in primary immunodeficiency diseases: phase 2/3 study design,” Clinical Immunology Society 2024, Minneapolis MN

Kanegane, H. et al. (2024), “ A Japanese Phase 3 Study of Subcutaneous Immunoglobulin 10% with Recombinant Human Hyaluronidase Every 3 or 4 Weeks for Primary Immunodeficiency Disease,” Japanese Society for Immunodeficiency and Autoinflammatory Diseases 2024, Tokyo JPN

Freedman, I., Roepcke, S. & Li, Z. (2024), "Population Pharmacokinetic Simulations in CIDP and MMN; Hyaluronidase Subcutaneous Immunoglobulin and Intravenous Immunoglobulin G 10%", poster in Peripheral Nerve Society Annual Meeting 2024, Montreal Québec CA

Freedman, I. (2023), "Chaos Synchronization and the Guaranteed Convergence of Estimators of Nonlinear Dynamical Systems," IEEE Philadelphia Section event, December 19, 2023

Freedman, I. (2023), "The Intellectual Heritage of Babylonian Astronomy: Music of the Spheres," IEEE Philadelphia Section event, December 5, 2023

Freedman, I. (2023), "The Intellectual Heritage of Babylonian Astronomy: Music of the Spheres," in *The Intellectual Heritage of the Ancient Near East. Proceedings of the 64th Rencontre Assyriologique Internationale and the 12th Melammu Symposium,* eds. Rollinger, R. et al., pubs. Austrian Academy of Sciences Press: Vienna Austria 2023. ISBN 978-3-7001-8574-1

Freedman, I. (2021), "Modern Methods of Chaos and Babylonian Mathematical Astronomy," Ronin Institute Public Seminar, https://www.youtube.com/watch?v=KIfNFzK8dyI

Freedman, I. (2021), "On ^{mul} SAG.ME.GAR," NABU 2021-2, Nr. 52

Wilkins, J. et al. (2019), ISoP Position Statement on "Guidance Document: Population Pharmacokinetics: July 2019", Docket No. FDA 2019-D-2398

Freedman, I. (2019), "Chaos Synchronization: Identifying & Predicting Noisy Complex Systems," in AAPS Forum to Connect Predictive Modelers, May 6-7, 2019, Boston MA

Pillai, N. et al. (2019), "Estimating parameters of nonlinear dynamic systems in pharmacology using chaos synchronization and grid search," Journal of Pharmacokinetics and Pharmacodynamics, 46(2), 193-210

van Oordt, C. W. M. H et al. (2018), "ImmunoPET imaging to assess target engagement: Experience from 89Zr-anti-HER3 mAb (GSK2849330) in patients with solid tumors", J Nucl Med jnumed.118.214726 published ahead of print February 7, 2019

Norris, D. et al. (2018), "Expansion Cohorts: Use in First-In-Human Clinical Trials To Expedite Development of Oncology," submitted to FDA, FDA-2018-D-2777-0001, DOI 10.17605/OSF.IO/NMGH8

Freedman, I. (2018), "IEEE Digital Health Standards," at FIS Global , November 27, 2018, Collegeville: PA

Freedman, I. (2018), "Chaos Synchronization: Estimating Parameters of Noisy Complex Systems," in IEEE Section Night, October 20, 2018, Philadelphia: PA

Schieke, M. & Freedman, I. (2018), "3D Topological Mapping of Macroanatomy and Microanatomy for Use in Big Data and Augmented Intelligence Systems," in IEEE-Standards Association Workshop on Digital Health Standards, September 11, 2018, Atlanta: GA

Freedman, I. (2018), "The Intellectual Heritage of Babylonian Astronomy: Music of the Spheres," presentation in *Rencontre Assyriologique Internationale 64*, July 14-20, 2018, Innsbruck Austria

Pillai, N. et al. (2018), "Chaos synchronization and Nelder-Mead search for parameter estimation in nonlinear pharmacological systems: estimating tumor antigenicity in a model of immunotherapy," PBMB Special Issue Volume 139, November 2018, Pages 23-30

Pillai, N. et al. (2018), "Estimating parameters of chaotic systems: chaos synchronization combined with Nelder-Mead search, " in PAGE 2018: May 31, 2018, Montreux CH

O'Connor, M. A. et al. (2017), "Elucidating Exposure-Response (Safety and Efficacy) of Adct-402 (Loncastuximab Tesirine), a Novel Pyrrolobenzodiazepine-Containing Antibody Drug Conjugate, for Recommended Phase 2 Dose Determination in Patients with Relapsed or Refractory Non-Hodgkin Lymphoma," in American Society of Hematology 59th Annual Meeting & Exposition, Atlanta GA Dec 9-12, 2017

Freedman, I., Bies, R. R., Pillai, N., Schwartz, S., Ho, T. (2017), "Estimating parameters of chaotic systems in pharmacology", invited talk in American Conference on Pharmacometrics 8, Oct 14-19, 2017, Ft. Lauderdale: FL

Nguyen, T-H-H et al. (2016), Model evaluation of continuous data pharmacometric models: metrics and graphics, CPT Pharmacometrics Syst Pharmacol. 2017 Feb; 6(2): 87–109

Freedman, I., Bies, R. R., Pillai, N. (2016), "Estimating parameters of chaotic systems," in American Association of Pharmaceutical Scientists Annual Meeting and Expositions, November 13-17, 2016, Denver CO

Freedman, I. (2016), "Babylonian astronomers predicted lunar and planetary positions using iterated maps, forerunners of deterministic chaos," in Rencontre Assyriologique Internationale 62, July 11-15, 2016, Philadelphia PA

Cataloluk, O. & Freedman, I. (2016), "The meaning of *armannu*," in Rencontre Assyriologique Internationale 62, July 11-15, 2016, Philadelphia PA

Freedman, I., Bies, R. R., Ho, T. (2015), "Estimating parameters for chaotic systems," in GlaxoSmithKline Biostatistics Annual Conference, November 13, 2015, Upper Providence PA

Freedman, I. (2015), "A QSP model for cancer incorporating HER-3," in GlaxoSmithKline Quantitative Systems Pharmacology Day, October 29, 2015, Upper Merion PA

Freedman, I. (2015), "Imaging & Microdialysis: Complementary Modalities?" invited panelist, in American Association of Pharmaceutical Scientists Annual Meeting and Exposition, October 24-29, 2015, Orlando: FL

Freedman, I. (2015), "On MA.NA EN.NUN in ^{mul} APIN," NABU 2015-2 Nr. 50

Freedman, I. (2015), "The Marduk Star Nēbiru," Cuneiform Digital Library Bulletin 2015:3, retrieved from http://cdli.ucla.edu/files/publications/cdlb2015_003.pdf

Freedman, I. (2015), "Short Note: On an Assyriological Interpretation of the Book of Daniel 5:1–28," Journal of Translation 11(1), 74

Bies, R. R., Ho, T. & Freedman, I. (2015), "Nonlinear dynamic physiological systems: Chaos Synchronization as an approach to parameter estimation" in *Kinetic and Dynamic Complexity in Drug Transit-Response in the Human Body*, satellite meeting to Population Approach Group Europe 2015, June 2, 2015, Hersonissos: GR

Freedman, I. (2015), "On an Assyriological interpretation of Constantine's *labarum*" retrieved from http://www.researchgate.edu

Freedman, I. (2015), "Model Evaluation and Selection," invited talk to American Association of Pharmaceutical Sciences Pharmacometrics Focus Group, March 26, 2015

Freedman, I., Moyer, B.R & Cheruvu, N.P.S. (2014), "Pharmaco-Imaging", in American Association of Pharmaceutical Sciences Translational Sciences 101 e-learning course, retrieved from http://www.aaps.org/TS101/

Freedman, I. (2014), "Role of Pharmacology in Anti-Cancer Antibody Development," invited talk, in American Association of Pharmaceutical Scientists Annual Meeting and Exposition, Nov 2-6, San Diego: CA

Willis, S. et al. (2014), "Martha Meets Shakespeare," musical performance, retrieved from http://www.youtube.com/

Austin, D., Freedman, I., Grove, R, Tolson, J. & Vanmeter, S. (2014), "The Relationship Between Peripheral B-Cell Levels and MRI Disease Activity in Relapsing-Remitting Multiple Sclerosis (RRMS)", dual presentation in American Academy of Neurology 66th Annual Meeting, Philadelphia: PA

Freedman, I. (2014), "On horses and bitumen in ^{mul}APIN," NABU 2014-1 Nr. 33
Freedman, I. (2014), "Pharmaco-Imaging in Translational Science and Research," in Pharmaco-Imaging in Drug and Biologics Development (Moyer, B.R., Cheruvu, N.P.S. & Hu, T. C.-C., eds.), Springer, Inc.

Freedman, I. & Bies, R. (2013),"Chaos Synchronization," invited talk, in 3rd Annual Indiana Clinical and Translational Sciences Institute Symposium on Disease and Therapeutic Response Modeling, Nov 5-6, 2013, Indianapolis, IN

Freedman, I. (2013), "Translational pharmacometric models of anti-cancer antibodies," invited talk, in 2nd International Conference on Translational & Personalized Medicine, Aug 5-7, 2013, Chicago IL

Freedman, I. (2013), "Writing on the Wall–Conspiracy in Babylon?" retrieved from http://www.researchgate.edu

Willis, S. et al. (2013), "Amy's Suite," musical performance, retrieved from http://www.youtube.com/

Freedman, I. (2012), "Case Studies in the Use of Imaging Agents in Proof-of-Concept and Early Proof-of-Efficacy Trials," invited talk, in National Biotechnology Conference, May 21-24, 2012, San Diego: CA

Freedman, I. (2012), "Regulatory Opportunities and Hurdles with Imaging Agents in the Era of Personalized Medicine," invited panelist, in National Biotechnology Conference, May 21-24, 2012, San Diego: CA

Freedman, I. (2012), "Translating Clinical and Pre-Clinical Information into Success," invited talk, in 3rd World PK/PD Summit, March 27-29, 2012, Boston: MA

Freedman, I. (2012), "Improving Integration of PK/PD Modeling into the Drug Development Process," invited panelist, in 3rd World PK/PD Summit, March 27-29, 2012, Boston: MA

Freedman, I. (2011), "The Impact of 3D Tissue Models on Drug Development and Predictive Safety," invited panelist, in Predictive Functional Tissue Models, November 17-18, 2011, Boston: MA

Derosier, F.J. et al. (2012), "Novel Uses," WIPO Publication WO2012096924 A1, priority date Jan 10, 2011

Freedman, I. (2011), "System for forecasting outcomes of clinical trials," United States Patent Application Publication 20110238317A1, priority date Aug 9, 2007.

Freedman, I. (2011), "Case Study–Optimizing Design of a Phase 2 Clinical Trial in Relapsing-Remitting Multiple Sclerosis," invited talk in Joint Statistical Meetings 2011, Miami FL July 30-August 4, 2011

Freedman, I. (2011), "Case Study–Bridging from Western to East Asian Population in Accordance with ICH E5," in American Conference on Pharmacometrics 2011, San Diego CA April 3-6, 2011
Larkin, J. et al. (2010), "Osteoarthritis Disease Modification Potential of GSK2394002" in GlaxoSmithKline Clinical Pharmacology Symposium, Philadelphia PA June 15, 2010

Larkin, J. et al. (2010), "ADAMTS-5 Neutralizing Monoclonal Antibody: Translational Pharmacology Approach Provides Reason to Believe," in GlaxoSmithKline Clinical Pharmacology Symposium, Philadelphia PA June 15, 2010

Freedman, I et al. (2009), "Bayesian Parametric Schild Analysis from a Whole Blood Stimulation Assay," in American Conference on Pharmacometrics 2009, Mystic CT October 4-7, 2009

Freedman, I. (2009), "Feasibility of Immunogenicity Assessment based on Models of Polyclonal Anti-Drug Antibody Immunodynamics," in American Conference on Pharmacometrics 2009, Mystic CT October 4-7, 2009

Austin, D. J., et al. (2008), "Translational Pharmacology from a Domain Antibody First Time in Human Study," in SciNovations Symposium, Verona IT, September 26, 2008

Freedman, I. (2007), "Video Compression," in The Handbook of Computer Networks (Hossein Bidgoli, ed.), John Wiley & Sons, Inc.

Freedman, I. (2006), “Examination on Challenge: A Proposal for Improving the Patent System”, *Disclosure*, Jan 2006 retrieved from http://www.napp.org

Freedman, I. (2005), “Erythropoetin Modeling & Simulation,” retrieved from http://www.freedmanpatent.com

Freedman, I. (2004), “Working with a Patent Agent,” in PatentCafe Magazine, May 11, 2004

Freedman, I. (2004), “Video,” in Encyclopedia of Human-Computer Interaction (William Sims Bainbridge, ed.), Berkshire Publishing Group LLC

Freedman, I. (2003), "Video Compression," in The Internet Encyclopedia (Hossein Bidgoli, ed.), John Wiley & Sons, Inc.

Freedman, I. (2002), "Comparison of algorithms for efficient approximation of heterogeneous multidimensional scientific data," Astronomical Data Analysis II. (Starck, J-L; Murtagh, F D, ed.) in Proc. SPIE, Volume 4847, pp. 313-321

Freedman, I. (1999), "Proposal for Preliminary Evaluation of an Approximate Data Compression Algorithm for the FITS Standard," BAAS, 194

Freedman, I. and Farrelle, P. M. (1996), "Systems Aspects of COBE Science Data Compression," ASP Conf. Ser. 101: Astronomical Data Analysis Software and Systems V

Sodroski T. et al. (1995), "The Ratio of H_2 Column Density to ^{12}CO Intensity in the Vicinity of the Galactic Center," Ap. J. 450

Freedman, I., Boggess, E. & Seiler, E. (1992), "COBE Science Data Compression," BAAS, 24
Freedman, I., Skard, J. A.J. & Kelsall, T. (1992), "COBE/DIRBE Sky Map Trending," in ASP Conf. Ser. 25: Astronomical Data Analysis Software and Systems I

Freedman, I., Raugh, A. C. & Cheng, E. S. (1992), "COBE Astronomical Databases," in ASP Conf. Ser. 25: Astronomical Data Analysis Software and Systems I

Freedman, I. et al. (1991), "COBE ground segment gyro calibration," in NASA Flight Mechanics/Estimation Theory Symposium

Kumar, V.K. et al. (1991), "COBE ground segment attitude determination," in NASA Flight Mechanics/Estimation Theory Symposium

Wright, E.L. et al. (1991), "COBE attitude refinement," in AIP Conf. Proc. 222: After the first three minutes

Wright, E. L. et al. (1989), "The COBE Attitude Determination System," BAAS 21

Jackson M. J. et al. (1986), "Efficiency of tesseral arithmetic for 2.5D image manipulation on serial computers and transputer arrays," in S.B.M. Bell and B.M. Diaz (eds.) Spatial Data Processing using Tesseral Methods, pp. 353-364.

Dogiel V. et al. (1983), "Can Cosmic Rays be Accelerated in Collapsing Molecular Clouds?" Proc. 18th International Cosmic Ray Conference, 2

Giler, M. et al. (1980), "A limit to the velocity of convection of cosmic rays out of the Galaxy," Proc. 16th International Cosmic Ray Conference, 2

Freedman, I. et al. (1980), "Derivation of the age distributions of cosmic rays in a Galaxy with a convective halo," AAP, 82

PROFESSIONAL AWARDS

Takeda Pharmaceuticals Recognition Awards (2022, 2023, 2024, 2025)
Daiichi Sankyo Recognition Award (2021)
Funaro Award (2015, 2018)
GlaxoSmithKline R&D Recognition Award (2012, 2013, 2016)
NASA Group Achievement Award (1990, 1992)
Hughes STX Achievement Award (1990)
IBM Fulcrum Award (1988)

PROFESSIONAL AFFILIATIONS

Institute of Electrical and Electronics Engineers (Senior Member; Chair, Communications & Information Theory Chapter, Philadelphia Section; 2018/2019 Vice-Chair, P2673 Standards Working Group; Trusted Analytic Exchange Sub-Group Chair, P2795 Standards Working Group; Member P1900.8 Standards Working Group; Former Secretary, Dynamic Spectrum Access Machine Learning Study Group)
Institute of Physics (Chartered Physicist)
International Society of Pharmacometrics (Member, Standards and Best Practices Committee Member, Model Evaluation Group)
Regulatory Affairs Professionals Society
National Coalition of Independent Scholars
Royal Society of Medicine (Fellow Member)

APPENDIX 2: MATERIALS CONSIDERED

Exhibit No.	Description
1001	U.S. Patent No. 11,805,267 to Ugur et al. ("the '267 Patent")
1002	Prosecution File History for the '267 Patent
1003	Declaration of Immanuel Freedman
1004	U.S. Patent Application Publication No. 2010/0322303 ("Wada")
1005	U.S. Patent Application Publication No. 2011/0007799 ("Karczewicz-I")
1006	Yan Ye et al., High Precision Interpolation and Prediction, VCEG-AI33, July 2008 ("AI33 submission")
1007	U.S. Patent Application Publication No. 2009/0257499 ("Karczewicz-II")
1008	Sridhar Srinivasan et al., An Overview of VC-1, Proceedings of SPIE, Vol. 5960 (July 2005) ("Srinivasan")
1009	U.S. Patent No. 9,344,744 ("Kirchhoffer-744")
1010	U.S. Patent Application Publication No. 2003/0112864 ("Karczewicz-864")
1011	Wiegand, Overview of the H.264/AVC Video Coding Standard
1012	Richardson, The H.264 Advanced Video Compression Standard
1013	U.S. Patent Application Publication No. 2008/0198935 ("Srinivasan-935")
1014	ITU-T Recommendation H.264.2: Reference software for H.264 advanced video coding (March 2005)
1015	U.S. Patent Application Publication No. 2013/0034158 ("Kirchhoffer-158")
1016	U.S. Patent No. 8,594,188 ("Demos")
1017	IPR2024-00626, Paper 13: Decision Granting Institution of Inter Partes Review
1018	IPR2024-00627, Paper 16: Decision Granting Institution of Inter Partes Review
1019	U.S. Patent Application Publication No. 2008/0181308
1020	Paulo J. Cordeiro et al., Efficient Constrained Video Coding for Low Complexity Decoding (2008)
1021	Patent Owner's Index for Complaint Appendix C in ITC Inv. No. 337-TA-1448

1022	Patent Owner's Appendix C0074 to Complaint in ITC Inv. No. 337-TA-1448
1023	Takeshi Chujoh et al., Internal Bit Depth Increase for Coding Efficiency, VCEG-AE13, January 2007
1024	Excerpts from Barry G. Haskell et al., Digital Video: An Introduction to MPEG-2 (Chapman and Hall, 1997)
1025	Joint Claim Construction Chart in ITC Inv. No. 337-TA-1448
1026	Declaration of Sylvia Hall-Ellis, Ph.D. ("Hall-Ellis")
1027	Thomas Sikora, MPEG Digital Video-Coding Standards, IEEE Signal Processing Magazine (September 1997)
1028	Gary J. Sullivan and Thomas Wiegand, Rate-Distortion Optimization for Video Compression, IEEE Signal Processing Magazine (November 1998)

APPENDIX 3: CHALLENGED CLAIMS

[1a]. A method for encoding a block of pixels, the method comprising:

[1b]. determining, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[1c]. using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[1d]. using said second reference block to obtain a second prediction, said second prediction having the second precision;

[1e]. obtaining a combined prediction based at least partly upon said first prediction and said second prediction;

[1f]. decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[1g]. encoding residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.

2. The method according to claim 1, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

3. The method according to claim 2, wherein said first prediction is obtained by interpolation using values of said first reference block by:

right shifting a sum of a P-tap filter using values of said first reference block.

4. The method according to claim 2, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

5. The method according to claim 1, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises:

inserting a rounding offset to the combined prediction before said decreasing.

6. The method according to claim 1, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[7a]. An apparatus for encoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:

[7b]. determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[7c]. use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[7d]. use said second reference block to obtain a second prediction, said second prediction having the second precision;

[7e]. obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[7f]. decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[7g]. encode residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.

8. The apparatus according to claim 7, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

9. The apparatus according to claim 8, wherein said first prediction is obtained by interpolation using values of said first reference block by:

right shifting a sum of a P-tap filter using values of said first reference block.

10. The apparatus according to claim 8, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

11. The apparatus according to claim 7, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by:

inserting a rounding offset to the combined prediction before said decreasing.

12. The apparatus according to claim 7, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[13a]. A computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:

[13b]. determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[13c]. use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[13d]. use said second reference block to obtain a second prediction, said second prediction having the second precision;

[13e]. obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[13f]. decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[13g]. encode residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.

14. The computer program product according to claim 13, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

15. The computer program product according to claim 14, wherein said first prediction is obtained by interpolation using values of said first reference block by:

right shifting a sum of a P-tap filter using values of said first reference block.

16. The computer program product according to claim 14, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

17. The computer program product according to claim 13, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:

insert a rounding offset to the combined prediction before said decreasing.

18. The computer program product according to claim 13, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[19a]. A method for decoding a block of pixels, the method comprising:

[19b]. determining, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[19c]. using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[19d]. using said second reference block to obtain a second prediction, said second prediction having the second precision;

[19e]. obtaining a combined prediction based at least partly upon said first prediction and said second prediction;

[19f]. decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[19g]. reconstructing the block of pixels based on the combined prediction.

20. The method according to claim 19, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

21. The method according to claim 20, wherein said first prediction is obtained by interpolation using values of said first reference block by:

right shifting a sum of a P-tap filter using values of said first reference block.

22. The method according to claim 20, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

23. The method according to claim 19, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises:

inserting a rounding offset to the combined prediction before said decreasing.

24. The method according to claim 19, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[25a]. An apparatus for decoding a block of pixels, the apparatus comprising:

at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:

[25b]. determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[25c]. use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[25d]. use said second reference block to obtain a second prediction, said second prediction having the second precision;

[25e]. obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[25f]. decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[25g]. reconstruct the block of pixels based on the combined prediction.

26. The apparatus according to claim 25, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

27. The apparatus according to claim 26, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.

28. The apparatus according to claim 26, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

29. The apparatus according to claim 25, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision

of said combined prediction by shifting bits of the combined prediction to the right, by:

inserting a rounding offset to the combined prediction before said decreasing.

30. The apparatus according to claim 25, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[31a]. A computer program product for decoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:

[31b]. determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[31c]. use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[31d]. use said second reference block to obtain a second prediction, said second prediction having the second precision;

[31e]. obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[31f]. decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[31g]. reconstruct the block of pixels based on the combined prediction.

32. The computer program product according to claim 31, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

33. The computer program product according to claim 32, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.

34. The computer program product according to claim 32, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

35. The computer program product according to claim 31, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:

insert a rounding offset to the combined prediction before said decreasing.

36. The computer program product according to claim 31, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

APPENDIX 4: UNDERSTANDING OF THE LAW

I have applied the following legal principles provided to me by counsel in arriving at the opinions set forth in this report.

Legal Standard for Prior Art

I am not an attorney. I have been informed by attorneys of the relevant legal principles and have applied them to arrive at the opinions set forth in this declaration.

I understand that the petitioner for inter partes review may request the cancelation of one or more claims of a patent based on grounds available under 35 U.S.C. § 102 and 35 U.S.C. § 103 using prior art that consists of patents and printed publications.

Anticipation and Prior Art

I understand that § 102 specifies when a challenged claim is invalid for lacking novelty over the prior art, and that this concept is also known as "anticipation." I understand that a prior art reference anticipates a challenged claim, and thus renders it invalid by anticipation, if all elements of the challenged claim are disclosed in the prior art reference. I understand the disclosure in the prior art reference can be either explicit or inherent, meaning it is necessarily present or implied. I understand that the prior art reference does not have to use the same words as the challenged claim, but all of the requirements of the claim must

be disclosed so that a person of ordinary skill in the art could make and use the claimed subject-matter.

In addition, I understand that § 102 also defines what is available for use as a prior art reference to a challenged claim.

Under § 102(a), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country before the challenged claim's date of invention.

Under § 102(b), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country more than one year prior to the challenged patent's filing date.

Under § 102(e), a challenged claim is anticipated if it was described in a published patent application that was filed by another in the United States before the challenged claim's date of invention or was described in a patent granted to another that was filed in the United States before the challenged claim's date of invention.

I understand that a challenged claim's date of invention is presumed to be the challenged patent's filing date. I also understand that the patent owner may establish an earlier invention date and "swear behind" prior art defined by § 102(a) or § 102(e) by proving (with corroborated evidence) the actual date on which the

named inventors conceived of the subject matter of the challenged claim and proving that the inventors were diligent in reducing the subject matter to practice.

I understand that the filing date of a patent is generally the filing date of the application filed in the United States that issued as the patent. However, I understand that a patent may be granted an earlier effective filing date if the patent owner properly claimed priority to an earlier patent application.

I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

I understand that when a challenged claim requires selection of an element from a list of alternatives, the prior art teaches the element if one of the alternatives is taught by the prior art.

Legal Standard for Obviousness

I understand that even if a challenged claim is not anticipated, it is still invalid if the differences between the claimed subject matter and the prior art are such that the claimed subject matter would have been obvious to a person of ordinary skill in the pertinent art at the time the alleged invention.

I understand that obviousness must be determined with respect to the challenged claim as a whole.

I understand that one cannot rely on hindsight in deciding whether a claim is obvious.

I also understand that an obviousness analysis includes the consideration of factors such as (1) the scope and content of the prior art, (2) the differences between the prior art and the challenged claim, (3) the level of ordinary skill in the pertinent art, and (4) "secondary" or "objective" evidence of non-obviousness.

Secondary or objective evidence of non-obviousness includes evidence of: (1) a long felt but unmet need in the prior art that was satisfied by the claimed invention; (2) commercial success or the lack of commercial success of the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise of the claimed invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the claimed invention; and (7) contemporaneous and independent invention by others. However, I understand that there must be a relationship between any secondary evidence of non-obviousness and the claimed invention.

I understand that a challenged claim can be invalid for obviousness over a combination of prior art references if a reason existed (at the time of the alleged invention) that would have prompted a person of ordinary skill in the art to combine elements of the prior art in the manner required by the challenged claim. I understand that this requirement is also referred to as a "motivation to combine,"

"suggestion to combine," or "reason to combine," and that there are several rationales that meet this requirement.

I understand that the prior art references themselves may provide a motivation to combine, but other times simple common sense can link two or more prior art references. I further understand that obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may come from market forces.

I understand obviousness to include, for instance, scenarios where known techniques are simply applied to other devices, systems, or processes to improve them in an expected or known way. I also understand that practical and common-sense considerations should be applied in a proper obviousness analysis. For instance, familiar items may have obvious uses beyond their primary purposes.

I understand that the combination of familiar elements according to known methods is obvious when it yields predictable results. For instance, obviousness bars patentability of a predictable variation of a technique even if the technique originated in another field of endeavor. This is because design incentives and other market forces can prompt variations of it, and predictable variations are not the product of innovation, but rather ordinary skill and common sense.

I understand that a particular combination may be obvious if it was obvious to try the combination. For example, when there is a design need or market

pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. This would result in something obvious because the result is the product not of innovation but of ordinary skill and common sense. However, I understand that it may not be obvious to try a combination when it involves unpredictable technologies.

It is further my understanding that a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill in the art, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

It is my understanding that the Manual of Patent Examining Procedure §2143 sets forth the following as exemplary rationales that support a conclusion of obviousness:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known technique to improve similar devices (methods, or products) in the same way;

- Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

A person of ordinary skill in the art looking to overcome a problem will often use the teachings of multiple publications together like pieces of a puzzle, even though the prior art does not necessarily fit perfectly together. Therefore, I understand that references for obviousness need not fit perfectly together like puzzle pieces. Instead, I understand that obviousness analysis takes into account inferences, creative steps, common sense, and practical logic and applications that a person of ordinary skill in the art would employ under the circumstances.

I understand that a claim can be obvious in light of a single reference, if the elements of the challenged claim that are not explicitly or inherently disclosed in the reference can be supplied by the common sense of one of skill in the art.

I understand that obviousness also bars the patentability of applying known or obvious design choices to the prior art. One cannot patent merely substituting one prior art element for another if the substitution can be made with predictable results. Likewise, combining prior art techniques that are interoperable with respect to one another is generally obvious and not patentable.

In order for a claim to be found invalid based upon a modification or combination of the prior art, there must be reasonable expectation that a person of ordinary skill would have successfully modified or combined the prior art to arrive at the claimed arrangement. This does not mean that it must be certain that a person of ordinary skill would have been successful – the law only requires that the person of ordinary skill in the art would have perceived a reasonable expectation of success in modifying or combining the prior art to arrive at the claimed invention.

In sum, my understanding is that obviousness invalidates claims that merely recite combinations of, or obvious variations of, prior art teachings using understanding and knowledge of one of skill in the art at the time and motivated by the general problem facing the inventor at the time. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor

at the time of the invention, can provide a reason for combining the elements of or attempting obvious variations on prior art references in the claimed manner.

Legal Standard for Claim Construction

I understand that before any invalidity analysis can be properly performed, the scope and meaning of the challenged claims must be determined by claim construction.

I understand that a patent may include two types of claims, independent claims and dependent claims. I understand that an independent claim stands alone and includes only the limitations it recites. I understand that a dependent claim depends from an independent claim or another dependent claim. I understand that a dependent claim includes all the limitations that it recites in addition to the limitations recited in the claim (or claims) from which it depends.

In comparing the challenged claims to the prior art, I have carefully considered the patent and its file history in light of the understanding of a person of skill at the time of the alleged invention.

I understand that to determine how a person of ordinary skill would have understood a claim term, one should look to sources available at the time of the alleged invention that show what a person of skill in the art would have understood disputed claim language to mean. It is my understanding that this may include what is called “intrinsic” evidence as well as “extrinsic” evidence.

I understand that, in construing a claim term, one should primarily rely on intrinsic patent evidence, which includes the words of the claims themselves, the remainder of the patent specification, and the prosecution history. I understand that extrinsic evidence, which is evidence external to the patent and the prosecution history, may also be useful in interpreting patent claims when the intrinsic evidence itself is insufficient. I understand that extrinsic evidence may include principles, concepts, terms, and other resources available to those of skill in the art at the time of the invention.

I understand that words or terms should be given their ordinary and accepted meaning unless it appears that the inventors were using them to mean something else or something more specific. I understand that to determine whether a term has special meaning, the claims, the patent specification, and the prosecution history are particularly important, and may show that the inventor gave a term a particular definition or intentionally disclaimed, disavowed, or surrendered claim scope.

I understand that the claims of a patent define the scope of the rights conferred by the patent. I understand that because the claims point out and distinctly claim the subject matter which the inventors regard as their invention, claim construction analysis must begin with and is focused on the claim language itself. I understand that the context of the term within the claim as well as other claims of the patent can inform the meaning of a claim term. For example, because

claim terms are normally used consistently throughout the patent, how a term is used in one claim can often inform the meaning of the same term in other claims. Differences among claims or claim terms can also be a useful guide in understanding the meaning of particular claim terms.

I understand that a claim term should be construed not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the entire specification. I understand that because the specification is a primary basis for construing the claims, a correct construction must align with the specification.

I understand that the prosecution history of the patent as well as art incorporated by reference or otherwise cited during the prosecution history are also highly relevant in construing claim terms. For instance, art cited by or incorporated by reference may indicate how the inventor and others of skill in the art at the time of the invention understood certain terms and concepts. Additionally, the prosecution history may show that the inventors disclaimed or disavowed claim scope, or further explained the meaning of a claim term.

With regard to extrinsic evidence, I understand that all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises, can also be considered. For example, technical dictionaries may indicate how one of skill in the art used or understood the claim

terms. However, I understand that extrinsic evidence is considered to be less reliable than intrinsic evidence, and for that reason is generally given less weight than intrinsic evidence.

I understand that in general, a term or phrase found in the introductory words or preamble of the claim, should be construed as a limitation if it recites essential structure or steps, or is necessary to give meaning to the claim. For instance, I understand preamble language may limit claim scope: (i) if dependence on a preamble phrase for antecedent basis indicates a reliance on both the preamble and claim body to define the claimed invention; (ii) if reference to the preamble is necessary to understand limitations or terms in the claim body; or (iii) if the preamble recites additional structure or steps that the specification identifies as important.

On the other hand, I understand that a preamble term or phrase is not limiting where a challenged claim defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention. I understand that to make this determination, one should review the entire patent to gain an understanding of what the inventors claim they invented and intended to encompass in the claims.

I understand that 35 U.S.C. § 112 ¶ 6 created an exception to the general rule of claim construction called a "means plus function" limitation. These types of

terms and limitations should be interpreted to cover only the corresponding structure described in the specification, and equivalents thereof. I also understand that a limitation is presumed to be a means plus function limitation if (a) the claim limitation uses the phrase "means for"; (b) the "means for" is modified by functional language; and (c) the phrase "means for" is not modified by sufficient structure for achieving the specified function.

I understand that a structure is considered structurally equivalent to the corresponding structure identified in the specification only if the differences between them are insubstantial. For instance, if the structure performs the same function in substantially the same way to achieve substantially the same result. I further understand that a structural equivalent must have been available at the time of the issuance of the claim.