UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

Petitioner

v.

NOKIA TECHNOLOGIES OY,

Patent Owner

Case: IPR2024-00626
IPR2024-00627

U.S. Patent No. 11,805,267

DECLARATION OF DR. IAIN RICHARDSON

TABLE OF CONTENTS

I. INTRODUCTION 1

II. EDUCATIONAL BACKGROUND AND QUALIFICATIONS 2

III. MATERIALS REVIEWED AND TIME OF INVENTION 5

IV. LEGAL STANDARDS 5

1. Overview 5
2. Patent Claims 6
3. Burden of Proof for Invalidity 7
4. Claim Construction 8
5. Person of Ordinary Skill in the Art 8
6. Anticipation 8
7. Obviousness 9

V. PERSON OF ORDINARY SKILL IN THE ART (POSITA) 12

VI. CLAIM CONSTRUCTION 13

VII. BACKGROUND OF THE RELEVANT TECHNOLOGY 14

1. Overview of Video Coding and Video Coding Standards 14
2. Overview of Motion Prediction in Video Coding 17
3. Technical Challenges in Motion Prediction 25

VIII. Overlap Between References Cited in Prosecution and the References Relied Upon in the Petition 34

IX. The Grounds in the Petition do not anticipate or render obvious the Challenged claims 47

1. Ground 1: Walker 47

d. Walker does not teach obtaining and combining high-precision predictions to obtain a combined prediction, as required by the claims 47

2. Ground 2: Karczewicz-I Combined with Karczewicz-II 59

Nokia Exhibit 2030, p.2
ASUS v. Nokia
IPR2025-01154

a. A POSITA would not be motivated to combine Karczewicz-I with Karczewicz-II as set out in the Petition 59

b. Without the modifications proposed in the petition, neither Karczewicz-I nor Karczewicz-II teach higher-precision predictions 67

X. CONCLUSION 68

Nokia Exhibit 2030, p.3
ASUS v. Nokia
IPR2025-01154

I, Dr. Iain Richardson, declare as follows:

I. INTRODUCTION

1. I have been retained on behalf of Nokia Technologies OY and asked to provide opinions in respect of two Inter Partes Review (IPR) proceedings, IPR2024-00626 and IPR2024-00627, both of which relate to U.S. Patent No. 11,805,267, the '267 Patent.

2. This Declaration summarises the independent opinions that I have formed to date, based on the material currently available to me. I reserve the right to modify my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this Declaration, including in response to any positions taken by Petitioners or their experts that I have not yet been provided with. I further understand that an institution decision is "all-or-nothing" and so I have focused my opinions on the independent claim at issue in this petition. Should the petition be instituted, I anticipate offering additional opinions regarding both the dependent and independent claims. I have also been asked to compare the art raised and applied in the petition to art that was before the PTO during the prosecution of the '267 Patent and its family members.

3. My findings, as explained below, are based on my study, experience, knowledge and background in the fields discussed below. I am informed by my

education and my extensive experience in the fields of video compression and multimedia coding and communications. I have also relied on my review and analysis of the prior art, information provided to me and information I have independently reviewed.

4. I am submitting the same declaration for both IPR2024-00626 and IPR2024-00627, as these petitions raise the same grounds for encoding (-00626) and decoding (-00627) claims in the same challenged patent, namely the '267 Patent. My cites to the petition herein are to the page numbers in IPR2024-00626.

II. EDUCATIONAL BACKGROUND AND QUALIFICATIONS

5. Details of my professional qualifications and background are summarized below and are set out in detail in my Curriculum Vitae, attached to this declaration.

6. I am Director and co-founder of Vcodex BV.

7. I have over 30 years of experience related to video and image compression and coding. I am qualified by education and experience to testify as an expert with respect to subject matter in the fields of video, image, multimedia and general-purpose data encoding, decoding, compression, decompression, and transmission.

8. I received a Master of Engineering (M.Eng.) degree in Electronic and Electrical Engineering from Heriot-Watt University in Edinburgh, Scotland in 1990.

I received a Doctor of Philosophy (Ph.D.) degree in Video Compression from Robert Gordon University in Aberdeen, Scotland in 1999. I worked as a Digital Signal Processing ("DSP") Hardware Designer with GEC Avionics Ltd. from 1990 to 1993.

9. In 1993, I assumed a post as a Lecturer, then Reader and eventually a Full Professor in the field of image and video compression in the School of Engineering at Robert Gordon University. In 2009, I was awarded the position of Honorary Professor with Robert Gordon University.

10. At Robert Gordon University, I founded and ran an image communication technology research laboratory. I carried out original research in the field of data, image, and video compression, initiated and managed research projects, and supervised research students.

11. I am the author of four books and over 70 journal and conference papers on video and image coding and communication, including two widely cited books on the H.264 / MPEG-4 industry standards for video compression. In my 2002 book, *Video Codec Design*, I discussed in detail the efficient implementation of image and video compression on software and hardware platforms.

12. In 1995, during my employment at Robert Gordon University, I co-founded and became Technical Director, equivalent of Chief Technology Officer, at 4i2i Communications Limited. I held this position until 2000. In my role at 4i2i, I was responsible for and engaged in the design of software and hardware modules for

compression and communication systems. The designs included video processing algorithms in software and hardware for implementation on processing devices and the design of compression algorithms such as block-based transforms, run-length coding, and variable-length coding in software and hardware. A modified version of one of my video compression hardware designs was successfully used by NASA to stream video from the International Space Station.

13. In 2007, I founded Vcodex Limited and currently serve as the CEO, through which I provide technical and strategic advice to technology companies particularly relating to video, image, and data compression for streaming.

14. In 2009, I founded Onecodec Limited and served as CEO, leading the company's development of innovative video, image, and multimedia data compression and storage software and systems. Onecodec's technology and business activities were merged with Vcodex Limited in 2015.

15. I have previously prepared expert reports and testified in a number of cases concerning video and image compression, graphics compression, general-purpose data compression, and multimedia streaming.

16. I am a named inventor on 11 patents or patent applications, including U.S. and non-US patents and applications.

17. Because of my background, training, and experience, I am qualified as an expert in the technical areas relevant to the '267 patent.

18. My company Vcodex BV is being compensated for my independent analysis as an expert with respect to this inter partes review (IPR) proceeding, but neither my compensation nor Vcodex BV's compensation is contingent in any way on the content of my analysis or the outcome of this proceeding. I have no financial interest in Petitioner (Amazon.com, Inc., Amazon.com Services LLC) or Patent Owner (Nokia Technologies Oy) and I have no financial interest in the challenged patent.

III. MATERIALS REVIEWED AND TIME OF INVENTION

19. The opinions contained in this Declaration are based on the documents that I have reviewed, including the Petition and all materials cited therein, my professional judgment, as well as my education, experience, and my knowledge regarding technologies related to the '267 Patent. I have also reviewed and considered Ye '503, Panchal, Ye '262, VCEG-AI33 and Karczewicz '057, which I understand have been designated Exhibits Ex-2007 – Ex-2011.

IV. LEGAL STANDARDS

1. Overview

20. I am not an attorney or a legal expert, and I do not opine on matters of law. All the legal standards, which are listed in this section and which I applied in reaching the conclusions expressed throughout this Declaration, were provided to me and explained to me by counsel. In preparing this Declaration, I have endeavored

to faithfully apply these legal standards. I do not intend to offer opinions on the law. Below, I describe the legal standards that I applied in my analysis of the claims of the ’267 Patent.

21. I have been informed by counsel that an issued patent is presumed valid and that the challenger of a patent’s validity bears the burden of proving invalidity of the patent.

2. Patent Claims

22. I have been informed by counsel that a patent claims describe the invention made by the inventors and describe what the patent owner owns and what the owner may prevent others from doing.

23. I have been informed by counsel that claims may describe products, such as machines or chemical compounds, or processes for making or using a product. Claims are usually divided into parts or steps called limitations.

24. I have been informed by counsel that there are two types of claims: independent claims and dependent claims. An independent claim sets forth all the requirements that must be met in order to be covered by that claim. Thus, it is not necessary to look at any other claim to determine what an independent claim covers. A dependent claim does not itself recite all of the requirements of the claim, but refers to another claim for some of its requirements. In this way, a dependent claim

depends on another claim. The law considers a dependent claim to incorporate all of the requirements of the claims to which it refers.

25. I have been informed by counsel that that a dependent claim then adds its own additional requirements. To determine what a dependent claim covers, it is necessary to look at both the dependent claim and any other claims to which it refers.

26. I have been informed by counsel that in a case where a dependent claim incorporates the limitations of an independent claim upon which it depends, if a reference fails to disclose any of the limitations of the independent claim, the claims which depend on that independent claim cannot be made invalid by such a reference. As a result, any of the references at issue in this proceeding that fail to disclose all of the limitations of an independent claim necessarily also fail to disclose all of the limitations of claims which depend on that independent claim.

3. Burden of Proof for Invalidity

27. I have been informed by counsel and I understand that the party challenging the validity of an issued U.S. patent in an IPR bears the burden of proving invalidity through facts supported by a preponderance of the evidence. I have been informed that a "preponderance of the evidence" means evidence sufficient to prove that a fact is more likely true than it is not.

4. Claim Construction

28. I have been informed by counsel and I understand that the first step in determining validity is to properly construe the claims. I have further been informed by counsel that, in this proceeding, the Board must determine the scope of the claims by using the same claim construction standard that United States District Courts use. Under this standard, the words of a claim are given their "ordinary" and "customary" meaning as would have been understood by a person of ordinary skill in the art at the time of the invention. I have been informed that such a person looks to the claims, specification, and file history in understanding the meaning of the claim terms. I have applied this understanding in this declaration.

5. Person of Ordinary Skill in the Art

29. I have been informed by counsel and I understand that a person having ordinary skill in the art may be defined by factors such as, when considered at the time of the alleged invention: (a) the levels of education and experience of the inventor and other persons actively involved in the field; (b) the types of problems encountered in the field; (c) prior art solutions to those problems; (d) rapidity with which innovations are made; and (e) the sophistication of the technology.

6. Anticipation

30. I have been informed by counsel and I understand that a patent claim can be found invalid as anticipated if each and every element of the claim, as

properly construed, is found either explicitly or inherently in a single prior art reference. If the single prior art reference does not disclose explicitly or inherently any one claim limitation, then that claim is not anticipated. I have further been informed that the single prior art reference cannot merely disclose each element; rather, it must disclose all of the elements as arranged in the claim. I have been informed that for a prior art reference to "inherently" disclose something, the reference must necessarily function in accordance with, or include the claimed limitations, to anticipate. The missing descriptive matter cannot merely probably or possibly include the claimed limitations. I have further been informed by counsel that, with respect to an anticipation analysis, it is improper to combine separate, distinct embodiments of a single prior art reference.

31. I have been informed that for a reference to be anticipatory, the reference must contain an enabling disclosure. I have been informed that merely naming or describing the subject matter is insufficient if it cannot be produced without undue experimentation.

7. Obviousness

32. I have been informed by counsel that a patent claim can be found invalid as obvious if the differences between the claim and the prior art references would have been obvious to a person of ordinary skill in the art at the time of the invention, taking into account (1) the scope and content of the prior art, (2) the

differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any objective indicia of non-obviousness. I have further been informed by counsel that a claim can be found to be obvious in light of either a single prior art reference or multiple prior art references. I have been informed that references qualify as prior art for an obviousness determination only when they are analogous to the claimed invention. Two separate tests define the scope of analogous prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed; and (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

33. I have been informed that to be obvious in light of a single prior art reference or multiple prior art references, there must be a reason either to modify the single prior art reference or to combine the two or more prior art references to achieve the claimed invention. I have been informed by counsel that this reason may come from a teaching, suggestion, or motivation to combine which may come from the reference or references themselves, the knowledge or "common sense" of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole.

34. I have been informed by counsel that the combination of familiar elements according to known methods is likely to be obvious when it does no more

than yield predictable results.

35. I have further been informed by counsel that there is no suggestion to combine, however, if a reference teaches away from its combination with another source. A reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant, or if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result sought by the applicant. If, when combined, the references would produce a seemingly inoperative device, then they teach away from their combination. I have been informed that it is improper to combine references where the references teach away from their combination.

36. I have been informed by counsel that the degree of teaching away will depend on the particular facts; in general, a reference will teach away if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result sought by the applicant. I have been informed that a reference teaches away, for example, if (1) the combination would produce a seemingly inoperative device, or (2) the references leave the impression that the product would not have the property sought by the applicant.

37. I have been informed by counsel that if the teachings of a prior art reference would lead one skilled in the art to make a modification that would render another prior art device inoperable, then such a modification would generally not be obvious. I have been informed that if a proposed modification would render the prior art device unsatisfactory for its intended purpose, then the proposed modification would generally not be obvious.

38. I have been informed by counsel that it is improper to use hindsight in making the obviousness determination.

39. I have been informed by counsel that some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness must be made. I have been informed that, for example, there may be no rational underpinning to support the legal conclusion of obviousness where systems are already complete and the party challenging validity seeks to mix and match components, which each exist in the separate systems, from these complete systems.

V. PERSON OF ORDINARY SKILL IN THE ART (POSITA)

40. Based on my review of the ’267 Patent, the types of problems encountered in the art at and prior to the time of the alleged invention, prior solutions to those problems, the pace with which innovations were made during the relevant time period, the sophistication of the technology, and the educational level of active

professionals in the field, I believe that a person of ordinary skill in the art, at the time of the invention, would have had at least a Bachelor's degree in electrical engineering, computer science, or computer engineering; familiarity with the relevant video coding standards as of this time; and at least two years of work experience in research, design, development, and/or testing of video coding hardware and/or software. More education could have substituted for less work experience and vice versa.

41. All of my opinions in this Declaration are from the perspective of a person of ordinary skill in the art, as I have defined it above and during the relevant time frame (e.g., 2011). During that time frame, I possessed at least the qualifications of a POSITA, as defined above.

VI. CLAIM CONSTRUCTION

42. I understand that claim terms are typically given their ordinary and customary meaning, as would have been understood by a person of ordinary skill in the art at the time of the alleged invention. In considering the meaning of the claims, I also understand that one must consider the claim language, specification, prosecution history, and the knowledge of a person of ordinary skill in the art at the relevant time.

VII. BACKGROUND OF THE RELEVANT TECHNOLOGY

1. Overview of Video Coding and Video Coding Standards

43. The advent and use of modern-day video coding and compression schemes involves a delicate balancing of competing concerns. On one hand, video technicians and consumers demand high fidelity and large resolution of the video images to make use of large high definition displays and wider availability of high-speed Internet connections. However, the competing concern becomes the efficient use of storage space on hard drives and memory cards as well as available bandwidth of Internet connections and computer processing demands. The use of modern video coding and compression schemes aim to represent the video images in a compressed form with as little data as possible, while maintaining as much quality of the original source images as possible after decoding and ensuring that the video image can be decoded and displayed in real time. Present-day applications of video coding often involve a trade-off between computation, compression and video quality. Increasing the amount by which data is compressed can save storage space and transmission bandwidth, but may result in degraded video quality to the point where the visual quality of decoded video is unacceptably poor when viewed by the consumer. More compression for the same video quality, or better video quality for the same level of compression, may be achieved at the expense of increased computational processing and/or increased memory requirements. However, an increase in computational

processing may require more sophisticated and perhaps more expensive processors and may lead to problems such as increased battery drain on mobile devices or reduced video frame rates.

Excerpt from Richardson 2010 book (Ex-2012[1]) at page 69:

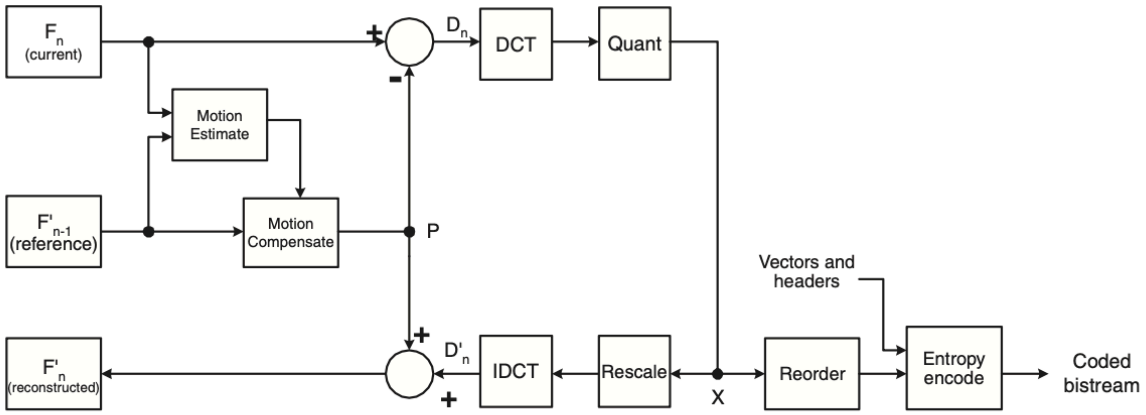


Figure 3.51 DPCM/DCT video encoder

……

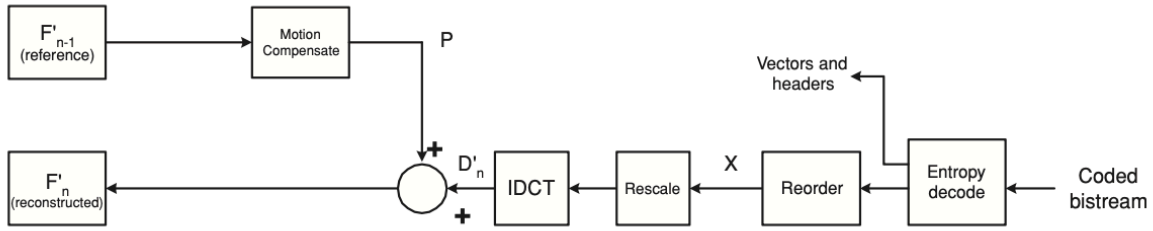


Figure 3.52 DPCM/DCT video decoder

44. A video codec comprises a video encoder and a video decoder, which

[1] I refer here to Exhibit 2012 which contains excerpts from my 2010 book, "The H.264 Advanced Video Compression Standard".

I illustrate in the above block diagrams from my 2010 book. A video encoder converts a sequence of input video frames (such as frame F'_n in the above diagram, video encoder) into a coded bitstream, which is a compressed representation suitable for storage or transmission and a video decoder converts such a compressed representation into a series of decoded video frames (such as frames F'_n in the above diagram, video decoder). (Ex-1001, 1:26-33). As the '267 patent explains, a video encoder forms a prediction for each block of pixels, labelled P in the above diagrams, using motion compensated prediction from an area in a previously coded video frame or spatial prediction from previously coded pixels in the same frame. The prediction is subtracted from the block of pixels to produce a residual or error, labelled D_n in the above encoder diagram. The residual or error is transformed, quantized and entropy encoded. (Ex-1001, 1:34-59). A video decoder entropy decodes the coded bitstream and rescales and inverse transforms block data in order to produce a decoded residual or error, labelled D'_n in the above decoder diagram. The decoder also decodes information that enables it to form the same prediction as the encoder, labelled P above, which it adds to the decoded residual in order to construct a decoded block of pixels and thus a decoded frame F'_n.

45. By 2011, a POSITA would have been familiar with video coding standards such as the ITU-T H.263 and H.264 standards discussed in the '267 patent (Ex-1001, 1:34-37). In common with other video coding standards, these standards

each specify a format for a compressed bitstream and a method of decoding such a bitstream to produce a decoded video sequence, but do not fully specify an encoding process. A video decoder conforming to a standard such as H.263 or H.264 decodes a bitstream that conforms to the specified bitstream format, in order to produce the same output as the specified decoding process. A video encoder encodes a bitstream that conforms to the specified bitstream format and that can be successfully decoded by the specified decoding process. Even though a standard such as H.263 or H.264 does not fully specify an encoder, in practical encoders the steps carried out when encoding video largely mirror those carried out when decoding video. For example, both encoder and decoder must construct an identical prediction P for each block, whether using spatial prediction (intraframe prediction) or motion compensated prediction (interframe prediction). The decoder's inverse transform (labelled IDCT in the diagrams above) is the inverse of the encoder's forward transform. The decoder's rescaling process mirrors (as far as is possible) the encoder's quantization process and the decoder's entropy decoding process reverses the encoder's entropy encoding process. Thus the specification of the decoder and the bitstream places certain constraints on the design of the encoder, in order for encoder and decoder to interoperate successfully.

2. Overview of Motion Prediction in Video Coding

46. A widely-used technique in video compression is the encoding of the

source video through motion compensated prediction. Ex-2012 at 28-38. This process can be carried out by dividing the individual pixel elements (pixels) of a video frame into blocks of pixels. Motion compensated prediction attempts to find a prediction or match for a block of pixels in a current video frame, in a correspondingly sized region in a previously coded frame. Such a match may be found in the same position as the current block or in an offset or shifted position in the previously coded frame. Such a shift in position may indicate, for example, that motion or other change is occurring in a particular area of the video scene. Motion compensated prediction has a very significant contribution to compression performance, since encoding the motion compensated differences between video frames typically results in much less data than encoding each video frame without such motion compensated prediction.

47. A video encoder or decoder may process video frame data in units of a macroblock, which corresponds to 16x16 displayed pixels. Each macroblock may be partitioned or split into smaller blocks for the purpose of constructing prediction blocks, where the smaller blocks may be square or rectangular in size. In the H.264 standard (Ex-1014), inter prediction blocks for the luma (brightness) component of a video image may have the following sizes: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4. Thus the smallest H.264 luma inter prediction block size is 4x4 and the largest inter prediction block size is 16x16. The choice of inter prediction block size is typically

made by the encoder. See E.g. Fig. 12 from Ex-1011, illustrating the inter prediction block sizes available according to the H.264 standard.

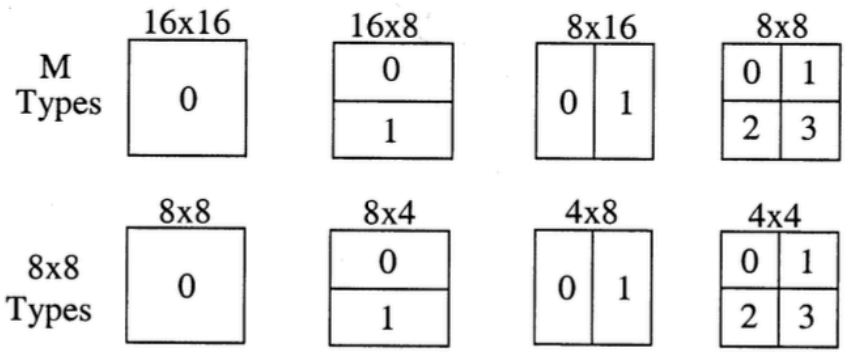


Fig. 12. Segmentations of the macroblock for motion compensation. Top: segmentation of macroblocks, bottom: segmentation of 8×8 partitions.

48. A video encoder or decoder creates a prediction block for each block of pixel values in a video frame, using image information such as previously-coded pixel values in the same image frame ('intra prediction') or previously-coded pixel values in a previous or later image frame in display order ('inter prediction'). (Ex-1001, 1:37-51). For each block of pixel values, the encoder subtracts a prediction block from the current block to form a residual or error block, which is encoded into the compressed bitstream. The decoder adds the prediction block to the decoded residual block in order to reconstruct a version of the original pixel block. Typically, both encoder and decoder create an identical prediction block (Ex-1001, 1:52-2:12).

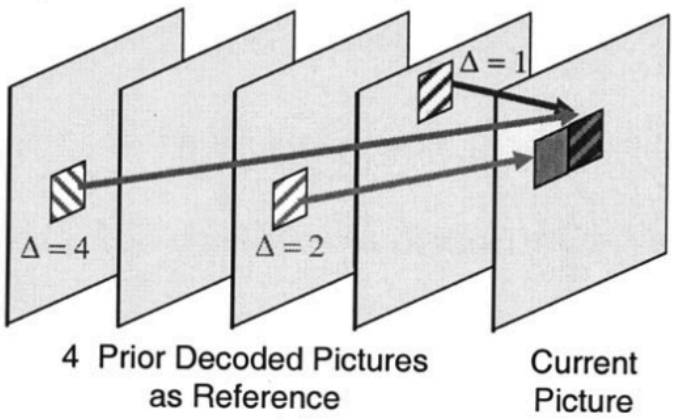
49. An inter prediction block may be constructed using Motion

Compensated Prediction (MCP), based on information in a previously encoded video frame. Both encoder and decoder typically store a set of reference pictures in a buffer or memory, each of which is a previously encoded and/or decoded video frame. When encoding or decoding a video sequence, both encoder and decoder maintain the same set of reference pictures, so that each can create the same inter predictions. The set of reference pictures may include previously coded frames that are displayed before to the current frame ('past' reference pictures) and/or previously coded frames that are displayed after the current frame ('future' reference pictures). An inter prediction block is formed by identifying a matching region in a reference picture, i.e. a region that may be the same size as the current block, either in the same position or offset in a horizontal and/or vertical direction. The offset between the current block position and the matching region is known as a motion vector. Motion vectors for each inter prediction block are coded into the compressed bitstream. The decoder can use the decoded motion vectors to reconstruct each prediction block (Ex-1001, 2:35-51).

50. An inter prediction block may be formed using MCP from a single reference picture (uni-prediction) or by combining predictions from two reference pictures (bi-prediction). The following Figure from Ex-1011 illustrates these two types of prediction. A prediction for the left-hand block in the illustrated Current Picture is constructed using uni-prediction from a single reference picture, two

frames previous to the current picture (illustrated as Δ=2). A prediction for the right-hand block in the Current Picture is constructed using bi-prediction from two reference pictures, respectively located four frames in the past (Δ=4) and one frame in the past (Δ=1). Note that in this example, all of the reference pictures are 'past' reference pictures, but in general the encoder and decoder may make use of 'past' and/or 'future' reference pictures to form block predictions.

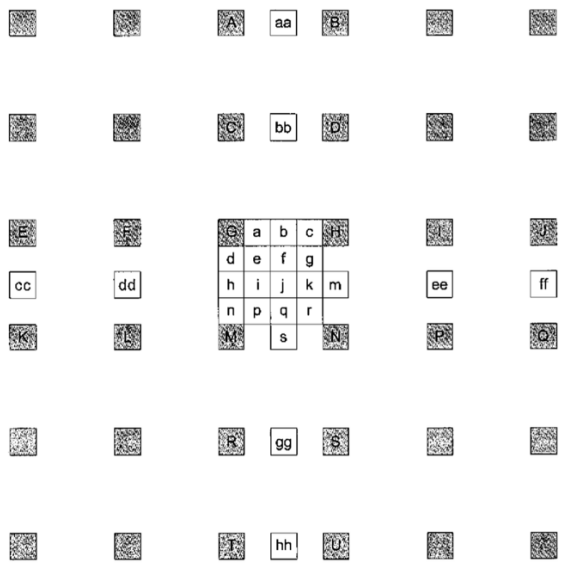
H.264 Overview (Ex-1011) at page 570:



51. In a video codec such as a codec operating according to the ITU-T H.264 video coding standard, each frame may be coded as an I-picture, P-picture or B-picture. An I-picture contains only intra predicted blocks. A P-picture contains intra predicted blocks or inter predicted blocks using uni-prediction only. A B-picture contains blocks that may be intra predicted, inter predicted using uni-prediction or inter predicted using bi-prediction. The 'Current Picture' in the above

Figure is an example of a B-picture, in which some blocks are inter predicted using uni-prediction and others are inter predicted using bi-prediction. For a uni-predicted block, the encoder chooses and encodes an index to one reference picture (e.g. indicating Δ=2 in the above example) and a motion vector. For a bi-predicted block, the encoder chooses and encodes the indices of two reference pictures (e.g. indicating Δ=1 and Δ=4 in the above example) and two motion vectors (Ex-1001, 2:35:59).

52. In some codecs, a motion vector may point to a fractional-pixel position in the reference picture, where the fractional-pixel locations refer to locations in between image pixels. This can enable the encoder and decoder to construct a prediction that is a better match than a prediction that is entirely formed using the actual pixels of the reference picture. The MCP process may involve applying interpolation filter(s) to the image pixels in a reference picture or frame, in order to generate interpolated fractional-pixel values (Ex-1001, 2:60-3:11). Fig. 12 of the '267 Patent illustrates examples of image pixel positions (also known as 'integer' pixel positions), such as G, H, M and N, and examples of fractional pixel positions such as a, b and c. In the example of Fig. 12, fractional pixel positions at ½ and ¼-pixel resolution may be generated. For example, position b is a ½-pixel position between G and H. Positions a and c are ¼-pixel and ¾-pixel positions respectively between G and H. (Ex-1001, 2:60-3:11 and Fig. 12).



53. When constructing a prediction block using bi-prediction, the encoder and decoder combine two motion compensated prediction blocks which may be located in two reference pictures, such as the two blocks located in reference frames at Δ=1 and Δ=4 in the earlier example. The two prediction blocks may be combined by averaging, in order to create a single bi-predicted prediction block (Ex-1001, 3:49-55).

54. As I explained earlier, in order to carry out motion compensated prediction, an encoder identifies a block of pixels in a reference picture, and determines the offset between the current block and the block of pixels in the reference picture, with the offset indicated using a data structure called a motion vector. The motion vector can either indicate an integer pixel offset (e.g. +1, -4) which indicates a block of pixels that exists in the reference picture, or a sub-pixel

offset (e.g. +1.25, -4.5) which indicates that an interpolation operation is to be performed on the pixels in the reference picture.

55. After identifying a potential motion vector to use for a prediction, an encoder constructs a prediction block, i.e. a block of predicted pixel values, using the motion vector and the indicated region of reference pixels (using interpolation if necessary) and subtracts the prediction block from the block to be coded, resulting in a prediction error signal or residual signal. The residual signal is coded and transmitted, with the corresponding motion vector, in the coded video bitstream.

56. The more similar the prediction is to the current block, the smaller the residual signal is and the fewer bits need to be included in the bitstream to represent that block. Accurate predictions, i.e. predictions that produce a small residual signal, are therefore very important for efficient video coding.

57. As part of the corresponding decoding process, both the motion vector and the prediction error signal are decoded from the coded video bitstream. The decoder uses the motion vector to construct the same prediction the encoder made, based on pixels stored in a reference picture and applying interpolation if indicated by the motion vector, and adds the decoded error prediction signal to the prediction to reconstruct a decoded version of the original block.

3. Technical Challenges in Motion Prediction

58. While both the encoder and decoder create predictions from blocks of reference pixels based on motion vectors, their tasks differ significantly. The encoder must find and construct the prediction to use for a given block, and use the prediction to make the residual value. The process of finding a suitable motion vector and prediction block to use for motion compensated prediction is called motion estimation.

59. Motion estimation and motion compensated prediction typically has high computation and storage costs at the encoder. The encoder must find a closely matching prediction for the block being coded. The better the match, the less error data must be sent, and the more efficient the bit stream. To do this, the encoder evaluates potential matches in a search area to find an effective match (Ex-1001, 2:26-29), typically a prediction block that minimizes the number of bits required to code both the residual and the motion vector information.

60. In order to find a near-optimal matching region, many potential matches in a search area around the current block must be generated and evaluated by the encoder. Each potential match in area searched by the encoder has an associated motion vector, which indicates the distance from the upper left corner of the current block to a pixel position that forms the upper left corner of a matching block of pixels. As I explain above, motion vectors can either identify an actual pixel

location or a location between the actual pixels. Motion vectors identifying actual pixels in the reference picture are called integer motion vectors while motion vectors identifying locations between pixels in the reference picture are called fractional or sub-pixel motion vectors. When a sub-pixel motion vector is chosen, generating the prediction block requires interpolation of pixel values in the reference picture to create interpolated values. Interpolation may require a computationally complex series of operations such as multiplications, additions, and divisions to arrive at a final interpolated value that can be used as a prediction (Ex-1001, 3:7-11). Generating potential prediction values for sub-pixel positions is therefore typically computationally expensive.

61. For each current block, an encoder typically selects one matching region in a reference picture, with a corresponding motion vector. However, certain integer-pixel or sub-pixel values may require to be searched for multiple current blocks. For example, potential matching regions that are offset to the right of a first current block may overlap with potential matching regions that are offset to the left of a second current block, and so on. Thus an encoder may choose to store interpolated pixel values (i.e. calculated sub-pixel values) for use in motion estimation for multiple current blocks. One strategy that may be efficient and which is used in practical implementations, is for the encoder to calculate all the possible interpolated pixel positions for a reference picture, prior to carrying out motion

estimation for blocks of a current picture.

62. The use of bi-prediction increases the computational complexity of the search for potential prediction blocks. The encoder typically has to search at least two reference pictures for suitable predictions, match the candidate predictions from each reference picture with candidate predictions from other reference picture(s) to identify a suitable set of predictions to combine, and then evaluate the residual signal that would result from using that combined prediction in order to identify the best pair of predictions, i.e. the best choice of first and second reference pictures and first and second motion vectors. Video coding standards such as H.264/AVC allow for lists of reference pictures, meaning that for each of the two prediction blocks that are combined to form a bi-predicted prediction block, it may be necessary or beneficial for an encoder to evaluate the candidates in the search spaces of multiple reference pictures. Thus, storing and saving interpolated pixel positions for potential use in forming multiple candidate prediction blocks, may result in a significant computational saving at the encoder side, compared with re-calculating each interpolated pixel position every time it is evaluated during motion estimation.

63. A video decoder must construct the identical prediction block to the block selected by the encoder. If the encoder has chosen a sub-pixel motion vector, indicating that a prediction block should be constructed using interpolation of sub-pixel positions, then the decoder must carry out the same interpolation process to

generate the same prediction block. However, the decoder typically has to carry out at most one interpolation process for each pixel position in a prediction block (or two if bi-prediction is used), rather than interpolating many possible sub-pixel positions as described above with respect to the encoder. Thus in general, motion compensated prediction at the decoder is computationally simpler than the combination of motion estimation and motion compensated prediction at the encoder. However, any increase in computational complexity due to the encoder implementing (for example) a more complex interpolation process, will have an impact on the computational complexity of the decoder, since the decoder has to construct the identical interpolated prediction. Thus, increasing the computational complexity of interpolation at the encoder has the result of increasing the computational complexity of interpolation at the decoder (or vice versa).

64. An important parameter in video coding is the bit depth or precision at which pixels are represented and at which calculations are performed. Pixel bit depth defines the range of possible values for the pixels in a video, such that more bit depth allows colors and contrasts in brightness to be more accurately represented. For example, representing pixel values in 8-bit precision enables each pixel value (such as a luma or brightness value) to take any one of 256 values (2^8 =256), for example in a range from 0-255. Representing a pixel value with 10-bit precision enables it to take any one of 1024 values, for example in a range from 0-1023. When

displayed on a suitable display device, a 10-bit image may provide a more faithful representation of the original image, for example with a greater contrast between dark and light areas and more nuanced brightnesses in between, and with greater colour fidelity. In terms of precision and quality of the displayed image, a higher bit depth may therefore be advantageous.

65. However, increasing the bit depth of pixel values has a number of implications for computational and memory requirements. Moving from 8 bits to 10 bits of bit depth, for example, increases the amount of memory (e.g. the number of stored bits) required to store each pixel or image sample by 25%. Encoding or decoding video requires multiple calculations to be carried out by a processor. When processing each pixel sample within an encoder or decoder, all 10 bits need to be made available before a calculation can take place, which requires more connections (wider data buses) and/or more clock cycles when processing each calculation, compared with processing 8-bit pixel values. When carrying out encoding or decoding processing, any digital logic for adding, subtracting, multiplying, dividing and comparing sample values must accommodate two additional bits. As a result, representing pixel values or other values at a higher precision in a video codec, comes at a cost of increased computation, increased memory requirements and increased rate of access to stored values ("memory bandwidth"). Computation, memory and memory bandwidth are at a premium in consumer electronic devices.

66. Computation and memory requirements are an important consideration in the design of video encoders and decoders (Ex-2012, Richardson H.264 book, 2010, pages 223-229). In recognition of this, well-known standards for video coding such as H.264 impose certain limits on computation and memory through the use of Profiles and Levels. A Profile is a set of processes or algorithms for video compression that limits the algorithmic capabilities of a video decoder. A Level places an upper limit on the computational and storage requirements of a video decoder. For example, H.264's Main Profile and Level 4.1 enables coding and decoding of High Definition video with a precision of 8 bits per luma pixel value (Ex-1014, A.2.2 for MP). H.264's High10 Profile extends this to a precision of up to 10 bits per luma pixel value (Ex-1014, A.2.5 for H10P). Thus the question of how many bits are used to represent pixel values and other values is an important consideration in the specification of video codecs and the design of practical video encoders and decoders.

67. In video coding, sub-pixel interpolation is used to generate predictions for motion compensated prediction using a set of weights applied to surrounding pixel values (Ex-1001. 3:6-11). Each full-pixel value considered contributes some fractional part of the new interpolated value. Applied to a block, interpolated values can create prediction blocks with better performance than the underlying blocks of full pixel values.

68. Interpolation in video coding is generally accomplished by an application of integer weights to underlying pixel values followed by a sum of the resulting weighted values, followed by a division by the sum of the weights. Interpolation is typically performed in this manner because multiplying by integers is relatively computationally simple in commonly-available digital hardware and division is relatively computationally complex. Further, choosing filter weights that add to a power of 2 allows the division to be carried out by a right-shift, which is a simple operation in digital logic. Before applying the division or right-shift, such an interpolation method produces an intermediate value at higher precision than the underlying pixel values. The H.264 video coding standard, specifies an interpolation process where certain higher-precision intermediate values (for example, intermediate values labelled b1, h1 and j1) are temporarily calculated to allow additional values to be interpolated using the intermediate values as inputs. These intermediate values (b1, h1, j1) are just that and are not themselves predictions or outputs of the interpolation process in the disclosure of H.264. Each output of the H.264 sub-pixel interpolation process (sub-pixel interpolated values a, b, c, d, e, f, g, h, i, j, k, n, p, q and/or r, H.264 Figure 8-4) is the same precision as the original input pixels, so that the output value can be used in motion compensated prediction. Ex-1014 at 190-192, including Figure 8-4 and equations 8-243 to 8-249. Portions of Ex-1014 at pages 190 and 191 are reproduced below:

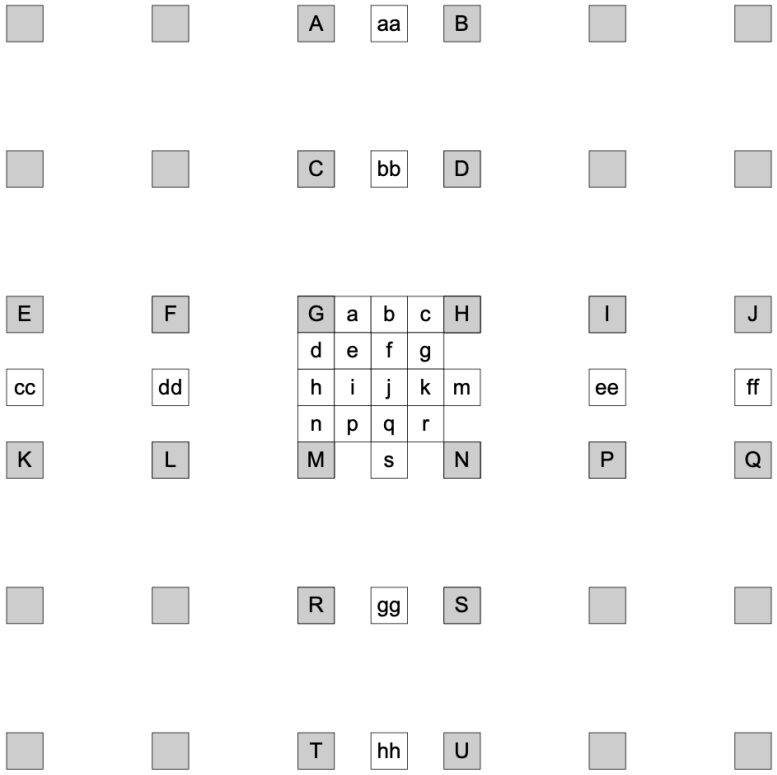


Figure 8-4 – Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation

Nokia Exhibit 2030, p.35
ASUS v. Nokia
IPR2025-01154

– The samples at half sample positions labelled b are derived by first calculating intermediate values denoted as b_1 by applying the 6-tap filter to the nearest integer position samples in the horizontal direction. The samples at half sample positions labelled h are derived by first calculating intermediate values denoted as h_1 by applying the 6-tap filter to the nearest integer position samples in the vertical direction:

$$b_1 = (E - 5 * F + 20 * G + 20 * H - 5 * I + J) \quad (8\text{-}243)$$
$$h_1 = (A - 5 * C + 20 * G + 20 * M - 5 * R + T) \quad (8\text{-}244)$$

The final prediction values b and h are derived using

$$b = \mathrm{Clip1}_Y((b_1 + 16) >> 5) \quad (8\text{-}245)$$
$$h = \mathrm{Clip1}_Y((h_1 + 16) >> 5) \quad (8\text{-}246)$$

– The samples at half sample position labelled as j are derived by first calculating intermediate value denoted as j_1 by applying the 6-tap filter to the intermediate values of the closest half sample positions in either the horizontal or vertical direction because these yield an equal result:

$$j_1 = cc - 5 * dd + 20 * h_1 + 20 * m_1 - 5 * ee + ff, \text{ or} \quad (8\text{-}247)$$
$$j_1 = aa - 5 * bb + 20 * b_1 + 20 * s_1 - 5 * gg + hh \quad (8\text{-}248)$$

where intermediate values denoted as aa, bb, gg, s_1 and hh are derived by applying the 6-tap filter horizontally in the same manner as the derivation of b_1 and intermediate values denoted as cc, dd, ee, m_1 and ff are derived by applying the 6-tap filter vertically in the same manner as the derivation of h_1. The final prediction value j are derived using

$$j = \mathrm{Clip1}_Y((j_1 + 512) >> 10) \quad (8\text{-}249)$$

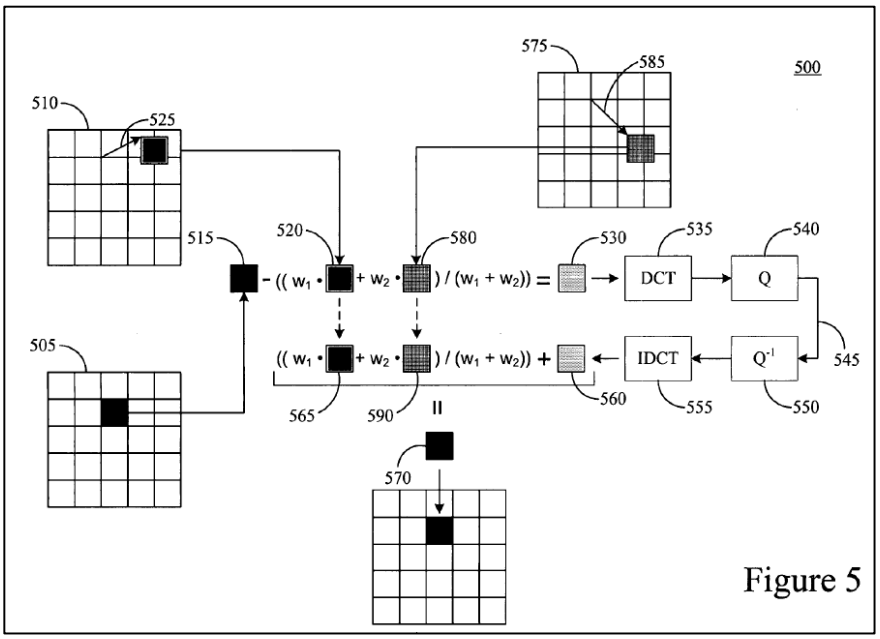
69. Relevant to the discussion herein, a POSITA would understand that the values of a prediction block and the values of a current block each have the same precision, so that an encoder can subtract the prediction block from the current block to form a residual and so that a decoder can add the prediction block to a residual in order to obtain a decoded block. For example, the H.264 standard specifies that prediction block values and current block values are each stored and used with the same precision, for example 8-bit precision when using H.264's Main Profile (Ex-1014 at page 198, specifying the process for constructing decoded block samples u_{ij} from prediction values $pred_L$ and residual values r_{ij}).

70. Arbitrarily increasing the bit depth of certain values such as uni-predicted prediction blocks can have a significant impact on decoder computation, memory and memory bandwidth.

VIII. OVERLAP BETWEEN REFERENCES CITED IN PROSECUTION AND THE REFERENCES RELIED UPON IN THE PETITION

a. The portions of the Walker reference relied upon in the Petition has substantially the same bi-prediction teaching as found in Ye '262, Panchal, and Karczewicz '057, references cited in prosecution and on the face of the '267 Patent.

71. The first ground of the petition relies on Walker for its disclosure of weighted prediction, as illustrated in Walker Figure 5, equation 13, and Table 2. Examples of these disclosures are reproduced below, each of which illustrates the same weighted bi-prediction concept.



Ex-1004 Figure 5

[0068] Case 3: If predFlagL0=1 and predFlagL1=1

$$Final_pred = Clip1[(((pred0)w0+(pred1)w1+2^{logWD})>>(\log WD+1))+((o_0+o_1+1)>>1)] \quad (13)$$

[0069] where descriptions, bit size and range of values for the variables are given in Table 1.

TABLE 1

Variables	Description	Range	No. of Bits
logWD	The base 2 logarithm of the denominator for all the luma or all the chroma weighting factors (luma and chroma weighting factors can be derived separately)	0 to 7	3
w0	The weighting factor applied to the luma or chroma prediction value for the first reference picture (list "0")	−128 to 127	8
w1	The weighting factor applied to the luma or chroma prediction value for the second reference picture (list "1")	−128 to 127	8

Ex-1004 ¶¶ 68-69 (including equation 13)

TABLE 2

Operation No.	Operation	Bitwidths Involved	Bitwidth of Operation Result
1	(pred0)w0, (pred1)w1	8 bits * 8 bits	16
2	2^{LWD-1}	logWD has a maximum value of 7, therefore LWD − 1 has a maximum value of 7 for case 3	7
3	(pred0)w0 + (pred1)w1 + 2^{LWD-1}	16 bits + 16 bits + 7 bits	18
4	(((pred0)w0 + (pred1)w1 + 2^{logWD}) >> (LWD + 1))	(18 bits) >> (8 bits)* *maximum value of LWD + 1	10
5	$(o_0 + o_1 + 1)$ >> (1)	(8 bits + 8 bits + 1) >> (1)	8
6	Clip1[Op. 4 + Op. 5] (for Clip1, see Eq. (11))	Clip1[10 bits + 8 bits]	8

Ex-1004 Table 2

72. The petition argues that Walker teaches a first and second prediction at a second precision, higher than a first precision, because in Walker the 8-bit predictions derived from individual reference pictures, pred0 and pred1, are multiplied by corresponding 8-bit weights, w0 and w1, before the resulting products are added together. Petition at 18-21.[2]

73. However, multiple references considered during prosecution taught the same weighted prediction scheme Petitioner relies upon in Walker.

74. For example, the Ye '262 reference that that formed the basis for multiple rejections during prosecution of the '893 application (Ex-1007, 146) taught an equation for bi-prediction that in my opinion is nearly identical to Walker's equation for bi-prediction. Walker's bi-prediction equation calculates a bi-predicted value Final_pred based on a first prediction pred0 and a second prediction pred1. Ye 262's bi-prediction equation calculates a bi-predicted value P(x,y) based on a first prediction $P_0(x,y)$ and a second prediction $P_1(x,y)$:

---

[2] Note I am submitting the same declaration for both IPR2024-00626 and IPR2024-00627, as these petitions raise the same grounds for encoding (-00626) and decoding (-00627) claims in the same patent. My cites to the petition are to the page numbers in IPR2024-00626.

Walker (Ex-1004) ¶ 68	Ye '262 (Ex-2009) ¶ 25
$Final_pred = Clip1[(((pred0)w0+(pred1)w1+2^{\log WD})>>(\log WD+1))+((o_0+o_1+1)>>1)]$	$P(x,y)=(w \cdot P_0(x,y)+(W-w) \cdot P_1(x,y)+W/2)>>S$

75. Compared to Ye '262, Walker's equation uses a function Clip1. Clip1 limits the Final_pred value to a range between 0 and 255 if the initial result (i.e. the portion of the equation within square parentheses […]) was outside of that range, but otherwise does nothing. Ex-1004 ¶ 66. The use of the Clip1 function in Walker to limit Final_pred to the allowed range does not appear to be relied upon in the petition to satisfy any claim limitations. I therefore conclude that the presence of the Clip1 function in Walker but not in Ye '262 is not relevant to the arguments raised in the petition.

76. The $+((o_0 + o_1 + 1)>>1)$ term, present in Walker's equation but absent from the equation in Ye '262, adds 1 if either or both offsets o_0 and o_1 are set to 1.This addition in Walker is not cited in the petition to satisfy any claim language. I therefore conclude that the presence of offsets o_0 and o_1 in Walker but not in Ye '262 is not relevant to the arguments raised in the petition.

77. Both Walker and Ye '262 equations multiply a first prediction (pred0 in Walker, $P_0(x,y)$ in Ye '262) by a first weight (w0 in Walker, w in Ye '262). Both equations multiply a second prediction (pred1 in Walker, $P_1(x,y)$ in Ye '262) by a

second weight (w1 in Walker, W-w in Ye ’262). Both equations add an offset for rounding (2^{logWD} in Walker, W/2 in Ye ’262). Both equations right-shift the result (>>(logWD+1) in Walker, >>S in Ye ’262, where >> denotes a binary right-shift). In the case that S = logWD+1, Walker’s right-shift amount logWD+1 (ignoring (o_0 + o_1 + 1)>>1) is identical to Ye ‘262’s right-shift amount S and Walker’s rounding offset 2^{logWD} is identical to Ye’s rounding offset W/2, since:

$W = 2^S$ (Ex-2009, Ye ‘262 at [0025])

$W/2 = 2^{S-1}$ (Ex-2009, Ye ‘262 at [0025])

2^{logWD} (Ex-1004, Walker at ¶ 68) = 2^{S-1} = W/2 (Ex-2009, Ye ‘262 at [0025])

78. Thus in my opinion there are no material differences between the weighted bi-prediction teaching in Walker relied upon in the petition and the weighted bi-prediction teaching in Ye ’262 which was central to the prosecution of the ’267 Patent’s original priority application.

79. Similar to Ye ‘262, the reference Panchal (Ex-2008), cited by the examiner in the priority ’893 Application prosecution (Ex-1007, 152) and then cited by Patent owner in every subsequent application, includes the same offsets and the two-stage right-shift found in Walker, with the parameter *r* in Panchal substituting for logWD in Walker and a slight variation in notation:

Walker (Ex-1004) ¶ 68	Panchal (Ex-2008) ¶ 61
$Final_pred=Clip1[(((pred0)w0+(pred1)w1+2^{\log WD})>>(\log WD+1))+((o_0+o_1+1)>>1)]$	$pred(i,j)=(pred0(i,j)*w0+pred1(i,j)*w1+2^r)>>(r+1)+((o1+o2+1)>>1)$

80. The exact manner in which offsets are rounded and in which right-shifts are applied does not affect the arguments put forward by Petitioner in respect of Ground 1. Petitioner relies on the specific calculation (pred0)w0 in Walker (Ex-1004, ¶ 68). This specific calculation is taught by Panchal and Ye '262, both of which were considered during prosecution. Walker's complete bi-prediction calculation is taught in substantially the same form in both Panchal and Ye '262. The only difference between the equations in Walker and Panchal is that the final value in Walker includes a clip function. As noted above, this clip function only serves to keep values within the allowed 0 to 255 bounds, and in my opinion is not material to any claim limitation at issue.

81. Karczewicz '057, cited to the examiner during the prosecution of the application that issued as the '267 Patent (Ex-1002, 75) also teaches the same equation for bi-prediction:

Walker (Ex-1004) ¶ 68	Karczewicz '057 (Ex-2011) at 13:11-12
$Final_pred=Clip1[(((pred0)w0+(pred1)w1+2^{\log WD})>>(\log WD+1))+((o_0+o_1+1)>>1)]$	Bidirectional prediction:$pred(i,j)=(pred0(i,j)*w0+pred1(i,j)*w1+2^r)>>(r+1)+((o1+o2+1)>>1)$

82. Karczewicz ’057, in an almost identical manner to Panchal, teaches the same weighted bi-prediction equation as Walker.

83. Thus the disclosure in Walker relied upon by Petitioner was included in at least three different references cited to the patent office during prosecution of the ’267 Patent, including a reference used as the basis for a rejection for one of the ’267 Patent’s priority applications.

b. **The portions of the Karczewicz-II reference relied upon in the Petition have substantially the same disclosure of interpolation taught in the Ye ’503 and VCEG-AI33 references cited in prosecution and on the face of the ’267 Patent.**

84. Petitioner’s Ground 2 relies on Karczewicz-II for its disclosure of sub-pixel interpolation using intermediate values with higher precision, as illustrated in Karczewicz-II tables 3, 4, 5, 6, and 8. These tables include the equations for Karczewicz-II’s higher precision interpolation method and an illustration of how to implement the method without exceeding the limits of 16 bit registers. As I explain below, Petitioner seeks to modify Karczewicz-II by choosing an intermediate value generated during the process described in Karczewicz-II and using that intermediate value as an input to the bi-prediction equations of Karczewicz-I, which also must be modified to accept the new higher-precision values as I explain later herein. Petition at 38-53, 59-68. Petitioner argues this modification is motivated by the statement in paragraph 99 of Karczewicz-II to “keep the highest possible precision through the

intermediate steps" of the interpolation process. Petition at 42.

85. Karczewicz-II, was considered by the Patent Office during the prosecution of the '267 Patent (Ex-1007, 74). In my opinion, Karczewicz-II contains substantially the same disclosure as other references the examiner reviewed multiple times in connection with the '267 Patent and its priority applications.

86. Ye '503, cited by the applicant during the prosecution of the '267 Patent and each of its five priority applications (see, e.g., Ex-1007, 88; Ex-1002, 74) includes the same interpolation equations and tables relied upon by the Petition as well as the same statements relied upon in the Petition to provide the motivation to combine. As I show below, Tables 3-8 in Ye '503 and Karczewicz-II match exactly:

Karczewicz-II (Ex-1006)	Ye '503 (Ex-2007)
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 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

Karczewicz-II (Ex-1006)	Ye '503 (Ex-2007)

TABLE 4

e = (b1 + h1 + 32) >> 6
g = (b1 + ee1 + 32) >> 6
m = (h1 + hh1 + 32) >> 6
o = (ee1 + hh1 + 32) >> 6

TABLE 4

e = (b1 + h1 + 32) >> 6
g = (b1 + ee1 + 32) >> 6
m = (h1 + hh1 + 32) >> 6
o = (ee1 + hh1 + 32) >> 6

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

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 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

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

Nokia Exhibit 2030, p.45
ASUS v. Nokia
IPR2025-01154

Karczewicz-II (Ex-1006)

TABLE 7

position {j} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Reg-ister size
r1 = y0	y0 is 1-D half-pixel aa1 before shifting down	-2550	10710	15s
r2 = y5	y5 is 1-D half-pixel jj1 before shifting down	-2550	10710	15s
r1 = r1 + r2	r1 is y0 + y5	-5100	21420	16s
r2 = y1	y1 is 1-D half-pixel bb1 before shifting down	-2550	10710	15s
r3 = y4	y4 is 1-D half-pixel ii1 before shifting down	-2550	10710	15s
r2 = r2 + r3	r2 is y1 + y4	-5100	21420	16s
r1 = r1 - r2	r1 is (y0 + y5) - (y1 + y4)	-26520	26520	16s
r1 = r1 >> 2	r1 is ((y0 + y5) - (y1 + y4)) >> 2	-6630	6630	14s
r3 = y2	y2 is 1-D half-pixel b1 before shifting down	-2550	10710	15s
r4 = y3	y3 is 1-D half-pixel hh1 before shifting down	-2550	10710	15s
r3 = r3 + r4	r3 is y2 + y3	-5100	21420	16s
r3 = max(-2372, r3)	clip r3 to avoid overflow (low side)	-2372	21420	16s
r3 = min(18640, r3)	clip r3 to avoid overflow (high side)	-2372	18640	16s
r2 = r3 - r4	r2 is (y2 + y3) - (y1 + y4)	-23792	23740	16s
r1 = r1 + r2	r1 is ((y0 + y5) - 5* (y1 + y4)) + 4* (y2 + y3)) >> 2	-30422	30370	16s
r1 = r1 >> 2	r1 is ((y0 + y5) - 5* (y1 + y4)) + 4* (y2 + y3)) >> 4	-7605	7592	14s
r1 = r1 + r3	r1 is ((y0 + y5) - 5* (y1 + y4)) + 20* (y2 + y3)) >> 4	-9977	26232	16s
j1 = r1	store result			
r1 = r1 + 32	r1 is ((y0 + y5) - 5* (y1 + y4)) + 20* (y2 + y3) + 512) >> 4	-9945	26264	16s
r1 = r1 >> 6	r1 is ((y0 + y5) - 5* (y1 + y4)) + 20* (y2 + y3) + 512) >> 10	-155	410	10s
r1 = max(0, r1)	clip r1 on the low side	0	410	9u
r1 = min(255, r1)	clip r1 on the high side	0	255	8u

Ye '503 (Ex-2007)

TABLE 7

position {j} of FIGS. 4A-4D

Operation	Comment	Min value	Max value	Register size
r1 = y0	y0 is 1-D half pixel aa1 before shifting down	-2550	10710	15s
r2 = y5	y5 is 1-D half pixel jj1 before shifting down	-2550	10710	15s
r1 = r1 + r2	r1 is y0 + y5	-5100	21420	16s
r2 = y1	y1 is 1-D half pixel bb1 before shifting down	-2550	10710	15s
r3 = y4	y4 is 1-D half pixel ii1 before shifting down	-2550	10710	15s
r2 = r2 + r3	r2 is y1 + y4	-5100	21420	16s
r1 = r1 - r2	r1 is (y0 + y5) - (y1 + y4)	-26520	26520	16s
r1 = r1 >> 2	r1 is ((y0 + y5) - (y1 + y4)) >> 2	-6630	6630	14s
r3 = y2	y2 is 1-D half pixel b1 before shifting down	-2550	10710	15s
r4 = y3	y3 is 1-D half pixel hh1 before shifting down	-2550	10710	15s
r3 = r3 + r4	r3 is y2 + y3	-5100	21420	16s
r3 = max (-2372,r3)	clip r3 to avoid overflow (low side)	-2372	21420	16s
r3 = min (18640,r3)	clip r3 to avoid overflow (high side)	-2372	18640	16s
r2 = r3 - r2	r2 is (y2 + y3) - (y1 + y4)	-23792	23740	16s
r1 = r1 + r2	r1 is ((y0 + y5) - 5* (y1 + y4) + 4* (y2 + y3)) >> 2	-30422	30370	16s
r1 = r1 >> 2	r1 is ((y0 + y5) - 5* (y1 + y4) + 4* (y2 + y3)) >> 4	-7605	7592	14s
r1 = r1 + r3	r1 is ((y0 + y5) - 5* (y1 + y4) + 20* (y2 + y3)) >> 4	-9977	26232	16s
j1 = r1	store result			
r1 = r1 + 32	r1 is ((y0 + y5) - 5* (y1 + y4) + 20* (y2 + y3) + 512) >> 4	-9945	26264	16s
r1 = r1 >> 6	r1 is ((y0 + y5) - 5* (y1 + y4) + 20* (y2 + y3) + 512) >> 10	-155	410	10s
r1 = max (0,r1)	clip r1 on the low side	0	410	9u
r1 = min (255,r1)	clip r1 on the high side	0	255	8u

Nokia Exhibit 2030, p.46
ASUS v. Nokia
IPR2025-01154

Karczewicz-II (Ex-1006)	Ye ’503 (Ex-2007)

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

TABLE 8-continued

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

Operation	Comment	Min value	Max value	Register size
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

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

TABLE 8-continued

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

Operation	Comment	Min value	Max value	Register size
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

87. The statements made in Karczewicz-II that Petitioner uses to support the motivation to modify and combine Karczewicz-I and Karczewicz-II are present in Ye ’503 verbatim. *Compare, e.g.,* Ex-1006 ¶ 99 with Ex-2007 ¶ 98.

88. VCEG-AI33, also by Yan Ye, Peisong Chen, and Marta Karczewicz and also considered by the examiner during prosecution (Ex-1007, 327; Ex-1002, 78), includes the same interpolation equations from Karczewicz-II and Ye ’503 tables 3 and 4. Ex-2010 at 1. VCEG-AI33 further includes step-by-step instructions for interpolation of the middle-subpixel (j) as well as adjacent quarter-pixel values showing the use of 16 bit registers, as also shown in tables 7 and 8 of Karczewicz-II

and Ye ’503. Ex-2010 at 3-4. VCEG-AI33 provides similar language to the motivation relied upon by Petitioner as motivation to modify and combine Karczewicz-I and Karczewicz-II, including the statement that “frequent and biased rounding effectively reduces precision of the interpolation filtering” and that, to resolve this issue, the authors propose “to postpone rounding and clipping until the last step not to compromise the final precision.” Ex-2010 at 1.

89. Accordingly, in addition to having actually been before the patent office for the prosecution of the challenged claims, the relied-upon disclosure from Karczewicz-II is included in at least two different references cited to the patent office during prosecution of the ’267 Patent and each of the ’267 Patent’s priority applications.

c. **The portions of the Karczewicz-I reference relied upon in the Petition have substantially the same disclosure of bi-prediction taught in the Ye ’262, Panchal, and Karczewicz 057 references cited in prosecution and on the face of the ’267 Patent.**

90. Petitioner relies on Karczewicz-I for its disclosure of bi-prediction and argues that a POSITA would have been motivated by Karczewicz-II’s teaching of avoiding rounding during interpolation to modify both Karczewicz-I and Karczewicz-II to used Karczewicz-II’s intermediate interpolation values as predictions for a modified version of Karczewicz-I’s bi-prediction equation. However, the same bi-prediction teaching relied upon in Karczewicz-I was before

the patent office in numerous other references during the prosecution of the ’267 Patent and its priority applications, including through Ye ’262, Panchal, and Karczewicz 057.

91. Ye ’262 teaches both default weighted bi-prediction and weighted bi-prediction in the same paragraph, and formed the basis for numerous rejections during the prosecution of the ’893 Application. Ex-2009 ¶ 25. See, e.g., Ex-1007, 146, 207. Panchal teaches the same default weighted, explicitly weighted, and implicitly weighted bi-prediction as Karczewicz-I. *Compare* Ex-2008 ¶¶ 60-62 with Ex-1005 ¶¶ 60-62. Karczewicz ’057 includes substantially the same overall disclosure as Karczewicz-I related to various forms and variations of bi-prediction. Compare Ex-2011 at 12:4-16:16 with Ex-1005 ¶¶ 57-76.

92. Accordingly, the relied-upon disclosure from Karczewicz-I is included in at least three different references cited to the patent office during prosecution of the ’267 Patent, including two references used as the basis for a rejection for one of the ’267 Patent’s priority applications.

IX. THE GROUNDS IN THE PETITION DO NOT ANTICIPATE OR RENDER OBVIOUS THE CHALLENGED CLAIMS

1. Ground 1: Walker

d. Walker does not teach obtaining and combining high-precision predictions to obtain a combined prediction, as required by the claims

93. The petition's first ground, based on Walker, relies on the interpretation that a weight, applied to a prediction, creates a higher precision prediction. Petition at 19-23. I disagree with this interpretation because it is inconsistent with the specification and prosecution history of the '267 Patent, the understanding of a person of skill in the art and the express teachings of Walker. Because the weights (w0 and w1) multiplied by the predictions (pred0 and pred1) are not themselves predictions with higher precision than the reference blocks used to obtain pred0 and pred1, Walker does not render at least elements 1[c], 1[d], 7[c], 7[d], 13[c], and 13[d] (626 Petition) and elements 19[c], 19[d], 25[c], 25[d], 31[c], and 31[d] (627 Petition) obvious. Each of these elements requires the first/second prediction have the second, higher precision.

94. As a POSITA would understand and as the '267 patent makes clear, the claimed first and second predictions are each [a] value obtained from a first or second reference frame and a first or second motion vector ('267 patent claim 1).

95. The '267 Patent states that, in motion compensated prediction, the

prediction is "the best match" from a reference frame, which can be "coded into a bit stream as [a] motion vector" at the encoder so the decoder can "reconstruct the exact prediction frame decoding the motion vector data encoded in the bitstream." Ex-1001, 2:26-32. The "source" of a prediction is a block of pixels in a previously coded image (optionally including interpolated sub-pixel positions), and a motion vector identifies that source and makes it possible for both encoder and decoder to construct the same prediction. Ex-1001, 3:14-18, 11:12-14.

96. Columns 12-14 of the '267 Patent provide an example of how the first and second predictions are calculated and combined in the preferred embodiment. The first and second prediction values are obtained using the block processor applied to pixel values from the first and second reference blocks based on the first and second motion vectors. Ex-1001, 12:44-55. If a motion vector indicates a fraction of a pixel, or sub-pixel, position in the reference frame, then an interpolation process is used to obtain the prediction. Ex-1001, 12:56-13:18. Example applications of a 6-tap filter applied to underlying pixel values are illustrated at Ex-1001, 14:19-22.

$$P1=(3*E_1-17*F_1+78*G_1+78*H_1-17*I_1+3*J_1+1)>>1$$

$$P2=(3*E_2-17*F_2+78*G_2+78*H_2-17*I_2+3*J_2+1)>>1$$

97. Here, P1 and P2 are the predictions because they are the values that

result from decoding the motion vector data encoded in the bitstream, applying the motion vector to a reference frame and interpolating pixel values of the reference frame to obtain a prediction value. An arbitrary intermediate value calculated within that process (e.g., $17*F_1$) does not constitute a prediction, even though such a value is calculated during the process of obtaining a prediction such as P1 or P2. Rather, an intermediate value such as for example $17*F_1$ is a component of the prediction P1, in a similar way that the price of an individual packet of food is a component of a final supermarket bill (grocery bill).

98. Applying weights to P1 and P2 does not change the precision of P1 or P2, and the resulting intermediate values (e.g., w1*P1 and w2*P2) are not themselves predictions. Instead, w1*P1 or w2*P2 are scaled contributions of P1 and P2 to the combined prediction, and are intermediate values calculated during the process of creating a combined prediction from predictions P1 and P2.

99. Petitioner proposes that applying a weight to a prediction pred0, such as w0*pred0, results in a value that is itself a prediction. Petition at 19-23. I disagree. A value such as w0*pred0 is at most an intermediate value, in a similar way that $17*F_1$ is an intermediate value in the discussion above. In the context of Walker equation 13, w0*pred0 is an intermediate value that contributes to the calculation of an actual prediction, namely Final_pred.

100. The ’267 Patent never addresses weighted bi-prediction, despite it being a part of multiple prior standards, including the H.264 standard referenced in the background of the specification. Instead, the ’267 patent refers only to obtaining a combined prediction from two component predictions. A POSITA would understand from that that any application of weights during the combination of two predictions is part of the combination process (e.g., the “+” in Fig. 11) and not a separate set of predictions.

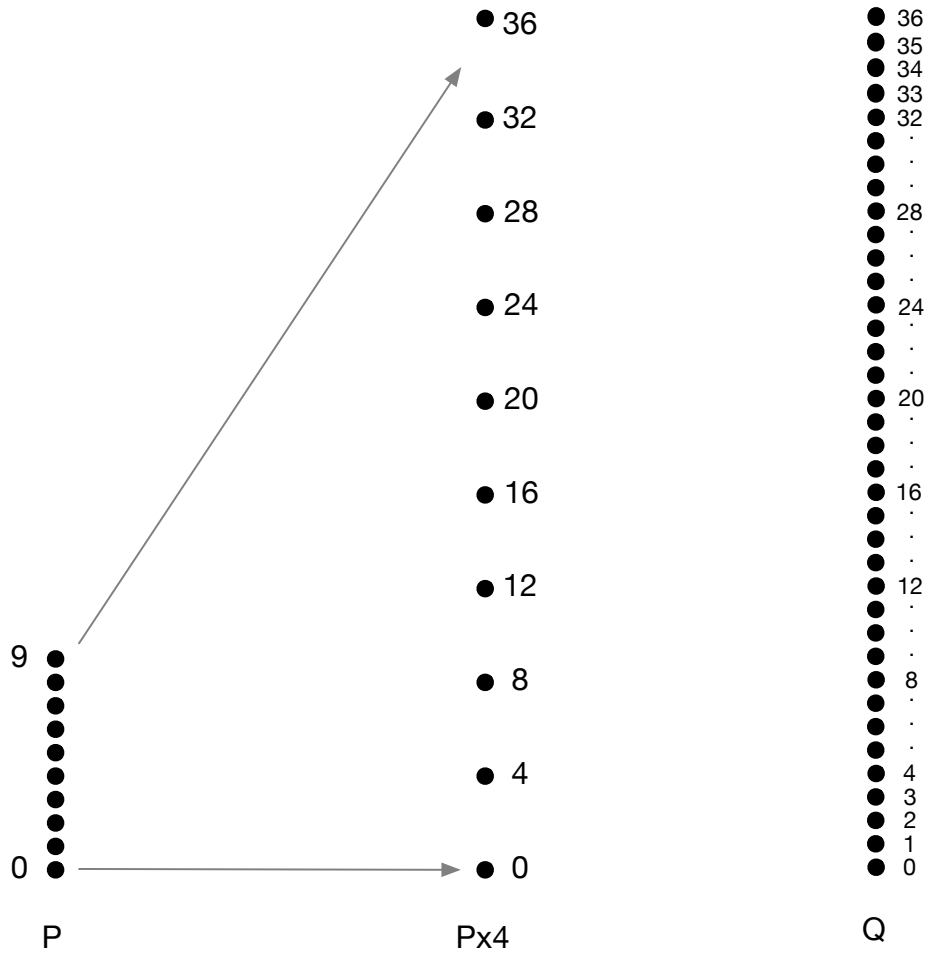
101. This is consistent with the prosecution history for the ’267 Patent’s family. As I discussed above, weighted prediction references were before the patent office during the prosecution of the ’267 Patent and its family members, and the weighted bi-prediction reference Ye ’262 formed the basis for several rejections of the ’267 Patent’s original priority application. However, even when considering a weighted bi-prediction reference such as Ye ‘262, the examiner recognized that weighted bi-prediction did not teach a first and second prediction with a second precision greater than the first precision (i.e., the precision of full pixel values). Ex-1007 at 252. The applicant, in response, made clear that applying a weight to a prediction did not increase the precision of the prediction, it instead mapped that prediction to a different range of values represented by the product of the weight and the prediction. Ex-1007 at 281. This operates in an analogous way to drawing a picture on a balloon and then inflating it. Inflating the balloon makes the original

image larger, but does not add precision to the original drawing.

102. Moreover, applying a weight to a prediction does not increase the precision of that prediction.

103. As an example, consider a variable P that can take an integer value in the range 0 to 9. Multiplying P by a weight of 4, i.e. (P×4), changes the range of values (from 0-9 to 0-36) but does not increase the precision of P. The weighted version P×4 can still only take one of 10 values: 0, 4, 8, 12, 16, 20, 24, 28, 32 or 36. A POSITA would understand that (P×4) has the same **precision** as P, even though its **range** is larger. In contrast, consider a variable Q that can take **any** integer value in the range 0 to 36. Unlike (P×4), Q has a **higher precision** than P, because it can take a wider range of values.

104. The relationship between P, (P×4) and Q is illustrated in the following Figure. The black circles indicate discrete possible values of each variable. Both P and (P×4) have exactly 10 discrete possible values, whereas Q has a higher precision and has 37 discrete possible values.



105. At the heart of Petitioner's argument with respect to Walker is this exact point. Petitioner points to Walker's (pred0)w0 (i.e. pred0 × w0) and suggests that it has a higher **precision** than pred0, which has an 8-bit precision and can therefore take one of 256 values, e.g. between 0 and 255. However, (pred0)w0 has the same **precision** as pred0, i.e. it can still only take one of 256 values, between 0 and (255 × w0). In the case of Walker, values of (pred0)w0 may have a larger **range** than pred0, necessitating a temporary register with a wider bit depth (such as 16

bits), but as I explain herein, (pred0)w0 does not have a higher precision than pred0.

106. In my opinion, Petitioner's interpretation of "prediction" is not consistent with the understanding of a POSITA. The petition points to no contemporary disclosure showing an understanding in the field that a prediction block multiplied by a weight as part of a bi-prediction process is itself a prediction. For example, in H.264, the results of the sub-pixel interpolation processes are called "prediction [luma/chroma] sample values" and are stored in arrays called "PredPart." Ex-1014 at 188 (§ 8.4.2.2). The result of applying weights to these prediction samples values, in contrast, is not referred to as a prediction sample value. Ex-1014 at 195 (Equation (8-278)). The petition merely takes the (pred0)w0 and (pred1)w1 values of Walker to be the claimed "predictions" because those values are stored using more than 8 bits. However, this is an interpretation driven by hindsight rather than based on any interpretation of was understood in the art, or of what Walker itself teaches.

107. Further, a POSITA would understand Walker itself to refer to pred0 and pred1 as predictions but not to refer to w0*pred0 or w1*pred1 as predictions.

108. Walker teaches three kinds of prediction: forward or uni-directional prediction, where predictions are made for a current picture from a single reference picture, bi-predictions, where predictions are made for a current picture from a

combination of predictions from two different reference pictures, and weighted bi-prediction, where predictions are made for a current picture from a weighted combination of predictions from two reference pictures.

109. For the first form of prediction, Walker describes finding "a best matching prediction region in a reference picture" and calculating "the difference between the current region and the best matching reference prediction region" to obtain a residual, or prediction error. Ex-1004 ¶ 30. A POSITA would understand this to show the prediction being obtained from the reference picture, being defined and communicated by the motion vector, and being used to determine the prediction error or residual. This is consistent with how the term prediction is used in the art in connection with motion compensated prediction. Walker Figure 3 below is annotated accordingly.

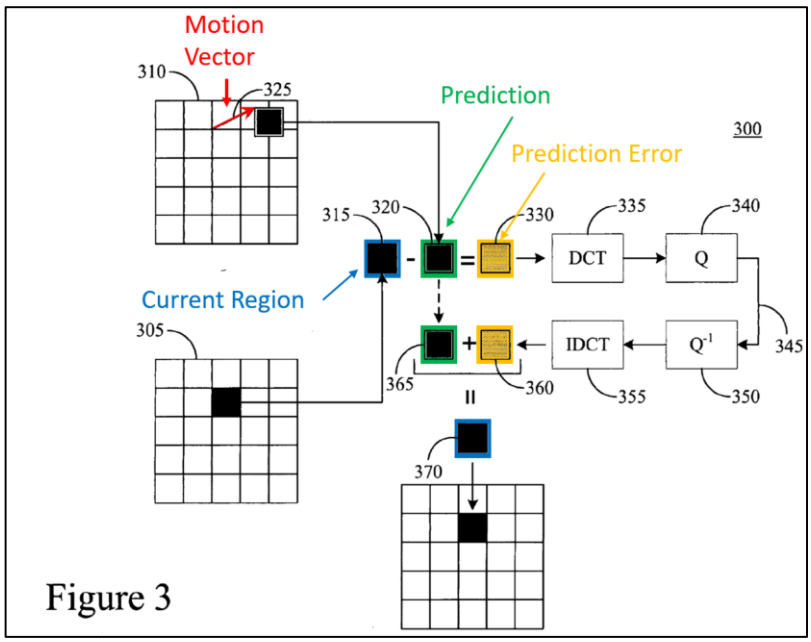
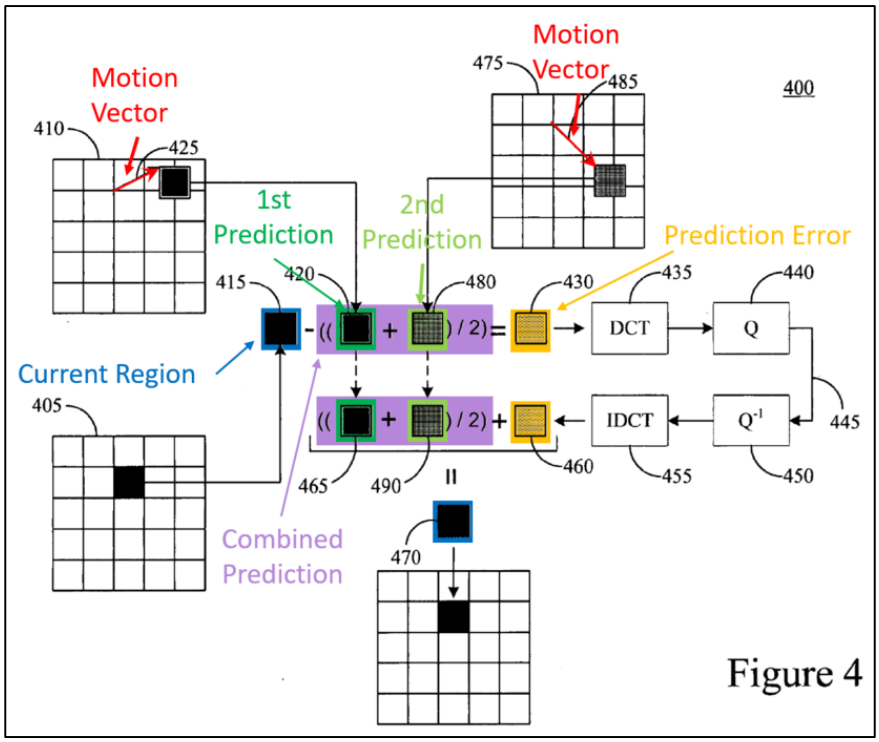


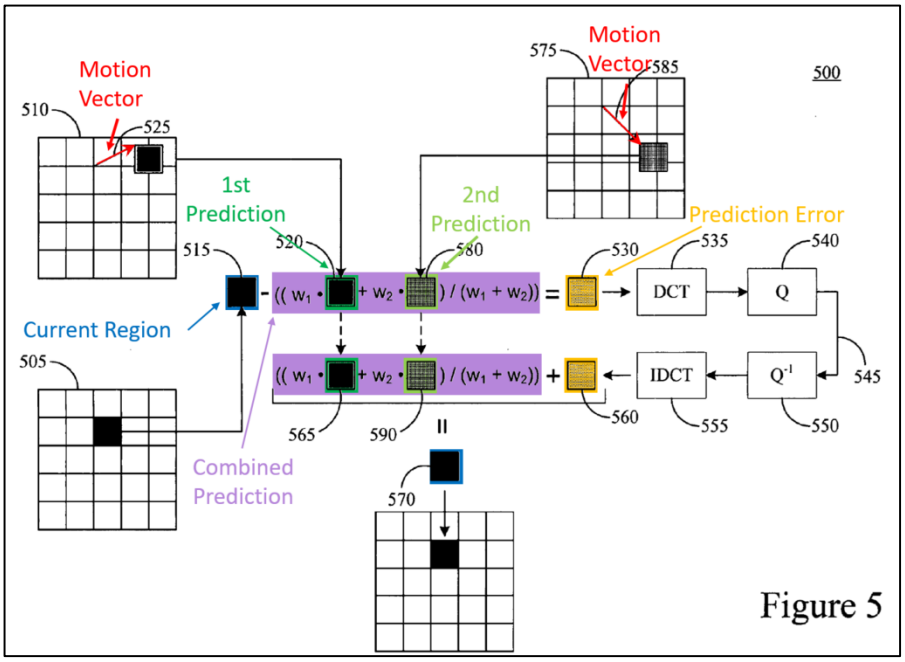
Figure 3

110. Walker then describes the use of bi-prediction (without weights) in connection with Figure 4. Both "a best matching prediction region in a previous picture" and "a best matching prediction region in a subsequent picture" are identified, each with a corresponding motion vector, and those two prediction regions are combined to form a "combined bi-prediction region." Ex-1004 ¶ 32. A POSITA would understand this description, consistent with Walker's description of forward prediction and Figure 3, to show predictions that are obtained from reference frames and that are defined and communicated based on their motion information. The annotated Walker Figure 4 below illustrates the operation of this bi-prediction mode.



111. Walker provides a similar description of weighted bi-prediction.

Suitable matching regions are found in both a previous and a subsequent reference picture and are combined, this time using weighting factors, to reduce the residual error signal of the combined prediction. Ex-1004 ¶ 34. Weighted bi-prediction in Walker uses two predictions, one from each of two reference frames, each identified by a motion vector, and then combines them to make a combined prediction. Walker Figure 5, annotated below, shows how predictions are combined in Walker when weighted bi-prediction is used.



112. Walker's description of prediction and weighted bi-prediction confirms that the predictions that are combined in bi-prediction are the prediction blocks identified and obtained by the motion vectors, not those same prediction blocks after they have had weighting factors applied. A simple description of the bi-prediction process is at Walker ¶¶ 48-49, where Walker describes the bi-prediction

equation $pred_{ij}=(pred1_{ij} + pred2_{ij} + 1) >> 1$ as "used to combine predictions from two reference frames," those predictions being $pred1_{ij}$ and $pred2_{ij}$, which are the "8-bit luminance and chrominance samples from prediction blocks from two reference frames." Ex-1004 ¶¶ 48-49; Walker describes unweighted prediction from H.264 in a similar way. Ex-1004 ¶¶ 59-60.

113. When describing weighted prediction, Walker describes the weighting factors w0 and w1 as being "applied to the luma or chroma value" for the corresponding reference picture, rather than as a factor used to obtain a prediction.

TABLE 1

Variables	Description	Range	No. of Bits
logWD	The base 2 logarithm of the denominator for all the luma or all the chroma weighting factors (luma and chroma weighting factors can be derived separately)	0 to 7	3
w0	The weighting factor applied to the luma or chroma prediction value for the first reference picture (list "0")	−128 to 127	8
w1	The weighting factor applied to the luma or chroma prediction value for the second reference picture (list "1")	−128 to 127	8

Ex-1004 ¶ 69, Table 1.

114. Walker's detailed step-by-step disclosure of its bi-prediction methods show that both the input predictions (pred0 or pred1) and the final values of each bi-prediction process have the same precision, namely 8 bits. This is true of Table 2, 3, and 4, each reproduced below with annotations. Ex-1004 ¶¶ 84, 89, 93; Tables 2-4.

TABLE 2

Operation No.	Operation	Bitwidths Involved	Bitwidth of Operation Result
1	(pred0)w0, (pred1)w1	8 bits * 8 bits	16
2	2^{LWD-1}	logWD has a maximum value of 7, therefore LWD − 1 has a maximum value of 7 for case 3	7
3	(pred0)w0 + (pred1)w1 + 2^{LWD-1}	16 bits + 16 bits + 7 bits	18
4	(((pred0)w0 + (pred1)w1 + 2^{logWD}) >> (LWD + 1))	(18 bits) >> (8 bits)* *maximum value of LWD + 1	10
5	$(o_0 + o_1 + 1)$ >> (1)	(8 bits + 8 bits + 1) >> (1)	8
6	Clip1[Op. 4 + Op. 5] (for Clip1, see Eq. (11))	Clip1[10 bits + 8 bits]	8

TABLE 3

Operation No.	Operation	Bitwidths Involved	Bitwidth of Operation Result
1	(pred0)w0', (pred1)w1'	8 bits * 9 bits	17
2	(pred0)w0' + (pred1)w1' + 2^7	17 bits + 17 bits + 7 bits	19
3	(((pred0)w0' + (pred1)w1' + 2^7) >> (8))	(19 bits) >> (8 bits)	11
4	$(o_0 + o_1 + 1)$ >> (1)	(8 bits + 8 bits + 1) >> (1)	8
5	Clip1[Op. 3 + Op. 4] (for Clip1, see Eq. (11))	Clip1[11 bits + 8 bits]	8

TABLE 4

Operation No.	Operation	Bitwidths Involved	Bitwidth of Operation Result
1	(pred0) w_A, (pred1) w_B	8 bits * 15 bits	23
2	((pred0) w_A) >> 6 + ((pred1)w_B) >> 6 + 2^7	17 bits + 17 bits + 7 bits	19
3	(((pred0) w_A) >> 6 + ((pred1)w_B) >> 6 + 2^7) >> (8)	(19 bits) >> (8 bits)	11
4	$(o_0 + o_1 + 1)$ >> (1)	(8 bits + 8 bits + 1) >> (1)	8
5	Clip1[Op. 3 + Op. 4] (for Clip1, see Eq. (11))	Clip1[11 bits + 8 bits]	8

115. The first and second predictions of Walker, pred0 and pred1, can each take one of only 256 possible values. These predictions each have a range from 0-255 and a precision of 8 bits. Each weighted prediction value identified by Petitioner, such as (pred0)w0, has a range that has been increased due to multiplication by a weight w0, but can still only take one of 256 possible values, (0×w0 to 255×w0). Therefore, as I explain above, the precision of the weighted value such as (pred0)w0 has not been increased.

116. Accordingly, Walker fails to teach at least elements 1[c], 1[d], 7[c], 7[d], 13[c], and 13[d] (626 Petition) and elements 19[c], 19[d], 25[c], 25[d], 31[c], and 31[d] (627 Petition) and the Petition has failed to show a reasonable likelihood that the challenged claims are invalid on this ground.

2. Ground 2: Karczewicz-I Combined with Karczewicz-II

a. A POSITA would not be motivated to combine Karczewicz-I with Karczewicz-II as set out in the Petition

117. The petition seeks not only to combine Karczewicz-I and Karczewicz-II, but to modify them both in the process. The petition fails to show that a POSITA would be motivated to make this modified combination and fails to show that a POSITA would have had a reasonable expectation of success in making it.

118. First, the Petition asserts that Karczewicz-II "teaches improved calculations for averaging interpolated pixel values, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate calculations" and that those "complimentary teachings" would "improve Karczewicz-I by reducing rounding inaccuracies." Petition at 39. However, Karczewicz-I already includes the higher-precision interpolation equations the petition relies upon due to its incorporation by reference of VCEG-AI33. Ex-1005 at ¶ 37, explaining that VCEG-AI33 is "hereby incorporated by reference in its entirety". As noted above, VCEG-AI33 includes the same teachings Petitioner relies

upon in Karczewicz-II, including the same equations for interpolation of quarter-pixel values using intermediate values for half-pixel positions. A POSITA would therefore understand Karczewicz-I to already include the improved interpolation techniques of Karczewicz-II, and would not look to Karczewicz-II for any further combination or modification.

119. Second, Karczewicz-I teaches that its input must be at the same precision as the current block of pixels and Karczewicz-II teaches that its output must be at the same precision as the current block of pixels. As a result, even if a POSITA were motivate to combine Karczewicz-I and Karczewicz-II, they would not combine Karczewicz-I and Karczewicz-II in the manner asserted by the petition, namely storing and using selected higher bit-depth intermediate values of Karczewicz-II as predictions, and using such selected higher bit-depth intermediate values as inputs for bi-prediction in Karczewicz-I.

120. Karczewicz-I refers to the result of its bi-predictions as "prediction data." Ex-1005 at ¶¶ 64, 69. Karczewicz-I further teaches that, once suitable prediction data is found, it is subtracted from the original block being coded (the current block) to create a residual block. Ex-1005 at ¶ 73. In a corresponding step at the decoder, the received residual block is added to the prediction data to reproduce the original data. Ex-1005 at ¶ 84. At ¶ 73 and ¶ 84, Karczewicz-I discloses what a POSITA would understand to be well-known arithmetic operations of subtraction

and addition respectively. Absent any special logic, which Karczewicz-I does not disclose, the current block pixels and the prediction data must have the same precision at the point at which they are added or subtracted. The unweighted bi-prediction equation relied upon by the Petition from Karczewicz-I has inputs of predictions pred0(i,j) and pred1(i,j), adds them and right-shifts by 1, then outputs prediction data pred(i,j). Ex-1005 ¶ 60. A POSITA would understand that according to this equation, the predictions pred0(i,j) and pred1(i,j) and the combined prediction pred(i,j) each have the same precision. As combined prediction pred(i,j) necessarily has the same precision as the current block pixel data (as I explain above) and as pred0(i,j), pred1(i,j) and the combined prediction pred(i,j) necessarily have the same precision as each other, predictions pred0(i,j) and pred1(i,j) in Karczewicz-I therefore have the same precision as the pixels of the current block.

121. Similarly, each of the interpolated values for sub-pixel positions in Karczewicz-II have the same precision as the original input pixels. The 6-tap filter used in Karczewicz-II has weights [1, -5, 20, 20, -5, 1] which add to 32, or 2^5. Ex-1006 ¶ 93. Thus intermediate values such as b1 and h1 that result from a single application of the 6-tap filter (without a right-shift) gain five bits of bit depth, and intermediate values such as j1 that result from two applications of the 6-tap filter (without a right-shift) gain 10 bits of bit depth. Thus for input pixels with 8-bit precision, b1 or h1 have 13 bits of precision and j1 has 18 bits of precision The

equations for sub-pixels a, c, d, l, f, i, k, n, e, g, m, and o in Tables 3 and 4 of Karczewicz-II each specify the addition of two input values with bit depths 5 greater than the original pixel bit depth (such as for example b1 + h1, sub-pixel e), followed by a right-shift of 6 binary places, which would return a result with the original pixel precision. Ex-1006 ¶¶ 99-100. The specific examples provided by Karczewicz-II, demonstrating how to implement these interpolation equations using 16-bit arithmetic, all specify 8-bit precision for input pixels such as x and for prediction outputs such as a, c, d, l and so on. Ex-1006 ¶¶ 103-105. Thus Karczewicz-II teaches using 8-bit input pixel values to generate 8-bit predictions whilst using higher precision for certain intermediate calculation values.

122. In summary, Karczewicz-II teaches generating as its output 8-bit prediction values corresponding to fractional motion vector positions by interpolation of 8-bit pixel values, and Karczewicz-I teaches combining such 8-bit prediction values (as inputs) to generate an 8-bit combined prediction value. Were a POSITA to combine Karczewicz-II and Karczewicz-I (without modifying both references) the result would therefore be a combined process that **at no point** required or generated a prediction value with a precision higher than that of the input pixel values (8 bit precision in the specific examples of the two references).

123. In its Ground 2 proposal, Petitioner suggests modifying both references in a manner not disclosed in either reference, namely storing and using

selected higher bit-depth intermediate values of Karczewicz-II as predictions, and using such selected higher bit-depth intermediate values as inputs for bi-prediction in Karczewicz-I.

124. As I explain above, Karczewicz-I discloses an equation for calculating a bi-prediction pred(i,j) from two input predictions pred0(i,j) and pred1(i,j), with all three predictions having a precision of 8 bits (Petition at p42, K-I at 60). Petitioner proposes replacing pred0(i,j) in this process with 'non-rounded (pred0(i,j))', a version of pred0(i,j) that was calculated and stored at a higher precision such as 15 bits (Petition at 47). Petitioner suggests that the resulting combined equation for pred(i,j) (Petition at 47) results from a combination of Karczewicz-II and Karczewicz-I. However, neither Petitioner's newly-created prediction ('non-rounded (pred0(i,j))') nor the combined equation (Petition at p47) are actually disclosed in Karczewicz-II. What is disclosed in Karczewicz-II is an intermediate value such as b1, calculated during averaging of a half-pixel and an integer-pixel value, which Petitioner arbitrarily chooses as an output of Karczewicz-II and an input of the bi-prediction equation of Karczewicz-I (Petition at p46-47).

125. To provide a basis for this modification, Petitioner points to paragraph 99 of Karczewicz-II in support of Petitioner's proposition that Karczewicz-II "teaches improved calculations for averaging interpolated pixel values, where rounding is delayed until later in the process, thereby maintaining higher precision

for intermediate calculations." Petition at 39. However, the process addressed by Paragraph 99 of Karczewicz-II is the process of interpolating final 8-bit values to be used as predictions in motion compensated prediction. Thus the process addressed by Karczewicz-II at Paragraph 99 does **not** disclose or suggest calculating or storing actual prediction values at a higher precision or bit-depth. Instead, such prediction values (the "very last step of the interpolation process" referred to in Karczewicz-II at Paragraph 99) are indeed subject to "shifting, rounding and clipping" in order to revert to the 8-bit precision Karczewicz-II requires for prediction values.

126. Petitioner never addresses, much less provides a motivation to incur, the significant computational, memory and memory bandwidth cost of calculating and storing predictions at a higher precision, necessarily required by its Karczewicz-I/Karczewicz-II combination. The advance of Karczewicz-II was the use of a limited number of 16-bit registers *during* the calculation of the 8-bit final prediction values for each sub-pixel position within a prediction block, using higher-precision intermediate values. Ex-1006 ¶¶ 53, 59. In my opinion, a POSITA would not be motivated to change the whole focus of Karczewicz-II in order to reduce rounding in bi-prediction, especially considering that Karczewicz-I already provides a solution in the form of rounding offsets. Ex-1005 ¶¶ 64-69. In other words, there is no need to modify Karczewicz-I to address the issue of rounding in bi-prediction, and significant potential technical disadvantages in terms of increased computation,

memory and memory bandwidth, and so a POSITA would not be motivated to make the change contemplated by the Petitioner.

127. Further to my discussion above, a POSITA would understand that the modifications proposed by Petitioner would have considerable technical costs and disadvantages in terms of computation, memory and memory bandwidth, and so would not be motivated to make them, without some external justification. For example, as I explained earlier, a video encoder typically generates and stores calculated prediction values for each of the 15 sub-pixel positions that surround each integer-pixel value (K-II, Fig. 4D), in order to search for a best matching region and thus an optimal motion vector, during the process of motion estimation. In the methods of Karczewicz-I and Karczewicz-II, such prediction values are generated and stored at the same precision as the integer pixel inputs, i.e. 8 bits per prediction value. A POSITA would understand that the proposed modifications would require the encoder to generate and store the prediction values for the 15 sub-pixel positions at a higher precision, requiring a significant increase in the storage and memory bandwidth demands for motion estimation and motion compensated prediction. Were Petitioner's proposed 16-bit precision to be adopted, calculations such as subtracting the prediction from the residual or combining predictions would require higher-precision arithmetic, thus incurring a computational cost. Further, these higher-bit depth prediction values would require at least twice the amount of

memory storage required for the methods of Karczewicz-I or Karczewicz-II. Further, memory bandwidth (the rate at which data is written to or read from memory) would need to be at least doubled to accommodate the additional precision of the stored intermediate values. This would result either a significant increase in memory resources and memory bandwidth on the video encoder device, resulting in a higher implementation cost, or a significant reduction in encoding speed due to the increase in storage and memory bandwidth requirements, or perhaps both. A POSITA would not be motivated to incur such processing and memory penalties in order to avoid rounding errors, absent a strong external motivation which the Petition does not provide.

128. Additionally, a POSITA would not be motivated to make the above contemplated changes to a decoder unless he or she was first motivated to make the corresponding changes at the encoder. As I explained above, an encoder's predictions are used to generate the residual data at the encoder that the decoder receives, together with information such as motion vectors indicating how the decoder should make its own corresponding prediction. The decoder then generates the prediction block and adds it to the residual information to reconstruct the block. If the encoder and decoder use different processes to generate their predictions, such as a higher-precision "non-rounded prediction" at the decoder but a lower-precision prediction at the encoder, then the values added to the residual data by the decoder

will not match the values subtracted at the encoder. This will introduce a distortion not only in the current picture, but also in subsequent pictures that refer back to the current picture, resulting in both an unacceptable reduction in decoded image quality and an unacceptable loss of compression performance due to the mismatched predictions. A POSITA would therefore not be motivated to increase or change the precision of predictions only at the decoder, due to the necessity for both encoder and decoder to create identical predictions.

b. **Without the modifications proposed in the petition, neither Karczewicz-I nor Karczewicz-II teach higher-precision predictions**

129. As I explain above, Karczewicz-II teaches generating 8-bit values for fractional motion vectors by interpolation of 8-bit pixel values and Karczewicz-I teaches combining those 8-bit values to generate an 8-bit combined prediction. Absent the modifications to both Karczewicz-I and Karczewicz-II contemplated by the petition (but not supported by the cited references or any extrinsic evidence that I am aware of), the predictions combined in Karczewicz-I have the same 8-bit precision as the pixels in the current block and reference blocks. The combination therefore fails to render obvious at least elements 1[c], 1[d], 7[c], 7[d], 13[c], and 13[d] (626 Petition) and elements 19[c], 19[d], 25[c], 25[d], 31[c], and 31[d] (627 Petition) and the Petition has failed to show a reasonable likelihood that the challenged claims are invalid on this ground.

X. CONCLUSION

130. This Declaration represents the opinions that I have formed to date based on the material currently available to me. I reserve the right to revise, supplement, and/or amend my opinions stated in this Declaration based on new information that becomes available to me and on my continuing analysis of the materials available to me.

131. 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, imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: June 27, 2024

Iain E. Richardson

Dr. Iain Richardson

Nokia Exhibit 2030, p.71
ASUS v. Nokia
IPR2025-01154