IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Kemal Ugur et al.	
U.S. Patent No.:	11,805,267	Attorney Docket No.: 54587-0013IP1
Issue Date:	October 31, 2023	
Appl. Serial No.:	17/328,750	
Filing Date:	May 24, 2021	
Title:	MOTION PREDICTION IN VIDEO CODING	

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

DECLARATION OF JOSEPH HAVLICEK IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 11,805,267

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

Date: 17 June 2025 By: ______________________

Joseph Havlicek

TABLE OF CONTENTS

I, Joseph Havlicek, do hereby declare that:

I. INTRODUCTION

1. My name is Joseph Havlicek, and I have been retained by counsel for ASUSTeK Computer Inc. and ASUS Computer International (collectively "ASUS" or "Petitioner") as an expert witness to assist in analyzing issues related to the patentability of certain claims of U.S. Patent No. 11,805,267 ("the '267 patent"). I understand that ASUS intends to submit this declaration in support of a petition for *inter partes* review ("IPR") of the '267 patent before the Patent Trial and Appeal Board ("PTAB") of the United States Patent and Trademark Office ("USPTO").

2. I am being compensated for my work in this matter at my standard hourly rate. My compensation in no way depends on the outcome of this proceeding or the content of my testimony.

II. QUALIFICATIONS

3. I received a Bachelor of Science degree in electrical engineering with minors in mathematics and computer science from Virginia Tech in 1986. I also received a Master of Science Degree in electrical engineering, also from Virginia Tech, in 1988. I received the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1996. My Ph.D. research was in the field of image processing.

4. From December 1984 to May 1987, I was a software engineer at

Management Systems Laboratories in Blacksburg, VA. My job responsibilities included developing software for nuclear materials management under contract with the United States Department of Energy.

5. From June 1987 to January 1997, I was an electrical engineer at the United States Naval Research Laboratory. For the period of June 1987 through August 1989, I was an on-site contractor affiliated with SFA, Inc., Landover, Maryland. From August 1989 through January 1997, I was a federal government employee. I was on leave without pay from August 1987 through July 1988 while completing my Master of Science degree. I was also on leave without pay for much of the period from August 1990 through January 1997 while I completed my Ph.D. degree. My main job responsibilities at the United States Naval Research Laboratory included designing digital and analog circuits to process real-time video signals and designing and implementing target detection, tracking, and identification algorithms for real-time video signals. I was a recipient of the 1990 Department of the Navy Award of Merit for Group Achievement for this work.

6. From January 1993 through December 1993, I was an on-site contractor at International Business Machines (IBM) Corporation, Austin, TX. My main job responsibilities included designing and implementing image compression and decompression algorithms (CODECs) for IBM products.

7. Since January 1997, I have been a regular faculty member in the

School of Electrical and Computer Engineering at the University of Oklahoma, Norman, OK. I was an Assistant Professor from January 1997 through June 2002. I was promoted to the rank of Associate Professor and granted tenure in July 2002. I was promoted to the rank of Professor in July 2007. I was appointed to the Williams Companies Foundation Presidential Professorship in April 2009. In April 2017, I was appointed to the Gerald Tuma Presidential Professorship.

8. My main job responsibilities at the University of Oklahoma include conducting academic research in electrical and computer engineering, teaching graduate and undergraduate courses in electrical and computer engineering, and performing professional and institutional service.

9. I am a member of several professional societies and organizations, including the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Signal Processing Society, the IEEE Computer Society, and the IEEE Intelligent Transportation Society. I am a Senior Member of the IEEE. From November 2015 through February 2018, I served as a Senior Area Editor for the IEEE Transactions on Image Processing. I was formerly an Associate Editor for the IEEE Transactions on Image Processing from December 2010 through October 2015. I have served as a Technical Area Chair for the IEEE International Conference on Image Processing in the area of Image & Video Analysis, Synthesis, and Retrieval (2012, 2013) and have served on the organizing committee of that conference

(2007). I have also served as a Technical Area Chair for the IEEE International Conference on Acoustics, Speech, and Signal Processing in the area of Image, Video, and Multidimensional Signal Processing (2012-2014).

10. For over 30 years, I have conducted research and taught classes in the field of image and video processing and analysis. My main scholarly contributions have been in the areas of modulation domain image models and image processing (AM-FM image models), video target tracking, and distributed control of video networks for intelligent transportation systems.

11. I have served as a supervisor or committee member for numerous Ph.D. dissertations and Master's theses. I have supervised 12 Ph.D. students to completion and am currently supervising three Ph.D. students. I have been a member of 68 additional doctoral dissertation committees. I have supervised 28 Master's students to completion. I am currently supervising one additional Master's students. I have been a member of 71 additional Master's thesis committees. A listing of my Ph.D. and Master's supervisions and committee memberships is found in my curriculum vitae in Appendix A.

12. I am co-founder and director of the University of Oklahoma Center for Intelligent Transportation Systems (CITS). Under my supervision, the Center has collaborated with the Oklahoma Department of Transportation since 1998 to design and implement the Oklahoma Statewide Intelligent Transportation System,

including a geographically distributed video network that is currently deployed on major highways and interstates across the entire State of Oklahoma.

13. I teach a variety of courses at the University of Oklahoma, including the required junior-level Signals and Systems course ECE 3793 (taught 21 times), the graduate level Digital Image Processing course ECE 5273 (taught 26 times), and the graduate level Digital Signal Processing course ECE 5213 (taught 18 times).

14. Since joining the University of Oklahoma in January 1997, I have been Principal Investigator or Co-Principal Investigator on over 110 externally funded grants and contracts with a total value of over $27M. My main research contributions have been in the areas of signal, image, and video processing, video target tracking, and intelligent transportation systems. I have been author or coauthor on over 130 scholarly publications in these areas. I was a recipient of the 1990 Department of the Navy Award of Merit for Group Achievement for my work in video target tracking. My research group at the University of Oklahoma originated the Virtual Traffic Management Center concept featured in a December 2014 FHWA technical report (Guidelines for Virtual Transportation Management Center Development) and a November 2014 FHWA national webinar with the same title. I have received a number of teaching awards, including the University of Oklahoma College of Engineering Outstanding Faculty Advisor Award (2005-

2006) and the University of Texas Engineering Foundation Award for Exemplary Engineering Teaching while Pursuing a Graduate Degree (1992).

15. Since joining the faculty of the University of Oklahoma in 1997, I have taught numerous classes at both the graduate and undergraduate levels. At the graduate level, I have taught the following courses: Digital Signal Processing (ECE 5213), Digital Image Processing (ECE 5273 and CS 5273), Multimedia Communications (ECE 5973), Kalman Filtering (ECE 6973), and Advanced Image Processing (ECE 6283). At the undergraduate level, I have taught the following courses: Digital Signals and Filtering (ECE 2713), Microcomputer System Design (ECE 3223), Signals and Systems (ECE 3793), Digital Signal Processing (ECE 4213), Digital Image Processing (ECE 4973), and Multimedia Communications (ECE 4793).

III. LEGAL STANDARDS

16. I have been asked to provide my opinions as to whether claims 1-36 of the ’267 patent would have been obvious to a person of ordinary skill in the art as of the earliest claimed priority date of the ’267 patent (January 7, 2011) (“Critical Date”).

17. I am an engineer by training and profession. The opinions I express in this declaration involve the application of my technical knowledge and experience to the evaluation of certain prior art with respect to the ’267 patent. In addition, I

understand that the following legal principles apply.

18. It is my understanding that, in determining whether claims of the ’267 patent are obvious in this proceeding, the claim terms are generally given their ordinary and customary meaning as understood by a person of ordinary skill in the relevant art. A person of ordinary skill in the art would read the claim terms in the context of the entire patent specification in which they appear, as well as the prosecution history of the patent.

19. It is my understanding that a claim is unpatentable under 35 U.S.C. § 103 if the claimed subject matter as a whole would have been obvious to a person of ordinary skill in the art at the time of the alleged invention. I also understand that an obviousness analysis takes into account the scope and content of the prior art, the differences between the claimed subject matter and the prior art, and the level of ordinary skill in the art at the time of the invention.

20. In determining the scope and content of the prior art, it is my understanding that a reference is considered relevant prior art if it falls within the field of the inventor’s endeavor. In addition, a reference is prior art if it is reasonably pertinent to the particular problem with which the inventor was involved. A reference is reasonably pertinent if it logically would have commended itself to an inventor’s attention in considering his problem. If a reference relates to the same problem as the claimed invention, that supports use of

the reference as prior art in an obviousness analysis.

21. To assess the differences between prior art and the claimed subject matter, it is my understanding that 35 U.S.C. § 103 requires the claimed invention to be considered as a whole. This "as a whole" assessment involves showing that one of ordinary skill in the art at the time of invention, confronted by the same problems as the inventor and with no knowledge of the claimed invention, would have selected the elements from the prior art and combined them in the claimed manner.

22. It is my further understanding that several rationales may be applied for combining references or modifying a reference to show obviousness of claimed subject matter. These rationales include: combining prior art elements according to known methods to yield predictable results; simple substitution of one known element for another to obtain predictable results; a predictable use of prior art elements according to their established functions; applying a known technique to a known device (method or product) ready for improvement to yield predictable results; choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify a prior art reference or to combine prior art teachings to arrive at the claimed invention.

IV. MATERIALS CONSIDERED

23. My analysis here is based on my years of education, research and experience, as well as my investigation and study of relevant materials, including those cited herein. I may rely upon these materials, my knowledge and experience, and/or additional materials to further explain and corroborate my analysis, and to respond to any critiques of my analysis that may be raised during the course of the IPR proceeding in which this declaration is submitted.

24. I understand that earlier IPR proceedings, IPR2024-00626 and IPR2024-00627, were instituted before the Patent Trial and Appeal Board concerning the ’267 patent, in which Amazon.com, Inc. and Amazon.com Services LLC (collectively, “Amazon”) were the petitioners (the “Amazon IPRs”). I understand that Dr. Immanuel Freedman submitted two declarations in the Amazon IPRs, which I have attached as Appendices B & C to the present declaration. I have reviewed Dr. Freedman’s declarations in their entirety, including the analysis, claim constructions, and supporting technical opinions presented therein. Based on my independent analysis of the ’267 patent and the materials cited herein, I agree with the technical opinions and substance of Dr. Freedman’s declarations from the Amazon IPRs for issues related to the grounds based on Karczewicz-I and Karczewicz-II, and I adopt them as my own unless otherwise noted. Dr. Freedman’s declarations are fully incorporated herein as they relate to the grounds

based on Karczewicz-I and Karczewicz-II.

25. In preparing this declaration, I considered the following materials in addition to Dr. Freedman's declarations:

ASUS Exhibit No.	Description
ASUS-1005	U.S. Patent Application Publication No. 2011/0007799 ("Karczewicz-I")
ASUS-1006	U.S. Patent Application Publication No. 2009/0257499 ("Karczewicz-II")
ASUS-1007	Prosecution History for U.S. Patent No. 9,432,693
ASUS-1008	U.S. Patent No. 9,344,744 ("Kirchhoffer")
ASUS-1009	Srinivasan, An Overview of VC-1
ASUS-1010	U.S. Patent Application Publication No. 2003/0112864 ("Karczewicz-864")
ASUS-1011	Wiegand, Overview of the H.264/AVC Video Coding Standard
ASUS-1012	Richardson, The H.264 Advanced Video Compression Standard
ASUS-1013	U.S. Patent Application Publication No. 2008/0198935 ("Srinivasan-935")
ASUS-1014	H.264 Advanced Video Coding for Generic Audiovisual Services (March 2009)
ASUS-1016	U.S. Patent No. 8,594,188 ("Demos")
ASUS-1024	Deposition Transcript of Dr. Iain Richardson

ASUS-1030	Deposition Transcript of Immanuel Freedman

V. OVERVIEW OF THE ’267 PATENT

26. The ’267 patent is directed to “utilizing motion prediction in video coding.” ASUS-1001, Abstract. Dr. Freedman’s declarations provide an overview of the subject matter of the ’267 patent, including background on digital video technologies, the field of art, the prosecution history, and the claims. See Appx. B, §§I-III; Appx. C, §§I-III. Rather than repeat these aspects of Dr. Freedman’s testimony, and to provide more focused testimony herein, I refer to Dr. Freedman’s declaration for further discussion of the ’267 patent.

27. For reference, I provide the following listing of challenged claim elements from the ’267 patent:

Claim 1	
[1a]	A method for encoding a block of pixels, the method comprising:
[1b]	determining, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;
[1c]	using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[1d]	using said second reference block to obtain a second prediction, said second prediction having the second precision;

[1e]	obtaining a combined prediction based at least partly upon said first prediction and said second prediction;
[1f]	decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[1g]	encoding residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.
Claim 2	
2	The method according to claim 1, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.
Claim 3	
3	The method according to claim 2, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 4	
4	The method according to claim 2, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 5	
5	The method according to claim 1, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises: inserting a rounding offset to the combined prediction before said decreasing.
Claim 6	
6	The method according to claim 1, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the

	second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.
Claim 7	
[7a]	An apparatus for encoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:
[7b]	determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;
[7c]	use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[7d]	use said second reference block to obtain a second prediction, said second prediction having the second precision;
[7e]	obtain a combined prediction based at least partly upon said first prediction and said second prediction;
[7f]	decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[7g]	encode residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.
Claim 8	
8	The apparatus according to claim 7, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.
Claim 9	

9	The apparatus according to claim 8, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 10	
10	The apparatus according to claim 8, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 11	
11	The apparatus according to claim 7, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by: inserting a rounding offset to the combined prediction before said decreasing.
Claim 12	
12	The apparatus according to claim 7, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.
Claim 13	
[13a]	A computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:
[13b]	determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[13c]	use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[13d]	use said second reference block to obtain a second prediction, said second prediction having the second precision;
[13e]	obtain a combined prediction based at least partly upon said first prediction and said second prediction;
[13f]	decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[13g]	encode residual data in a bitstream, wherein the residual data is determined based upon a difference between the combined prediction and the block of pixels.
Claim 14	
14	The computer program product according to claim 13, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.
Claim 15	
15	The computer program product according to claim 14, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 16	
16	The computer program product according to claim 14, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 17	
17	The computer program product according to claim 13, wherein the program code instructions configured to decrease said precision of said

	combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to: insert a rounding offset to the combined prediction before said decreasing.
Claim 18	
18	The computer program product according to claim 13, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.
Claim 19	
[19a]	A method for decoding a block of pixels, the method comprising:
[19b]	determining, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;
[19c]	using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[19d]	using said second reference block to obtain a second prediction, said second prediction having the second precision;
[19e]	obtaining a combined prediction based at least partly upon said first prediction and said second prediction;
[19f]	decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[19g]	reconstructing the block of pixels based on the combined prediction.
Claim 20	
20	The method according to claim 19, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

Claim 21	
21	The method according to claim 20, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 22	
22	The method according to claim 20, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 23	
23	The method according to claim 19, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises: inserting a rounding offset to the combined prediction before said decreasing.
Claim 24	
24	The method according to claim 19, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.
Claim 25	
[25a]	An apparatus for decoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:
[25b]	determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[25c]	use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[25d]	use said second reference block to obtain a second prediction, said second prediction having the second precision;
[25e]	obtain a combined prediction based at least partly upon said first prediction and said second prediction;
[25f]	decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[25g]	reconstruct the block of pixels based on the combined prediction.
Claim 26	
26	The apparatus according to claim 25, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.
Claim 27	
27	The apparatus according to claim 26, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 28	
28	The apparatus according to claim 26, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 29	
29	The apparatus according to claim 25, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the

	combined prediction to the right, by: inserting a rounding offset to the combined prediction before said decreasing.
Claim 30	
30	The apparatus according to claim 25, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.
Claim 31	
[31a]	A computer program product for decoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to:
[31b]	determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;
[31c]	use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;
[31d]	use said second reference block to obtain a second prediction, said second prediction having the second precision;
[31e]	obtain a combined prediction based at least partly upon said first prediction and said second prediction;
[31f]	decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and
[31g]	reconstruct the block of pixels based on the combined prediction.
Claim 32	

32	The computer program product according to claim 31, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.
Claim 33	
33	The computer program product according to claim 32, wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.
Claim 34	
34	The computer program product according to claim 32, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.
Claim 35	
35	The computer program product according to claim 31, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to: insert a rounding offset to the combined prediction before said decreasing.
Claim 36	
36	The computer program product according to claim 31, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

VI. PERSON OF ORDINARY SKILL IN THE ART

28. It is my understanding the patentability of the claims of the ’267 patent must be assessed from the perspective of a person of ordinary skill in the art

at the time of the alleged invention ("POSITA"). For purposes of my analysis in this declaration, I have taken the earliest claimed priority date of the '267 patent (January 7, 2011) as the date of the alleged invention ("Critical Date"). I understand that the factors considered in determining the ordinary level of skill in a field of art include the level of education and experience of persons working in the field; the types of problems encountered in the field; the teachings of the prior art, and the sophistication of the technology at the time of the alleged invention. I understand that a POSITA is not a specific real individual, but rather is a hypothetical individual having the qualities reflected by the factors above. I understand that a POSITA would also have knowledge from the teachings of the prior art, including the art cited below.

29. Taking these factors into consideration, it is my opinion that one of ordinary skill in the art in the field of digital video coding as of the Critical Date, would have had a 1) a bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa.

30. I have possessed the qualifications of a POSITA since the Critical Date of the '267 patent, and long before.

VII. CLAIM CONSTRUCTION

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

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

A. "precision"

33. I have carefully reviewed Dr. Freedman's analysis of the term "precision," and I agree with and adopt his analysis as my own. *See* Appx. A, §IV.A; Appx. B, §IV.A.

VIII. GROUND 1: CLAIMS 1-36 ARE OBVIOUS BASED ON KARCZEWICZ-I IN VIEW OF KARCZEWICZ-II

34. In Section V.B of his declarations, Dr. Freedman provides an overview of the Karczewicz-I and Karczewicz-II prior art references. Dr. Freedman then analyzed these prior art references and explained in detail why a POSITA would have found it obvious to combine the teachings of Karczewicz-I and Karczewicz-II to arrive at the alleged inventions described in claims 1-36 of the

’267 patent. I have carefully reviewed Dr. Freedman’s analysis in this regard, and I agree with and adopt his analysis as my own. It is clear that a predictable combination of Karczewicz-I and Karczewicz-II would have rendered claims 1-36 of the ’267 patent obvious before the Critical Date for the reasons articulated in Dr. Freedman’s declarations.

35. In addition, it is my opinion that Karczewicz-I and Karczewicz-II teach the same calculation of averaging interpolated pixel values. Karczewicz-I teaches bi-prediction techniques for H.264 with two motion vectors pointing to two blocks of pixels or integer values that are averaged together. ASUS-1005, ¶53, ¶35, ¶44, ¶60. Karczewicz-II optimizes this calculation by preserving higher-precision intermediate values. The combination simply applies the exact optimization of Karczewicz-II to modify the corresponding calculation in Karczewicz-I and uses a known technique to improve a similar device/method. The references share the same architecture, and the combination does not change the architecture; it simply uses more bits for intermediate calculations, as Karczewicz-II teaches.

36. Karczewicz-II teaches optimizations for interpolating and averaging integer, half-, and center-pixels, which a POSITA would have been motivated to apply to at least three scenarios for Karczewicz-I’s teachings that follow the exact optimization scenarios taught by Karczewicz-II for averaging integer, half, and center pixel values. The modifications for each scenario implement Karczewicz-

II's optimization of preserving higher-precision intermediate values. Across the three scenarios, Karczewicz-II treats each type of motion vector the same. For example, when a motion vector points to a half-pixel, all three scenarios use a non-rounded half-pixel value. ASUS-1006, ¶103, ¶105, Tables 5, 6, 8; ASUS-1024, 118:14-121:4 (confirming that Karczewicz-II teaches one method of calculating non-rounded half-pixel value). Therefore, in the combination, only one value needs to be stored for the first motion vector.

37. Since H.264 already included bi-prediction and interpolation—with two motion vectors that could point to integer, half-pixel, or center-pixel positions—the three scenarios of motion vectors pointing to different permutations of integer or sub-pixel locations were already present for H.264. ASUS-1024, 81:3-12, 101:6-102:7; ASUS-1006, ¶¶93-102; ASUS-1014, 190-193. The combination simply uses more bits when calculating these scenarios that were encountered during pre-existing H.264 encoding/decoding. *Id*. Therefore, even if code branches or logic costs were needed to handle these three scenarios for the combination, they are generic costs that were already needed for H.264 even without the combination. ASUS-1024, 271:7-16, 267:9-16 (admitting that code branches are ubiquitous, stating "I don't think I could write a video codex even a very simple one without using conditional execution."); ASUS-1030, 102:7-103:22 ("it would be convenient and simple to maintain the higher [precision] throughout

the sequence of calculations"; the modification of Karczewicz-I involves "a tiny change" and "a few lines of code."). The '267 patent's discussion of code branches as "Background Information" proves that separate code branches were known and used in the prior art, despite their alleged inefficiency or cost. ASUS-1001, 4:14-43.

38. Further, Karczewicz-II already provides motivation to use its optimizations for averaging interpolated pixel values, applied to three scenarios, and Karczewicz-II provides motivation to use its teachings despite Karczewicz-II's calculations being carried out millions of times per second and despite the encoder's need to repeatedly test potential predictions, which is present regardless of whether higher-precision intermediate values are used and are not caused by the modifications of Karczewicz-I. ASUS-1024, 145:21-146:18. A POSITA would have recognized that there were ways of minimizing the complexity and limiting the number of searches to a small number. ASUS-1024, 159:5-160:11. For video codecs, performance is often improved at the expense of increased computational complexity. ASUS-1024, 271:17-272:8. Therefore, even if there were a tradeoff, it would not obviate the motivation to combine.

39. Moreover, the combination rounding occurs for calculations that already included rounding. ASUS-1005, ¶60, ¶55; ASUS-1006, ¶¶96-106, Tables 1-8. The affected calculations already involve rounding. *See id*. The calculations

themselves include basic mathematical and logical operations, such as binary arithmetic, addition, rounding, and bit shifting. This minor implementation detail would not have changed the principle of operation of Karczewicz-I or Karczewicz-II.

IX. CONCLUSION

40. In conclusion, I find the claims of the ’267 patent addressed herein to be rendered obvious in their entirety, based upon the prior art references Karczewicz-I and Karczewicz-II and the combination of these prior art references.

41. The findings and opinions set forth in this declaration are based on my work and examinations to date.

42. I may continue my examinations. I may also receive additional documentation and other factual evidence over the course of this IPR that will allow me to supplement and/or refine my opinions. I reserve the right to add to, alter, or delete my opinions and my declaration upon discovery of any additional information. I reserve the right to make such changes as may be deemed necessary.

43. In signing this declaration, I recognize that the declaration will be filed as evidence in an IPR before the PTAB. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for

cross-examination within the United States during the time allotted for cross-examination.

Appendix A

Joseph P. Havlicek

The University of Oklahoma, School of Electrical & Computer Engineering
110 W. Boyd, DEH 150, Norman, OK 73019
E-mail: joebob@ou.edu Gmail: joseph.p.havlicek@gmail.com
http://www.ou.edu/content/coe/ece/faculty_directory/dr_havlicek.html

Title:	Gerald Tuma Presidential Professor & Williams Companies Foundation Presidential Professor
Unit:	School of Electrical and Computer Engineering
Director:	OU Center for Intelligent Transportation Systems
Member:	OU Institute for Biomedical Engineering, Science, and Technology

- **Citizenship:** USA

- **Education:**

PhD EE	The University of Texas at Austin, 1996. Dissertation: "AM-FM Image Models." Advisor: Prof. Alan C. Bovik.
MSEE	Virginia Tech, 1988. Thesis: "Median Filtering for Target Detection in an Airborne Threat Warning System." Advisor: Prof. John C. McKeeman.
BSEE	Virginia Tech, 1986. Minors in Mathematics, Computer Science.

- **Professional Experience:**

1/97 - present: School of Electrical & Computer Engineering, Univ. OK, Norman, OK

Gerald Tuma Presidential Professor:	4/17 - present
Williams Companies Foundation Presidential Professor:	4/09 - present
Professor:	7/07 - present
Associate Professor:	7/02 - 6/07
Assistant Professor:	1/97 - 6/02

Held tenure track position requiring research, teaching, and service, as well as establishment of strong, externally funded research programs in signal, image, and video processing and intelligent transportation systems. Director and co-founder, OU Center for Intelligent Transportation Systems. Member, OU Institute for Biomedical Engineering, Science, and Technology. Total external grants and contracts exceeding $27M.

6/87 - 1/97: U.S. Naval Research Laboratory, Washington, DC
Electrical Engineer
(Was affiliated with SFA, Inc., Landover MD, from 6/87-8/89)
(Was on *leave without pay* during semesters spent at UT Austin)
Engineering member of the team that developed the Navy's first two-color infrared missile warning receiver (Fly's Eye). The production version of this system protected Navy and Marine helicopters from surface to air missile attacks in Afghanistan and Iraq. Received the Department of the Navy Award of Merit for Group Achievement for this work. Designed and analyzed new algorithms for infrared target detection, tracking, and identification. Designed digital architectures for real-time implementation. Conducted experimental work on airborne and ground-based platforms. Extensive field experience at China Lake Naval Weapons Center, Miramar Naval Air Station, Patuxent River Naval Air Station, and Sandia National Laboratories.

6/93 - 12/96: Dept. Electrical & Computer Engineering, University of Texas, Austin, TX
Assistant Director, Laboratory for Vision Systems
Senior student administrator of laboratory whose members include approximately 12 research-supported graduate students. Authored and integrated grant proposals. Briefed sponsors. Authored contract reports. Reviewed papers for journals and conferences. Advised graduate students. Supervised honors undergraduate projects. Substitute lecturer for both graduate and undergraduate courses in the systems area.

1/93 - 12/93: Dept. E51, Still Video Products, IBM Corporation, Austin, TX
Software Developer
(on-sight contractor affiliated with Ralph Kirkley Associates, Austin, TX)
Developed C code for IBM PS/2 computers under OS/2 and MS Windows to port an implementation of the JPEG image compression/decompression standard from the IBM M/ACPA card to the IBM AudioVation card.

8/87 - 8/88: Bradley Department of Electrical Engineering, VPI & SU, Blacksburg, VA
Graduate Research Assistant
Under contract with NRL, led 9-man team in chip-level simulation of a real-time nonlinear image filter. Under contract with IBM, investigated the feasibility and performance of networks of LEO store-and-forward communication satellites.

12/84 - 5/87: Management Systems Laboratories, Blacksburg, VA
Software Engineer
Under contract with DOE, designed and implemented management decision support software for nuclear materials management on IBM mainframe computers.

► **Expert Testimony History:**

- In the matter of *Unified Patents, LLC Request for Ex Parte Reexamination Against U.S. Patent No. 10,574,982 assigned to Dolby Video Compression, LLC*. Provided opinions and testimony at Examiner interview. Retained by Fish & Richardson P.C. on behalf of Dolby Video Compression, LLC, San Francisco, CA, 12/24 - present.
- In the matter of *University of British Columbia v. Caption Health, Inc., et al.*, Case No. 5:24-cv-03200-EKL, U.S. District Court for the Northern District of California. Reviewed source code and provided infringement analysis. Retained by Perkins Coie LLP on behalf of University of British Columbia, Vancouver, BC, Canada, 12/24 - present.
- In the matter of *Omnitracs v. Motive Technologies*, Case No. 3:23-cv-05261, U.S. District Court for the Northern District of California. Provided infringement analysis, patent benefit analysis, and noninfringement allegation test specifications. Retained by Kirkland & Ellis LLP on behalf of Omnitracs, LLC, Westlake, TX, XRS Corporation, Burnsville, MN, and SmartDrive Systems, Inc., San Diego, CA, 7/24 - 9/24.
- In the matter of *Amazon.com, Inc. and Amazon.com Services LLC v. Nokia Technologies Oy*, petition for IPR of U.S. Patent No. 8,050,321, USPTO Case No. IPR2024-00691. Provided IPR expert declaration, testified at deposition. Retained by Perkins Coie LLP on behalf of Amazon.com, Inc. and Amazon.com Services LLC, Seattle, WA, 11/23 - present.
- In the matter of *Amazon.com, Inc. and Amazon.com Services LLC v. Nokia Technologies Oy*, petition for IPR of U.S. Patent No. 8,204,134, USPTO Case No. IPR2024-00725. Provided IPR expert declaration. Retained by Perkins Coie LLP on behalf of Amazon.com, Inc. and Amazon.com Services LLC, Seattle, WA, 11/23 - 4/24.
- In the matter of *Amazon.com, Inc. and Amazon.com Services LLC v. Nokia Technologies Oy*, petition for IPR of U.S. Patent No. 7,532,808, USPTO Case No. IPR2024-00847, IPR2024-00848. Provided IPR expert declaration. Retained by Perkins Coie LLP on behalf of Amazon.com, Inc. and Amazon.com Services LLC, Seattle, WA, 11/23 - present.
- In the matter of *Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, Cameras, and Components and modules Thereof*, USITC Investigation No. 337-TA-1379. Provided two declarations. Retained by Perkins Coie LLP on behalf of Amazon.com, Inc. and Amazon.com Services LLC, Seattle, WA, 11/23 - 5/24.
- In the matter of *Certain Electronic Devices, Including Smartphones, Computers, Tablet Computers, and Components Thereof*, USITC Investigation No. 337-TA-1373. Provided nonin-

fringement analysis and one declaration. Retained by Fish & Richardson P.C. on behalf of Intel Corporation, Santa Clara, CA, and Lenovo Group Limited, Hong Kong S.A.R., China, Lenovo (United States) Inc., Morrisville, NC, and Motorola Mobility LLC, Chicago, IL, 11/23 - 3/24.

- In the matter of *Unified Patents, LLC Request for Reexamination Against U.S. Patent No. 7,739,714 assigned to Distributed Media Solutions LLC (an affiliate of IP Investments).* Provided one *Ex Parte* Reexam declaration. Retained by Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 10/23 - 12/23.
- In the matter of *Advanced Coding Technologies LLC v. Samsung Electronics Co. LTD and Samsung Electronics America, Inc., Case No. 2:22-cv-00499, U.S. District Court for the Eastern District of TX.* Provided two expert reports, testified at deposition. Retained by Fish & Richardson P.C. on behalf of Samsung Electronics Co. LTD, Suwon, South Korea, and Samsung Electronics America, Inc., Ridgefield Park, NJ, 5/23 - 8/24.
- In the matter of *State of Texas v. Meta Platforms, Inc.*, Cause No. 22-0121. Provided technical consulting to support Meta's defense related to facial recognition software. Retained by Gibson, Dunn & Crutcher LLP on behalf of Meta Platforms, Inc., Menlo Park, CA, 6/23 - 3/24.
- In the matter of *Certain Video Processing Devices and Products Containing Same*, USITC Investigation No. 337-TA-1323. Provided one expert report, testified at deposition. Retained by Fish & Richardson P.C. on behalf of ASUSTek Computer Inc., Taipei, Taiwan, and ASUS Computer International, Fremont, CA, 2/23 - 5/23.
- In the matter of *Unified Patents, LLC Request for Reexamination Against U.S. Patent No. 9,497,469 assigned to Velos Media LLC.* Provided one *Ex Parte* Reexam declaration. Retained by Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 12/22 - 2/23.
- In the matter of *Certain Video Processing Devices and Products Containing Same*, USITC Investigation No. 337-TA-1323. Provided noninfringement analysis. Retained by Perkins Coie LLP on behalf of Intel Corporation, Santa Clara, CA, 10/22 - 12/22.
- In the matter of *TCL Electronics Holdings Ltd. v. LG Electronics Inc.*, petition for IPR of U.S. Patent No. 7,839,452, USPTO Case No. IPR2023-00461. Provided one IPR declaration. Retained by PV Law LLP on behalf of TCL Electronics Holdings Ltd. and associated companies, Huizhou, Guangdong, China, 8/22 - 5/23.
- In the matter of *PerDiemCo LLC v. CalAmp Corp.*, Case No. 1:20-cv-01397-VAC-SRF, U.S. District Court for the District of DE. Provided noninfringement analysis; case settled. Retained by Barnes & Thornburg LLP on behalf of CalAmp Corp., Irvine, CA, 6/22 - 4/23.
- In the matter of *DigiMedia Tech, LLC v. Lenovo (United States) Inc. and Motorola Mobility LLC*, Case No. 1:21-cv-00227-MN, U.S. District Court for the District of DE. Provided one declaration. Retained by Kilpatrick, Townsend & Stockton LLP on behalf of Lenovo (United States) Inc., Morrisville, NC, and Motorola Mobility LLC, Chicago, IL, 1/22 - 3/22.
- In the matter of *EyesMatch Ltd. and Memomi Labs Inc. v. Facebook, Inc., Instragram, LLC, and WhatsApp LLC*, Case No. 1-21-cv-00111, U.S. District Court for the District of DE. Provided two declarations. Retained by Cooley LLP on behalf of Facebook Inc., Instagram, LLC, and WhatsApp LLC, Menlo Park, CA, 1/22 - 7/22.
- In the matters of certain petitions for IPR associated with *Certain Fitness Devices, Streaming Components Thereof, and Systems Containing the Same*, USITC Investigation No. 337-TA-1265. Worked on five IPR petitions that were ultimately not filed. Retained by Cooley LLP on behalf of Peloton Interactive, Inc. (New York), lululemon athletica Inc. (Vancouver, BC) and Curiouser Products Inc. (New York) d/b/a MIRROR, and iFIT Inc. (Logan, UT), FreeMotion Fitness, Inc. (Logan, UT) and NordicTrack, Inc. (Logan, UT), 9/21 - 4/22.
- In the matter of *Unified Patents, LLC Request for Reexamination Against U.S. Patent No. 10,244,252 assigned to Electronics and Telecommunications Research Institute*. Provided one *Ex Parte* Reexam declaration. Retained by Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 8/21 - 10/21.
- In the matter of *Indect USA Corp. v. Park Assist, LLC*, Case No. 3:18-cv-2409-BEN-MDD, U.S. District Court for the Southern District of CA. Provided one expert report, testified at deposition, testified at trial. Retained by Foley & Lardner LLP on behalf of Indect USA Corp., Denver, CO, 4/21 - 9/22.

- In the matter of *Unified Patents LLC v. GE Video Compression LLC*, Petition for *Ex Parte* Reexamination of U.S. Patent No. 6,795,583. Provided one *Ex Parte* Reexam declaration. Retained by Desmarais LLP on behalf of Unified Patents LLC, San Jose, CA, 9/20 - 04/21.
- In the matter of *Unified Patents LLC v. Electronics and Telecommunications Research Institute, Kwangwoon University Research Institute for Industry Cooperation, Industry-Academia Cooperation Group of Sejong University*, petition for IPR of U.S. Patent No. 9,736,484, Case No. IPR2021-00368. Provided one IPR declaration. Retained by Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 9/20 - 12/20.
- In the matter of *Certain Electronic Devices, Including Computers, Tablet Computers, and Components and Modules Thereof*, USITC Investigation No. 337-TA-1208. Provided two expert reports, testified at deposition. Retained by WilmerHale LLP on behalf of Lenovo (United States) Inc., Morrisville, NC, 8/20 - 3/21.
- In the matter of *Park Assist, LLC v. San Diego County Regional Airport Authority and Ace Parking Management, Inc.*, Case No. 3:18-cv-02068-BEN-MDD, U.S. District Court for the Southern District of CA. Testified at deposition, provided one claim construction declaration. Retained by Morrison & Foerster LLP on behalf of SDCRAA, San Diego, CA, 7/20 - 03/21.
- In the matter of *LG Electronics Inc. v. Hisense Electronics Manufacturing Company of America Corp.*, Civil Case No. 2:19-cv-09474-JAK, U.S. District Court for the Central District of CA, Western Division. Provided one claim construction declaration, testified at deposition. Retained by Covington & Burling LLP on behalf of Hisense Electronics Manufacturing Company of America, Inc., Suwanee, GA, 7/20 - 1/21.
- In the matter of *Renesas Electronics Corporation v. Broadcom Corporation*, petition for IPR of U.S. Patent No. 8,284,844, USPTO Case No. IPR2019-01040. Provided one IPR declaration. Retained by Steptoe & Johnson LLP on behalf of Broadcom Limited, San Jose, CA, 6/20 - 7/20.
- In the matter of *Hisense Electronics Manufacturing Company of America v. LG Electronics Inc.*, petition for IPR of U.S. Patent No. 7,839,452, USPTO Case No. IPR2020-01208. Provided one IPR declaration. Retained by Covington & Burling LLP on behalf of Hisense Electronics Manufacturing Company of America, Inc., Suwanee, GA, 3/20 - 1/21.
- In the matter of *Nokia Technologies v. Lenovo (Shanghai) Electronics Tech. Co. Ltd, et al., 19-CV-0427 (E.D.N.C.) and related Nokia v. Lenovo cases including in Germany and India*. I was briefly retained to perform analysis of reference picture management in H.264 and H.265. Retained by Cooley LLP on behalf of Nvidia Corp., Santa Clara, CA, 1/20 - 3/20.
- In the matter of *Intel Corporation v. Dynamic Data Technologies, LLC*. Provided one IPR declaration; case settled before filing. Retained by Perkins Coie LLP on behalf of Intel Corporation, Santa Clara, CA, 3/19 - 6/19.
- In the matter of *Unified Patents LLC v. Velos Media, LLC*, petition for IPR of U.S. Patent No. 8,885,956, USPTO Case No. IPR2019-01130. Provided one IPR declaration. Retained by Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 2/19 - 12/19.
- In the matter of *Unified Patents LLC v. Velos Media, LLC*, petition for IPR of U.S. Patent No. 10,110,898, USPTO Case No. IPR2019-00763. Provided two declarations, testified at deposition. Retained by Winston & Strawn LLP and Greenberg Traurig LLP on behalf of Unified Patents LLC, San Jose, CA, 10/18 - 04/20.
- In the matter of *Avago Technologies General IP (Singapore) Pte. Ltd. v. Nintendo of Europe GmbH*. Provided written opinions to the German Federal Patent Court. Retained by Freshfields Bruckhaus Deringer LLP on behalf of Avago Technologies General IP (Singapore) Pte. Ltd., 9/18 - 10/18.
- In the matter of *Certain Infotainment Systems, Components Thereof, and Automobiles Containing the Same*, USITC Investigation No. 337-TA-1119. Provided infringement analysis, one declaration. Retained by Steptoe & Johnson LLP on behalf of Broadcom Limited, San Jose, CA, 3/18 - 2/19.
- In the matter of *Cisco Systems, Inc. v. Realtime Adaptive Streaming, LLC*, petition for IPR of U.S. Patent No. 8,934,535, USPTO Case No. IPR2018-01384. Provided one IPR declaration. Retained by Winston & Strawn LLP on behalf of Cisco Systems, Inc., San Jose, CA, 3/18 - 8/18.
- In the matter of *Avago Technologies General IP (Singapore) Pte. Ltd. v. Audi AG*. Provided written opinion to the German Federal Patent Court. Retained by Grünecker Patent- und

Rechtsanwälte PartG mbB on behalf of Avago Technologies General IP (Singapore) Pte. Ltd., 3/18 - 11/18.
- In the matter of *Certain Semiconductor Devices and Consumer Audiovisual Products Containing the Same*, USITC Investigation No. 337-TA-1047. Testified at deposition and trial. Provided three expert reports, two declarations, and three witness statements. Retained by Steptoe & Johnson LLP and Kilpatrick, Townsend & Stockton LLP on behalf of Broadcom Limited, San Jose, CA, 3/17 - 12/17.
- In the matter of *Certain Semiconductor Integrated Circuits and Products Containing Same*, USITC Investigation No. 337-TA-840. Infringement analysis and one written declaration. Retained by Covington & Burling LLP on behalf of Microchip Technology, Chandler, AZ, 2/17/12 - 3/21/12.

▶ Honors & Awards:

- Top Reviewer Recognition, 2024 IEEE International Conference on Image Processing.
- Outstanding Reviewer Recognition Award, 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing.
- Best Reviewer Award, 2020 IEEE International Conference on Image Processing.
- Top Reviewer Certificate, 2020 IEEE International Conference on Image Processing, awarded to top 3% out of over 700 reviewers.
- Named to the University of Oklahoma Gerald Tuma Presidential Professorship, 2017.
- *2014 IEEE International Conference on Image Processing Top 10% Paper Award*, for C.T. Nguyen and J.P. Havlicek, "On the amplitude and phase computation of the AM-FM image model."
- Named to the University of Oklahoma Williams Companies Foundation Presidential Professorship, 2009.
- Oklahoma Highway Safety Office Project Director's Award, FY 2009, co-recipient with Dr. R.D. Barnes, for implementing police electronic crash reporting in the State of Oklahoma.
- IEEE Maximum Impedance Award, OU School of ECE, 2007.
- University of Oklahoma College of Engineering Outstanding Faculty Advisor Award, 2005-2006.
- Oklahoma Highway Safety Office Award of Excellence, FY 2005, presented to the OU ITS Lab for enhancing traffic records management through project SAFE-T.
- Oklahoma Highway Safety Office Project Director's Award, FY 2003, co-recipient with Dr. J.J. Sluss, Jr., for enhancing highway safety through ITS projects.
- University of Oklahoma College of Engineering Brandon H. Griffith Faculty Award, 2003.
- Listed at number 22 in OU *FY 99 Awards – Top 25 Faculty/Staff – Norman Campus*.
- IEEE Favorite Instructor Award, OU School of ECE, 1998, 2000.
- University of Texas Engineering Foundation Award for Exemplary Engineering Teaching while Pursuing a Graduate Degree, 1992.
- Department of the Navy Award of Merit for Group Achievement, 1990.
- Management Systems Laboratories Outstanding Student Employee Scholarship, 1987.
- Eta Kappa Nu Honor Society
- Tau Beta Pi Honor Society
- Phi Kappa Phi Honor Society
- Listed in *Who's Who in America*, 2002 Ed.

▶ Professional Memberships:

- Institute of Electrical and Electronics Engineers (IEEE), Senior Member
- IEEE Signal Processing Society
- IEEE Intelligent Transportation Systems Society
- IEEE Computer Society

▶ Professional Service:

- National Science Foundation, Proposal Review Panelist: 2022, 2020, 2012.
- Senior Area Editor, *IEEE Transactions on Image Processing*, Nov. 2015 - Feb. 2018.
- Associate Editor, *IEEE Transactions on Image Processing*, Dec. 2010 - Oct. 2015.
- Associate Editor, *IEEE Transactions on Industrial Informatics*, Jan. 2010 - Jul. 2013.

- *IEEE International Conference on Image Processing (ICIP)*
  - Reviewer (1998 - present).
  - 2016: Paper Awards Committee.
  - 2013: Technical Area Chair for EDICS 6.1: Image & Video Analysis, Synthesis, and Retrieval.
  - 2012: Technical Area Chair for EDICS 6.2: Image & Video Analysis, Synthesis, and Retrieval; Session Chair.
  - 2007: Publications Chair, Organizing Committee, and Session Chair.
- *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*
  - Reviewer (2005-present).
  - 2012, 2013, 2014: IVMSP Technical Area Chair
- *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*
  - 2024: Technical Program Committee, Session Chair.
  - 2020: Technical Program Committee, Session Chair.
  - 2016: Technical Program Committee.
  - 2012, 2014: Technical Program Committee, Session Chair.
  - 2010: General Co-Chair (with Prof. Scott Acton, University of Virginia).
  - 2008: Technical Program Co-Chair (with Prof. Scott Acton, University of Virginia).
  - 2006: Technical Program Co-Chair (with Prof. Til Aach, RWTH Aachen University, Germany).
  - 2004: Technical Program Co-Chair (with Prof. Til Aach, Medical University of Luebeck, Germany).
  - 2002: Publicity Chairman, Technical Program Committee, Session Chair.
  - 2000: Publicity Chairman, Technical Program Committee, Session Chair.
  - 1998: Technical Program Committee, Session Chair.
- *IEEE International Conference on Intelligent Transportation Systems (ITSC)*
  - 2013: Reviewer
  - 2011: Session Chair, reviewer.
  - 2009: Technical Program Committee, Special Session Organizer, Session Chair.
- *IEEE Workshop on Perception Beyond the Visible Spectrum*
  - 2014, 2015: Technical Program Committee.
- *IEEE Int'l. Workshop on Object Tracking and Classification Beyond the Visible Spectrum*
  - 2009, 2013: Technical Program Committee.
- *European Signal Processing Conference (EUSIPCO)*
  - 2015, 2016, 2017, 2018: Reviewer
- *45th IEEE Midwest Symposium on Circuits and Systems (2002)*: Session Organizer and Session Chair.
- *IEEE Asilomar Conference on Signals, Systems, and Computers*
  - 2000, 2001: special session organizer
- Presently serving or have served as a reviewer for *IEEE Transactions on Signal Processing*; *IEEE Transactions on Image Processing*; *IEEE Signal Processing Letters*; *IEEE Transactions on Pattern Analysis and Machine Intelligence*; *IEEE Transactions on Circuits and Systems II*; *IEEE Transactions on Communications*; *IEEE Transactions on Industrial Informatics*; *IEEE Transactions on Parallel and Distributed Systems*; *IEEE Transactions on Education*; *IEEE Transactions on Information Technology in Biomedicine*; *Journal of the Optical Society of America – A*; *IEE Proceedings – Vision, Image & Signal Processing*; *IEE Electronics Letters*; *EURASIP Journal on Applied Signal Processing*; *Journal of Electronic Imaging*; *Pattern Recognition Letters*; *Multidimensional Systems and Signal Processing*; *Signal Processing*.

► **Committee Assignments and University Service:**

- Committee A, School of Electrical & Computer Engineering (tenure and promotion/executive committee) (Aug 14 - Aug 16, Nov 04 - Aug 08, Aug 23 - present)

- Chairman, Graduate Studies Committee, School of Electrical & Computer Engineering (Aug 08 - Jul 13)
- School of Electrical & Computer Engineering Graduate Liaison (Aug 08 - Jul 13)
- Graduate Studies Committee, School of Electrical & Computer Engineering (Aug 08 - Jul 13, Aug 97 - Aug 06)
- Chairman, Undergraduate Studies Committee, School of Electrical & Computer Engineering (Dec 21 - Aug 23)
- School of Electrical & Computer Engineering Undergraduate Program Committee (May 19 - May 20)
- Chairman, College of Engineering PP03 Faculty Task Force (Mar 12 - Apr 13) (task force to revise and rewrite policies and procedures for faculty tenure, promotion, annual evaluations, and workload)
- University of Oklahoma Conflict of Interest Advisory Committee (Aug 15 - present, Co-Chair Jan 21 - present).
- University of Oklahoma Conflict of Interest Officer Search Committee (Oct 21 - Jan 22)
- University of Oklahoma Graduate Council (Aug 10 - Jun 13)
- College of Engineering E-Club Faculty Co-Advisor (May 00 - May 04), Advisor (May 04 - Jan 06) (*this is the largest student organization on the OU campus*)
- Faculty Senate (Aug 02 - May 05)
- College of Engineering Academic Misconduct Board and Grade Appeals Board (Jun 03 - Jun 05)
- Coordinator, Systems Area Faculty Interest Group (FIG) (Dec 08 - present, Oct 00 - Aug 02)
- School of Music piano faculty search committee (Sep 19 - Mar 21, Sep 16 - Dec 16, Sep 12 - Dec 12)
- School of Electrical & Computer Engineering Director Search Committee (Oct 04 - Jun 05)
- School of ECE Faculty Search Committee (97, 02, 03, 05, 06, 07, 14, 15, 17, 18)

► **Teaching:**

1/97 - present: School of Electrical & Computer Eng., University of OK, Norman, OK

- ECE2713, Digital Signals and Filtering (SP 18, SP 19, SP 20, SP 21, SP 22, SP 23, SP 24, SP25)
- ECE3223, Microcomputer System Design (FA 97)
- ECE3793, Signals and Systems (SP 97, FA 98, SP 99, FA 99, SP 00, FA 00, SP 01, SP 02, FA 02, SP 03, FA 03, SP 04, FA 04, SP 05, FA 05, SP 06, SP 07, SP 08, SP 15, SP 16, SP 17)
- ECE3960, Honors Reading (SP 00)
- ECE3980, Honors Research (FA 01, SP 02, SP 03, FA 11, SP 12, SP 19, SP 25)
- ECE4213, Digital Signal Processing (FA 02, FA 06, FA 07, FA 08, FA 09, FA 10, FA 11, FA 12, FA 14, FA 15, FA 16, FA 17, FA 18, FA 19, FA 20, FA 21, FA 22, FA 23)
- ECE4973, Digital Image Processing (SP 98)
- ECE4990, Special Studies (various semesters SP 98 – present)
- ECE5213, Digital Signal Processing (FA 02, FA 06, FA 07, FA 08, FA 09, FA 10, FA 11, FA 12, FA 14, FA 15, FA 16, FA 17, FA 18, FA 19, FA 20, FA 21, FA 22, FA 23)
- CS5273, Digital Image Processing (SP 98, FA 00, SP 02, SP 03, SP 04, SP 05)
- ECE5273, Digital Image Processing (SP 98, FA 00, SP 02, SP 03, SP 04, SP 05, SP 06, SP 07, SP 08, SP 09, SP 10, SP 11, SP 12, SP 13, SP 14, SP 15, SP 16, SP 17, SP 18, SP 19, SP 20, SP 21, SP 22, SP 23, SP 24, SP 25)
- ECE5973/ECE4973, Multimedia Communications (FA 98)
- ECE5973, Kalman Filtering (FA 99, FA 03, FA 05)
- ECE5980, Thesis Research (SP 99 – present)
- ECE5990, Special Problems (various semesters FA 97 – present)
- ECE6283, Advanced Image Processing (FA 04)
- ECE6973, Advanced Image Processing (FA 01)
- ECE6980, Dissertation Research (SP 00 – present)

9/90 - 6/93: Dept. Electrical & Computer Eng., University of Texas, Austin, TX

- EE464K, Senior Design Projects (FA 90 – Summer 93)

1/91 - 12/96: Dept. Electrical & Computer Eng., University of Texas, Austin, TX

- EE381K, Topic 10: Image Processing (substitute lecturer)
- EE381K, Topic 8: Digital Signal Processing (substitute lecturer)
- EE380L, Topic 7: Computer Vision (substitute lecturer)
- EE351K, Probability and Random Processes (substitute lecturer)

► **Graduate Degree Production:**

Ph.D. Supervisions Completed:

1. Peter Tay, "An Optimally Well Localized Multi-Channel Parallel Perfect Reconstruction Filter Bank," October, 2003.
2. Guangwei Mu "WAAS Error, Integrity and Availability Modeling for GPS-based Aircraft Landing System," April, 2004 (co-supervised with Dr. Jim Sluss).
3. Hengqing Wen, "Anti-Spoof Design for TDMA Based GPS/LAAS Landing Aid," December, 2004.
4. Yunhua Wang, "Multiplierless CSD Techniques for High Performance FPGA Implementations of Digital Filters," April, 2007 (co-supervised with Dr. Linda DeBrunner).
5. Osama Alkhouli, "Hirschman Optimal Transform Least Mean Square Adaptive Filters," October, 2007 (co-supervised with Dr. Victor DeBrunner).
6. Ngao D. Mamuya, "Biometric Classification with Factor Analysis," May, 2010.
7. Nicholas A. Mould, "Neighborhood-Level Learning Techniques for Nonparametric Scene Models," May, 2012.
8. Chuong T. Nguyen, "Modulation Domain Image Processing," May, 2012.
9. Ekasit Vorakitolan, "Video CODEC with Adaptive Frame Rate Control for Intelligent Transportation System Applications," May, 2014.
10. Patrick Adrian Campbell, "High-Fidelity and Perfect Reconstruction Techniques for Synthesizing Modulation Domain Filtered Images," December, 2016.
11. Johnathan D. Williams, "Extended Observation Particle Filter with SVD Template Generation Implemented for GPU," December, 2018.
12. John R. Junger III, "Object Detection in Dual-Band Infrared," November 2023.

Ph.D. Supervisions in Progress:

- Elnaz Aghdaei
- Obada Muhammad (Biomedical Engineering)

Additional Ph.D. Committees Served on:

1. Madhavi Kadiyala, "Design of Optimal Subband Filter Banks for Image Discrimination," October, 1999.
2. Mohamed Allali, "Digital Signal Processing on the Unit Sphere via a Ramanujan Set of Rotations and Planar Wavelets" (interdisciplinary: Electrical Engineering and Mathematics), July, 2000.
3. Yunxiang Wu, "Iterative Decoding for Magnetic Recording Channels," September, 2000.
4. Helen Jun Xing, "Performance Evaluation of CDMA Systems," April, 2001.
5. Pamela Pike, "Leisure Piano Lessons: A Case Study of Lifelong Learning" (Music – DMA), May, 2001.
6. Longji Wang, "Active Vibration Control Systems in the Frequency and Sub-Band Domain," July, 2001.
7. Sebastian Torres, "Estimation of Doppler and Polarimetric Variables for Weather Radars," October, 2001.
8. Valliappa Lakshmanan, "A Hierarchical, Multiscale Texture Segmentation Algorithm for Real-World Scenes," October, 2001.
9. Richard Todd, "Design of Low-Density Parity Check Codes for Magnetic Recording Channels," December, 2002.
10. Guoping Wang, "A High-Performance Inner-Product Processor for Real and Complex Numbers," April, 2003.
11. Leslie Fife, "TriM: Tri-Modal Data Communication in Mobile Ad-Hoc Network Database Systems" (Computer Science), December, 2003.

12. Kuo-Liang Li, "Usage and Development of Piano Method Books in Tiawan: Interviews and Observations with Piano Teachers" (Music – DMA), April, 2004.
13. Weijun Tan, "Low-Density Parity-Check Coding for High-Density Magnetic Recording Systems," July, 2004.
14. Haitao Xia, "Error-Correction Coding for High-Density Magnetic Recording Channels," September, 2004.
15. Yongshen Ni, "Fuzzy Correlation and Regression Analysis," April, 2005.
16. Dayong Zhou, "Adaptive Nonlinear System Compensation Techniques and their Applications to Digital Communication and Control Systems," April, 2005.
17. Xiaojuan Hu, "FIR Filter Design for Area Efficient Implementation," May, 2005.
18. Lesley Sisterhen, "The Use of Imagery, Mental Practice, and Relaxation Techniques for Musical Performance Enhancement" (Music – DMA), June, 2005.
19. Su Yang, "Design of PHY & MAC Layer Protocols for Inter-Vehicle Communications," October, 2005.
20. Rob Sulman, "Affine Group Actions on Euclidean Space" (Mathematics), April, 2006.
21. Peng Yan, "A Study on Mobile Ad Hoc Networks Equipped with Free-Space Optical Capabilities," December, 2006.
22. Yan Zhai, "Improved Nonlinear Filtering for Target Tracking," April, 2007.
23. Cheng Zhong, "Efficient Soft-Decision Decoding of Reed-Solomon Codes," May, 2008.
24. Yih-Ru Huang, "Optoelectronics Three-Dimensional Tracking System for Collision Risk Model," April, 2009.
25. Mari Iida, "The Acceptance of Western Piano Music in Japan and the Career of Takahiro Sonoda" (Music – DMA), April, 2009.
26. Yong Ma, "Multi-Modal Behavior and Clustering in Dynamical Systems with Applications to Wind Farms," April, 2009.
27. Yuzhen Xue, "Identification and Estimation of Multi-Modal Complex Dynamic System," May, 2009.
28. B.H.M. Priyantha Wijesinghe, "Development of a Prototype In-Situ Fatigue Sensor for Structural Health Monitoring of Highway Bridges" (Civil Engineering), April, 2010.
29. Han Wang, "Parallel Subspace Subcodes of Reed-Solomon Codes for Magnetic Recording Channels," May, 2010.
30. Yahia Tachwali, "Cognitive Radio Solution for IEEE 802.22," July, 2010.
31. Wei Guan, "Some Local and Global Aspects of Mathematical Digital Signal Processing" (Mathematics), August, 2010.
32. Molly Donovan Wong, "Development and Characterization of a High Energy Phase Contrast X-Ray Imaging System Prototype," June, 2011.
33. Chenxi Lin, "Problems in the Design and Operation of Uncertain Complex Engineering Systems," July, 2011.
34. Jie Lu, "Distributed Computation and Optimization over Networks," July, 2011.
35. Rodney Keele, "Advances in Modeling and signal processing for Bit-Patterned Magnetic Recording Channels with Written-In Errors," April, 2012.
36. Di Wang, "Learning Visual Features for Grasp Selection and Control" (Computer Science), April, 2012.
37. Phuong Pham, "Target Tracking Using Wireless Sensor Networks," November, 2012.
38. Lina Sawalha, "Exploiting Heterogeneous Multicore Processors through Fine-Grained Scheduling and Low-Overhead Thread Migration," December, 2012.
39. Nickolas LaSorte, "The Coexistence of Wireless Medical Devices in the Presence of Heterogeneous Wireless Networks," April, 2013.
40. Shang Wang, "Waveform and Transceiver Optimization for Multi-Functional Airborne Radar Through Adaptive Processing," May, 2013.
41. Enfeng Jiang, "Channel Detection on Two-Dimensional Magnetic Recording," July, 2013.
42. David Sandmann, "Design and Implementation of a Precision Three-Dimensional Binocular Image Tracker for Departing Aircraft," November, 2013.
43. Min Zhu, "EEG/MEG Sparse Source Imaging and its Application in Epilepsy," December, 2013.
44. Seyed Hossein Hosseini, "Revealing Additional Information About Electricity Market Underlying Power System Using Power System Principles and Published Market Results," Septem-

ber, 2014.

45. James M. Kurdzo, "Pulse Compression Waveforms and Applications for Weather Radar" (Meteorology), October, 2015.
46. Peng F. Tang, "Analysis of Backbone Technique: A Hilbert Transform and Discrete Hilbert Transform-Based Technique," December, 2015.
47. Benjamin P. Carlson, "Phenotype Operators for Improved Performance of Heuristic Encoding within Genetic Algorithms" (Computer Science), April, 2016.
48. Erik Petrich, "Real-Time 3-D Scene Reconstruction," May, 2016.
49. Kristina Henckel, "A Pianistic Analysis of Bedřich Smetana's Piano Cycle *Dreams, Six Characteristic Pieces for Piano*" (Music – DMA), November, 2016.
50. Milad Javadi, "New Implication of Short Circuit Analysis in Assessing Impact of Renewable Energy Resources on System Strength of a Power Grid," June, 2017.
51. Xining Yu, "Digital Signal Processing Based Real-Time Phased Array Radar Backend System and Optimization Algorithms," October, 2017.
52. Muhammad Usman Ghani, "Optimization of a High-Energy X-Ray Inline Phase Sensitive Imaging System for Diagnosis of Breast Cancer," April, 2018.
53. Chuang Li, "Reconstructing Resting State Networks from EEG," August, 2018.
54. Craig Edwards, "The Enumeration Problem on Numerical Monoids" (Mathematics), May, 2019.
55. Faranak Aghaei, "Developing Novel Computer-Aided Detection and Diagnosis Systems of Medical Images," November, 2019.
56. Elizabeth Pacheco, "New Simple Representations of Leavitt Path Algebras" (Mathematics), December, 2019.
57. John Price, "From Bagatelles to Capriolen: Eugen d'Albert and his Later Keyboard Works" (Music – DMA), July, 2020.
58. Shajid Islam, "Probe-Based, Quasi-Near-Field Phased Array Calibration," December, 2020.
59. Morteza Heidari, "Applying Novel Machine Learning Technology to Optimize Computer-Aided Detection and Diagnosis of Medical Images," April, 2021.
60. Seyedehnafiseh Mirniaharikandehei, "Developing Novel Quantitative Imaging Analysis Schemes Based on Machine Learning for Cancer Research," April, 2021.
61. Fauzia Ahmed, "Evaluation of Transfemoral Prosthesis Performance Control Using Artificial Neural Network Controllers," April, 2021.
62. David Marvel, "Selected Songs of Nadia Boulanger: Formal Analysis and Adaptation for Brass Chamber Music" (Music – DMA), December, 2021.
63. Ali Khan, "Diffuse Optical Tomography of Spontaneous Brain Fluctuations in Humans" (Biomedical Engineering), April, 2022.
64. Farid Omoumi, "Subjective Evaluation of the In-Line Phase-Sensitive Imaging Systems in Breast Cancer Screening and Diagnosis," July, 2022.
65. Wenwen Li, "Multi-Persistence Homology and Topological Robotics" (Mathematics), April, 2023.
66. Hyeri Kim, "Robust Velocity Unfolding for Weather Radar Based on Convolutional Neural Networks," April, 2023.
67. Precious K. Jatau, "Machine Learning for Classifying Biological Radar Echos with S-Band Polarimetric Radar," November, 2023.

M.S. Supervisions Completed:

1. Santha Parameswaran, "Modulation Domain Forecasting of Nonstationary and Chaotic Time Series," March, 2000 (co-supervised with Dr. Monte Tull).
2. Tanachit Tangsukson, "AM-FM Texture Segmentation," May, 2000.
3. Altaf Ahmed, "Designing a Global IP Routing Strategy," July, 2001 (co-supervised with Dr. Jim Sluss).
4. Igor Ivić, "Demonstration of an Efficient Method for Estimating Spectral Moments," November, 2001.
5. Chee-Hong Gan, "Design of a GIS-Based Traffic Management Center Software Control Platform for Oklahoma Department of Transportation," April, 2002 (co-supervised with Dr. Jim Sluss).
6. Kok-Hoong Chow, "MPLS Modeling and Simulation in Optical Networks," July, 2002 (co-

supervised with Dr. Jim Sluss).

7. Fabrice Ouandji, "Modulation Domain Texture Features for Content-Based Image Retrieval (CBIR)," July, 2004.
8. Ekasit Vorakitolan, "Work Zone Features for Oklahoma's Statewide Intelligent Transportation System," July, 2004.
9. Nantapol Kitiyanan, "AM-FM Fingerprint Reference Point Detection and Matching," November, 2004.
10. Krishnapraveen Suri, "Phase Reconstruction from Multicomponent AM-FM Image Representations," April, 2005.
11. Roy Sivley, "Perfect Reconstruction AM-FM Image Models," March, 2006.
12. Prakash K. Parthasarathy, "Minimum Entropy Based FIR Filter Estimation," December, 2006 (co-supervised with Dr. Victor DeBrunner).
13. Chuong Nguyen, "Dual-Domain Target Tracking," June, 2007.
14. Linda Ouandji, "Advanced Voice and Multimedia Communications System for the ODOT ITS Network," October, 2008.
15. Adrian Campbell, "AM-FM Image Processing Toolbox," December, 2008.
16. Colin Johnston, "Advanced Multi-Channel Dual Domain Constrained Adaptation Particle Filter for Infrared Target Tracking," April, 2009.
17. Anagha Wankhede, "Orientation Selective Perfect Reconstruction Filterbank Toolbox," May, 2010.
18. Basel Kilani, "Statewide Console for Distributed Control of Intelligent Transportation Systems," December, 2010.
19. Sahithi Peddireddy, "Reduction of Beat Type Digital Video Noise Using AM-FM Image Filters," December, 2011.
20. Shawna Ong, "Auxiliary Particle Filter for Modulation Domain Infrared Target Tracking," May, 2012.
21. John R. Jünger III, "The Comparison of Taylor Series and Unscented Transform Kalman Filters," May, 2012 (co-supervised with Dr. S. Lakshmivarahan).
22. Md. Ridwanul Alam, "Tissue Classification-Based Automated Threshold Selection (TCATS) for Segmentation of Bone in Marrow Proliferation Assessments," May, 2015.
23. Jesyca Fuenmayor Bello, "A State Vector Augmentation Method for Including Velocity Information in the Likelihood Function of the SIR Video Target Tracking Filter," July, 2016.
24. Hesham Makhlouf, "Police Electronic Citation Mobile System for Statewide Deployment in Oklahoma," April, 2018.
25. Rodrigo Collao Benitez, "Developing Affordable Smart Solutions for Police Reporting," July, 2018.
26. Brandon Carson, "Automatic Bone Structure Segmentation of Under-Sampled CT/FLT-PET Volumes for HSCT Patients," July, 2021.
27. Favio Hurtado, "Multiclass Bone Segmentation of PET/CT Scans for Automatic SUV Extraction," December, 2021.

M.S. Supervisions in Progress:

- Tristan N. Arian
- Lucas J. Powers

Additional M.S. Committees Served on:

1. Kirankumar Govindarajan, "Implementation of a Wavelet Vocoder," July, 1997.
2. Tod Bussert, "Using Artificial Neural Networks to Improve the Mechanical Signature Analysis Test," December, 1997.
3. Georgios Lezos, "Neural Network and Fuzzy Logic Techniques for Time Series Forecasting," December, 1998.
4. Chetan Anantharaman, "Implementation of Generic Subband/Wavelet Architectures for Image Coding," April, 1999.
5. Mir Sayed Ali, "A CORSIM Traffic Model to Support ITS and DTA in Oklahoma City," February, 2000.
6. Aaron Bansemer, "Retrieval and Analysis of the Electric Field in Thunderstorms" (Meteorology), April, 2000.

7. James Shields, "Design and Implementation of a High-Speed Multiplexer-Based Parallel Multiplier," May, 2000.
8. Rick Pendergraft, "A Performance Evaluation of an Augmented GPS Landing System," September, 2001.
9. Sudhir Rai, "Signal Analysis of Heart Rate Variability Data," December, 2001.
10. Rupa Balan, "Neural Network Modeling of Heart Rate Variability," April, 2002.
11. Anand Mohan, "Low Power and Low Space FIR Filter Design," June, 2002.
12. Alan Harris, "A Fiber Bragg Grating Load Cell," July, 2002.
13. Mahmuda Afroz, "A Design to Measure the Strain of a Large Structure Using Fiber Bragg Gratings," July, 2002.
14. Santiago Rendón, "A Statistical Evaluation of a Protected Service Volume Using an Augmented GPS Landing System," August, 2002.
15. Yuan Chen, "Effects of Digital Watermarking on Digital X-Ray Images," January, 2003.
16. Scott Graham, "A Video System for LAAS/WAAS Data Analysis," May, 2003.
17. Ewa Matusiak, "Uncertainty Principles for Finite Abelian Group and Applications" (interdisciplinary program in Signal Processing, Computational & Applied Mathematics — *SigCAM*), May, 2003.
18. Totrakool Khongsap, "Quantization on a Sphere" (interdisciplinary program in Signal Processing, Computational & Applied Mathematics — *SigCAM*), May, 2003.
19. Minh Quang Ta, "Minimum Entropy Estimation of FIR Filters," May, 2003.
20. Eric Wainright, "Wavelength Diversity in Free-Space Optics to Alleviate Fog Effects," December, 2003.
21. Benjamin Mohr, "Design, Implementation and Testing of a New Curved Path Navigator for LAAS and WAAS," April, 2004.
22. Erik Petrich, "Image Processing Methods for Product Label Identification on Cylindrical Surfaces," July, 2004.
23. John Paul Nguyenkim, "Implementation of a Redundant Binary Co-Processor onto an FPGA for Complex Arithmetic Signal Processing," September, 2004.
24. Anil Babu Chalamalasetti, "Analysis of Radar Signals with Oversampling in Range," September, 2004.
25. Yih-Ru Huang, "Evaluation of a Real Time DGPS (LAAS) Landing System for Missed Approaches and Guided Missed Approaches," September, 2004.
26. Wei Zhang, "Efficient Multiplierless Filter Implementations for Embedded Systems," October, 2004.
27. Ashish Parajuli, "Speech Enhancement Based on Perceptual Wavelet Thresholding and Auditory Masking," December, 2004.
28. Ayodeji Fajebe, "A Software Methodology for Embedded Intelligent Systems," February, 2005.
29. Abderrahmane Bennis, "Division and Square-Root Based on Redundant Binary Numbers," April, 2005.
30. Roland Ferenczhalmy, "Analysis of Adsorption and Desorption Kinetics of Volatile Analytes Using Mid-Infrared Laser Absorption Spectroscopy," August, 2005.
31. Deepak V. Bhogaraju, "Entropy Uncertainty in FIR Filter Implementations," September, 2005.
32. Benjamin Blevins, "Stereoscopic Tracking of Approaching Aircraft," December, 2005.
33. Brian Birk, "The Design and Implementation of a Fault Tolerant LAAS Base Station," May, 2006.
34. Nicholas Mould, "Reconfigurable Computing Architectures: Dynamic and Steering Vector Methods," May 2006.
35. Rodolfo Salas, "Control Electronics for Laser Absorption Spectroscopy," May, 2006.
36. Matthew S. Falk, "Developing a New Airway Criteria Using Aircraft's Required Navigational Performance," December, 2006.
37. Hieu Thai, "System Identification of Bridges Under a Moving Load and Implementation of the Bridge Monitoring System," March, 2007.
38. Kevin Ford, "Computer Hardware for Vibration Mitigation and Monitoring," March, 2007.
39. Molly Donovan, "Performance Evaluation of a Phase Contrast X-Ray Imaging Prototype System," June, 2007.

40. Kyle Sparger, “Roadside Data Collection and Monitoring using GPRS Cellular Network,” July, 2007.
41. Patrick Macklin, “Development and Integration of a Power Management Board for the Collision Risk Model,” September, 2007.
42. Adriana Sofia Otero, “Adaptive Localized Route Maintenance Mechanism to Improve Performance of VoIP Over Ad Hoc Networks,” April, 2010.
43. Jasper Staab, “Binary Mimicry in the Executable File,” May, 2010.
44. Yasmin Jahir, “AODVH: Multipath Routing Protocol for Hybrid Nodes in Disaster Area Wireless Network (DAWN),” July, 2010.
45. Jordan Kuehn, “FPGA Real-Time Motion Control and Automation of Biped Robot,” December, 2010.
46. Jacob Henderson, “Application of Magnetic Field Distortion Characteristics for use in Autonomous Location Detection,” May, 2011.
47. Sonya Wolff, “Pre-Execution: An Elegant Approach to the Memory Wall,” July, 2011.
48. Feng Nai, “Wind Turbine Clutter Mitigation for Weather Radars,” November, 2011.
49. Vasiliy Mayer, “Redefining Airway Constraints Based on En Route Flight Tests,” December, 2011.
50. Sina Asadallahi, “Distributed Adaptive Backoff Reservation Protocol for 802.11 Wireless Networks,” July, 2012.
51. Nathan McVay, “Sensitivity Analysis of Long Term Bias Error in the Global Positioning System,” October, 2012.
52. Timothy Wilson, “Remote Desktop Capability for Labview Programs on an Android Platform,” November, 2012.
53. Muhammad Usman Ghani, “Quantitative Analysis of Contrast to Noise Ratio Using a Phase Contrast X-Ray Imaging Prototype,” October, 2013.
54. Marcin Rutkowski, “Glitching-Aware Model Characterization Methodology for Power Estimation Techniques in CMOS Arithmetic Structures,” May, 2014.
55. Kevin Windham, “Subsampling Effects on Range Migration Correction in SAR Imaging,” July, 2014.
56. Milad Javadi, “Identification of Simultaneously Congested Transmission Lines in Power Market,” December, 2014.
57. Nastaran Emaminejad, “Exploring the new CT Image Features to Improve Lung Cancer Diagnosis and Treatment Efficacy Assessment,” April, 2015.
58. Faranak Aghaei, “Computer-Aided Breast MR Image Feature Analysis for Prediction of Tumor Response to Chemotherapy,” April, 2015.
59. David Schvartzman Cohenca, “Weather Radar Spatio-Temporal Saliency (WR-STS),” June, 2015.
60. Lesya Borowska, “Experiments on Electromagnetic Leakage from Laptops,” May, 2017.
61. Jiaxi Zhu, “Low-Cost, Software Defined FMCW Radar for Observations of Drones,” May, 2017.
62. Johnny O’Keeffe, “Neuroimaging Features of Adults with and without Amnestic Mild Cognitive Impairment,” July, 2017.
63. Lucia R. Fitzmorris, “Learning Assisted Decoupled Software Pipelining (LA-DSWP),” April, 2018.
64. Precious Jatau, “A Fuzzy Logic Algorithm for Separating Radar Echos from Birds and Insect at S-band,” July, 2018.
65. Bradley Gregory, “Objective Characterization of In-Line Phase Contrast X-Ray Imaging Prototype Using a Mid-Energy Beam,” July, 2019.
66. Brian Carlton, “Nonlinear Amplifier Amplitude Modulation Distortion Mitigation Techniques,” April, 2022.
67. Trey T. Crump, “An Analysis of the Information Content of Radar Detection,” July, 2022.
68. Roman A. Munoz, “A Study on Diffusion Probabilistic Models for Image Generation,” November, 2023.
69. Aminat B. Oyeleke, “Distributed Matrix Analysis and Computation Over Networks,” April, 2024.
70. Erfan Seifi, “Developing an Algorithm Integrating Voice and Imaging Analysis to Recognize Facial Features and Deficiencies After Oral Surgery,” April, 2024.

71. Summer Edwards, “Multimodal Imaging Approaches Using Functional Near-Infrared Spectroscopy, Electroencephalography and Transcranial Magnetic Stimulation” (Biomedical Engineering), July, 2024.

► **Externally Funded Grants and Contracts:**

1. R.D. Barnes (PI) and J.P. Havlicek, “Police Automated Records Information System FY24,” State of Oklahoma, Highway Safety Office, **$121,614**, 10/1/23-9/30/24. OU Pink Sheet Credit: 50% ($60,807).
2. J.P. Havlicek (PI), M. Atiquzzaman, and R.D. Barnes, “SAFE-T: Statewide Analysis for Engineering & Technology,” State of Oklahoma, Department of Transportation, **$117,867**, 1/1/23-6/30/24. OU Pink Sheet Credit: 34% ($40,075).
3. J.P. Havlicek (PI) and R.D. Barnes, “Oklahoma Intelligent Transportation System CY 2024,” State of Oklahoma, Department of Transportation, **$381,862**, 12/1/23-6/30/24. OU Pink Sheet Credit: 50% ($190,931).
4. S.M. Schaefer, S. Hampton, F. Cianfarani, J. Havlicek, and R. Barnes, “Oklahoma State Housing Assessment,” Oklahoma Housing Finance Authority, **$925,487**, 1/1/23-12/31/27. OU Pink Sheet Credit: 20% ($185,097).
5. R.D. Barnes (PI) and J.P. Havlicek, “Police Automated Records Information Systems FY23,” State of Oklahoma, Highway Safety Office, **$97,427**, 10/1/22-9/30/23. OU Pink Sheet Credit: 50% ($48,714).
6. J.P. Havlicek (PI) and R.D. Barnes, “Oklahoma Intelligent Transportation System FFY 2023,” State of Oklahoma, Department of Transportation, **$500,299**, 7/1/22-10/31/24. OU Pink Sheet Credit: 50% ($250,150).
7. R.D. Barnes (PI) and J.P. Havlicek, “PARIS FY22,” State of Oklahoma, Highway Safety Office, **$194,855**, 10/1/21-9/30/22. OU Pink Sheet Credit: 50% ($97,427).
8. J.P. Havlicek (PI), M. Atiquzzaman, and R.D. Barnes, “SAFE-T: Statewide Analysis for Engineering & Technology,” State of Oklahoma, Department of Transportation, **$117,867**, 10/1/21-9/30/22. OU Pink Sheet Credit: 34% ($40,075).
9. J.P. Havlicek (PI) and R.D. Barnes, “Oklahoma Intelligent Transportation System FFY 2022,” State of Oklahoma, Department of Transportation, **$770,000**, 7/1/21-12/31/22. OU Pink Sheet Credit: 50% ($385,000).
10. K.M. Williams (PI), J.L. Holter Chakrabarty, Y. Yanik, S.K. Vesely, and J.P. Havlicek, et al., including Emory University/Children’s Hospital of Atlanta, OU Health Sciences Center, OU Norman Campus, and University of Michigan, “Multi-institutional Prospective Pilot **R**esearch of Imaging and blood biomarker **EV**aluation of **E**ngraftment after **AL**ogeneic Hematopoietic Stem Cell Transplantation in Children and Adults (REVEAL),” NIH Title: “Imaging and Blood Biomarkers to Predict Graft Failure after HSCT,” US Dept. Health and Human Services, National Institutes of Health, National Heart, Lung, and Blood Institute. **$2,913,713**. 6/1/20-7/31/25. Prime contractor: Emory University/CHOA; subcontract awarded to OU Norman Campus: $34,476 for Period 4 (8/01/23-7/31/24); $34,478 for Period 3 (8/01/22-7/31/23); $40,562 for Period 2 (8/01/21-7/31/22); $40,562 for Period 1 (8/15/20-8/31/21). Subcontract PI: J.P. Havlicek. OU Pink Sheet Credit: 100% ($150,078).
11. J.P. Havlicek (PI) and R.D. Barnes, “PARIS D360 Database Update,” State of Oklahoma, Department of Public Safety, **$28,246**, 5/7/21-6/30/21. OU Pink Sheet Credit: 50% ($14,123).
12. J.P. Havlicek (PI), M. Atiquzzaman, and R.D. Barnes, “SAFE-T: Statewide Analysis for Engineering & Technology,” State of Oklahoma, Department of Transportation, **$117,867**, 10/1/20-9/30/21. OU Pink Sheet Credit: 34% ($40,075).
13. R.D. Barnes (PI) and J.P. Havlicek, “Drive Oklahoma: Oklahoma’s Intelligent Transportation System FY21,” State of Oklahoma, Department of Transportation, **$700,000**, 7/1/20-12/31/21. OU Pink Sheet Credit: 50% ($350,000).
14. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, “Electronic Police Records 2020 (Supplement),” State of Oklahoma, Highway Safety Office, **$112,140**, 10/1/19-9/30/20. OU Pink Sheet Credit: 40% ($44,856).
15. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, “Electronic Police Records 2020,” State of Oklahoma, Highway Safety Office, **$112,140**, 10/1/19-9/30/20. OU Pink Sheet Credit: 40% ($44,856).

16. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "SAFE-T: Statewide Analysis for Engineering & Technology," State of Oklahoma, Highway Safety Office, **$98,830**, 10/1/19-9/30/20. OU Pink Sheet Credit: 20% ($19,766).
17. R.D. Barnes (PI) and J.P. Havlicek, "Intelligent Transportation Systems 2020," State of Oklahoma, Department of Transportation, **$700,000**, 7/1/19-6/30/20. OU Pink Sheet Credit: 50% ($350,000).
18. J.P. Havlicek (PI), R.D. Barnes, and M. Atiquzzaman, "Oklahoma Impaired Driver Database," State of Oklahoma, Highway Safety Office, **$36,297**, 10/1/18-9/30/19. OU Pink Sheet Credit: 34% ($12,341).
19. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "Police Automated Records Information System (PARIS)," State of Oklahoma, Highway Safety Office, **$199,088**, 10/1/18-9/30/19. OU Pink Sheet Credit: 40% ($79,635).
20. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "SAFE-T: Statewide Analysis for Engineering & Technology," State of Oklahoma, Highway Safety Office, **$98,196**, 10/1/18-9/30/19. OU Pink Sheet Credit: 20% ($19,639).
21. R.D. Barnes (PI) and J.P. Havlicek, "Traffic Incident Management (TIM) Report Analysis," State of Oklahoma, Department of Transportation, **$60,000**, 9/27/18-7/31/19. OU Pink Sheet Credit: 50% ($30,000).
22. J.L. Holter Chakrabarty (PI), J.P. Havlicek, and S.K. Vesely, "FLT Imaging to Detect Relapse in Leukemia Patients Following Transplantation," US Dept. Health and Human Services, National Institutes of Health. **$49,799**. 9/1/18-6/30/19. Prime contractor: University of Oklahoma Health Sciences Center; subcontract awarded to OU Norman Campus: $13,218. Subcontract PI: J.P. Havlicek. OU Pink Sheet Credit: 100% ($13,218).
23. R.D. Barnes (PI) and J.P. Havlicek, "Hardware and Software for Next Generation ITS," State of Oklahoma, Department of Transportation, **$700,000**, 7/1/18-6/30/19. OU Pink Sheet Credit: 50% ($350,000).
24. R.D. Barnes (PI) and J.P. Havlicek, "Expanding PARIS+ to Regional Police Agencies," Southern Plains Transportation Center, **$39,738**, 10/15/17-5/30/18. OU Pink Sheet Credit: 50% ($19,869).
25. J.P. Havlicek (PI), R.D. Barnes, and M. Atiquzzaman, "OU Impaired Driver Database Hosting and Support," State of Oklahoma, Highway Safety Office, **$36,000**, 10/1/17-9/30/18. OU Pink Sheet Credit: 34% ($12,240).
26. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "PARIS Software Development and Integration," State of Oklahoma, Highway Safety Office, **$200,000**, 10/1/17-9/30/18. OU Pink Sheet Credit: 40% ($80,000).
27. M. Atiquzzaman (PI), R.D. Barnes, and J.P. Havlicek, "SAFE-T Data Improvement Project," State of Oklahoma, Highway Safety Office, **$85,920**, 10/1/17-9/30/18. OU Pink Sheet Credit: 20% ($17,184).
28. R.D. Barnes (PI) and J.P. Havlicek, "Engineering and Design of Intelligent Transportation System," State of Oklahoma, Department of Transportation, **$635,000**, 10/1/16-9/30/17. OU Pink Sheet Credit: 50% ($317,500).
29. J.P. Havlicek (PI), R.D. Barnes, and M. Atiquzzaman, "Operation of Oklahoma Statewide Impaired Driver Database," State of Oklahoma, Highway Safety Office, **$39,811**, 1/1/17-9/30/17. OU Pink Sheet Credit: 34% ($13,536).
30. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "Police Automated Records Information System and Collision Reporting System," State of Oklahoma, Highway Safety Office, **$233,977**, 10/1/16-9/30/17. OU Pink Sheet Credit: 40% ($93,591).
31. M. Atiquzzaman (PI), R.D. Barnes, and J.P. Havlicek, "Statewide Analysis for Engineering and Technology," State of Oklahoma, Highway Safety Office, **$88,877**, 10/1/16-9/30/17. OU Pink Sheet Credit: 20% ($17,775).
32. R.D. Barnes (PI) and J.P. Havlicek, "Intelligent Transportation System Engineering and Design," State of Oklahoma, Department of Transportation, **$668,819**, 10/1/15-9/30/16. OU Pink Sheet Credit: 50% ($334,410).
33. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "Police Automated Records Information System and DUI Tracking Database," State of Oklahoma, Highway Safety Office, **$379,128**, 10/1/15-9/30/16. OU Pink Sheet Credit: 40% ($151,651).
34. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "Statewide Analysis for Engineering

and Technology," State of Oklahoma, Highway Safety Office, **$88,877**, 10/1/15-9/30/16. OU Pink Sheet Credit: 20% ($17,775).

35. L. Ding (PI), J.P. Havlicek, and D.T. Liu, "RII Track-2 FEC: Innovative, Broadly Accessible Tools for Brain Imaging, Decoding and Modulation," National Science Foundation, **$1,357,173** (subcontract to the University of Rhode Island; prime contract award amount: $5,999,853), 8/1/15-7/31/19. OU Pink Sheet Credit: 33% ($447,867).
36. J.P. Havlicek (PI) and R.D. Barnes, "Oklahoma Bureau of Narcotics and Dangerous Drugs (OBNDD) PARIS System," Oklahoma Bureau of Narcotics and Dangerous Drugs, **$7,201**, 5/1/15-12/31/15. OU Pink Sheet Credit: 50% ($3,601).
37. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "OU Intelligent Transportation Systems FY15," Oklahoma Department of Transportation, **$400,000**, 10/1/14-9/30/15. OU Pink Sheet Credit: 40% ($160,000).
38. J.P. Havlicek (PI), M. Atiquzzaman, and R.D. Barnes, "SAFE-T System Expert System Functionality: Option III," Oklahoma Department of Transportation, **$232,127**, 1/1/15-12/31/16. OU Pink Sheet Credit: 34% ($78,923).
39. M.B. Yeary (PI), R.D. Palmer, and J.P. Havlicek, "System and Software Support for CGI (Supplement)," CGI Federal, Inc., **$34,224**, 11/7/14-3/8/15, OU Pink Sheet Credit: 25% ($8,556).
40. M. Atiquzzaman (PI), R.D. Barnes, and J.P. Havlicek, "Enhancing Driver Safety During Severe Weather Conditions," Southern Plains Transportation Center, **$199,998**, 7/1/14-6/30/16. OU Pink Sheet Credit: 30% ($59,999).
41. R.D. Barnes (PI), J.P. Havlicek, and M. Atiquzzaman, "Police Automated Records and Information System," State of Oklahoma, Highway Safety Office, **$368,500**, 10/1/14-9/30/15. OU Pink Sheet Credit: 40% ($147,400).
42. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$74,825**, 10/1/14-9/30/15. OU Pink Sheet Credit: 20% ($14,965).
43. Joseph P. Havlicek, "PET Image Analysis Using a Novel Radioisotope Fluorothymidine for Identification of Bone Marrow Repopulation following Myeloablative Transplantation: Supplement," University of Oklahoma Health Sciences Center, Stephenson Cancer Center, **$16,966**, 10/1/14-4/30/15. OU Pink Sheet Credit: 100% ($16,966).
44. J.L. Holter Chakrabarty (PI), J.P. Havlicek, and S.K. Vesely, "PET Image Analysis Using a Novel Radioisotope Fluorothymidine for Identification of Bone Marrow Repopulation following Myeloablative Transplantation," Oklahoma Shared Clinical and Translational Resources pilot grant funded by US Dept. Health and Human Services, National Institutes of Health. **$50,000**. 1/8/14-6/30/14. Prime contractor: University of Oklahoma Health Sciences Center; subcontract awarded to OU Norman Campus: $25,788. Subcontract PI: J.P. Havlicek. OU Pink Sheet Credit: 100% ($25,788).
45. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, and J.P. Havlicek, "ITS System Engineering and Integration," Oklahoma Department of Transportation, **$344,000**. 10/1/13-9/30/14. OU Pink Sheet Credit: 35% ($120,400).
46. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "University of Oklahoma SAFE-T Project," State of Oklahoma, Highway Safety Office, **$174,000**, 10/1/13-9/30/14. OU Pink Sheet Credit: 20% ($34,800).
47. R.D. Barnes (PI), M. Atiquzzaman, and J.P. Havlicek, "OU TraCS/PARIS Project," State of Oklahoma, Highway Safety Office, **$238,000**, 10/1/13-9/30/14. OU Pink Sheet Credit: 40% ($95,200).
48. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, and J.P. Havlicek, "ITS System Engineering and Integration," Oklahoma Department of Transportation, **$344,000**. 10/1/12-9/30/13. OU Pink Sheet Credit: 37% ($127,280).
49. R.D. Barnes (PI), M. Atiquzzaman, and J.P. Havlicek, "Police Automated Records Information System," State of Oklahoma, Highway Safety Office, **$155,000**, 10/1/12-9/30/13. OU Pink Sheet Credit: 40% ($62,000).
50. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$55,000**, 10/1/12-9/30/13. OU Pink Sheet Credit: 20% ($11,000).
51. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, and J.P. Havlicek, "ITS System Engi-

neering and Integration," Oklahoma Department of Transportation, **$312,150**. 10/1/11-9/30/12. OU Pink Sheet Credit: 35% ($109,253).

52. J.P. Havlicek (PI) and R.D. Barnes, "GPS Location Data Enhancement in Electronic Traffic Records," Oklahoma Transportation Center, **$100,000**, 10/1/11-12/31/12. OU Pink Sheet Credit: 50% ($50,000).
53. R.D. Barnes (PI) and J.P. Havlicek, "Fatality Analysis Reporting System and Roadway Inventory Correlation," Oklahoma Transportation Center, **$100,000**, 10/1/11-12/31/12. OU Pink Sheet Credit: 50% ($50,000).
54. R.D. Barnes (PI), M. Atiquzzaman, and J.P. Havlicek, "OU Software Development & Integration Project," State of Oklahoma, Highway Safety Office, **$220,000**, 10/1/11-9/30/12. OU Pink Sheet Credit: 40% ($88,000).
55. M. Atiquzzaman (PI), J.P. Havlicek, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$54,660**, 10/1/11-9/30/12. OU Pink Sheet Credit: 20% ($10,932).
56. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, J. Basara, and M.P. Tull, "A Mobile Intelligent Transportation System (ITS) Platform," Oklahoma Transportation Center, **$341,352**. 1/1/11-2/29/12. OU Pink Sheet Credit: 20% ($68,270).
57. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "ITS System Engineering Crash Diagram Supplement," Oklahoma Department of Transportation, **$17,645**. 10/1/10-6/30/12. OU Pink Sheet Credit: 25% ($4,411).
58. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "Intelligent Transportation System (ITS) Engineering and Integration Services," Oklahoma Department of Transportation, **$341,000**. 10/1/10-6/30/12. OU Pink Sheet Credit: 25% ($82,250).
59. R.D. Barnes (PI), M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "OU Software Development & Integration Project," State of Oklahoma, Highway Safety Office, **$234,573**, 10/1/10-9/30/11. OU Pink Sheet Credit: 30% ($70,372).
60. M. Atiquzzaman (PI), J.P. Havlicek, M.P. Tull, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$64,879**, 10/1/10-9/30/11. OU Pink Sheet Credit: 22% ($14,273).
61. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "Intelligent Transportation System (ITS) Engineering and Integration Services," Oklahoma Department of Transportation, **$220,000**. 10/1/09-9/30/10. OU Pink Sheet Credit: 25% ($55,000).
62. R.D. Barnes (PI), M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "OU Software Development & Integration Project," State of Oklahoma, Highway Safety Office, **$150,000**, 10/1/09-9/30/10. OU Pink Sheet Credit: 30% ($45,000).
63. M. Atiquzzaman (PI), J.P. Havlicek, M.P. Tull, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$55,000**, 10/1/09-9/30/10 OU Pink Sheet Credit: 22% ($12,100).
64. R.D. Barnes (PI), James J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "Roadway Weather Information System and Automatic Vehicle Location (AVL) Coordination," Oklahoma Transportation Center (OTC), **$145,433**, 6/1/08-5/31/10. OU Pink Sheet Credit: 20% ($29,087).
65. R.D. Barnes (PI), James J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and M.P. Tull, "Roadway Weather Information System and Automatic Vehicle Location (AVL) Coordination (Matching Funds)," Oklahoma Department of Transportation, **$55,000**, 6/1/08-5/31/10. OU Pink Sheet Credit: 20% ($11,000).
66. R.D. Barnes (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, M.P. Tull, and H. Refai, "ITS System Engineering and Integration Supplement," Oklahoma Department of Transportation, **$33,000**. 11/1/08-10/31/09. OU Pink Sheet Credit: 10% ($3,300).
67. R.D. Barnes (PI), J.J. Sluss, Jr., J.P. Havlicek, and M.P. Tull, "Intelligent Transportation System (ITS) Engineering and Integration Services," Oklahoma Department of Transportation, **$155,000**. 10/1/08-9/30/09. OU Pink Sheet Credit: 25% ($38,750).
68. J.P. Havlicek (PI) and G. Fan, "Multiple Domain Particle Filters for Integrated Tracking and Recognition in IR Imagery," Department of Defense, Army Research Office, **$474,000**, 7/1/08-6/30/11. OU Pink Sheet Credit: 100% ($474,000).
69. R.D. Barnes (PI), J.P. Havlicek, and M.P. Tull, "OU Software Development & Integration Project," State of Oklahoma, Highway Safety Office, **$150,000**, 10/1/08-9/30/09. OU Pink

Sheet Credit: 33% ($49,500).

70. R.D. Barnes (PI), J.P. Havlicek, and M.P. Tull, "OU Software Development & Integration Project Supplement," State of Oklahoma, Highway Safety Office, **$5,000**, 7/15/09-9/30/09. OU Pink Sheet Credit: 33% ($1,650).
71. M. Atiquzzaman (PI), J.P. Havlicek, M.P. Tull, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis System," State of Oklahoma, Highway Safety Office, **$54,745**, 10/1/08-9/30/09. OU Pink Sheet Credit: 22% ($12,044).
72. J.P. Havlicek (PI), M.P. Tull, and R.D. Barnes, "OU Software Development & Integration Project (TraCS) Supplement," State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/07-9/30/08. OU Pink Sheet Credit: 40% ($20,000).
73. J.P. Havlicek (PI), M.P. Tull, and R.D. Barnes, "OHP Troop S Civil Assessment System," State of Oklahoma, Department of Public Safety, **$50,000**, 4/15/08-4/14/09. OU Pink Sheet Credit: 34% ($17,000).
74. J.P. Havlicek (PI), M.P. Tull, and R.D. Barnes, "Automated Driver License Testing System," State of Oklahoma, Department of Public Safety, **$108,035**, 10/1/07-9/30/08. OU Pink Sheet Credit: 40% ($43,214).
75. J.P. Havlicek (PI), M.P. Tull, and R.D. Barnes, "OU Software Development & Integration Project (TraCS)," State of Oklahoma, Highway Safety Office, **$150,000**, 10/1/07-9/30/08. OU Pink Sheet Credit: 40% ($60,000).
76. J.P. Havlicek (PI), M. Atiquzzaman, M.P. Tull, and R.D. Barnes, "University of Oklahoma Crash Reporting and Analysis System (SAFE-T)," State of Oklahoma, Highway Safety Office, **$53,171**, 10/1/07-9/30/08. OU Pink Sheet Credit: 30% ($15,951).
77. M.P. Tull (PI), J.J. Sluss, Jr., J.P. Havlicek, and R.D. Barnes, "ITS System Engineering and Integration Services to be Provided by the OU ITS Lab as Part of the Oklahoma Transportation Center, FY 2008," Oklahoma Department of Transportation, **$219,976**, 10/1/07-9/30/08. OU Pink Sheet Credit: 30% ($65,993).
78. J.P. Havlicek (PI), J.J. Sluss, Jr., and M.P. Tull, "TraCS: Traffic and Criminal Software (continuation of OU Mobile Data Collection System Pilot Project)," State of Oklahoma, Highway Safety Office, **$182,467**, 10/1/06-9/30/07. OU Pink Sheet Credit: 40% ($72,987).
79. M.P. Tull (PI), J.J. Sluss, Jr., and J.P. Havlicek, "ITS System Engineering and Integration," Oklahoma Department of Transportation, **$208,000**, 10/1/06-9/30/07. OU Pink Sheet Credit: 45% ($93,600).
80. M.P. Tull (PI), J.J. Sluss, Jr., M. Atiquzzaman, J.P. Havlicek, and T. Runolfsson, "Advanced Voice and Multimedia Communications System for the ODOT ITS Network," State of Oklahoma, Department of Transportation (Oklahoma Transportation Center), **$81,000**, 10/1/06-9/30/07. OU Pink Sheet Credit: 30% ($24,300).
81. J.P. Havlicek (PI), J.J. Sluss, Jr., M. Atiquzzaman, M.P. Tull, and T. Runolfsson, "University of Oklahoma Crash Reporting and Analysis," State of Oklahoma, Highway Safety Office, **$50,000**. 10/1/06-9/30/07. OU Pink Sheet Credit: 25% ($12,500).
82. J.P. Havlicek (PI), J.J. Sluss, Jr., M.P. Tull, and T. Runolfsson, "OU Mobile Data Collection System Pilot Project (Continuation)," State of Oklahoma, Highway Safety Office, **$45,751**, 10/1/06-9/30/07. OU Pink Sheet Credit: 25% ($11,438).
83. J.J. Sluss, Jr. (PI), J.P. Havlicek, M.P. Tull, and T. Runolfsson, "Truck Weight Enforcement Using Advanced Weigh-in-Motion Systems," Oklahoma Transportation Center, **$78,223**, 5/1/06-4/30/07. OU Pink Sheet Credit: 25% ($19,556).
84. T. Landers (PI), with 19 Co-PI's including J.P. Havlicek, "Inter-Modal Containerized Freight Security: FY 06 Allocation," Oklahoma Department of Transportation, **$2,083,151**, 7/1/06-6/30/07. OU Pink Sheet Credit: 6% ($124,989).
85. J.P. Havlicek (PI), J.J. Sluss, Jr., M.P. Tull, and T. Runolfsson, "OU Mobile Data Collection Project (CDL)," State of Oklahoma, Highway Safety Office, **$105,277**, 3/1/06-9/30/06. OU Pink Sheet Credit: 25% ($26,319).
86. J.J. Sluss, Jr. (PI), J.P. Havlicek, M.P. Tull, and T. Runolfsson, "Intelligent Transportation System (ITS) Engineering and Integration Services," Oklahoma Department of Transportation, **$225,000**, 10/1/05-9/30/06. OU Pink Sheet Credit: 25% ($56,250).
87. J.P. Havlicek (PI), M.P. Tull, and J.J. Sluss, Jr., "SAFE-T: State-Wide Analysis for Enhancing Transportation," State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/05-9/30/06. OU Pink Sheet Credit: 33% ($16,500).

88. R. Mc Pherson (PI), J.J. Sluss, Jr., J. Snow, J.P. Havlicek, J. Basara, M. Wolfinbarger, and C. Friebrich, "Clarus Weather System Design," Mixon/Hill, Inc. (prime contractor; flow-through from U.S. DoT – FHWA), **$411,769**, 6/1/05-2/28/07. OU Pink Sheet Credit: 10% ($41,177).
89. J.P. Havlicek (PI), J.J. Sluss, Jr., M.P. Tull, and T. Runolfsson, "OU Mobile Data Collection System Pilot Project," State of Oklahoma, Highway Safety Office, **$208,000**, 4/25/05-3/31/06. OU Pink Sheet Credit: 25% ($52,000).
90. J.P. Havlicek (PI) and J.J. Sluss, Jr., "University of Oklahoma Crash Reporting and Analysis System (FMCSA Supplement)," State of Oklahoma, Highway Safety Office, **$75,000**, 1/1/05-9/30/05. OU Pink Sheet Credit: 50% ($37,500).
91. J.P. Havlicek (PI) and J.J. Sluss, Jr., "University of Oklahoma Crash Reporting and Analysis System," State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/04-9/30/05. OU Pink Sheet Credit: 50% ($25,000).
92. J.J. Sluss, Jr. (PI) and J.P. Havlicek, "Intelligent Transportation System Engineering and Integration Services," Oklahoma Department of Transportation, **$222,356**, 10/1/04-9/30/05. OU Pink Sheet Credit: 50% ($111,178).
93. J.P. Havlicek (PI) and G. Fan, "Integrated Detection, Tracking, Classification, and Learning for Dual-Band Infrared Imagery," Department of Defense, Army Research Office, **$465,897**, 7/1/04-6/30/07. OU Pink Sheet Credit: 100% ($465,897).
94. J.J. Sluss, Jr. (PI) and J.P. Havlicek, "Design and Integration of ITS (Intelligent Transportation Systems) Project in Oklahoma," Oklahoma Department of Transportation, **$164,500**, 10/1/03-9/30/04. OU Pink Sheet Credit: 50% ($82,250).
95. J.P. Havlicek (PI) and J.J. Sluss, Jr., "A Statewide Crash Reporting and Analysis System," State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/03-9/30/04. OU Pink Sheet Credit: 50% ($25,000).
96. J.J. Sluss, Jr. (PI) and J.P. Havlicek, "Design and Integration of ITS (Intelligent Transportation Systems) Project in Oklahoma (Year 0)," Oklahoma Department of Transportation, **$41,000**, 7/1/03-9/30/03. OU Pink Sheet Credit: 50% ($20,500).
97. J.J. Sluss, Jr. (PI), J.P. Havlicek, and S. Radhakrishnan, "Development of a 511 Traveler Information Program Deployment Plan for Oklahoma," Oklahoma Department of Transportation, **$50,000**, 1/1/03-6/30/04. OU Pink Sheet Credit: 33% ($16,500).
98. J.P. Havlicek (PI) and J.J. Sluss, Jr., "A Statewide Accident Reporting and Analysis System," Oklahoma Transportation Center, **$30,000**, 1/1/03-9/30/03. OU Pink Sheet Credit: 50% ($15,000).
99. J.P. Havlicek (PI) and J.J. Sluss, Jr., "ITS Features for Enhanced Highway Safety in Work Zones," State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/02-9/30/03. OU Pink Sheet Credit: 50% ($25,000).
100. J.J. Sluss, Jr. (PI) and J.P. Havlicek, "Design and Integration of ITS (Intelligent Transportation Systems) Project in Oklahoma," Oklahoma Department of Transportation, **$145,000**, 6/18/02-9/30/03. OU Pink Sheet Credit: 50% ($72,500).
101. J.E. Fagan (PI), J.P. Havlicek, and G.R. Schaumburg, "Determining the Required Navigational Performance of the GPS, WAAS, and LAAS Systems for Precision Simple and Complex Approaches and the Development of Models for the Prediction of the Operational Performance of these Navigation Systems," Federal Aviation Administration, **$545,000**, 5/1/02-6/30/03. OU Pink Sheet Credit: 30% ($163,500).
102. J.J. Sluss, Jr. (PI), J.P. Havlicek, and S. Radhakrishnan, "Oklahoma Statewide ITS Strategic Plan and ITS/CVO Plan," Federal Highway Administration/Oklahoma Department of Transportation subcontract; prime contractor: P.B. Farradyne, Inc., **$32,692**, 3/1/02-3/31/03. OU Pink Sheet Credit: 33% ($10,788).
103. J.E. Fagan (PI), J.P. Havlicek, and G.R. Schaumburg, "Determining the Required Navigational Performance of the GPS, WAAS, and LAAS Systems for Precision Simple and Complex Approaches and the Development of Models for the Prediction of the Operational Performance of these Navigation Systems in a Wide Variety of Aircraft (Global Positioning System Wide and Local Area Augmentation System)," Federal Aviation Administration, **$240,000**, 2/1/00-6/30/02. OU Pink Sheet Credit: 30% ($72,000).
104. J.P. Havlicek (PI), "Decentralized Image Retrieval for Education \(DIRECT\)," National Science Foundation subcontract; prime contractor: University of Virginia, PI: S.T. Acton,

$63,171, 1/1/02-12/31/03. OU Pink Sheet Credit: 100% ($63,171).

105. J.P. Havlicek (PI) and J.J. Sluss, Jr., “System Development and Testing for ITS,” State of Oklahoma, Highway Safety Office, **$50,000**, 10/1/01-9/30/02. OU Pink Sheet Credit: 50% ($25,000).
106. M.P. Tull (PI), J.P. Havlicek, J.J. Sluss, Jr., and J. Cheung, “Artificial Intelligence Based Forecasting,” Lucent Technologies, **$39,943**, 1/1/01-5/31/01. OU Pink Sheet Credit: 37% ($14,779).
107. P. Pulat (PI), J.J. Sluss, Jr., J.P. Havlicek, S. Radhakrishnan, and S.A. Moses, “Design and Evaluation of a Hierarchical Highway Network Structure and a Decision Support System with Surveillance Information to Enhance Business Partnerships in the E-Marketplace,” National Science Foundation, **$100,001**, 8/15/00-8/14/01. OU Pink Sheet Credit: 20% ($20,000).
108. J.P. Havlicek (PI) and J.J. Sluss, Jr., “System Development, Integration, and Component Testing for Oklahoma City’s Intelligent Transportation System,” State of Oklahoma Highway Safety Office, **$50,000**, 10/1/00-9/30/01. OU Pink Sheet Credit: 50% ($25,000).
109. M.P. Tull (PI), J.J. Sluss, Jr., J.P. Havlicek, and S. Radhakrishnan, “Artificial Intelligence Based Inventory and Forecasting,” Lucent Technologies, **$248,428**, 1/1/00-12/31/00. OU Pink Sheet Credit: 33% ($81,981).
110. J.P. Havlicek (PI) and J.J. Sluss, Jr., “System Development, Integration, and Component Testing for Oklahoma City’s Intelligent Transportation System,” State of Oklahoma Highway Safety Office, **$50,001**, 10/1/99-9/30/00. OU Pink Sheet Credit: 50% ($25,001).
111. J.J. Sluss, Jr. (PI) and J.P. Havlicek, “An Intelligent Transportation System for Oklahoma City,” State of Oklahoma Department of Transportation, **$80,000**, 7/1/99-8/15/00. OU Pink Sheet Credit: 50% ($40,000).
112. J.E. Fagan (PI), J.P. Havlicek, J.J. Sluss, Jr., and G.R. Schaumburg, “A Proposal for Research to Determine the Required Navigational Performance of the GPS, WAAS, and LAAS Systems for Simple and Complex Approaches and the Development of Models for the Prediction of the Operational Performance of these Navigation Systems in a Wide Variety of Aircraft,” Federal Aviation Administration, **$866,300**, 4/16/99-6/30/01. OU Pink Sheet Credit: 30% ($259,890).
113. M.P. Tull (PI), J.J. Sluss, Jr., and J.P. Havlicek, “Extended Artificial Intelligence Based Forecasting and Inventory Planning Models,” Lucent Technologies, **$232,754**, 1/1/99-12/31/99. OU Pink Sheet Credit: 33.3% ($77,507).
114. J.J. Sluss, Jr. (PI) and J.P. Havlicek, “System Architecture Design for Oklahoma City’s Intelligent Transportation System,” State of Oklahoma Department of Transportation, **$49,776**, 5/13/98-10/31/98. OU Pink Sheet Credit: 50% ($24,888).
115. M.P. Tull (PI), J.J. Sluss, Jr., J.P. Havlicek, V.E. DeBrunner, L.S. DeBrunner, S.C. Lee, and S. Radhakrishnan, “Artificial Intelligence Based Forecasting and Inventory Planning Models,” Lucent Technologies, **$229,298**, 11/1/97-12/31/98. OU Pink Sheet Credit: 21% ($48,153).

- **Total External Funding: $27,565,129**
- **Total Attributable to J.P. Havlicek (OU Pink Sheet Credit): $9,211,089**
- **Internally Funded Grants:**

1. H. Liu (PI), J. Holter Chakrabarty, and J.P. Havlicek, “Development of a Predictive Imaging Model for Prediction of Relapse Following Allogeneic Bone Marrow Transplantation,” University of Oklahoma Bioengineering Center seed funding for interdisciplinary Research, $47,172. 12/15/13-12/14/14.
2. P.S. Harvey, R.W. Floyd, L. Gruenwald, J.P. Havlicek, Y. Li, and J.-S. Pei, “Safer School Buildings for Wind and Earthquakes: A Multidisciplinary Approach,” University of Oklahoma College of Engineering seed funding for interdisciplinary Research, $10,000. 6/1/15-5/31/16.

Total Internal Funding: $57,172

► **Invited Lectures:**

1. J.P. Havlicek, "Designing Perceptually-Based Image Filters in the Modulation Domain," School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, May 3, 2011.
2. J.P. Havlicek, "Designing Perceptually-Based Image Filters in the Modulation Domain," Dept. Automation, Shanghai Jiao Tong University, Shanghai, China, September 25, 2010.
3. J.P. Havlicek, "Infrared Target Tracking in the Modulation Domain," Dept. Electrical & Computer Engineering, University of New Mexico, Albuquerque, NM, March 28, 2008.
4. J.P. Havlicek, "Multidimensional AM-FM Models with Image Processing Applications," School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, November 22, 2002.
5. J.P. Havlicek, "Image Texture Retrieval Using Joint Amplitude-Frequency Modulation Models," Dept. Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, July 22, 2002.
6. J.P. Havlicek, "Modulation Models for Image Processing and Machine Vision," Dept. Electrical Engineering, The Ohio State University, Columbus, OH, March 31, 1998.
7. J.P. Havlicek, "Modulation Models for Image Processing and Machine Vision," School of Electrical & Computer Engineering, Oklahoma State University, Stillwater, OK, March 26, 1998.
8. J.P. Havlicek, "Wideband Frequency Excursions in Multicomponent AM-FM Models," School of Electrical & Computer Engineering Colloquium Seminar Series, the University of Oklahoma, Norman, OK, September 18, 1997.
9. J.P. Havlicek, "AM-FM Image Models," IEEE Oklahoma City Section meeting, Oklahoma City, OK, March 20, 1997.
10. J.P. Havlicek, "AM-FM Image Models," School of Electrical & Computer Engineering, the University of Oklahoma, Norman, OK, July 18, 1996.
11. J.P. Havlicek, "AM-FM Image Analysis," Dept. Electrical Engineering, University of Washington, Seattle, WA, May 14, 1996.
12. J.P. Havlicek, "AM-FM Image Analysis," Dept. Electrical Engineering, The Pennsylvania State University, University Park, PA, April 22, 1996.

► **Conference Presentations Without Proceedings:**

1. H. Soltani, M. Muraleetharan, and J. Havlicek, "Effects of ground improvement zone dimensions on the modal characteristics of pile founded structures," *Engineering Mechanics Institute Conference 2019*, California Institute of Technology, Pasadena, CA, Jun. 18-21, 2019.
2. K.M. Williams, J.L. Holter Chakrabarty, L. Lindenberg, S. Adler, J. Gea-Banacloche, B. Blacklock-Schuver, F.T. Hakim, D.D. Hickstein, J.N. Kochenderfer, J. Wilder, T. Chinn, K. Kurdziel, S.M. Steinberg, H. Khuu, F.I. Lin, D.H. Fowler, D. Halverson, D.N. Avila, G. Selby, T.N. Taylor, J. Mann, J. Hsu, R.B. Epstein, S.L. Anderson, C.T. Nguyen, J. Havlicek, S. Li, T. Pham, T. Kraus, S.K. Vesely, PhD, S.Z. Pavletic, C.M. Bollard, P. Choyke, and R.E. Gress, "FLT imaging reveals kinetics and biology of engraftment after myeloablative HSCT," *56'th American Society of Hematology (ASH) Annual Meeting and Exposition*, San Francisco, CA, Dec. 6-9, 2014.
3. K.M. Williams, J.L. Holter, L. Lindenberg, S. Adler, J. Gea-Banacloche, B. Blacklock-Schuver, F. Hakim, D. Hickstein, J. Kochenderfer, J. Wilder, T. Chinn, K. Kurdziel, S. Steinberg, H. Khuu, D. Fowler, F.I. Lin, D. Halverson, D.N. Avila, G. Selby, S.L. Anderson, C.T. Nguyen, J.P. Havlicek, T.N. Taylor, J. Mann, J. Hsu, R. Epstein, S.K. Vesely, S. Li, T. Kraus, T. Pham, S.Z. Pavletic, C. Bollard, P. Choyke, and R.E. Gress, "Novel imaging reveals early engraftment and stem cell homing," *NIH Blood and Marrow Transplant (BMT) Consortium: 20th Anniversary Allogeneic Stem Cell Transplant at NIH Conference and Celebration*, Washington, DC, Sep. 11-12, 2014.

Publications

A. Archival Journal Papers:

1. J.P. Wright, P.F. Tang, J.-S. Pei, F. Gay-Balmaz, and J.P. Havlicek, "On computing the analytic-signal backbone of the unforced harmonic oscillator," *J. Comput. Appl. Math.*, vol. 385, 16 pp., Article 113206, Mar. 15, 2021, published online Sep. 22, 2020.
2. E.D. Ross, S.S. Gupta, A.M. Adnan, T.L. Holden, J. Havlicek, and S. Radhakrishnan, "Neurophysiology of spontaneous facial expressions: II. Motor control of the right and left face is partially independent in adults," *Cortex*, vol. 111, pp. 164-182, Feb. 2019, published online Nov. 10, 2018.
3. K.M. Williams, J. Holter-Chakrabarty, L. Lindenberg, Q. Dong, S.K. Vesely, C.T. Nguyen, J.P. Havlicek, K. Kurdziel, J. Gea-Banacloche, F.I. Lin, D.N. Avila, G. Selby, C.G. Kanakry, S. Li, T. Scordino, S. Adler, C.M. Bollard, P. Choyke, and R.E. Gress, "Imaging of subclinical haemopoiesis after stem-cell transplantation in patients with haematological malignancies: A prospective pilot study," *The Lancet Haematology*, vol. 5, no. 1, pp. e44-e52, Jan. 2018, published online Dec. 13, 2017.
4. E.D. Ross, S.S. Gupta, A.M. Adnan, T.L. Holden, J. Havlicek, and S. Radhakrishnan, "Neurophysiology of spontaneous facial expressions: I. Motor control of the upper and lower face is behaviorally independent in adults," *Cortex*, vol. 76, pp. 28-42, Mar. 2016.
5. L. Yu, G. Fan, J. Gong, and J.P. Havlicek, "Joint infrared target recognition and segmentation using a shape manifold-aware level set," *Sensors*, special issue on sensors in new road vehicles, vol. 15, no. 5, pp. 10118-10145, Apr. 2015.
6. J. Gong, G. Fan, L. Yu, J.P. Havlicek, D. Chen, and N. Fan, "Joint target tracking, recognition and segmentation for infrared imagery using a shape manifold-based level set," *Sensors*, vol. 14, no. 6, pp. 10124-10145, Jun. 2014.
7. J. Gong, G. Fan, L. Yu, J.P. Havlicek, D. Chen, and N. Fan, "Joint view-identity manifold for infrared target tracking and recognition," *Comput. Vision, Image Understand.*, vol. 118, pp. 211-224, Jan. 2014.
8. N. Mould and J.P. Havlicek, "Neighborhood-level learning techniques for nonparametric scene models," *Signal, Image, Video Process.*, vol. 8, no. 6, pp. 1015-1029, Sep. 2014. doi: 10.1007/s11760-013-0571-x.
9. V. Venkataraman, G. Fan, J.P. Havlicek, X. Fan, Y. Zhai, and M. Yeary, "Adaptive Kalman filtering for histogram-based appearance learning in infrared imagery," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4622-4635, Nov. 2012.
10. V. Venkataraman, G. Fan, L. Yu, X. Zhang, W. Liu, and J.P. Havlicek, "Automated target tracking and recognition using coupled view and identity manifolds for shape representation," *EURASIP J. Advances Signal Process.*, v. 2011, doi: 10.1186/1687-6180-2011-124, 17 pp., Dec. 7, 2011.
11. P.C. Tay, J.P. Havlicek, S.T. Acton, and J.A. Hossack, "Properties of the magnitude terms of orthogonal scaling functions," *Digital Signal Process.*, vol. 20, no. 5, pp. 1330-1340, Sep. 2010.
12. X. Fan, G. Fan, and J.P. Havlicek, "Generative model for maneuvering target tracking," *IEEE Trans. Aerospace, Elect. Sys.*, vol. 46, no. 2, pp. 635-655, Apr. 2010.
13. Y. Zhai, M. Yeary, J. Havlicek, and G. Fan, "A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems," *IEEE Trans. Instrumentation, Measurement*, vol. 57, no. 10, pp. 2377-2387, Oct. 2008.
14. V. DeBrunner, J.P. Havlicek, T. Przebinda, and M. Özaydın, "Entropy-based uncertainty measures for $L^2(\mathbb{R}^n)$, $\ell^2(\mathbb{Z})$, and $\ell^2(\mathbb{Z}/N\mathbb{Z})$ with a Hirschman optimal transform for $\ell^2(\mathbb{Z}/N\mathbb{Z})$," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2690-2699, Aug. 2005.
15. S.T. Acton, D.P. Mukherjee, J.P. Havlicek, and A.C. Bovik, "Oriented texture completion by AM-FM reaction-diffusion," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 885-896, Jun. 2001.
16. J.P. Havlicek and P.C. Tay, "Determination of the number of texture segments using wavelets," *Electron. J. Diff. Eqns.*, vol. Conf. 07, 2001, pp. 61-70, `http://ejde.math.swt.edu/conf-proc/07/toc.html`. Also published in *Proc. 16th Conf. Appl. Math.*, Edmond, OK, Feb. 23-24, 2001, pp. 61-70.
17. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Multidimensional quasi-eigenfunction approximations and multicomponent AM-FM models," *IEEE Trans. Image Process.*, vol. 9, no. 2,

pp. 227-242, Feb. 2000.

18. J.P. Havlicek, D.S. Harding, and A.C. Bovik, “Multicomponent multidimensional signals,” *Multidimensional Syst. and Signal Process.*, vol. 9, no. 4, *invited paper*, pp. 391-398, Oct. 1998.
19. J.P. Havlicek, “The evolution of modern texture processing,” *Elektrik, Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 5, no. 1, special issue on image processing, pp. 1-28, 1997.
20. A.C. Bovik, J.P. Havlicek, M.D. Desai, and D.S. Harding, “Limits on discrete modulated signals,” *IEEE Trans. Signal Process.*, vol. 45, no. 4, pp. 867-879, Apr. 1997.
21. J.P. Havlicek, D.S. Harding, and A.C. Bovik, “The mutlicomponent AM-FM image representation,” *IEEE Trans. Image Process.*, special issue on nonlinear image processing, vol. 5, no. 6, pp. 1094-1100, Jun. 1996.
22. J.P. Havlicek, J.C. McKeeman, and P.W. Remaklus, “Networks of low-earth orbit store-and-forward satellites,” *IEEE Trans. Aerospace and Elect. Sys.*, vol. 31, no. 2, pp. 543-554, Apr. 1995.
23. J.P. Havlicek, G.R. Katz, and J.C. McKeeman, “Even length median filters in optimal signal processing,” *Electron. Letters*, vol. 28, no. 13, pp. 1258-1260, Jun. 18, 1992.
24. J.P. Havlicek, K.A. Sarkady, G.R. Katz, and J.C. McKeeman, “Fast efficient median filters with even length windows,” *Electron. Letters*, vol. 26, no. 20, pp. 1736-1737, Sep. 27, 1990.

B. Book Chapters:

1. O. Alkhouli, V. DeBrunner, and J. Havlicek, “Hirschman Optimal Transform (HOT) DFT Block LMS Algorithm,” in *Adaptive Filtering*, L. Garcia, ed., ISBN: 978-953-307-158-9, In-Tech, Sep. 2011, pp. 135-152.
2. G. Fan, V. Venkataraman, X. Fan, and J.P. Havlicek, “Appearance Learning for Infrared Tracking with Occlusion Handling,” in *Machine Vision Beyond Visible Spectrum*, R.I. Hammoud, G. Fan, R.W. McMillan, and K. Ikeuchi, ed., Augmented Vision and Reality Series, Springer, Berlin, Heidelberg, Jun. 2011, pp. 33-64.
3. G. Fan, X. Fan, V. Venkataraman, and J.P. Havlicek, “Vehicle Tracking and Recognition,” in *Intelligent Video Surveillance: Systems and Technology*, Y. Ma and G. Qian, ed., CRC Press - Taylor & Francis Group, Oxford, Dec. 2009, pp. 149-179.
4. G. Fan, V. Venkataraman, L. Tang, and J.P. Havlicek, “On Boosted and Adaptive Particle Filters for Affine-Invariant Target Tracking in Infrared Imagery,” in *Augmented Vision Perception in Infrared: Algorithms and Applied Systems* (Advances in Pattern Recognition), R.I. Hammoud, ed., Springer-Verlag, London, 2009, pp. 441-466.
5. J.P. Havlicek, P.C. Tay, and A.C. Bovik, “AM-FM Image Models: Fundamental Techniques and Emerging Trends,” in *Handbook of Image and Video Processing, 2 ed.*, A.C. Bovik, ed., Elsevier Academic Press, Burlington, MA, 2005, pp. 377-395.
6. M.S. Pattichis, J.P. Havlicek, S.T. Acton, and A.C. Bovik, “Multidimensional AM-FM Models with Image Processing Applications,” in *Advances in Image Processing and Understanding: A Festschrift for Thomas S. Huang*, A.C. Bovik, C.W. Chen, and D. Goldgof, ed., Series in Machine Perception and Artificial Intelligence, vol. 52, World Scientific Publishing, Singapore, 2002, pp. 277-305.
7. M.P. Tull, J.J. Sluss, Jr., and J.P. Havlicek, “Product Demand Forecasting Using Genetic Programming,” in *Fuzzy Engineering Expert Systems with Neural Network Applications,* A.B. Badiru and J.Y. Cheung, John Wiley & Sons, New York, 2002, pp. 274-281.
8. J.P. Havlicek and A.C. Bovik, “Image Modulation Models,” in *Handbook of Image and Video Processing,* A.C. Bovik, ed., Communications, Networking, and Multimedia Series by Academic Press, San Diego, 2000, pp. 305-316.
9. J.P. Havlicek, A.C. Bovik, and D. Chen, “AM-FM Image Modeling and Gabor Analysis,” in *Visual Information Representation, Communication, and Image Processing,* C.W. Chen and Y. Zhang, ed., Optical Engineering Series by Marcel Dekker, Inc., New York, 1999, pp. 343-385.

C. Refereed Conference Papers:

1. B.D. Carson, F. Hurtado, J.P. Havlicek, L.J. Powers, L. Lindenberg, D.N. Avila, C.G. Kanakry, P. Choyke, K. Kurdziel, P. Eclarinal, K.M. Williams, and J. Holter Chakrabarty, "Approximate vertebral body instance segmentation by PET-CT fusion for assessment after hematopoietic stem cell transplantation," in *Proc. 23'rd IEEE Int'l. Conf. Bioinformatics and Bioengineering*, Dayton, OH, Dec. 4-6, 2023, pp. 62-69.
2. J.P. Havlicek, T.N. Arian, H. Soltani, T. Przebinda, and M. Özaydın, "A preliminary case for Hirschman transform video coding," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Mar. 29-31, 2020, pp. 104-107.
3. H. Soltani and J.P. Havlicek, "An improved subspace identification of a nonlinear soil-structure system," in *Proc. 12'th Int'l. Workshop Structural Health Monitoring*, Stanford, CA, Sep. 10-12, 2019, 8 pp.
4. H. Soltani, K.K. Muraleetharan, and J.P. Havlicek, "Characterizing dynamics of pile-supported structures using system and subsystem identification methods," in *Proc. 11'th Int'l. Workshop Structural Health Monitoring*, Stanford, CA, Sep. 12-14, 2017, 8 pp.
5. C.T. Nguyen, J.P. Havlicek, Q. Duong, S. Vesely, R. Gress, L. Lindenberg, P. Choyke, J. Holter Chakrabarty, and K. Williams, "An automatic 3D CT/PET segmentation framework for bone marrow proliferation assessment," in *Proc. IEEE Int'l. Conf. Image Process.*, Phoenix, AZ, Sep. 25-28, 2016, pp. 4126-4130.
6. C.T. Nguyen, J.P. Havlicek, J. Holter Chakrabarty, Q. Duong, and S.K. Vesely, "Towards automatic 3D bone marrow segmentation," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Mar. 6-8, 2016, pp. 9-12.
7. J.C. Fuenmayor Bello and J.P. Havlicek, "A state vector augmentation technique for incorporating indirect velocity information into the likelihood function of the SIR video target tracking filter," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Mar. 6-8, 2016, pp. 109-112.
8. C.T. Nguyen and J.P. Havlicek, "Color to grayscale image conversion using modulation domain quadratic programming," in *Proc. IEEE Int'l. Conf. Image Process.*, Québec City, Canada, Sep. 27-30, 2015, pp. 4580-4584.
9. J. Holter Chakrabarty, C.T. Nguyen, J. Havlicek, S.K. Vesely, L. Lindenberg, S. Adler, B.A.J. Blacklock-Schuver, K. Kurdziel, F. Lin, D. Avila, G.B. Selby, J. Mann, J. Hsu, A. Chai, R.B. Epstein, N.Q.T. Duong, S. Li, T. Kraus, S. Hopps, T. Pham, C.M. Bollard, P. Choyke, R. Gress, and K.M. Williams, "Novel imaging analysis of the marrow compartment after myeloablative HSCT reveals the kinetics and degree of myeloablation and cell Recovery," *Proc. 2015 American Society for Blood and Marrow Transplantation (BMT) Tandem Meetings*, San Diego, CA, Feb. 11-15, 2015. Published in *Biology of Blood and Marrow Transplantation*, vol. 21, no. 2, pp. S319-S320, Feb. 2015.
10. C.T. Nguyen, J.P. Havlicek, G. Fan, J.T. Caulfield, and M.S. Pattichis, "Robust dual-band MWIR/LWIR infrared target tracking," in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2-5, 2014, pp. 78-83.
11. C.T. Nguyen and J.P. Havlicek, "On the amplitude and phase computation of the AM-FM image model," in *Proc. IEEE Int'l. Conf. Image Process.*, Paris, France, Oct. 27-30, 2014, pp. 4318-4322.
12. C.T. Nguyen and J.P. Havlicek, "Linear adaptive infrared image fusion," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, San Diego, CA, Apr. 6-8, 2014, pp. 117-120.
13. P.A. Campbell and J.P. Havlicek, "Frequency guided phase unwrapping for improved AM-FM reconstruction," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, San Diego, CA, Apr. 6-8, 2014, pp. 129-132.
14. E. Vorakitolan, J.P. Havlicek, R.D. Barnes, and A.R. Stevenson, "Simple, effective rate control for video distribution in heterogeneous intelligent transportation system networks," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, San Diego, CA, Apr. 6-8, 2014, pp. 37-40.
15. J. Gong, G. Fan, J.P. Havlicek, N. Fan, and D. Chen, "Infrared target tracking, recognition and segmentation using shape-aware level set," in *Proc. IEEE Int'l. Conf. Image Process.*, Melbourne, Australia, Sep. 15-18, 2013, pp. 3283-3287.
16. L. Yu, G. Fan, J. Gong, and J.P. Havlicek, "Simultaneous target recognition, segmentation, and pose estimation," in *Proc. IEEE Int'l. Conf. Image Process.*, Melbourne, Australia,

Sep. 15-18, 2013, pp. 2655-2659.

17. P.A. Campbell, J.P. Havlicek, A.R. Stevenson, and R.D. Barnes, "Realization of ITS applications through mapping technologies: a survey of advanced traveler information systems," in *Proc. 15th Int'l. IEEE Conf. Intel. Transportation Syst.*, Anchorage, AK, Sep. 16-19, 2012, pp. 788-795.
18. C.T. Nguyen and J.P. Havlicek, "AM-FM models, partial Hilbert transform, and the monogenic signal," in *Proc. IEEE Int'l. Conf. Image Process.*, Orlando, FL, Sep. 30 - Oct. 3, 2012, pp. 2337-2340.
19. N. Mould and J.P. Havlicek, "A stochastic learning algorithm for pixel-level background models," in *Proc. IEEE Int'l. Conf. Image Process.*, Orlando, FL, Sep. 30 - Oct. 3, 2012, pp. 1233-1236.
20. J. Gong, G. Fan, L. Yu, J.P. Havlicek, and D. Chen, "Joint view-identity manifold for target tracking and recognition," in *Proc. IEEE Int'l. Conf. Image Process.*, Orlando, FL, Sep. 30 - Oct. 3, 2012, pp. 1357-1360.
21. N. Mould and J.P. Havlicek, "A conservative scene model update policy," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Ap. 22-24, 2012, pp. 145-148.
22. C.T. Nguyen and J.P. Havlicek, "Gradient-based texture cartoon decomposition," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Apr. 22-24, 2012, pp. 85-88.
23. S. Peddireddy, N. Mould, and J.P. Havlicek, "AM-FM picture carrier beat type noise filters," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Apr. 22-24, 2012, pp. 209-212.
24. C.T. Nguyen, J.D. Williams, J.P. Havlicek, and M. Özaydın, "FM processing with generalized amplitude & phase: application to modulation domain geometric image transformations," in *Proc. IEEE Int'l. Conf. Image Process.*, Brussels, Belgium, Sep. 11-14, 2011, pp. 81-84.
25. E. Vorakitolan, N. Mould, J.P. Havlicek, R.D. Barnes, and A.R. Stevenson, "A low-cost distributed instrumentation system for monitoring, identifying and diagnosing irregular patterns of behavior in critical ITS components," in *Proc. 14th Int'l. IEEE Conf. Intel. Transportation Syst.*, Washington, DC, Oct. 5-7, 2011, pp. 2045-2050.
26. P.A. Campbell, J.R. Junger, J.P. Havlicek, A.R. Stevenson, and R.D. Barnes, "Pathfinder: Oklahoma's advanced traveler information system," in *Proc. 14th Int'l. IEEE Conf. Intel. Transportation Syst.*, Washington, DC, Oct. 5-7, 2011, pp. 402-407.
27. B.H. Kilani, P.A. Campbell, J.P. Havlicek, M.P. Tull, A.R. Stevenson, and R.D. Barnes, "Peer-based communication for a distributed, virtual Traffic Management Center," in *Proc. 14th Int'l. IEEE Conf. Intel. Transportation Syst.*, Washington, DC, Oct. 5-7, 2011, pp. 2068-2073.
28. E. Vorakitolan, J.P. Havlicek, M. Atiquzzaman, and R.D. Barnes, "Exploiting trunked radio to support ITS network expansion and redundancy," in *Proc. 22nd IEEE Int'l. Symp. Personal, Indoor, Mobile Radio Comm.*, Toronto, Canada, Sep. 11-14, 2011, pp. 761-766.
29. V. Venkataraman, G. Fan, L. Yu, X. Zhang, W. Liu, and J.P. Havlicek, "Joint target tracking and recognition using view and identity manifolds," in *Proc. 8th Joint IEEE Int'l. Workshop Object Tracking, Class., in and Beyond the Visible Spectrum*, in conjunction with the *2011 IEEE Computer Soc. Conf. Comput. Vision, Pattern Recog.*, Colorado Springs, CO, Jun. 20-25, 2011, pp. 33-40.
30. J.T. Caulfield and J.P. Havlicek, "Advanced image processing techniques for extracting regions of interest using multimode IR processing," in *Proc. 39th IEEE Applied Imagery Pattern Recog. Workshop*, Washington, DC, Oct. 13-15, 2010, pp. 238-241.
31. C.T. Nguyen and J.P. Havlicek, "Modulation domain texture decomposition," in *Proc. IEEE Int'l. Conf. Image Process.*, Hong Kong, Sep. 26-29, 2010, pp. 2741-2744.
32. M. Shook, J. Junger, N. Mould, and J.P. Havlicek, "Quantifying infrared target signature evolution using AM-FM features," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Austin, TX, May 23-25, 2010, pp. 189-192.
33. C.T. Nguyen and J.P. Havlicek, "Coherent texture decomposition using AM-FM model," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Austin, TX, May 23-25, 2010, pp. 81-84.
34. C.T. Nguyen, P.A. Campbell, and J.P. Havlicek, "FM filters for modulation domain image processing," in *Proc. IEEE Int'l. Conf. Image Process.*, Cairo, Egypt, Nov. 7-11, 2009,

pp. 3973-3976.

35. C.M. Johnston, N.A. Mould, and J.P. Havlicek, "Multichannel dual domain infrared target tracking for highly evolutionary target signatures," in *Proc. IEEE Int'l. Conf. Image Process.*, Cairo, Egypt, Nov. 7-11, 2009, pp. 4117-4120.
36. B.H. Kilani, E. Vorakitolan, J.P. Havlicek, M.P. Tull, and A.R. Stevenson, "Distributed ITS control and the Oklahoma virtual TMC," in *Proc. 12th Int'l. IEEE Conf. Intel. Transportation Syst.*, St. Louis, MO, Oct. 4-7, 2009, pp. 785-790.
37. J.R. Junger, J.P. Havlicek, R.D. Barnes, and M.P. Tull, "Prediction aggregation of remote traffic microwave sensors speed and volume data," in *Proc. 12th Int'l. IEEE Conf. Intel. Transportation Syst.*, St. Louis, MO, Oct. 3-7, 2009, pp. 672-678.
38. C.M. Johnston, N. Mould, J.P. Havlicek, and G. Fan, "Dual domain auxiliary particle filter with integrated target signature update," in *Proc. 6th Joint IEEE Int'l. Workshop Object Tracking, Class., in and Beyond the Visible Spectrum*, in conjunction with the *2009 IEEE Computer Soc. Conf. Comput. Vision, Pattern Recog.*, Miami, FL, Jun. 20-25, 2009, pp. 54-59.
39. V. Venkataraman, G. Fan, X. Fan, and J.P. Havlicek, "Appearance learning by adaptive Kalman filters for FLIR tracking," in *Proc. 6th Joint IEEE Int'l. Workshop Object Tracking, Class., in and Beyond the Visible Spectrum*, in conjunction with the *2009 IEEE Computer Soc. Conf. Comput. Vision, Pattern Recog.*, Miami, FL, Jun. 20-25, 2009, pp. 46-53.
40. C.T. Nguyen and J.P. Havlicek, "AM-FM image filters," in *Proc. IEEE Int'l. Conf. Image Process.*, San Diego, CA, Oct. 12-15, 2008, pp. 789-792.
41. N.A. Mould, C.T. Nguyen, C.M. Johnston, and J.P. Havlicek, "Online consistency checking for AM-FM target tracks," in *Proc. SPIE/IS&T Conf. Computational Imaging VI*, C.A. Bouman, E.L. Miller, and I. Pollak, ed., *SPIE* **v. 6814**, Jan. 26-31, 2008, pp. 681413-1 – 681413-12.
42. C.T. Nguyen, R.A. Sivley, and J.P. Havlicek, "First results in perceptually-based AM-FM image filtering," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Mar. 24-26, 2008, pp. 77-80.
43. N.A. Mould, C.T. Nguyen, and J.P. Havlicek, "Infrared target tracking with AM-FM consistency checks," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Mar. 24-26, 2008, pp. 5-8.
44. Y. Wang, L.S. DeBrunner, J.P. Havlicek, and D. Zhou, "Iterative radix-8 multiplier structure based on a novel real-time CSD recoding," in *Proc. 41st IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 4-7, 2007, pp. 977-981.
45. H. Thai, V. DeBrunner, L.S. DeBrunner, J. Havlicek, K. Mish, K. Ford, and A. Medda, "Deterministic-stochastic subspace identification for bridges," in *Proc. 2007 IEEE Statistical Signal Process. Workshop*, Madison, WI, Aug. 26-29, 2007, pp. 749-753.
46. C.T. Nguyen, J. Havlicek, and M. Yeary, "Modulation domain template tracking," in *Proc. 4th Joint IEEE Int'l. Workshop Object Tracking, Class., in and Beyond the Visible Spectrum*, in conjunction with the *2007 IEEE Computer Soc. Conf. Comput. Vision, Pattern Recog.*, Minneapolis, MN, Jun. 17-22, 2007 (8 pages).
47. R.A. Sivley and J.P. Havlicek, "Perfect reconstruction AM-FM image models," in *Proc. IEEE Int'l. Conf. Image Process.*, Atlanta, GA, Oct. 8-11, 2006, pp. 2125-2128.
48. C.T. Nguyen and J.P. Havlicek, "Modulation domain features for discriminating infrared targets and backgrounds," in *Proc. IEEE Int'l. Conf. Image Process.*, Atlanta, GA, Oct. 8-11, 2006, pp. 3245-3248.
49. Y. Zhai, M. Yeary, J. -C. Noyer, J. Havlicek, S. Nemati, and P. Lanvin, "Visual target tracking using improved and computationally efficient particle filtering," in *Proc. IEEE Int'l. Conf. Image Process.*, Atlanta, GA, Oct. 8-11, 2006, pp. 1757-1760.
50. G. Fan, V. Venkataraman, L. Tang, and J. Havlicek, "A comparative study of boosted and adaptive particle filters for affine-invariant target detection and tracking," in *Proc. 3rd Joint IEEE Int'l. Workshop Object Tracking, Class., in and Beyond the Visible Spectrum*, in conjunction with the *2006 IEEE Computer Soc. Conf. Comput. Vision, Pattern Recog.*, New York, NY, Jun. 17-22, 2006, p. 138 (8 pages).
51. J.P. Havlicek, C.T. Nguyen, and M. Yeary, "Modulation domain infrared target models," in *Targets and Backgrounds XII: Characterization and Representation*, W.R. Watkins and D. Clement, ed., *SPIE* **v. 6239**, May 4, 2006, pp. 62390D-1 – 62390D-11.

52. J.P. Havlicek, C.T. Nguyen, G. Fan, and V.B. Venkataraman, "Integration of a dual-band IR data acquisition system using low-cost PV320 cameras," in *Infrared Technology and Applications XXXII*, B.F. Andersen, G.F. Fulop, and P.R. Norton, ed., *SPIE* **v. 6206**, May 18, 2006, pp. 62061U-1 – 62061U-9.
53. K. Suri and J.P. Havlicek, "Phase algorithm for blocking artifact reduction in reconstructions from analysis-only AM-FM models," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Denver, CO, Mar. 26-28, 2006, pp. 6-10.
54. R.A. Sivley and J.P. Havlicek, "A spline-based framework for perfect reconstruction AM-FM models," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Denver, CO, Mar. 26-28, 2006, pp. 198-202.
55. Y. Wang, L.S. DeBrunner, J.P. Havlicek, and D. Zhou, "Signal exclusive adaptive average filter for impulse noise suppression," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Denver, CO, Mar. 26-28, 2006, pp. 51-55.
56. R.C. Huck, J.P. Havlicek, J.J. Sluss, Jr., and A.R. Stevenson, "A low-cost distributed control architecture for intelligent transportation systems deployment in the State of Oklahoma," in *Proc. 8th Int'l. IEEE Conf. Intel. Transportation Syst.*, Vienna, Austria, Sep. 13-16, 2005, pp. 919-924.
57. P.C. Tay and J.P. Havlicek, "JPEG 2000 scalar quantization using an optimally frequency localized modulated lapped transform," in *Proc. IEEE Int'l. Conf. Image Process.*, Genoa, Italy, Sept. 11-14, 2005, vol. I, pp. 93-96.
58. R.A. Sivley and J.P. Havlicek, "Multidimensional phase unwrapping for consistent APF estimation," in *Proc. IEEE Int'l. Conf. Image Process.*, Genoa, Italy, Sept. 11-14, 2005, vol. II, pp. 458-461.
59. Y. Zhai, M. Yeary, J.P. Havlicek, J.-C. Noyer, and P. Lanvin, "Visual tracking using sequential importance sampling with a state partition technique," in *Proc. IEEE Int'l. Conf. Image Process.*, Genoa, Italy, Sept. 11-14, 2005, vol. III, pp. 876-879.
60. Y. Zhai, M. Yeary, V. DeBrunner, J.P. Havlicek, and O. Alkhouli, "Image restoration using a hybrid combination of particle filtering and wavelet denoising," in *Proc. IEEE Int'l. Conf. Image Process.*, Genoa, Italy, Sept. 11-14, 2005, vol. II, pp. 790-793.
61. N. Kitiyanan and J.P. Havlicek, "Modulation domain reference point detection for fingerprint recognition," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Lake Tahoe, NV, Mar. 28-30, 2004, pp. 147-151.
62. D. Zhou, V. DeBrunner, and J.P. Havlicek, "A spatially selective filter based on the undecimated wavelet transform that is robust to noise estimation error," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Lake Tahoe, NV, Mar. 28-30, 2004, pp. 162-166.
63. P.C. Tay and J.P. Havlicek, "Frequency implementation of discrete wavelet transforms," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Lake Tahoe, NV, Mar. 28-30, 2004, pp. 167-171.
64. H. Wen, J. Havlicek, and J. Fagan, "B-value research for FAA LAAS station integrity and fault detection," in *Proc. ION 2004 Nat. Tech. Meeting*, San Diego, CA, Jan. 26-28, 2004, pp. 817-822.
65. G. Mu, J.P. Havlicek, and J. Fagan, "A new position domain algorithm to improve WAAS availability and continuity while maintaining integrity," in *Proc. ION 2004 Nat. Tech. Meeting*, San Diego, CA, Jan. 26-28, 2004, pp. 145-156.
66. J.P. Havlicek, J. Tang, S.T. Acton, R. Antonucci, and F.N. Ouandji, "Modulation domain texture retrieval for CBIR in digital libraries," in *Proc. 37th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 9-12, 2003, vol. 2, pp. 1580-1584.
67. P.C. Tay and J.P. Havlicek, "Joint uncertainty measures for maximally decimated M-channel prime factor cascaded wavelet filter banks," in *Proc. IEEE Int'l. Conf. Image Process.*, Barcelona, Spain, Sept. 14-17, 2003, vol. 1, pp. 1033-1036.
68. P. Tay, J.P. Havlicek, and V. DeBrunner, "Discrete wavelet transform with optimal joint localization for determining the number of image texture segments," in *Proc. IEEE Int'l. Conf. Image Process.*, Rochester, NY, Sept. 22-25, 2002, vol. 3, pp. 281-284.
69. M.J. Lipsey and J.P. Havlicek, "On the Teager-Kaiser energy operator 'low frequency error'", in *Proc. 45th IEEE Midwest Symp. Circuits, Syst.*, Tulsa, OK, Aug. 4-7, 2002, vol. 3, pp. 53-56.
70. P. Tay, J.P. Havlicek, and V. DeBrunner, "Image watermarking using wavelets," in *Proc. 45th*

IEEE Int'l. Symp. Circuits, Syst., Tulsa, OK, Aug. 4-7, 2002, vol. 3, pp. 258-261.

71. P. Tay, J.P. Havlicek, and V. DeBrunner, "A novel translation and modulation invariant discrete-discrete uncertainty measure," in *Proc. IEEE Int'l. Conf. Acoust., Speech, Signal Process.*, Orlando, FL, May 13-17, 2002, vol. 2, pp. 1461-1464.
72. P. Tay, J.P. Havlicek, and V. DeBrunner, "A wavelet filter bank which minimizes a novel translation invariant discrete uncertainty measure," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Santa Fe, NM, Apr. 7-9, 2002, pp. 173-177.
73. T.B. Yap, J.P. Havlicek, and V. DeBrunner, "Bayesian segmentation of AM-FM texture images," in *Proc. 35th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 4-7, 2001, vol. 2, pp. 1156-1160, (*invited paper*).
74. C. Zhong and J.P. Havlicek, "LDPC codes for robust transmission of images over wireless channels," in *Proc. 35th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 4-7, 2001, vol. 1, pp. 797-800.
75. N. Ray, J. Havlicek, S. Acton, and M. Pattichis, "Active contour segmentation guided by AM-FM dominant component analysis," in *Proc. IEEE Int'l. Conf. Image Process.*, Thessaloniki, Greece, Oct. 7-10, 2001, vol. 1, pp. 78-81.
76. T.B. Yap, T. Tangsukson, P.C. Tay, N.D. Mamuya, and J.P. Havlicek, "Unsupervised texture segmentation using dominant image modulations," in *Proc. 34th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 29-31, 2000, vol. 2, pp. 911-915 (*invited paper*).
77. V. DeBrunner, L. DeBrunner, J. Havlicek, and M. Tull, "Introduction to digital signals and filtering: implementing DSP First at the University of Oklahoma," in *Proc. 9th IEEE DSP Workshop*, Hunt, TX, Oct. 15-18, 2000, 6 pp., (*invited paper*).
78. J.P. Havlicek, P.C. Tay, and J.J. Sluss, Jr., "Signals and systems: a consistent, unified approach," in *Proc. 30th ASEE/IEEE Frontiers in Education Conf.*, Kansas City, MO, Oct. 18-21, 2000, , vol. 2 pp. F4E/1-F4E/6.
79. T. Tangsukson and J.P. Havlicek, "AM-FM image segmentation," *Proc. IEEE Int'l. Conf. Image Process.*, Vancouver, Canada, Sep. 10-13, 2000, vol. 2, pp. 104-107.
80. V. DeBrunner, M. Özaydın, T. Przebinda, and J. Havlicek, "The optimal solutions to the continuous- and discrete-time versions of the Hirschman uncertainty principle," in *Proc. IEEE Int'l. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 5-9, 2000, vol. 1, pp. 81-84.
81. T. Tangsukson and J.P. Havlicek, "Modulation domain image segmentation," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Austin, TX, Apr. 2-4, 2000, pp. 46-50.
82. S.D. Sloan, R.W. Saw, J.J. Sluss, M.P. Tull, and J.P. Havlicek, "Genetic algorithm forecasting for telecommunications products," in *Smart Engineering Systems: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems, Proc. ANNIE 1999*, St. Louis, MO, Nov. 7-10, 1999, ed. C.H. Dagli, *et al.*, ASME Press, Fairfield, NJ, 1999, pp. 361-368.
83. G. Lezos, M. Tull, J. Havlicek, and J. Sluss, "Predicting the future with the appropriate embedding dimension and time lag," in *Proc. Int'l. Joint Conf. Neural Networks*, Washington, DC, Jul. 10-16, 1999, vol. 4, pp. 2509-2513.
84. J.P. Havlicek, J.W. Havlicek, N.D. Mamuya, and A.C. Bovik, "Skewed 2D Hilbert transforms and computed AM-FM models," in *Proc. IEEE Int'l. Conf. Image Process.*, Chicago, IL, Oct. 4-7, 1998, vol. 1, pp. 602-606.
85. J.P. Havlicek, D.S. Harding, N.D. Mamuya, and A.C. Bovik, "Wideband frequency excursions in computed AM-FM image models," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, Tucson, AZ, Apr. 6-7, 1998, pp. 211-216.
86. J.P. Havlicek, J.W. Havlicek, and A.C. Bovik, "The analytic image," in *Proc. IEEE Int'l. Conf. Image Process.*, Santa Barbara, CA, Oct. 26-29, 1997, vol. 2, pp. 446-449.
87. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Extracting Essential Modulated Image Structure," in *Proc. 30th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 3-6, 1996, vol. 2, pp. 1014-1018 (*invited paper*).
88. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Discrete quasi-eigenfunction approximation for AM-FM image analysis," in *Proc. IEEE Int'l. Conf. Image Process.*, Lausanne, Switzerland, Sep. 16-19, 1996, vol. 1, pp. 633-636.
89. J.P. Havlicek, M.S. Pattichis, D.S. Harding, A.C. Christofides, and A.C. Bovik "AM-FM Image Analysis Techniques," in *Proc. IEEE Southwest Symp. Image Anal. & Interp.*, San

Antonio, TX, Apr. 8-9, 1996, pp. 195-200.

90. D.S. Harding, J.P. Havlicek, and A.C. Bovik, "Recent advances in multi-component AM-FM image modeling," in *Proc. 9th IEEE Int'l. Workshop Image, Multidimensional Signal Process.*, Belize City, Belize, March 3-6, 1996, pp. 68-69.
91. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Reconstruction from the multi-component AM-FM image representation," in *Proc. IEEE Int'l. Conf. Image Process.*, Washington, DC, Oct. 22-25, 1995, vol. 2, pp. 280-283.
92. J.P. Havlicek, A.C. Bovik, M.D. Desai, and D.S. Harding, "The discrete quasi-eigenfunction approximation", in *Proc. Int'l. Conf. on Digital Signal Process.*, Limassol, Cyprus, June 26-28, 1995, pp. 747-752.
93. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Multi-component signal demodulation and reconstruction using AM-FM modulation models", in *Proc. 1995 IEEE Workshop Nonlin. Signal and Image Process.*, Neos Marmaras, Halkidiki, Greece, June 20-22, 1995, pp. 41-45.
94. J.P. Havlicek and A. C. Bovik, "Multi-component AM-FM image models and wavelet-based demodulation with component tracking", in *Proc. IEEE Int'l. Conf. Image Process.*, Austin, TX, Nov. 13-16, 1994, vol. 1, pp. 41-45.
95. A.C. Bovik, J.P. Havlicek, and M.D. Desai, "Theorems for discrete filtered modulated signals", in *Proc. IEEE Int'l. Conf. Acoust., Speech, Signal Process.*, Minneapolis, MN, Apr. 27-30, 1993, vol. 3, pp. 153-156.
96. J.P. Havlicek, A.C. Bovik, and P. Maragos, "Modulation models for image processing and wavelet-based image demodulation," in *Proc. 26th IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 26-28, 1992, vol. 2, pp. 805-810.
97. T.L. Arnow, B. Pierce, J. Havlicek, and H.G. Longbotham, "Comparison of four methods of image decomposition," in *Proc. Canadian Conference on Electrical and Computer Engineering*, Toronto, Ont., Canada, Sept. 13-16, 1992.
98. M. Pauli, M. Cordray, J. Havlicek, G. Katz, M. Kruer, K. Sarkady, M. Satyshur, J. Caulfield, J. Hunt, S. Michaels, E. Wilder, R. York, and G. Stamm, "Analysis of recent Fly's Eye IR threat warning sensor measurements," in *Proceedings of the IRIS Symposium on Passive Sensors,* March 1990.
99. G. Katz, M. Kruer, M. Pauli, M. Satyshur, D. Scribner, E. Takken, J. Havlicek, S. Michaels, K. Sarkady, E. Wilder, R. York, J. Caulfield, J. Hunt, K. Norwood, and G. Stamm, "Recent measurements and processed imagery from a dual-band 128x128 mid-wave infrared staring data measurement sensor (Fly's Eye)," in *Proceedings of the IRIS Symposium on Passive Sensors,* March 1989.
100. D.A. Scribner, I.B. Schwartz, D. Ilg, J. Havlicek, M. Pauli, G. Katz, and R. Priest, "3-D LMS filtering techniques for detection of moving targets against clutter backgrounds," in *Proceedings of the IRIS Symposium on Targets, Backgrounds, and Discrimination,* February 1989.

D. Technical Reports:

1. J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Computation of the multi-component AM-FM image representation," Tech. Rept. TR-96-001, Center for Vision and Image Sciences, The University of Texas at Austin, May, 1996.
2. J.P. Havlicek and A. C. Bovik, "AM-FM models, the analytic image, and nonlinear demodulation techniques", Tech. Rept. TR-95-001, Center for Vision and Image Sciences, The University of Texas at Austin, March, 1995.
3. J.P. Havlicek, J.H. Polaha, C.W. Bostian, and J.C. McKeeman, "IBM/Virginia Tech Joint Development Project Final Report on Low-Earth-Orbit Communications Satellites," Doc. Num. 8/05/88, Bradley Dept. Elect. Engr., Virginia Polytechnic Institute and State University, July, 1988.

E. Theses:

1. J.P. Havlicek, "AM-FM image models", *Ph.D. dissertation, The University of Texas at Austin,* 1996.
2. J.P. Havlicek, "Median filtering for target detection in an airborne threat warning system," *M.S. thesis, Virginia Tech,* 1988.

▶ **Short Vita:**

Joseph P. Havlicek is Gerald Tuma Presidential Professor of Electrical & Computer Engineering and Williams Companies Foundation Presidential Professor of Electrical & Computer Engineering at the University of Oklahoma. He is director and co-founder of the OU Center for Intelligent Transportation Systems. He is also a full member of the OU Institute for Biomedical Engineering, Science, and Technology. He received the B.S. degree in 1986 and the M.S. degree in 1988 in Electrical Engineering from Virginia Tech, Blacksburg, VA, and the Ph.D. degree in Electrical and Computer Engineering in 1996 from the University of Texas at Austin.

From 1984 to 1987, he was with Management Systems Laboratories, Blacksburg, VA, as a software engineer developing decision support software for US DoE nuclear materials management. From 1987 to 1989, he was affiliated with SFA, Inc., Landover, MD, and from 1987 to 1997 he was with the Naval Research Laboratory, Washington, DC, where he worked on infrared missile warning receivers for Navy aircraft, including the Navy's first two-color infrared system. In 1990 he was a recipient of the Department of the Navy Award of Merit for Group Achievement for this work which led to production combat systems deployed in both Afghanistan and Iraq. Throughout 1993 he was a programmer–analyst with Ralph Kirkley Associates, Austin, TX, developing image CODECS on-site in the multimedia division of IBM, Austin. He joined the University of Oklahoma as an assistant professor in January, 1997, where he currently holds the rank of Professor and the Gerald Tuma and Williams Companies Foundation Presidential Professorships. His research interests include signal, image, and video processing, modulation domain signal processing, target tracking, medical imaging, and intelligent transportation systems. He is author or co-author of over 130 scholarly publications in these areas. Since joining the University of Oklahoma in 1997, he has been PI or co-PI on more than 110 externally funded grants and contracts totaling over $27M.

Dr. Havlicek is a senior member of the IEEE. He served as a Senior Area Editor of IEEE Transactions on Image Processing from Nov. 2015 through Feb. 2018. Prior to that, he served as an Associate Editor for five years. He is also a past Associate Editor of IEEE Transactions on Industrial Informatics. He served as Publications Chair on the Organizing Committee of the 2007 IEEE International Conference on Image Processing (ICIP), as a Technical Area Chair for ICIP 2012 and ICIP 2013, and as a Technical Area Chair for the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2012 and 2013. He has been a member of the Technical Program Committee and Organizing Committee of the IEEE Southwest Symposium on Image Analysis and Interpretation since 1998, serving as General Co-Chair (2010), Technical Program Co-Chair (2004, 2006, 2008), and Publicity Chair (2000, 2002). He was chairman of the Electrical and Computer Engineering Graduate Studies Committee at the University of Oklahoma from 2008 to 2013 and chairman of the Undergraduate Studies Committee from 2021 to 2023.

He was recipient of the University of Oklahoma College of Engineering Outstanding Faculty Advisor Award in 2006, the University of Oklahoma College of Engineering Brandon H. Griffith Faculty Award in 2003, the University of Oklahoma IEEE Favorite Instructor Award in 1998 and 2000, and the 1992 University of Texas Engineering Foundation Award for Exemplary Engineering Teaching while Pursuing a Graduate Degree. Dr. Havlicek is a member of Tau

Beta Pi, Phi Kappa Phi, and Eta Kappa Nu.

- **Document Last Updated:** 12 February 2025

Appendix B

IN THE 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 No. IPR2024-00626

U.S. Patent 11,805,267

____

Declaration of Immanuel Freedman

TABLE OF CONTENTS

I, Immanuel Freedman, declare as follows:

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

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

I. OVERVIEW OF THE TECHNOLOGY

A. Video Compression Basics

3. Video encoding, also referred to as video compression, exploited redundancies in video data to reduce the size of video. Since the early 1990s, major video coding standards such as MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263, and H.264 have applied the same model, where video encoders have a

motion estimation and compensation front end, a transform stage such as Discrete Cosine Transform ("DCT"), and an entropy encoder at the back end for generating the coded bitstream. At the decoder, the inverse process was used to decode the video. In 2003, the H.264 standard, also known as Advanced Video Coding (AVC), was introduced. This standard quickly became a prevalent and widely adopted video format. The model of a typical general video encoder is illustrated below. Ex-1005, Fig. 2. This fundamental model has been used by major video encoding standards since the 1990s.

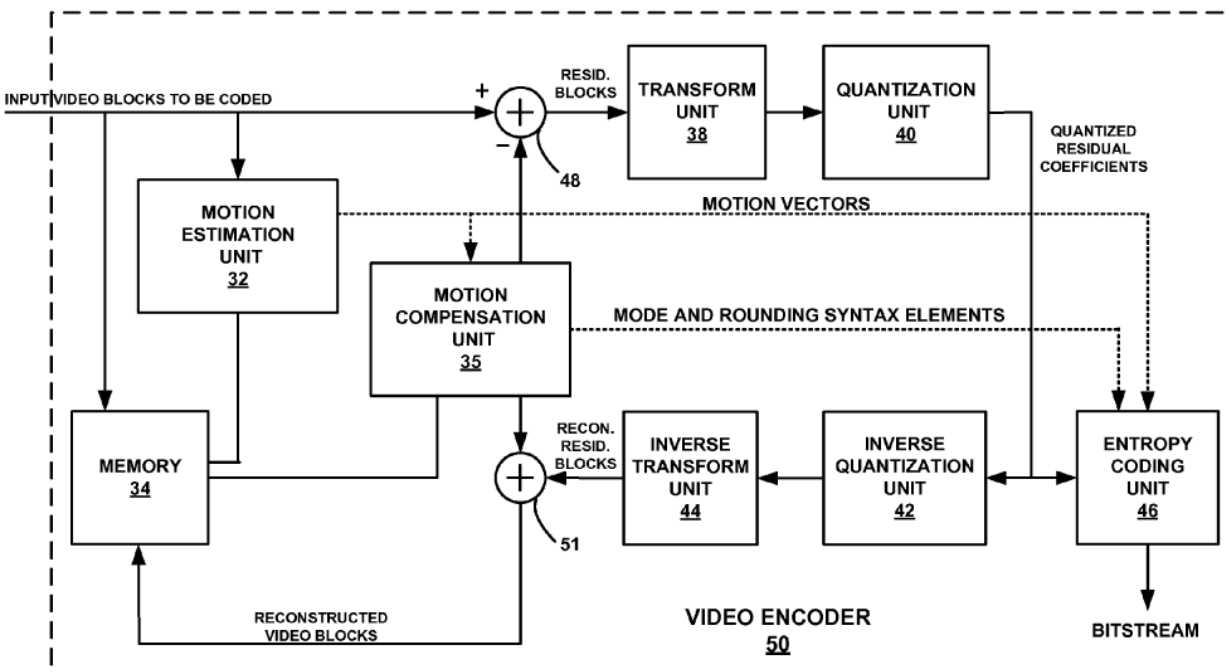


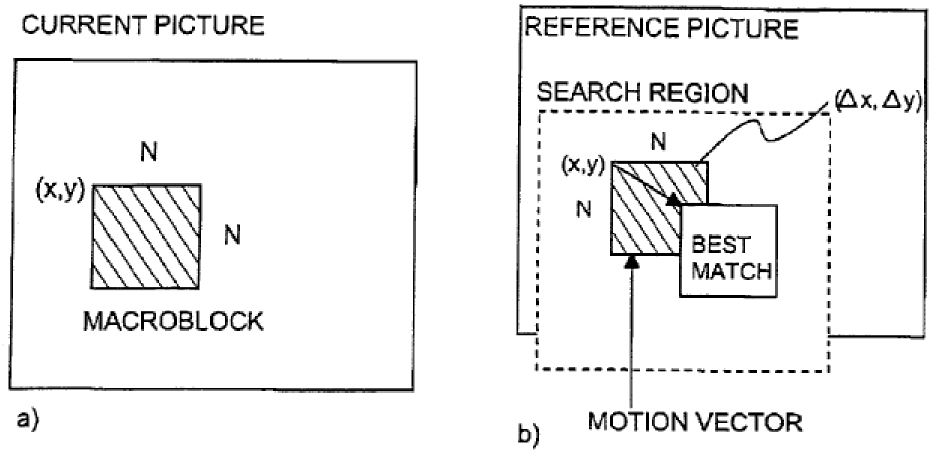
FIG. 2

4. Video was made up of a series of pictures known as frames. Each frame was segmented into blocks of pixels (e.g., referred to as video blocks,

macroblocks, sub-macroblocks, etc.). Each block contained a group of pixels, such as 16x16, 8x8, or 4x4 pixels.

B. Motion Estimation and Compensation

5. Video blocks were encoded with reference to each other. This was known as predictive coding. Two main types of predictive coding were intra-frame and inter-frame encoding. Intra-frame encoding used predictive coding within the same frame, where a block was encoded with reference to another block within the same frame. This took advantage of similarities within the same frame. Inter-frame coding allowed a block to be encoded with reference to blocks in other frames. This type of temporal prediction, called motion estimation and compensation, took advantage of similarities between different frames. For example, when an object appeared in successive frames, inter-frame prediction encoded and transmitted information for a first frame, and encoded subsequent frames by reference to reference blocks in the first frame. A motion vector indicated the displacement of a current block with respect to a reference block, for example indicating that a block moved to the right 5 pixels and moved down 3 pixels between frames. Ex-1010, ¶18, Fig. 4:



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

6. In many video standards, including H.264, blocks encoded with only intra-frame encoding were known as "I" blocks. Conversely, there were two types of inter-coded blocks: "P" and "B" blocks. "P" blocks allowed unidirectional prediction to other frames, while "B" blocks allowed bidirectional prediction, meaning blocks within the frame could be predicted in the forward *and* backwards directions. Ex-1011, 000002, 000007; Ex-1012, 000198-200.

7. Motion estimation and compensation involved identifying the movement of objects or regions between successive frames in a video sequence. In bidirectional prediction for a target block, the encoder searched for similar blocks in two reference frames, such as a past/previous reference frame and a future/subsequent reference frame, that best match the target block. This process

resulted in two motion vectors, each pointing to a different block in a different reference frame. Ex-1012, 000194-195, 000200, 000062, 000090.

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

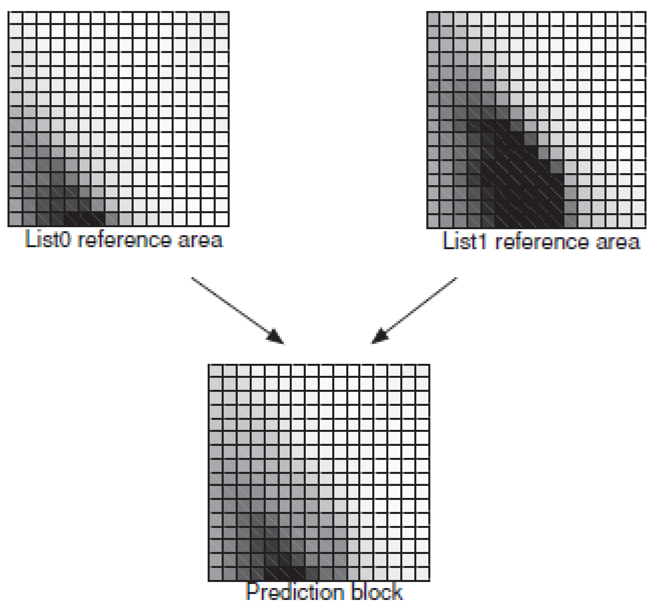


Figure 6.35 Biprediction example

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

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

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

11. The decoder then combined these blocks in the same manner as the encoder to reconstruct the bidirectional prediction of the target block, which included a prediction value for each pixel of the target block. Ex-1012, 000062, 000074. For example, when a weighted bidirectional prediction method was used at the encoder, the decoder used the same weights to combine the two reference

blocks. The decoder also extracted and decoded the residual block that was transmitted from the encoder. The original target block was reconstructed by adding the decoded residual block to the reconstructed bidirectional prediction. Ex-1012, 000059-60, 000062, 000074, 000123.

C. Subpixels and Interpolation

12. When a motion vector pointed to an integer pixel position, the values of the block at that position were used to generate a predicted block. But the H.264 video compression standard, along with multiple other standards, allowed motion vectors to have more granular subpixel resolution by pointing to fractional pixel positions (e.g., half-pixel or quarter-pixel positions), resulting in more accurate motion estimation and compensation. Ex-1011, 000002; Ex-1012, 000184. This situation arose when the best match for a target block in a reference frame was not located at an exact integer pixel position, for example where an object moved exactly one pixel between frames, but rather at a fractional (subpixel) position, e.g., where an object moved a half or quarter pixel between frames.

13. When a motion vector points to an integer pixel position, the values of the reference block are used to generate the predicted block. When a motion vector pointed to a subpixel position, the encoder/decoder used interpolation to generate the predicted block. Interpolation involved creating new pixel values at the subpixel positions based on the surrounding integer pixel values. For example, in

half-pixel interpolation, the value at a half-pixel position could be calculated as the average of six adjacent integer pixel values. Ex-1011, 000010; Ex-1012, 000187. In summary, bi-prediction used two motion vectors, which could point to integer or sub-pixel positions. Therefore, bi-prediction often involved different permutations of motion vectors that could point to integer or sub-pixel positions, with for example both vectors pointing to interpolated blocks, or one vector pointing to an interpolated block and the other pointing to an integer pixel position. *See* Ex-1011, 000010; Ex-1012, 000184-189:

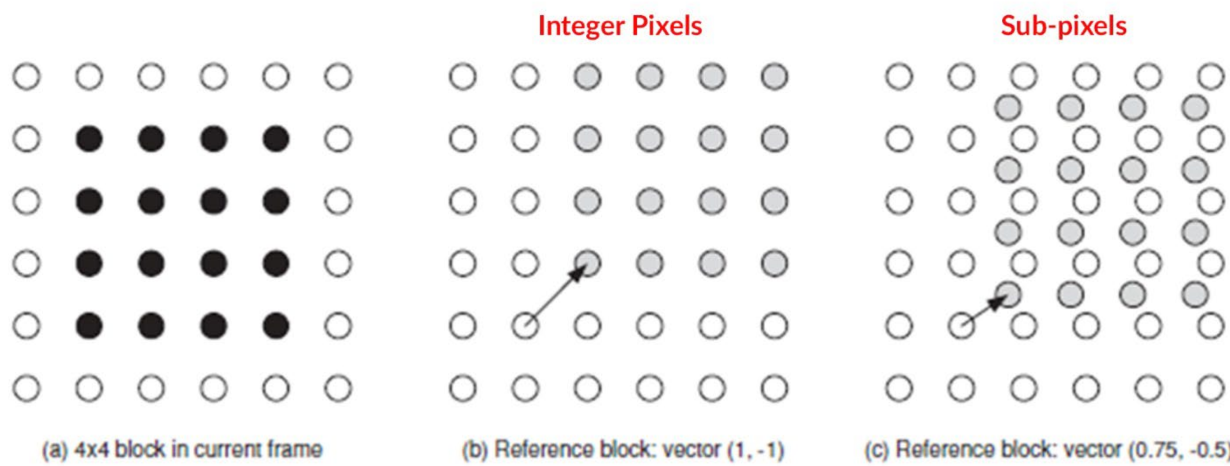


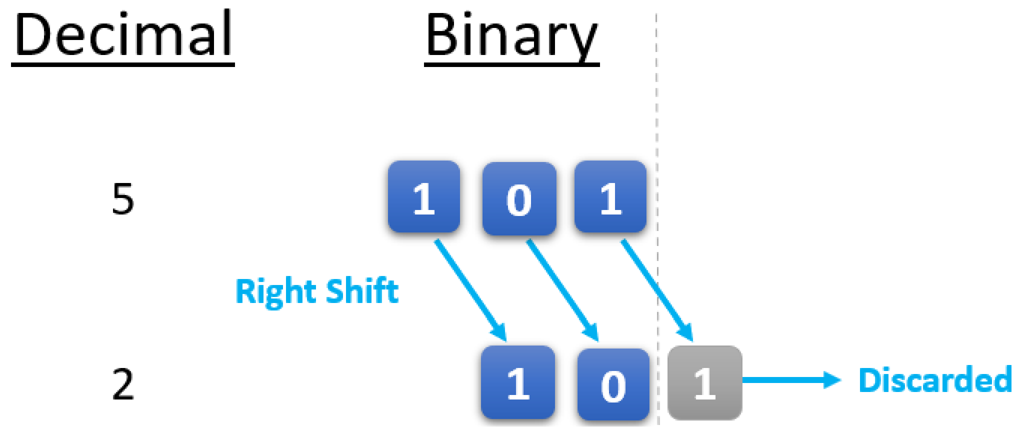
Figure 6.18 Example of integer and sub-pixel prediction

D. Precision and Bit Shifting

14. In computers, numeric values, such as pixel values (predicted or otherwise), were represented in registers or memories as binary numbers. An uncompressed binary number includes a series of positive or negative powers of 2. In a binary number, each digit is a 0 or 1. The precision of the binary number is the

number of terms in that series, which is also the number of bits needed to represent its value in this form. When a calculation may result in multiple possible values, the precision of the result of the calculation corresponds to the number of bits needed to represent the possible values in an uncompressed binary form, which could be changed by mathematical operations. For example, an uncompressed variable having values that range from 0 to 3 can be represented by a binary number of 2 bits. When multiplied by two, the result could range from 0 to 6 and thus required 3 bits of precision to represent the largest possible value 6 (110 in binary). Since multiplication and addition made numbers larger, they may have required more bits (higher precision) to represent the results. Conversely, since division made numbers smaller, it may have reduced the number of bits needed to represent the result. Higher precision allowed for a more accurate representation of a value but required more bits, while lower precision reduced the number of bits needed but might lead to a loss of accuracy. Ex-1009, 000005; Ex-1008, 7:4-19, 16:22-17:36, 20:42-47; Ex-1010, ¶¶61-62. In video, pixel values were often represented using 8 bits, which represented 2^8=256 possible values (from 0-255). A pixel on a two-color (black and white) display can be represented with a single bit having two possible values: 0 or 1. A pixel with 2 bits of precision would represent values from 0-3. A pixel with 3 bits of precision would represent values from 0-7, and so on. Motion prediction and sub-pixel interpolation involved mathematical calculations, including multiplication and addition, that increased the number of bits needed to represent the intermediate results of those calculations. For example, when two 8-bit integer pixel values were averaged together, they were first added together, resulting in a sum that may require 9 bits to represent. The sum is next divided by two, resulting in an 8-bit number. Division can cause a rounding error because, in integer arithmetic, any remainder from the division is discarded. I explain this in more detail below.

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



16. The inverse of truncation can be applied to a binary number by shifting its bits to the left by the desired number of positions. Ex-1008, 20:42-47; Ex-1013, ¶91, ¶116, ¶131. Left shifting effectively performs multiplication by 2 for each position shifted. Ex-1004, ¶46; Ex-1008, 20:42-47; Ex-1013, ¶91, ¶116, ¶131. The newly added least significant bits were filled with zeros. Each left shift effectively doubled the value and also increased the precision by one bit, as a new zero bit was added on the right.

17. Truncating a binary number is sometimes referred to as rounding the binary number. This rounding operation might cause a rounding error due to the difference between the original number and the rounded number. For example, truncating the binary number 101 (5 in decimal) by one bit results in 10 (2 in decimal). This is equivalent to dividing 5 by 2, which equals 2.5. Since the remainder is discarded, there is a rounding error of 0.5. This could occur when averaging 2 and 3, which normally would result in 2.5, but with binary numbers

results in $\frac{2+3}{2} = \frac{5}{2} = 2$. Rounding errors could accumulate in calculations, especially when many operations were performed sequentially. This could lead to significant discrepancies between the computed result and the true value. In this example, if the above average is calculated twice and then added together, the expected result would be 2.5 + 2.5 = 5. However, because two right-shift (division) operations occur during the average, the result is 2 + 2 = 4, meaning the rounding error is 1. To prevent this error, it was known in the art to maintain higher precision for the calculations, e.g., by delaying rounding (division/right-shift) operations. *E.g.*, Ex-1008, 7:4-19. Applying that concept to this example, using basic arithmetic, the division in the averaging operation would be delayed until the end, thereby reaching the expected result of 5 and preventing rounding error:

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

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

19. One well-known and commonly used approach to address accumulation of rounding errors was to maintain higher precision in intermediate steps of calculations. Ex-1008, 7:4-19 ("An advantage of embodiments according

to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."), 16:22-17:36 ("[A] higher bit-depth prediction[] and a higher bit-depth reconstruction residual information[] may lead to higher precise reconstructed samples[] of the block[], and therefore to a smaller needed residual information, as in systems, wherein a rounding of prediction samples and of reconstructed residual samples occurs before the prediction and residual reconstruction process."); Ex-1009, 000005 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding and clamping. It is possible to achieve additional accuracy by retaining a higher precision result after the first stage of filtering. … The two shifts are chosen so as to (a) add up to the required shift for normalizing the filters and (b) to allow for a 16 bit implementation - where the intermediate values in the second filtering operation are within 16 bits."); Ex-1010, ¶¶61-62 ("[T]runcation of the ¼ resolution sub-pixel values has a deleterious effect on the precision of some of the ¼ resolution sub-pixel values. Specifically, the ¼ resolution sub-pixel values are less precise than they would be if calculated from values that had not been truncated and clipped. … . In the encoder the interpolation method according to TML6 works like the previously described TML5 interpolation method, except that maximum precision is retained throughout. This is achieved by using intermediate values which are neither rounded nor clipped.”).

20. Since each rounding step had the potential to discard information and thereby introduce rounding errors, this practice delayed rounding and used more bits to represent numbers during the computation process than were used in the final output. By doing so, the accumulation of rounding errors was minimized, as intermediate operations had a finer granularity and could represent values more accurately. *Id*.

II. THE ’267 PATENT

A. Overview

21. The ’267 patent is directed to “[a]pparatuses, methods and computer programs … for utilizing motion prediction in video coding.” Ex-1001, Abstract. The ’267 patent discusses a process for generating a bi-directional prediction for a current block, including “us[ing] motion vector information to determine which block is used as a first reference block for the current block and which block is used as a second reference block for the current block,” “us[ing] some pixel values of the first reference block to obtain first prediction values and some pixel values

of the second reference block to obtain second prediction values," and combining "the two prediction values." *See, e.g.*, Ex-1001, 12:41-55, 13:43-55, Fig. 10:

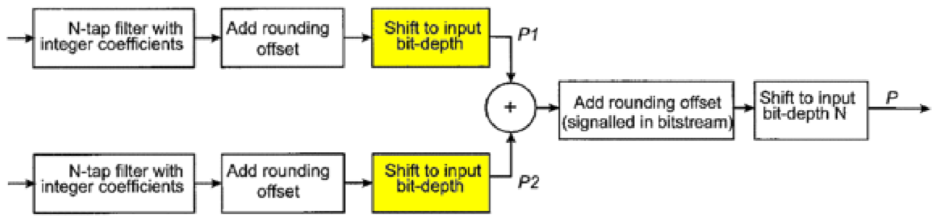


Fig. 10

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

22. The '267 patent does not purport to invent this conventional process of bi-directional prediction, admitting that video coding processes according to standards such as MPEG-2, H.263, and H.264 were known in the art. Ex-1001, 1:34-46 ("Background Information" section). The '267 patent states that "Background" art includes motion compensated prediction and specifically "bi-directional prediction" (e.g., Ex-1001, 2:35-59), where reference blocks are determined based on motion vectors (e.g., Ex-1001, 2:20-34, 3:12-18), predictions are determined based on reference blocks (e.g., Ex-1001, 1:34-46), a bi-directional prediction is obtained by combining two predictions based on two reference blocks

(e.g., Ex-1001, 3:49-55, 3:66-4:20), and residual data is calculated as a difference between the prediction and the current block, encoded, and later used to reconstruct the current block (e.g., Ex-1001, 1:52-59, 3:25-30, 2:1-12).

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

24. The purportedly inventive concept of the ’267 patent is to maintain prediction signals “in a higher precision during the prediction calculation” and reduce the precision “after the two or more prediction signals have been combined

with each other." Ex-1001, 4:29-43, 6:51-57, 12:41-13:55. By doing so, the '267 patent claims to "enable[] reducing the effect of rounding errors in bi-directional and multi-directional prediction." Ex-1001, 4:29-35, 6:51-57, Fig. 11:

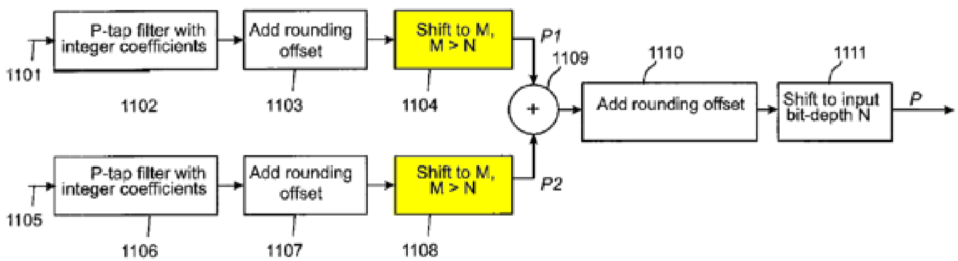


Fig. 11

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

B. Prosecution History

25. The '267 patent was allowed after one Office Action, which included an obviousness-type double patenting rejection, in response to which the Applicant submitted a terminal disclaimer. Ex-1002, 000123-130, 000155-156, 000159.

26. The application for U.S. Patent No. 9,432,693, a parent of the '267 patent, received Office Actions with substantive prior-art rejections. The prosecution history of the '693 patent includes three Office Actions, which present §103 rejections based on U.S. 2013/0142262 ("Ye"), U.S. 2009/0087111 ("Noda"), and U.S. 2010/0086027 ("Panchal"). Ex-1007, 000139-157, 000200-218, 000246-261. The original claims recited, among other limitations,

"determining a block of pixels of a video representation encoded in a bitstream, values of said pixels having a first precision" and "using said first reference pixel location to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision." Ex-1007, 000036.

27. The Examiner initially cited to Ye's teachings of integer pixel precision and fractional pixel precision as respectively teaching the recited "first precision" and "second precision." Ex-1007, 000146-148, 000203-204, 000207-208. In response, the Applicant distinguished the cited teachings of Ye by amending the claims to recite "wherein the first precision indicates the number of bits needed to represent values of said pixels" and "wherein the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction[.]" Ex-1007, 000236, 000243.

28. The Examiner cited Ye's weighted prediction teachings (e.g., the equation $P(x,y) = (w \cdot P_0(x,y)+(W-w) \cdot P_1(x,y)+ W/2) >> S)$ in the next Office Action, asserting that multiplying $P_0(x,y)$ with a weight w "increase[s] the precision of $P_0(x,y)$" and that this suggests the second precision. Ex-1007, 000249-251. However, the Examiner acknowledged that Ye does not have explicit teachings that "said first prediction having a second precision, which is <u>higher than said first precision</u>, wherein the second precision indicates <u>the number of bits needed to represent values of said first prediction and values of said second</u>

prediction[,]" among other limitations. Ex-1007, 000252 (emphasis in original). The Examiner cited to Noda's teaching of increasing pixel bit depth to address Ye's acknowledged deficiencies.

29. In response, the Applicant argued that neither reference teaches that "the reference blocks are at a first precision and the first and second predictions are at a second precision." Ex-1007, 000280-282. Regarding Ye, the Applicant did not dispute that the weighted prediction (e.g., $w \cdot P_0(x,y)$, $(W-w) \cdot P_1(x,y)$) teaches the claimed first or second prediction. The Applicant argued that "$w \cdot P_0(x,y)$ and $(W-w) \cdot P_1(x,y)$ do not have increase[d] precision relative to $P_0(x,y)$ and $P_1(x,y)$, respectively, and, instead, simply serve to change the range of values," and that "[s]uch an increase in the range of values as in Ye does not teach or suggest that the precision increases such that Ye fails to teach or suggest any increase in precision from the reference blocks to the first and second predictions." Ex-1007, 000281.

30. Regarding Noda, the Applicant asserted that "Noda discloses increasing the bit depth of each pixel of an input image having an N bit depth to a reference image of (N+M) bit depth, and only then generating a prediction image of the (N+M) bit depth from the reference image of the (N+M) bit depth." Ex-1007, 000281-282. Based on this characterization of Noda, the Applicant argued that "Noda fails to teach or suggest any increase in precision from the reference

blocks to the first and second predictions. Instead, both the reference block from which the prediction image is generated as well as the prediction image itself have the same precision, that is, a (N+M) bit depth." *Id*. The Examiner allowed the application for the '693 patent after this response. Ex-1007, 000294-311.

C. Priority Date

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

D. Challenged Claims

32. I understand that Petitioner is challenging the validity of claims 1-18 of the '267 patent in the Petition for *Inter Partes* Review to which this declaration will be attached. Those claims are reproduced in Appendix 3. While the Petition and this declaration are directed to the challenged claims, I have considered all

claims 1-36 of the ’267 patent, as well as portions of the ’267 patent prosecution history in forming my opinions.

III. LEVEL OF ORDINARY SKILL IN THE ART

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

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

35. I also considered the sophistication of the technologies involved, and the educational background and experience of those actively working in the field at the time. I also considered the level of education that would be necessary to understand the ’267 patent. Finally, I placed myself back in the relevant period of time and considered the engineers and programmers that I have worked with and led in the field of video encoding/decoding.

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

IV. CLAIM CONSTRUCTION

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

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

A. “precision”

39. Based on my review of the claims and specification of the ’267 patent, it is my opinion that a POSITA would have understood “precision” is satisfied by,

but is not necessarily limited to, "a number of bits needed to represent possible values."

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

41. This interpretation is confirmed by the specification of the '267 patent, which uses the term "precision" to refer to a number of bits representing possible values, with examples of the number of bits used to represent the possible pixel and prediction values. *See, e.g.*, Ex. 1001, 12:41-13:18 ("The precision M is higher than the precision of the expected prediction value. For example, *pixel values and the prediction values may be represented by N bits* wherein M>N. In some example implementations N is 8 bits and M is 16 bits but it is obvious that

also other bit lengths can be used with the present invention."), 14:4-10 ("For example, if a motion vector of one of the prediction directions point to an integer sample, the bit-depth of prediction samples with integer accuracy may be increased by shifting the samples to the left so that the filtering can be performed with *values having the same precision*."), 13:19-55. These passages of the specification demonstrate that the precision describes a number of bits needed to represent possible values.

42. The interpretation is further confirmed by the Applicant's statement during prosecution of a parent application (U.S. Patent Application No. 13/344,893, issued as U.S. Patent No. 9,432,693) of the '267 patent. *See, e.g.*, Ex-1007, 000280 ("*The first precision indicates the number of bits needed to represent values of said pixels*. … The second precision indicates the number of bits needed to represent values of the first prediction and values of the second prediction."), 000243.

V. INVALIDITY GROUNDS

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

44. Based on my review and analysis of the materials cited herein, my opinions regarding the understanding of a POSITA in the relevant timeframe

(*supra* §II.C), and my training and experience, it is my opinion that the challenged claims of the ’267 patent are invalid in view of the following grounds:

Grounds	Claims	Statutory Basis	Prior Art
1	1-18	§ 103	Walker
2	1-18	§ 103	Karczewicz-I in view of Karczewicz-II

A. Ground 1: Claims 1-18 Are Rendered Obvious by Walker

45. It is my opinion that a POSITA would have found Claims 1-18 obvious based on the teachings of Walker.

1. U.S. Patent Application Publication No. 2005/0281334 (“Walker”) (Ex-1004)

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

47. Walker was published December 22, 2005, and was filed as application 11/120,513 on May 2, 2005. Therefore, I understand Walker is prior art under at least pre-AIA §102(b) because it was published more than one year before the earliest possible filing date for the ’267 patent, its provisional application’s filing date of January 7, 2011. I further understand Walker is prior art under at least

pre-AIA §§102(a) and 102(e) because it was filed and published before January 7, 2011.

48. Walker is directed to "methods and apparatus for decoding compressed video data where various weighted prediction methods were used for encoding the video data." Ex-1004, ¶3. Walker "allows decoding of multiple weighted bi-directional encoding schemes with a single decoder[,]" including, for example, encoding schemes under MPEG-4 and H.264. Ex-1004, ¶8, ¶¶46-49, ¶¶58-72. For example, Walker teaches weighted prediction under H.264, where a combined prediction is obtained based upon a weighted combination of predictions of two reference blocks. Ex-1004, ¶¶58-70. On the decoding side, Walker proposes "a universal formula that is used by embedded hardware, … to decode weighted prediction frames" encoded in the various implementations described therein. Ex-1004, ¶72, ¶92. Moreover, Walker teaches obtaining predictions for subpixels via interpolation and that "[p]ixel interpolation can be used to improve the performance of motion compensated predictive coding." Ex-1004, ¶111, ¶114, Figs. 8-9.

49. Walker includes multiple figures and corresponding teachings directed to aspects of video encoding and decoding. The teachings and figures of Walker's embodiments, as relied on by this Declaration, are explained as aspects of video encoding/decoding that are used in conjunction with each other. For

example, Figure 1 shows "a general communications system for encoding and decoding streaming pictures" that includes "multiple types of encoder devices … and a decoder device[.]" Ex-1004, Fig. 1, ¶25. Walker then explains interrelated aspects of those video encoders and decoders.

50. Walker teaches that the encoder devices of Figure 1 each "performs weighted bi-directional prediction by one of a plurality of methods." Ex-1004, ¶25; *see also* ¶45 ("A versatile decoder, such as decoder device 155 depicted in FIG. 1, should be able to decode video that was encoded by multiple implementations with various encoded bit configurations and various types of weighted/non-weighted prediction methods."). Walker discusses "[t]he most prevalent weighted bi-directional prediction implementations" including the "weighted bi-directional prediction in H.264[.]" Ex-1004, ¶46, ¶72 ("The four implementations presented above [e.g., H.264] are all widely used and accepted forms of video compression."). Therefore, Walker teaches at least one encoder device of Figure 1 performs weighed bi-directional prediction according to the H.264 standard.

51. Figure 5 illustrates "an example of a weighted B Frame construction process[,]" which is carried out in both encoding and decoding processes. Ex-1004, Fig. 5, ¶¶34-35. Walker teaches the process of Figure 5 being performed by the encoder devices and decoder device of Figure 1. Ex-1004, ¶29 ("FIGS. 3, 4, 5 and 7 illustrate various inter-coding processes including those used for constructing P

Frames, B Frames, weighted B Frames and H.264 predicted frames. The encoder devices 105, 110 and 115 and the decoder device 155 depicted in FIG. 1 can perform these processes in whole or in part."). Because Walker teaches at least one encoder device of Figure 1 performing weighed bi-directional prediction according to the H.264 standard, which is a technique for constructing B frames, Walker's disclosure with respect to Figure 5 encompasses this technique.

52. Figure 8 illustrates "an example of a decoder process for decoding multiple encoder implementation of weighted bi-directional predicted video data." Ex-1004, Fig. 8, ¶110. Walker teaches the process of Figure 8 being carried out by the decoder device of Figure 1. Ex-1004, ¶110 ("Process 800 could be carried out with a device such as decoder device 155 depicted in FIG. 1."). A POSITA would have understood that, when the decoder device of Figure 1 carries out the process of Figure 8, the process is used to decode weighted bi-directional predicted video data generated by the encoder devices of Figure 1. Because Walker teaches at least one encoder device of Figure 1 performing weighed bi-directional prediction according to the H.264 standard, Walker teaches the process of Figure 8 being used to decode video data encoded according to the H.264 standard.

53. Walker teaches using pixel interpolation as part of the process of Figure 8. Ex-1004, ¶111. Walker provides further details about the pixel

interpolation process in Figure 9, which illustrates "an example of half-pixel interpolation for use in motion compensation." Ex-1004, ¶114, Fig. 9.

54. In short, the teachings and figures of Walker's embodiments are used in conjunction with each other. A POSITA would have been motivated to combine Walker's teachings, as explained for its embodiments, because Walker presents those teachings as complementary aspects of video encoders and decoders that are meant to be used together. Ex-1004, ¶25, ¶29, ¶¶34-35, ¶¶45-46, ¶72, ¶¶110-111, ¶114, Figs. 1, 5, 8-9. Moreover, it was known in the art that weighted bi-directional prediction and sub-pixel interpolation were used together because widespread industry standards including the ubiquitous H.264 standard utilized those concepts together. *See, e.g.*, Ex-1006, ¶¶46-47, ¶51, ¶¶68-69, ¶74, ¶86, ¶88, ¶93; Ex-1014, 000188-193.

55. Walker is in the same field of endeavor as the '267 patent because it is directed to video encoding/decoding, and in particular motion prediction. *See supra* §III; Ex-1004, ¶3 ("This invention relates to methods and apparatus for decoding compressed video data where various weighted prediction methods were used for encoding the video data."). Walker teaches and improves upon the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex-1004, ¶¶46-49, ¶¶58-72, ¶112 *with* Ex-1001, 2:60-3:11, 9:26-41.

56. Walker teaches limitations that the Examiner found missing in Ye. As an example, Walker includes explicit teachings about the numbers of bits needed to represent its pixel values, weighted predictions, and combined predictions. *Infra* §§V.A.2[1b-1d, 1f]/[7b-7d, 7f]/ [13b-13d, 13f]. This type of explicit teaching was acknowledged by the Examiner to be missing from Ye. *Supra* §II.B; Ex-1007, 000252. As explained below, Walker teaches the limitations of claims 1-18 at least partly based on the teachings that Ye lacks.

2. Independent Claims 1, 7, and 13

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

57. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that Walker teaches the preamble.

58. Walker "relates to methods and apparatus for decoding compressed video data where various weighted prediction *methods* were used for *encoding* the video data." Ex-1004, ¶3.[1] Walker teaches "a general communications system for encoding and decoding streaming pictures." Ex-1004, ¶25, Fig. 1:

[1] All annotations/emphasis added unless otherwise noted.

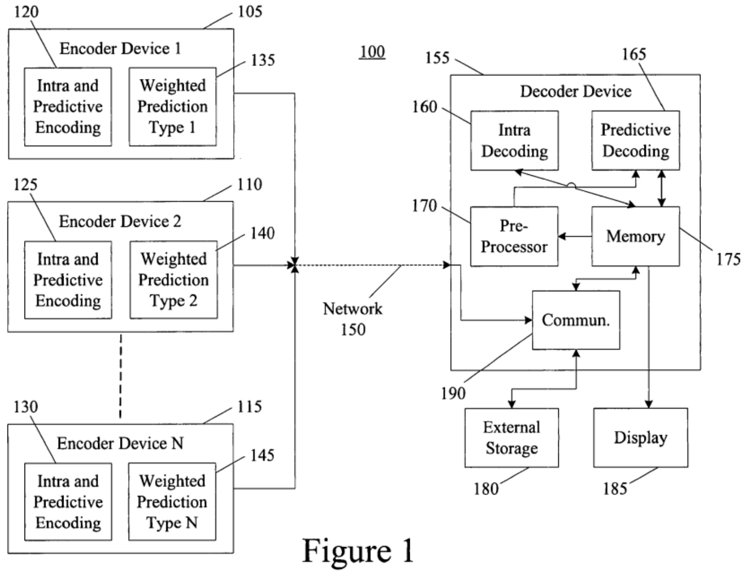


Figure 1

Walker's teachings include "intra-coding and predictive coding modules" that "perform the various types of encoding including intra-coded and inter-coded pictures" and "weighted prediction module [that] performs weighted bi-directional prediction by one of a plurality of methods." Ex-1004, ¶25.

59. Walker encodes a block of pixels, including with bi-directional motion prediction on macroblocks. *See, e.g.*, Ex-1004, ¶34 ("Encoding macroblock 515 of current picture 505 is predicted in reference to previous reference picture 510 at a previous time point than current picture 505 and in reference to subsequent reference picture 575 at a subsequent time point."), ¶60 ("where pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future) …"), ¶111 ("An encoder can perform pixel interpolation to locate the best matching reference macroblock (or any size section) and point to the pixel or

interpolated pixel with a motion vector."). A macroblock is a block of pixels. Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels."). Therefore, Walker teaches a method for encoding a block of pixels.

60. As explained below, operations related to weighted bi-directional prediction according to the H.264 standard, as taught by Walker, teach the limitations of claim 1. *Infra* §§V.A.2[1b-1g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1. Therefore, Walker teaches a method for encoding a block of pixels, comprising the operations explained below for limitations [1b]-[1g].

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

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

62. As explained above for [1a], Walker teaches a method for encoding a block of pixels, including operations as described for limitations [1b]-[1g]. *Supra* §V.A.2[1a]. As explained below, Walker teaches an encoder device for encoding a block of pixels that performs operations as described for limitations [7b]-[7g].

Infra §§V.A.2[7b-7g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1.

63. Walker teaches implementing its methods using a processor and memory that includes computer program code (e.g., software module). Ex-1004, ¶123 ("The steps of a method or algorithm described in connection with the examples disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art."). Because Walker's encoder device performs its method of encoding a block of pixels, Walker teaches implementing the encoder device as an apparatus that comprises such a processor and memory.

64. A POSITA would have been knowledgeable about basic computer architecture and understood that, in a conventional computing device, the processor executes computer program code in the memory to cause the device to carry out its functionalities. Therefore, Walker's encoder device teaches an apparatus for encoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one

processor, cause the apparatus to perform operations as described for limitations [7b]-[7g].

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

65. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that Walker teaches the preamble.

66. Walker teaches implementing its methods using program code (e.g., software module) residing on computer readable storage media, including memory types, hard disks and CD-ROMs that are executable by a processor. Ex-1004, ¶123 ("The steps of a method or algorithm described in connection with the examples disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art."). A POSITA would have understood that the types of storage media disclosed by Walker, including RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory,

registers, hard disk, a removable disk, a CD-ROM, are forms of non-transitory computer readable storage medium, as opposed to transitory signals. Ex-1004, ¶123. A POSITA would have found it obvious that the software module, which is stored in non-transitory computer readable storage medium and executable by a processor, includes computer executable program code portions that comprise program code instructions.

67. As explained above, Walker teaches an apparatus with program code that implements its teachings (*supra* §V.A.2[7a]), stored on non-transitory medium. Therefore, Walker applies its teachings to a computer-program product. The operations taught by Walker, which are performed by the software module, teach the limitations of claim 13. *Infra* §§V.A.2[13b-13g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1. Additionally, a POSITA would have found it obvious to have a computer program product that includes a storage medium storing the software module, since computer program products have almost universally stored their software program code in non-transitory mediums (e.g., hard disk, memory, or CD-ROM) for decades. *See* Ex-1004, ¶123. For example, software program code instructions saved on a CD-ROM had been a very common form of computer program product since the 1990s. Therefore, Walker teaches a computer program product for encoding a block of pixels, the computer program product comprising

at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to perform the operations recited in claim 13.

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

68. Walker teaches limitations [1b], [7b], and [13b]. First, Walker teaches **determining, for a current block** (e.g., Walker's "current macroblock" 515)**, a first reference block** (e.g., "best matching macroblock" 520 in previous reference picture 510) **based on a first motion vector** (e.g., motion vector 525) **and a second reference block** (e.g., "best matching macroblock" 580 in subsequent reference picture 575) **based on a second motion vector** (e.g., motion vector 585). Walker's Figure 5 teaches "an example of a weighted B Frame construction process" for video encoding of macroblocks. Ex-1004, ¶34, Fig. 5:

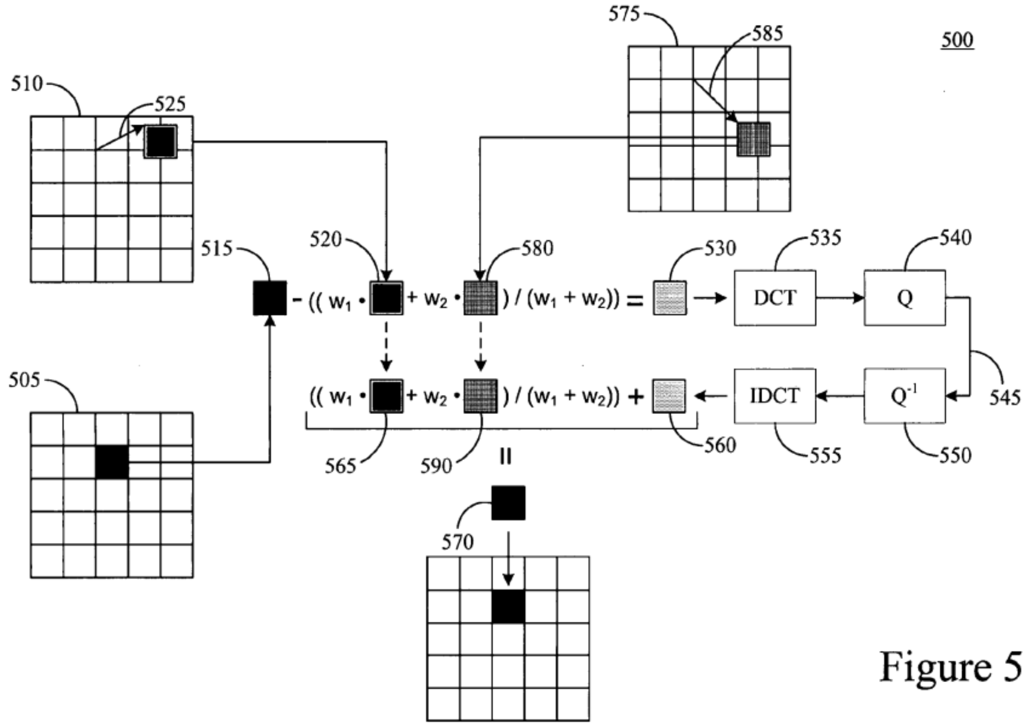


Figure 5

Walker teaches the encoder carrying out the process of Figure 5. Ex-1004 ¶29 ("FIGS. 3, 4, 5 and 7 illustrate various inter-coding processes including those used for constructing P Frames, B Frames, weighted B Frames and H.264 predicted frames. The encoder devices 105, 110 and 115 and the decoder device 155 depicted in FIG. 1 can perform these processes in whole or in part.").

69. For each macroblock of a B frame, Walker teaches determining two "best matching" reference blocks based on two motion vectors: forward and backward. Ex-1004, ¶32, ¶34. Each B frame combines forward and backward motion vectors that reference blocks in I or P frames. Ex-1004, ¶28 ("Each B frame can combine forward and backward motion vectors and residual errors referenced to I frame 22A or predicted P frames 24[.]").

70. Walker determines a *first*, best-matching reference block for the current macroblock based on a first, *backward*-pointing motion vector. The backward motion vector points to a previous frame (reference picture). For the backwards motion vector, "[a] search is made in previous reference picture 510 to locate *best matching macroblock 520 that is closest to current macroblock 515* being encoded." Ex-1004, ¶34. "The location of the best matching macroblock 520 is encoded in motion vector 525." *Id*.

71. Walker determines a *second*, best-matching reference block for the current macroblock based on a second, *forward*-pointing motion vector. For the forward-pointing motion vector, Walker applies a similar approach to the one explained above for the backwards vector: "A search is made in subsequent reference picture 575 to locate best matching macroblock 580 that is closest to current macroblock 515. The location of best matching macroblock 580 is encoded in motion vector 585." *Id.*

72. Walker determines both reference blocks for the current macroblock based on their corresponding motion vectors.[2] The location of each reference block

[2] The '267 patent admits that determining reference blocks based on motion vectors were known in the art by describing it in the "Background Information" section. *See* Ex-1001, 2:20-34, 3:12-18 ("Each of these motion vectors represents the displacement of the image block in the picture to be coded (in the encoder) or decoded (at the decoder) and the prediction source block in one of the previously coded or decoded images (or pictures).").

is identified by its motion vector; Walker encodes the location of each reference block in the form of a motion vector. Ex-1004, ¶34. Therefore, the determination of which block to use as a reference block is based on the motion vector that identifies the location of that block. *See* Ex-1004, ¶34; *see also* ¶32 ("The locations of the best matching prediction region in the subsequent reference picture and the best matching prediction region in the previous reference picture can be encoded in two motion vectors.").

73. Second, Walker teaches that **the pixels of the current block, the first reference block, and the second reference block have values with a first precision** (e.g., 8 bits). Walker teaches that the pixels in its macroblocks have 8-bit luminance and chrominance values. Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels. Pixels can be defined by an 8-bit luminance value (Y) and two 8-bit chrominance values (Cr and Cb)."), ¶60 ("where pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future)[.]"). Therefore, Walker teaches that 8 bits are needed to represent the possible values of the pixels of the current block, the first reference block, and the second reference block.

74. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because 8 bits

are needed to represent the possible luminance and chrominance values of the pixels of the current block, the first reference block, and the second reference block, these pixels all have values with a same "first precision."

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

75. It is my opinion that Walker teaches limitations [1c], [7c], and [13c]. As explained above, Walker teaches at least one encoder device of Figure 1 performing weighted bi-directional prediction according to the H.264 standard, which is applicable to the weighted B frame construction process of Figure 5. *Supra* §V.A.1.

76. First, Walker teaches **using said first reference block** (e.g., a "prediction block" from a past frame) **to obtain a first prediction** (e.g., "(pred0)w0"). Walker explains that "the H.264 video compression standard offers weighted and non-weighted prediction for both single directional and bi-directional predicted regions." Ex-1004, ¶59 ("Weighted prediction is invoked in H.264 by setting one or both variables 'predFlagL0' and 'predFlagL1' equal to 1."). Equation 13 governs weighted predictions (e.g., Case 3, where "both of the two reference partitions are to be weighted"):

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

Ex-1004, ¶¶67-68.[3]

77. Here, the value "(pred0)w0" is the weighted prediction from the first reference block: "pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future) and Final_pred is the resultant prediction[.]" Ex-1004, ¶60. The prediction blocks from the two reference frames (one past, one future) refer to best-matching macroblocks in previous and subsequent reference pictures mentioned elsewhere in Walker, which satisfy the recited "first reference block" and "second reference block." *Supra* §V.A.2[1b]/[7b]/[13b]; Ex-1004, ¶34. A POSITA would have understood this because "frames" and "pictures" were used interchangeably in video coding to refer to an image in a sequence of images of a video. Likewise, "past/future" were used interchangeably with "previous/subsequent," respectively, in describing relative locations of frames or pictures. *Supra* §I.B. Moreover, it is clear from the context of Walker and consistent with how bi-predicted blocks had worked since the 1990s.

[3] The "Clip1" function limits the result of Equation 13 to 0-255. Ex-1004, ¶66:

$$Clip1(x) = Clip3(0, 255, x) \quad (11)$$

$$Clip3(x, y, z) = \begin{cases} x; z < x \\ y; z > y \\ z; \text{otherwise} \end{cases} \quad (12)$$

78. The coefficients and constants in this equation are explained 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
o_0	The additive offset applied to the luma or chroma prediction value for the first reference picture (list "0")	−128 to 127	8
o_1	The additive offset applied to the luma or chroma prediction value for the first reference picture (list "0")	−128 to 127	8

Ex-1004, ¶69.

79. Walker teaches executing a series of computational operations based on Equation 13 when both reference values are weighted. Ex-1004, ¶¶67-68. The operations are outlined in Table 2.[4]

[4] Walker Table 2 and equation 14 represent a general formulation that includes Equation 13. Walker explains three cases: the first reference block is weighted; the second reference block is weighted; or both reference blocks are weighted as described for Equation 13. Ex-1004, ¶¶61-69. To simplify all three cases into one equation, Walker teaches a generalized form in Equation 14, which therefore

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, ¶84.

80. Operation No. 1 computes the values "(pred0)w0" and "(pred1)w1." Ex-1004, ¶68, ¶84. The weighted prediction value "(pred0)w0" is the product of a sample value (e.g., pred0) from a first reference block (e.g., from a past frame)

---

encompasses the case of Equation 13. Ex-1004, ¶74, ¶84. The only difference between Equations 13 and 14 is that the terms 2^{logWD} and logWD+1 in Equation 13 are respectively replaced with 2^{LWD-1} and LWD in Equation 14, respectively. *Compare* Ex-1004, ¶68 *with* ¶74. Walker explains that, in the case where Equation 13 applies (i.e., Case 3), the variable LWD=logWD+1, which causes Equation 14 to be identical to Equation 13 because LWD-1=logWD and that $2^{LWD-1}=2^{logWD}$. Ex-1004, ¶82. Therefore, Equation 14 is equivalent to Equation 13 for Case 3. Indeed, Walker uses Equation 14 to illustrate its teachings for decoding, and as was well known to a POSITA, motion prediction must be determined in the same manner when encoding and decoding video to ensure that the decoding process restores the video to its original form. Therefore, a POSITA would have understood that Table 2 describes the computational operations related to Equation 13.

multiplied by a weight (e.g., w0), and is thus obtained using the first reference block. Ex-1004, ¶60, ¶69.

81. Walker's "(pred0)w0" constitutes a first prediction because it is a value calculated by multiplying a pixel value (e.g., pred0) from a reference block with a scaling factor (e.g., weight w0), which is used for motion prediction of a current block.[5] Ex-1004, ¶60, ¶¶68-69.

---

[5] This is consistent with the usage of "prediction" in the '267 patent specification, which includes embodiments where predictions are calculated by performing mathematical operations on pixel values in reference blocks, such as applying a scaling factor (e.g., weights or other coefficients) on the pixel values. For example, the '267 patent describes "us[ing] some pixel values of the first reference block to obtain first prediction values and some pixel values of the second reference block to obtain second prediction values." Ex-1001, 12:41-13:18. "[I]f a first motion vector points to a fraction of a pixel, … a P-tap filter such as a six-tap filter in which P pixel values of the reference block are *used to calculate the prediction value*" by, e.g., multiplying each pixel value with a weight (e.g., 1, -5, 20). *Id*. ("Hence, the filter 1102 would receive 1101 the pixel values of pixels E, F, G, H, I and J and filter these values by the equation P1=(E1−5*F1+20*G1+20*H1−5*I1+J1)."); *see also* Ex-1001, 13:19-42, 14:11-22. The '267 patent further states that "if a motion vector of one of the prediction directions point to an integer sample, the bit-depth of prediction samples with integer accuracy may be increased by shifting the samples to the left[.]" Ex-1001, 14:4-10. Shifting a binary number to the left is mathematically equivalent to multiplying the binary number by a scaling factor. *Supra* §I.D.; *infra* §V.A.5. For example, shifting a binary number to the left by 5 bits is equivalent to multiplying the number by 2^5 or 32. *Id*. The '267 patent thus describes multiplying the value of a pixel sample from a reference block by a scaling factor to obtain a prediction. During prosecution, the Applicant did not dispute that weighted predictions teach first and second predictions. Ex-1007, 000280-281; *supra* §II.B.

82. Second, Walker teaches that **said first prediction** (e.g., the value "(pred0)w0") **having a second precision** (e.g., 16 bits)**, which is higher than said first precision** (e.g., 8 bits). In Walker's Table 2, the column named "Bitwidth of Operation Result" includes the number of bits needed to store the result of each operation based on the possible values of the result. Ex-1004, ¶84. The row corresponding to Operation No. 1 indicates that "(pred0)w0" has a bitwidth of 16, which means that 16 bits are needed to represent the possible values of "(pred0)w0." *Id*. This is consistent with binary arithmetic because pred0 has 8 bits, w0 has 8 bits, and therefore, 16 bits (8+8) would be needed to represent the possible values of their product. *Id.*

83. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because 16 bits are needed to represent the possible values of the first prediction (e.g., "(pred0)w0"), the first prediction has a second precision of 16 bits, which is higher than the first precision of 8 bits. *Supra* §V.A.2[1b]/[7b]/[13b].

84. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶34. "The weighted best matching forward macroblock and the weighted best matching backward macroblock are combined to form a weighted combined bi-directional macroblock…" *Id.* Walker's weighting factors, as explained above, are chosen "such that a weighted linear combination of the

best matching subsequent and best matching previous macroblocks results in a smaller residual error…" *Id.*

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

85. It is my opinion that Walker teaches limitations [1d], [7d], and [13d]. Walker teaches **using said second reference block** (e.g., a "prediction block" from a future frame) **to obtain a second prediction** (e.g., "(pred1)w1")**, said second prediction having the second precision** (e.g., 16 bits).

86. As explained with respect to limitations [1c], [7c], and [13c], Walker's Equation 13 governs weighted predictions, including Case 3, where "both of the two reference partitions are to be weighted." Ex-1004, ¶¶67-68; *supra* §V.A.2[1c]/[7c]/[13c]. Operation No. 1 in the process of executing this equation computes the values "(pred0)w0" and "(pred1)w1." *Id*. The value "(pred1)w1" is the product of a sample value (e.g., pred1) from a second reference block (e.g., from a future frame) multiplied by a weight (e.g., w1), and is thus obtained using said second reference block. Ex-1004, ¶60, ¶69; *supra* §V.A.2[1c]/[7c]/[13c]. For the same reasons as explained above for why "(pred0)w0" is a first prediction, the value "(pred1)w1" is a second prediction. *Supra* §V.A.2[1c]/[7c]/[13c]. Furthermore, the second prediction (pred1)w1 has a precision of 16 bits, which is the second precision. Ex-1004, ¶84; *supra* §V.A.2[1c]/[7c]/[13c].

87. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶34. "The weighted best matching forward macroblock and the weighted best matching backward macroblock are combined to form a weighted combined bi-directional macroblock…" *Id.* Walker's weighting factors, as explained above, are chosen "such that a weighted linear combination of the best matching subsequent and best matching previous macroblocks results in a smaller residual error…" *Id.*

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

88. It is my opinion that Walker teaches limitations [1e], [7e], and [13e]. As explained above, Equation 13 governs weighted predictions, e.g., Case 3, where "both of the two reference partitions are to be weighted":

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

Ex-1004, ¶¶67-68. Walker teaches executing a series of computational operations outlined in Table 2 based on Equation 13:[6]

[6] *Supra* §V.A.2[1c]/[7c]/[13c], n.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

Ex-1004, ¶84; *supra* §§V.A.2[1c-1d]/[7c-7d]/[13c-13d].

89. Walker teaches **obtaining a combined prediction based at least partly upon said first prediction** (e.g., "(pred0)w0") **and said second prediction** (e.g., "(pred1)w1"). As explained above, Operation No. 1 calculates the first and second predictions (e.g., "(pred0)w0" and "(pred1)w1"). *Supra* §§V.A.2[1c-1d]/[7c-7d]/[13c-13d]. Operation No. 2 calculates a rounding offset (e.g., 2^{LWD-1} or 2^{logWD})[7]. *Infra* §V.A.6. Operation No. 3 calculates a sum by adding up the first

[7] As explained above, 2^{LWD-1} and 2^{logWD} are equivalent in the case where Equation 13 applies. *Supra* §§V.A.2[1c-1d]/[7c-7d]/[13c-13d], n.4; Ex-1004, ¶82.

prediction (e.g., "(pred0)w0"), the second prediction (e.g., "(pred1)w1"), and the rounding offset (e.g., 2^{LWD-1} or 2^{logWD}).

90. This sum of Operation No. 3 (e.g., "(pred0)w0+(pred1)w1+2^{logWD}") is a combined prediction because it combines the first and second predictions. The combined prediction is calculated based on said first prediction (e.g., "(pred0)w0") and said second prediction (e.g., "(pred1)w1"), as well as the rounding offset (e.g., 2^{LWD-1} or 2^{logWD}). Therefore, the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

91. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶34. "The weighted best matching forward macroblock and the weighted best matching backward macroblock are combined to form a weighted combined bi-directional macroblock…" *Id.* Walker's weighting factors, as explained above, are chosen "such that a weighted linear combination of the best matching subsequent and best matching previous macroblocks results in a smaller residual error…" *Id.* These weighting factors are applied for each pixel of the block (e.g., in Operation 1), before obtaining a combined prediction for each of the pixels of the block. *Id.*

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

92. It is my opinion that Walker teaches limitations [1f], [7f], and [13f].

93. As explained above, Equation 13 governs weighted predictions, e.g., Case 3, where "both of the two reference partitions are to be weighted":

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

Ex-1004, ¶¶67-68. Walker teaches executing a series of computational operations outlined in Table 2 based on Equation 13:[8]

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 + $2^{\log WD}$) >> (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

[8] *Supra* §V.A.2[1c]/[7c]/[13c], n.4.

Ex-1004, ¶84; *supra* §§V.A.2[1c-1e]/[7c-7e]/[13c-13e]. Walker teaches obtaining a combined prediction based on its calculation of the sum $(pred0)w0+(pred1)w1+2^{logWD}$ in Operation No. 3. *Supra* §V.A.2[1e]/[7e]/[13e].

94. Walker teaches **shifting bits of the combined prediction to the right** (e.g., ">>(logWD+1)"). Operation No. 4 teaches shifting bits of the sum to the right as emphasized above. Ex-1004, ¶68, ¶84. In this expression, the symbol >> means right bit-shift. Ex-1004, ¶46 ("Digital signal processing functional symbols such as left bit shift (<<) and right bit shift (>>) will be used extensively in this discussion. Such symbols are well known in the art."). (logWD+1) indicates the number of bits shifted.

95. Walker teaches that shifting bits to the right **decreases a precision of said combined prediction**[9] (e.g., from 18 bits to 10 bits). As indicated by the

---

[9] Walker teaches performing additional mathematical operations on the combined prediction. Ex-1004, ¶68, ¶84. For example, Operation No. 5 computes an additive offset (o_0+o_1+1). Ex-1004, ¶69, ¶84. Operation No. 6 adds the additive offset to the combined prediction and uses a "Clip1" function to further limit the bitwidth of the combined prediction. Ex-1004, ¶66, ¶84. These additional operations are allowed by the claims 1, 7, and 13 because each claim recites the transitional term "comprising," which I have been informed indicates that the claims are open-ended and do not exclude additional, unrecited elements or steps. This understanding is confirmed by dependent claims 5, 11, and 17, which like Walker, recite an additional step of "[inserting/insert] a rounding offset to the combined prediction before said decreasing." This additional step is not excluded by the independent claims and does not change the nature of the combined prediction—regardless of whether additional operations are performed (e.g., per claim 5), the combined prediction remains the combined prediction. Therefore, the additional operations

"Bitwidth of Operation Result" column of Table 2, after Operation No. 3, the number of bits needed to represent possible values of the combined prediction is 18. Ex-1004, ¶84. After the right bit shift of Operation No. 4, the number of bits needed to represent possible values of the combined prediction becomes 10. *Id*. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent the combined prediction is decreased from 18 to 10, Walker teaches decreasing a precision of said combined prediction.

96. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶34. "The weighted best matching forward macroblock and the weighted best matching backward macroblock are combined to form a weighted combined bi-directional macroblock…" *Id.* Walker's weighting factors, as explained above, are chosen "such that a weighted linear combination of the best matching subsequent and best matching previous macroblocks results in a smaller residual error…" *Id.* These weighting factors are applied for each pixel of the block (e.g., in Operation 1), before obtaining a combined prediction for each of the pixels of the block and then decreasing the precision, as explained above. *Id.*

---

with respect to the combined prediction, as taught by Walker, are not excluded by claims 1, 7, and 13 and do not affect Walker's teaching of relevant limitations.

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

97. It is my opinion that Walker teaches limitations [1g], [7g], and [13g].

98. First, Walker teaches **determining residual data** (e.g., Walker's "residual error") **based upon a difference between the combined prediction** (e.g., "weighted combined bi-directional macroblock") **and the block of pixels** (e.g., "current macroblock"). As explained above, Walker teaches obtaining a combined prediction and decreasing a precision of the combined prediction using a weighted bi-directional prediction method. *Supra* §§V.A.2[1e-1f]/[7e-7f]/[13e-13f]. The value of the combined prediction is obtained for each pixel of the current block, which includes multiple pixels, based on pixel values of the reference blocks. *See* Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels. Pixels can be defined by an 8-bit luminance value (Y) and two 8-bit chrominance values (Cr and Cb)."), ¶60 ("where pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future)[.]"), ¶34.

99. As was well known to those skilled in the art, when the value of the combined prediction is determined for each pixel of the current macroblock using a weighted bi-directional prediction method based on two reference blocks, the combined prediction for the current block is determined. *Supra* §I.B. The

collective combined prediction values are referred to as "a weighted combined bi-directional macroblock" in Walker. Ex-1004, ¶34 ("The weighted best matching forward macroblock and the weighted best matching backward macroblock are combined to form a weighted combined bi-directional macroblock that is subtracted from current macroblock 515 resulting in residual error 530."). Walker teaches that this "weighted combined bi-directional macroblock [] is subtracted from current macroblock 515 resulting in residual error." *Id*. Thus, the residual error is determined based upon a difference between the combined prediction and the block of pixels. *See* Ex-1004, ¶32 ("The difference between the current picture region and the best matching combined bi-directional prediction region is a residual error (or prediction error)."), ¶34 ("Weighting factors w1 and w2 can be chosen such that a weighted linear combination of the best matching subsequent and best matching previous macroblocks results in a smaller residual error than if equal weights were used…"). Walker's residual error constitutes "residual data" because the residual error is encoded as part of a data stream, as explained below.

100. Second, Walker teaches **encoding the residual data** (e.g., "residual error") **in a bitstream** (e.g., a data stream that includes B frames). Walker teaches encoding the residual error. Ex-1004, ¶34 ("Residual error 530 is encoded with DCT 535 and then quantized 540. The quantized coefficients of residual error 530, motion vectors 525 and 585, weights and reference frame identifying information,

are encoded information representing current macroblock 515."). The encoded residual error is placed in a data stream. Ex-1004, ¶28 ("FIG. 2B is a diagram illustrating a conventional encoded data stream including bi-directional predicted frames, which depicts the frame dependencies of a GOP. … Each B frame can combine forward and backward motion vectors and residual errors referenced to I frame 22A or predicted P frames 24[.]"), Fig. 2B:

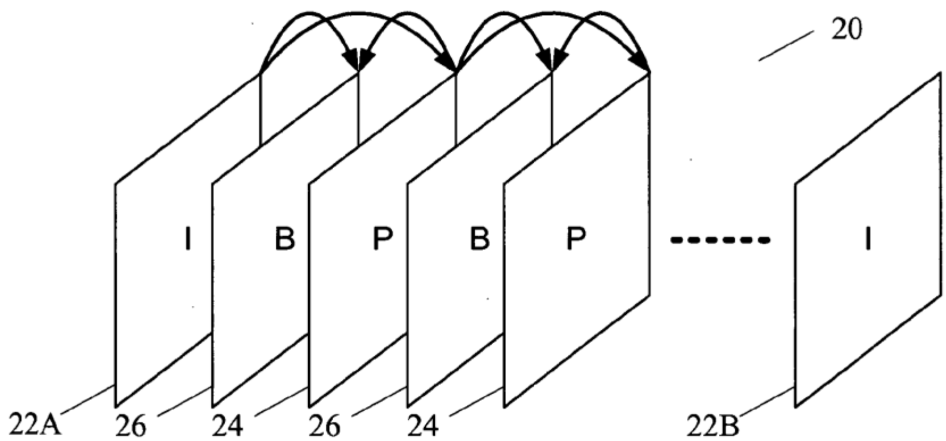


Figure 2B

Because data streams are stored and transmitted as a sequence of bits, they are often referred to as "bitstreams." Therefore, Walker teaches encoding the residual data in a bitstream.

3. Dependent Claims 2, 8, and 14

2. The method according to claim 1,

8. The apparatus according to claim 7,

14. The computer program product according to claim 13,

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

101. Walker teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 2, 8, and 14.

102. First, Walker teaches **an instance in which said first motion vector points to a subpixel** (e.g., an "interpolated pixel"). Walker teaches an encoder[10] using a motion vector to point to either a "pixel or interpolated pixel." Ex-1004, ¶111 ("An encoder can perform pixel interpolation to locate the best matching reference macroblock (or any size section) and point to the pixel or interpolated pixel with a motion vector."), Fig. 8. As further explained below, Walker teaches that these interpolated pixels are subpixels positioned between integer pixels, e.g., half pixels, quarter pixels, or eighth pixels. *See e.g.*, Ex-1004, ¶114, Fig. 9. These subpixels are interpolated from integer pixels, which is why Walker refers to them

[10] As explained above, Walker teaches the process of Figure 8 being used to decode weighted bi-directional predicted video data generated by the encoder devices of Figure 1. *Supra* §V.A.1.

as interpolated pixels. I note that integer pixels are not interpolated. *Supra* §I.C. Thus, Walker's teachings encompass the motion vector pointing to a subpixel.

103. Second, Walker teaches that in an instance in which said first motion vector points to a subpixel, **said first prediction is obtained by interpolation using pixel values of said first reference block**. Walker teaches that "[p]ixel interpolation can be used to improve the performance of motion compensated predictive coding." Ex-1004, ¶114, ¶111 ("Pixel interpolation, step 835, is used to achieve better matching reference regions for motion compensation.").

104. Walker teaches obtaining a prediction for a subpixel by interpolation (e.g., half pixel, quarter pixel, eighth pixel). Ex-1004, ¶114 ("FIG. 9 is an illustration of an example of half-pixel interpolation for use in motion compensation. The example shown is half pixel interpolation where one interpolated pixel is located between each of the original integer pixels. Integer pixels 910 are depicted as circles labeled upper case 'A' to 'I' and the interpolated or half-pixels 920 are depicted as squares labeled lower case 'a' to 'o'. … Other orders of pixel interpolation are supported by various standards. H.264 supports quarter pixel interpolation as well as eighth pixel interpolation. Those of ordinary skill in the art would understand these other pixel interpolation methods and they are not discussed in greater detail herein."), Fig. 9.

105. The half pixel, quarter pixel, and eighth pixel taught by Walker are "subpixels," which refer to fractions of pixels located between full pixels. Ex-1004, ¶114 ("Integer pixels 910 are depicted as circles labeled upper case 'A' to 'I' and the interpolated or half-pixels 920 are depicted as squares labeled lower case 'a' to 'o'."), Fig. 9. I note this is consistent with the use of the term "subpixels" in the '267 patent, which likewise refers to a fraction of a pixel or a pixel position that is between two full pixels. *Compare* Ex-1004, Fig. 9 *with* Ex-1001, 12:56-63, Fig. 12.

<u>Walker</u>

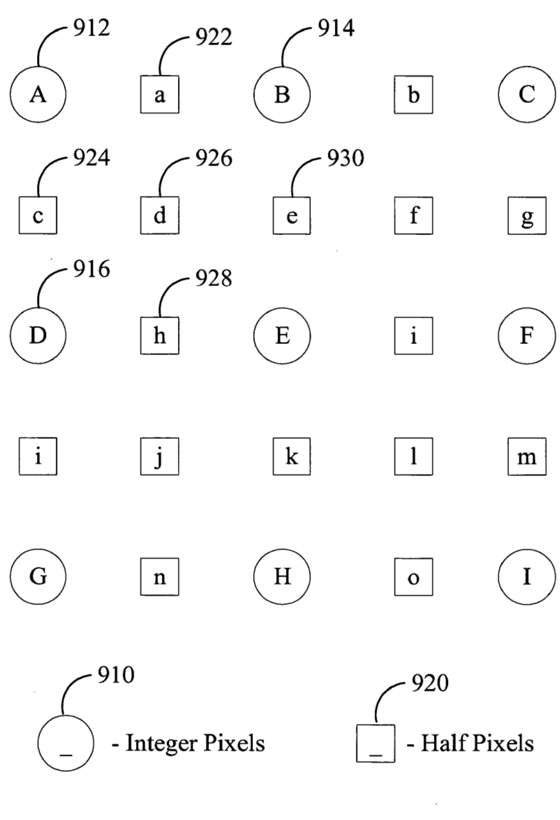


Figure 9

<u>'267 patent</u>

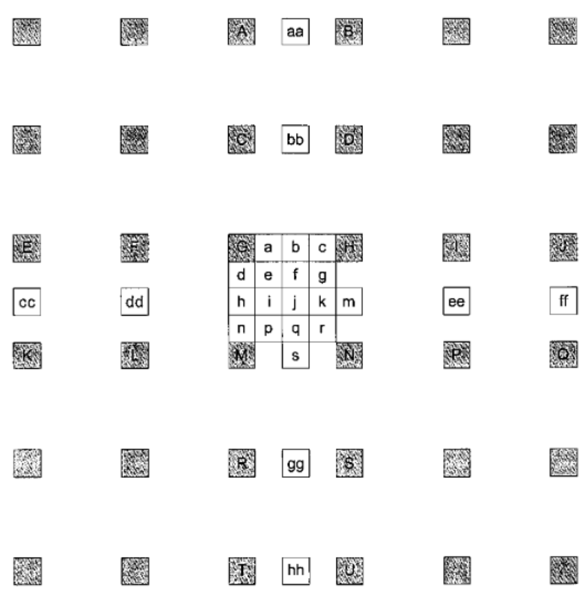


Fig. 12

106. Walker teaches an example of performing interpolation with a 2-tap FIR filter using pixel values of integer pixels neighboring a half pixel. Ex-1004, ¶114 ("Half pixel interpolation can be carried out with a bilinear filter such as, for example, a 2-tap FIR filter with weights [0.5 0.5]. For example, interpolated pixel 922 can be calculated as the average of integer pixel 912 and integer pixel 914, interpolated pixel 924 can be the average of integer pixel 912 and integer pixel 916, and interpolated pixel 926 can be the average of two interpolated pixels (for example, 922 and 928 or 924 and 930)."), Fig. 9. As indicated in Walker's figures, the integer pixels are the pixels in the reference block. *Id.*, Fig. 9. They are the input to the FIR filter that is used for interpolating sub-pixels, including half pixels. Thus, Walker teaches obtaining said first prediction by interpolation using pixel values of said first reference block.

107. As explained for claim 1, Walker teaches using the first motion vector and the first prediction in a bi-directional weighted motion prediction process (*supra* §V.A.2). Walker further teaches "improv[ing] the performance of motion compensated predictive coding" using sub-pixel interpolation. Ex-1004, ¶114. Walker teaches or at least suggests that the interpolation results are used in

determining a weighted prediction for motion prediction.[11] Ex-1004, ¶111 ("The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights wA and wB respectively[.]"). This confirms that it would have been obvious for the first motion vector to point to a subpixel and for the first prediction to be obtained by interpolation.[12]

4. Dependent Claims 3, 9, and 15

---

[11] While Walker's paragraph 111 discusses weighted prediction as part of a decoding process, a POSITA would have understood that this also teaches, or at least suggests, that a matching weighted prediction process is used for encoding because decoding is the inverse process of encoding. *Supra* §I.A. A POSITA would have found this obvious because Walker teaches this (e.g., Ex-1004, Fig. 1) and it has generally been the case for video encoding/decoding since the 1990s. *Supra* §I.A.

[12] Moreover, the '267 patent admits that the limitations of claims 2, 8, and 14 were known in the prior art. Ex-1001, 2:60-3:11. The '267 patent states that, in the "Background" of the patent, including for MPEG-2 and H.264, there were instances where a motion vector points to a subpixel, e.g., "fractional-pixel positions" (Ex-1001, 2:60-65), and in such an instance, the prediction was obtained by interpolation using pixel values of the reference block, e.g., neighboring samples at full-pixel locations. Ex-1001, 2:65-3:11 ("In order to obtain samples at fractional-pixel locations, interpolation filters may be used in the MCP [Motion Compensated Prediction] process. Conventional video coding standards describe how a decoder can obtain samples at fractional-pixel accuracy by defining an interpolation filter. In MPEG-2, for example, motion vectors can have at most, half-pixel accuracy, where the samples at half-pixel locations are obtained by a simple averaging of neighboring samples at full-pixel locations. The H.264/AVC video coding standard supports motion vectors with up to quarter-pixel accuracy. Furthermore, in the H.264/AVC video coding standard, half-pixel samples are obtained through the use of symmetric and separable 6-tap filters, while quarter-pixel samples are obtained by averaging the nearest half or full-pixel samples.").

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

108. Walker teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.A.3. Walker further teaches that said first prediction is obtained by interpolation using values of said first reference block. *Id*. As explained below, it is my opinion that Walker further teaches the additional limitation of claims 3, 9, and 15.

109. Walker teaches performing interpolation for a half pixel using a "2-tap FIR filter," which computes the average of two pixel values with equal weights. Ex-1004, ¶114, Fig. 9. Walker's 2-tap FIR filter is a P-tap filter, where P is 2.[13] And because Walker teaches performing the interpolation using values of said first reference block (*supra* §V.A.3), it teaches a P-tap filter using values of said first

[13] A POSITA would have understood that the term "P-tap filter" was commonly used to describe a filter with multiple taps, where P is a variable that can take an integer value. *See, e.g.*, Ex-1001, 12:60-63 ("a P-tap filter such as a six-tap filter"), 16:25-29.

reference block. As was known in the art, the popular H.264 standard used 6-tap filters for interpolation, where P is 6.[14] *See* Ex-1004, ¶114.

110. Walker teaches one mathematical implementation where this interpolation uses equal filter weights of 0.5 and 0.5 to average nearby integer pixels. Ex-1004, ¶114 ("Half pixel interpolation can be carried out with a bilinear filter such as, for example, a 2-tap FIR filter with weights [0.5 0.5]. For example, interpolated pixel 922 can be calculated as the average of integer pixel 912 and integer pixel 914, interpolated pixel 924 can be the average of integer pixel 912 and integer pixel 916, and interpolated pixel 926 can be the average of two interpolated pixels (for example, 922 and 928 or 924 and 930)."), Fig. 9. This performs the averaging function using weights of 0.5 to effectively divide values by two (multiply by 1/2) when averaging them. *See id.*

111. Walker further teaches an optimization where bit-shifting to the right is used instead of division. Ex-1004, ¶37 ("Another way of normalizing without a division operation is by use of bit shifting. The weights can be derived with a common denominator and the division can be represented by a right shift of the combined weighted prediction a number of bits based on the base 2 logarithm of

---

[14] This was admitted by the '267 patent. Ex-1001, 2:60-3:11 ("Furthermore, in the H.264/AVC video coding standard, half-pixel samples are obtained through the use of symmetric and separable 6-tap filters, while quarter-pixel samples are obtained by averaging the nearest half or full-pixel samples.").

the denominator. For example, w1 could be equal to 12 and w2 could be equal to 4 and the denominator could be 16. The denominator of 16 would translate to a right shift of 4 bits. A right shift of 4 bits is equivalent to dividing by 16, thus w1 would translate to a normalized weight of 0.75 and w2 would translate to a normalized weight of 0.25."), ¶46. When applied to Walker's interpolation teachings (Ex-1004, ¶114), it would have been obvious to avoid multiplication and division by using an FIR filter with equal weights [1 1] and right-shifting the results by 1 bit (e.g., dividing the sum by 2). This calculates the average of the two pixels as Walker teaches (Ex-1004, ¶114) while reducing computational complexity and avoiding multiplication and division as Walker also teaches (Ex-1004, ¶37).

112. A POSITA would have been motivated to apply Walker's teachings in this manner because, in computing, multiplication and division were known to be far more computationally complex than addition and bit-shifting, which is why calculations were often optimized in the art by bit-shifting rather than multiplying or dividing values. *See id.*

5. Dependent Claims 4, 10, and 16

4. The method according to claim 2,

10. The apparatus according to claim 8,

16. The computer program product according to claim 14,

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

113. Walker teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.A.3. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 4, 10, and 16.

114. Walker teaches **instances in which said second motion vector points to an integer sample**. As explained above, Walker teaches that the motion vector may point to an integer pixel or a subpixel and therefore Walker's teachings encompass instances in which said second motion vector points to an integer sample. *Supra* §V.A.3; Ex-1004, ¶111, ¶114, Figs. 8-9. The use of integer pixels was well-known in the art.[15]

115. Walker further teaches, in an instance in which said second motion vector points to an integer sample, **said second prediction is obtained by shifting values of said second reference block to the left**. For both integer and sub-pixels,

---

[15] The '267 patent admits that integer ("full") pixels were known in the art. Ex-1001, 2:60-3:11 ("The motion vectors are not limited to having full-pixel accuracy, but could have fractional-pixel accuracy as well.").

Walker teaches obtaining the second prediction (e.g., "(pred1)w1") as the product of a sample value (e.g., pred1) from a second reference block (e.g., from a future frame) multiplied by a weight (e.g., "w1"). *Supra* §V.A.2[1d]/[7d]/[13d]. This calculation is applied to all the pixels of the block. *E.g.*, Ex-1004, ¶34; *supra* §V.A.2[1d]/[7d]/[13d]. Walker teaches examples where the weight takes a value that is a power of 2 (e.g., 4). Ex-1004, ¶37 ("w2 could be equal to 4"). Walker further teaches that multiplication and bit-shifting are equivalent mathematical operations. Ex-1004, ¶46 ("Those of ordinary skill in the art would understand that the bit shifting operations could be accomplished by other methods such as, for example, applying a scaling factor through multiplication or division."). Therefore, it would have been obvious to implement Walker's teachings, where the sample pixel value from the reference block is multiplied by 4, by left-shifting the pixel value to the left by 2 bits. A POSITA would have found it obvious and been motivated to do so because, in computing terms, left-shifting is a simpler operation than multiplication and can be performed without the need for a multiplication unit.

116. In fact, left-shifting to multiply a value by a coefficient and increase the precision of intermediate values in calculations was well known in the art. For example, in U.S. Patent Application Publication No. 2008/0198935 ("Srinivasan-935") (Ex-1013), "the input is pre multiplied by 8 (i.e. left shifted by 3 bits)" "[f]or

the sake of improved coding performance by the reduction of rounding errors[.]" Ex-1013, ¶91; *see also Id.*, ¶116, ¶124, ¶129 ("[T]o minimize the damage of truncation errors and thus maximize transform performance, input data to a transform needs to be left shifted several bits."), ¶131 ("One way to reduce the damage of truncation errors is to left shift the input data[.]"). As another example, U.S. Patent Application Publication No. 2013/0034158 ("Kirchhoffer-158") (Ex-1015) discusses increasing the bit-depth representation of a value by left-shifting the value with a predetermined number of bits. Ex-1015, ¶84 ("A precision may also be called a bit-depth representation, wherein a higher precision corresponds to a higher bit-depth representation. A bit-depth representation of a value may be increased by left-shifting the value with a predetermined number of bits. A left-shift corresponds to a multiplication with 2."); *see also Id.*, ¶85 ("[A]n increase in the precision of the reconstructed reference image samples[] may be obtained by left-shifting each value of these reconstructed reference image samples[.]"), ¶67, ¶103. The explicit teachings of Srinivasan-935 and Kirchhoffer-158 further confirm that it would have been obvious to perform left-shifting in calculating the weighted prediction.

6. Dependent Claims 5, 11, and 17

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

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

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

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

117. Walker teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 5, 11, and 17.

118. Walker teaches **inserting a rounding offset** (e.g., 2^{LWD-1} or 2^{logWD}) **to the combined prediction** as part of Operation No. 3 **before said decreasing** of the precision in Operation No. 4. As explained above, Walker teaches obtaining a combined prediction via its disclosure of calculating a sum by adding up the first prediction (e.g., (pred0)w0), the second prediction (e.g., (pred1)w1), and the rounding offset (e.g., 2^{LWD-1}, which equals 2^{logWD} for Case 3, *supra* §V.A.2[1c]/[7c]/[13c], n.4) in Operation No. 3 of Table 2. *Supra*

§V.A.2[1e]/[7e]/[13e]; Ex-1004, ¶68, ¶84. Therefore, Walker inserts a rounding offset to the combined prediction before its next step, in Operation No. 4 of Table 2, where Walker decreases a precision of said combined prediction by shifting bits of the combined prediction to the right. *Supra* §V.A.2[1f]/[7f]/[13f]; Ex-1004, ¶68, ¶84:

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

119. Walker explains that 2^{LWD-1} is a **rounding offset**. Walker refers to variables in Equation 14, including 2^{LWD-1} as "weighting factor variables and offset variables[.]" Ex-1004, ¶74. The value 2^{LWD-1} is an offset variable. A POSITA would have understood this because it is added to, rather than multiplied with,

sample values, thereby providing an offset for the sample value. Moreover, the offset variable 2^{LWD-1} is inserted into the combined prediction before a right-shift operation, which causes values to round to the closest integer after the right shift, rather than always rounding down which would be the result absent the offset. In light of this effect on the rounding operation, a POSITA would have understood that the offset variable 2^{LWD-1} is a rounding offset. As explained above, for bi-directional motion prediction of Case 3, $2^{LWD-1}=2^{logWD}$. *Supra* §V.A.2[1c]/[7c]/[13c], n.3. Therefore, 2^{LWD-1} and 2^{logWD} are both rounding offsets.

120. Walker teaches this rounding offset as part of its rounding process, which decreases the precision as recited by the claims. Ex-1004, ¶68, ¶84. Additionally, it was obvious for said decreasing to include the rounding offset (claim 5) because the insertion of the rounding offset is performed right before and in conjunction with the right-shifting to affect the direction of the rounding. Walker includes a rounding offset to control rounding error resulting from the right-shift operation that decreases precision. This was common in the art. *Supra* §I.D.

121. Regarding claim 11, as explained above, Walker teaches at least one memory and computer program code are configured to cause the apparatus to perform the operations recited in limitations [7e]-[7f]. *Supra* §V.A.2[7a]. Since Walker applies its teachings to a computer implementation, Walker also applies the

above-described teachings, for decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, to a computer implementation. Therefore, it would have been obvious that **the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right by** inserting a rounding offset to the combined prediction before said decreasing (claim 11).

122. Likewise, for claim 17, as explained above, Walker applies its teachings to a computer implementation, including for program code instructions that are configured to perform the operations recited in claim 13, including limitations [13e]-[13f]. *Supra* §V.A.2[13a]. Therefore, for reasons explained above, Walker teaches that **the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right further comprise program code instructions configured to** inserting a rounding offset to the combined prediction before said decreasing (claim 17).

7. Dependent Claims 6, 12, and 18

6. The method according to claim 1,

12. The apparatus according to claim 7,

18. The computer program product according to claim 13,

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

123. Walker teaches the method according to claim 1, the apparatus according to claim 7, and the computer program product according to claim 13. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 6, 12, and 18.

124. As explained above, Walker teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision because 8 bits are needed to represent the possible pixel values of these blocks. *Supra* §V.A.2[1b]/[7b]/[13b]. Here, **the first precision indicates a number of bits needed to represent the values of the pixels**.

125. As explained above, Walker teaches said first prediction having a second precision, which is higher than said first precision, and said second prediction having the second precision because 16 bits are needed to represent the possible values of the first prediction and the second prediction. *Supra* §§V.A.2[1c-1d]/[7c-7d]/[13c-13d]. Here, **the second precision indicates the**

number of bits needed to represent values of said first prediction and values of said second prediction.

B. Ground 2: Claims 1-18 Are Rendered Obvious by Karczewicz-I in View of Karczewicz-II

126. It is my opinion that a POSITA would have found Claims 1-18 obvious based on the combination of Karczewicz-I and Karczewicz-II.

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

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

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

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

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

2. U.S. Patent Application Publication No. 2009/0257499 Karczewicz (“Karczewicz-II”)

131. I have reviewed the Karczewicz-II reference. I understand that Karczewicz-II is cited on the face of the ’267 patent. Karczewicz-II was cited in an Information Disclosure Statement filed June 7, 2021 along with 76 other references. Ex-1007, 000074-78. Other than acknowledging the Information Disclosure Statement, the Examiner did not discuss Karczewicz-II during the prosecution history or use Karczewicz-II in any rejections.

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

133. Karczewicz-II teaches block-based techniques for motion prediction. *E.g.*, Ex-1006, ¶8, ¶¶35-36, ¶46, ¶54. Karczewicz-II teaches improved calculations

for predictions involving interpolated fractional pixel positions. Ex-1006, ¶2, ¶10, ¶¶93-106. Karczewicz-II teaches that prediction values for quarter-pixel positions can be calculated as an average of two adjacent integer or half-pixel positions. Ex-1006, ¶¶96-102. Karczewicz-II teaches such calculation for multiple different scenarios. For example, Karczewicz-II teaches a quarter-pixel position between an integer pixel and a half-pixel position is calculated as an average of the predictions of the integer pixel and the half-pixel position. Ex-1006, ¶96, ¶99, ¶103, Tables 1, 3, 5. In addition, Karczewicz-II teaches a quarter-pixel position between a center-pixel position (i.e., the position that is at the center of four integer pixels) and a half-pixel position is calculated as an average of predictions for the two positions. Ex-1006, ¶96, ¶99, ¶108, Tables 1, 3, 8. Moreover, Karczewicz-II teaches a quarter-pixel position between two half-pixel positions is calculated as an average of predictions for the two positions. Ex-1006, ¶97, ¶101, ¶103, Tables 2, 4, 6.

134. Karczewicz-II teaches an improved method for averaging interpolated pixel values. Specifically, Karczewicz-II discloses maintaining higher precision for intermediate values (e.g., integer or half-pixel predictions) during calculation and delaying rounding until later in the process in order to reduce rounding inaccuracies. *See, e.g.*, Ex-1006, ¶10, ¶39, ¶53, ¶59, ¶¶99-106.

135. Karczewicz-II is in the same field of endeavor as the ’267 patent (video encoding/decoding). *See supra* §III. Karczewicz-II is directed to “digital

video coding” and “interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding.” Ex-1006, ¶2, Abstract. Karczewicz-II uses and improves the same known standards (e.g., H.264) as the ’267 patent for video encoding and decoding. *Compare* Ex-1006, ¶¶46-47, ¶82, ¶88, ¶¶93-0106 *with* Ex-1001, 2:60-3:11, 9:26-41.

3. Motivation to Combine and Reasonable Expectation of Success

136. Karczewicz-I and Karczewicz-II are Qualcomm patent applications by the same inventors Marta Karczewicz, Peisong Chen, and Yan Ye. Both are directed to video coding and apply their teachings to similar architectures. Ex-1005, ¶2, Fig. 1; Ex-1006, ¶2, Fig. 1

<u>**Karczewicz-I**</u> <u>**Karczewicz-II**</u>

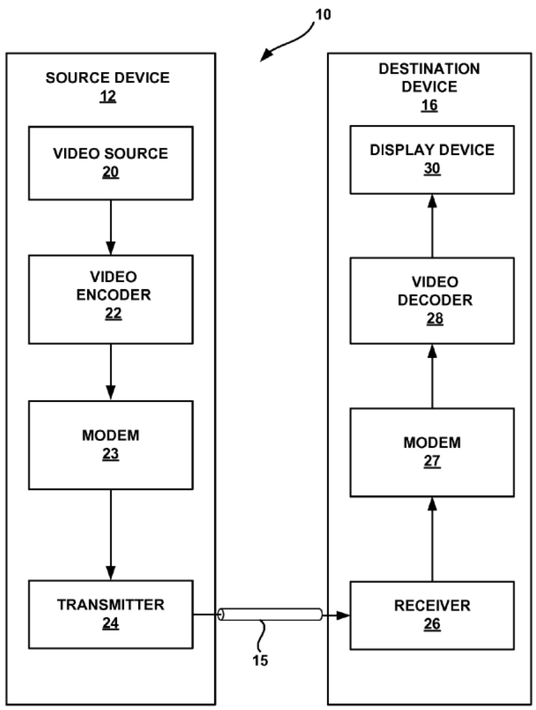


FIG. 1

Ex-1005, Fig. 1, ¶¶29-50.

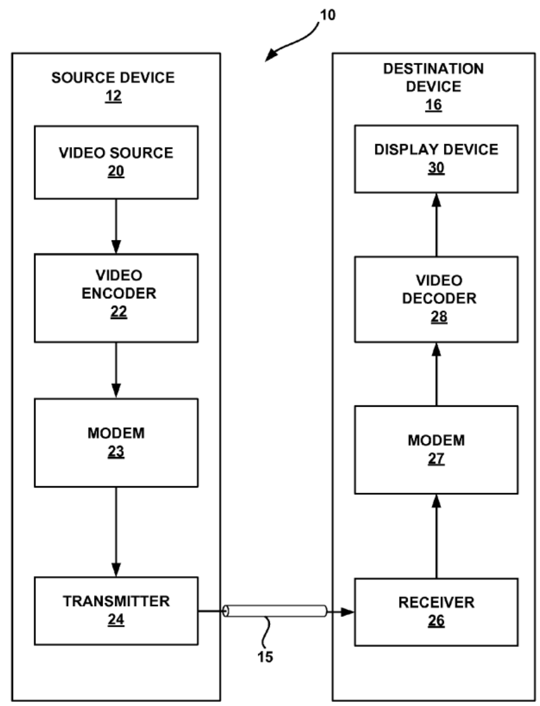


FIG. 1

Ex-1006, Fig. 1, ¶¶40-53.

137. Karczewicz-I and Karczewicz-II are both directed to block-based, e.g., H.264, techniques for motion prediction. Ex-1005, ¶35, ¶44, ¶¶55-60; Ex-1006, ¶8, ¶¶35-36, ¶46, ¶54. Both include teachings for a video encoder that performs motion estimation and compensation for inter-predictive coding:

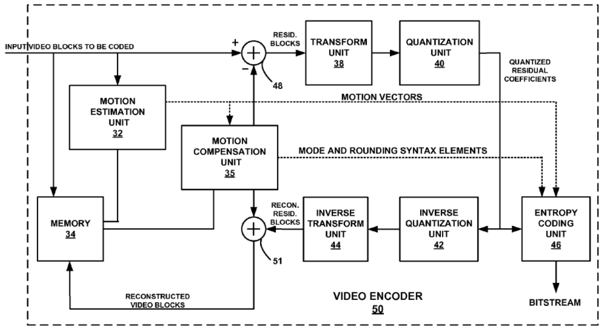


FIG. 2

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

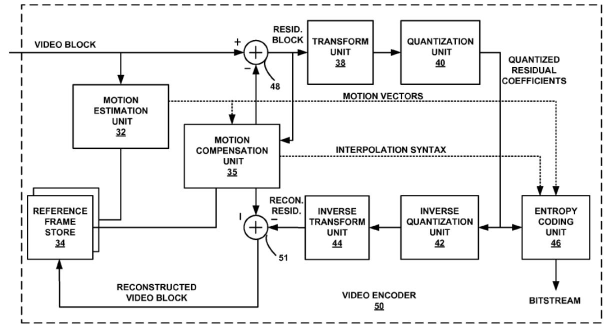


FIG. 2

Ex-1006, Fig. 2, ¶56 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding.").

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

techniques taught by Karczewicz-I and techniques taught by Karczewicz-II using that common architecture.

139. This would have been a combination of prior art elements according to known methods. Karczewicz-I teaches bi-prediction techniques for the H.264 standard with two motion vectors pointing to two blocks of pixels values that are averaged together. Ex-1005, ¶53, ¶35, ¶44, ¶60. For interpolated sub-pixels, this calculation averages the interpolated values together. Beyond integer pixels, Karczewicz-II teaches that, for H.264, motion vectors can also point to fractional sub-pixels (Ex-1006, ¶¶56-58, ¶¶93-102),[16] and Karczewicz-II teaches an improved calculation for averaging interpolated pixel values, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate calculations (E.g., Ex-1006, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-106). Therefore, Karczewicz-II provides complementary teachings that improve Karczewicz-I's teachings for averaging predicted pixel values.

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

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

accuracy of the average. *See, e.g.*, Ex-1006, ¶102 (“By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate.”), ¶10 (“In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values.”), ¶39, ¶53, ¶59. Therefore, a POSITA who understood fundamental computer logic concepts (e.g., binary arithmetic, bit shifting, precision control, rounding and offsetting, and error analysis) would have been motivated to apply Karczewicz-II’s improved calculations to Karczewicz-I’s teachings.

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

same calculations for interpolating and averaging pixel values when motion vectors point to interpolated sub-pixels. Therefore, since Karczewicz-II teaches an improved implementation of the mathematical calculations taught by Karczewicz-I, a POSITA would have been motivated to use the known technique of Karczewicz-II to improve similar devices/methods, as taught by Karczewicz-I, in the same way, using higher precision when combining interpolated pixel values, as Karczewicz-II teaches. This would have achieved the same benefits taught by Karczewicz-II, e.g., eliminating the propagation of rounding inaccuracies and improving prediction accuracy. *E.g.*, Ex-1006, ¶102, ¶10. And it would have had predictable results because it applies teachings from Karczewicz-II to corresponding mathematical calculations used for Karczewicz-I, using similar video encoding architectures as explained above. A POSITA would have recognized the applicability of Karczewicz-II to the corresponding calculations in Karczewicz-I, which was a simple matter given the level of ordinary skill. Notably, a POSITA would have understood fundamental computer logic concepts (e.g., binary arithmetic, bit shifting, precision control, rounding and offsetting, and error analysis) and mathematical calculations for known motion estimation and compensation techniques (e.g., bi-directional prediction, determination and use of motion vectors, interpolation for fractional pixels) because they are integral to working with video codecs.

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

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

143. Karczewicz-I's discussion focuses on methods for combining two predictions, relying on known techniques, such as the techniques under the H.264 standard, for determining the two predictions to be combined. *See, e.g.*, Ex-1005, ¶35 ("Video encoder 22 and video decoder 28 may operate according to a video compression standard, such as the ITU-T H.264 standard[.]"). The H.264 standard provides that, when a motion vector points to a fractional pixel/subpixel, interpolation is used to calculate a prediction. *Supra* §I.C; Ex-1006, ¶¶68-69, ¶74, ¶93. Consistent with H.264, Karczewicz-I teaches that the inter-predictive coding

process, where motion vectors point to different frames, includes interpolation, and therefore provides express teaching, suggestion, and motivation ("TSM") to combine with known teachings for interpolation. Ex-1005, ¶41 ("Following inter-based predictive encoding (which includes interpolation and the techniques of this disclosure to efficiently select a prediction algorithm or mode by which to predict a coded unit) …").

144. Karczewicz-II teaches improved calculations for predictions involving interpolated fractional pixel positions in H.264, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate values. *See, e.g.*, Ex-1006, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."), ¶99, ¶10. Karczewicz-II explains that, according to the H.264 standard, predictions for half-pixel positions are calculated using 6-tap interpolation filters that interpolate the sub-pixel based on nearby pixels in the same row (e.g., half-pixel "b") or column (e.g., half-pixel "h"). Ex-1006, ¶74, ¶¶93-94, Fig. 4A-4B:

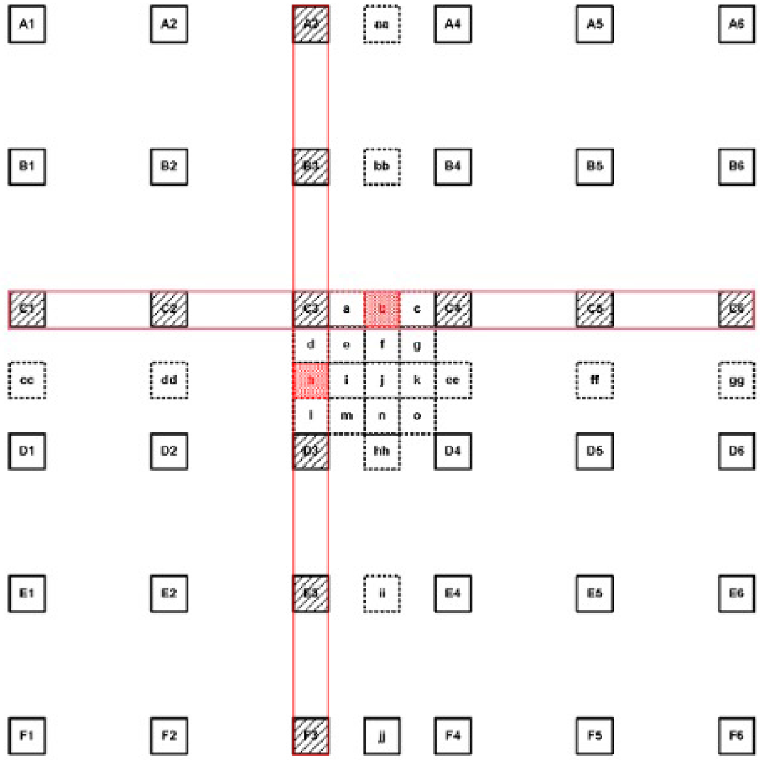


FIG. 4B

145. This process involves two steps. First, a 6-tap filter multiplies the six pixels in the row (or column) by the filter values (1, −5, 20, 20, −5, 1), adds the products together to produce a non-rounded prediction (e.g., b1). Second, the result is rounded down, e.g., using a right-shift operation (">>") that decreases the number of bits needed to represent the prediction back down to the original precision.[17] *Supra* §I.D. Here is the process for interpolating sub-pixel "b":

---

[17] A POSITA would have understood that the right shifting operation is equivalent to dividing the weighted sum by the total weight. In binary computation, right

b1=C1−5*C2+20*C3+20*C4−5*C5+C6

…

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

Ex-1006, ¶¶93-94, Fig. 4B. This produces a rounded prediction (e.g., b).

146. For half-pixel positions that are in the center of four integer pixels on two dimensions (e.g., j) (referred to as a "center-pixel position" hereinafter), the interpolation process is applied in two rounds: one to interpolate a middle row of half pixels, and another to interpolate that middle row into the center pixel. In other words, the 6-tap interpolation filter is applied on values of six half-pixel predictions. Ex-1006, ¶95, Fig. 4C.

147. Karczewicz-II teaches calculating quarter-pixel predictions by averaging the predictions of the two nearest integer or half-pixel positions as explained above. Ex-1006, ¶¶96-97. Karczewicz-II teaches an improvement to reduce coding inefficiencies and increase the accuracy of pixel calculations by "keep[ing] the highest possible precision through the intermediate steps" and avoiding "any shifting, rounding and clipping operations[] until the very last step of the interpolation process." Ex-1006, ¶99. This prevents the propagation of rounding errors. Ex-1006, ¶102 ("By preserving the full precision of the

---

shifting by a number of bits is equivalent to division by 2 to the power of the number of bits shifted. For example, right shifting by 5 bits is equivalent to dividing the weighted sum by 2^5, which equals 32; here, 32 is the sum of the six weights of the 6-tap filter (i.e., 1-5+20+20-5+1=32).

intermediate values, the interpolated sub-pixels will be more accurate.”), ¶10 (“In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values.”), ¶39, ¶53, ¶59.

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

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

$$\text{pred}(i,j)=(\text{pred0}(i,j)+\text{pred1}(i,j)+1)>>1$$

Ex-1005, ¶¶58-60. As explained above, H.264 allows motion vectors to point to integer pixels or fractional pixels, such as half pixels, and uses interpolation to calculate predictions for fractional pixels. *See e.g.*, Ex-1005, ¶41; Ex-1006 ¶¶93-

102 ("A sub-pixel motion vector refers to a sub-pixel position in a reference picture which needs to be interpolated."), ¶¶56-58; *supra* §I.C.[18] Therefore, the predictions pred0(i,j) and pred1(i,j) encompass scenarios that include integer pixel prediction, a half-pixel prediction, or a center-pixel prediction, and Karczewicz-I calculates an average of these pixel values.

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

151. Scenario 1 (with the first motion vector pointing to a half-pixel position and the second motion vector pointing to an integer pixel position). Beyond integer pixels, Karczewicz-II explains how, for H.264, motion vectors can also point to half-pixel positions. *E.g.*, Ex-1006, ¶¶93-102, ¶¶56-58. When the two motion vectors in Karczewicz-I's bi-directional prediction point to a half-pixel position and an integer pixel position, the default weighted prediction is calculated as an average of the half-pixel and the integer pixel. *See* Ex-1005, ¶60, ¶55.

---

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

152. Karczewicz-II teaches an improved calculation for averaging a half-pixel value with an integer pixel value. While Karczewicz-II uses this calculation to interpolate a quarter-pixel position, it is the same calculation as Scenario 1 because it averages an integer and half pixel. *See* Ex-1006, ¶96, Fig. 4D. Karczewicz-II explains the conventional calculation for averaging two numbers: adding the half-pixel value (e.g., “b”) with the integer pixel value (e.g., C3) and a rounding offset 1, then dividing by 2 (using a right-shift “>>” operation that is mathematically equivalent to dividing by 2). Ex-1006, Fig. 4D, Table 1:

TABLE 1

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

153. Karczewicz-II improves this conventional approach by “keep[ing] the highest possible precision through the intermediate steps[.]” Ex-1006, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex-1006, 99. The integer pixel value (e.g., C3) is multiplied by 32 (by bit-shifting to the left 5 bits, which is mathematically equivalent), taking its precision from 8 to 13 bits. Ex-1006, Table 5. Instead of using a rounded 8-bit half-pixel (e.g., b), Karczewicz-II delays the

rounding step and instead uses a *non-rounded* half-pixel prediction (e.g., b1) that is 15 bits. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (">>6"). *Id.*, Table 3:

TABLE 3

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

154. The operations for this improved approach are shown in Table 5 of Karczewicz-II. Ex-1006, ¶103 ("The following Tables show the interpolation process for other sub-pixels in sixteen bit storage elements. In the Tables below, the operations defined in each column are performed sequentially through the respective table."), Table 5:

TABLE 5

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

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

155. Since Karczewicz-II teaches this optimization for averaging an interpolated half-pixel and an integer pixel, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of an integer pixel and an interpolated half-pixel in Scenario 1. For example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex-1005, ¶60):

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

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

156. For Scenario 1, the two predictions would be a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)). *See* Ex-1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel prediction (pred0(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied (Ex-1006, ¶¶93-94). To match this higher precision, the integer pixel (pred1(i,j)) is left-shifted, as Karczewicz-II teaches. Ex-1006, ¶99, Table 3. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶96, ¶99, Tables 1 and 3; Ex-1005, ¶60.

157. <u>Scenario 2</u> (with the first motion vector pointing to a center-pixel position and the second motion vector pointing to a half-pixel position). Karczewicz-II explains how, for H.264, motion vectors point to center- and half-pixel positions. *E.g.*, Ex-1006, ¶¶93-95. When the two motion vectors in Karczewicz-I's bi-directional prediction point to a center- and half-pixel position,

the default weighted prediction is calculated as an average of the center- and half-pixels. *See* Ex-1005, ¶60, ¶55.

158. Karczewicz-II teaches an improved calculation for averaging a center- and half-pixel. Ex-1006, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 2 because it averages a center and half pixel. As discussed above, Karczewicz-II explains the conventional calculation for averaging two numbers: adding the center-pixel value (e.g., "j") with the half-pixel value (e.g., "b") and a rounding offset 1, then dividing by 2 (using a right-shift ">>" operation that is mathematically equivalent to dividing by 2). Ex-1006, ¶96, Fig. 4D, Table 1:

TABLE 1

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

159. Karczewicz-II improves this conventional approach by "keep[ing] the highest possible precision through the intermediate steps[.]" Ex-1006, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex-1006, ¶99. Whereas Table 1 used a fully-rounded center-pixel (e.g., "j"), Table 3 uses a partially-rounded pixel

(e.g., “j1>>5”) that remains 5 bits longer than the rounded value in Table 1.[19] *Infra* §V.B.6. Likewise, Table 3 replaces the rounded half-pixel (e.g., “b”) with a non-rounded half-pixel (e.g., “b1”) that is 15 bits. These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (“>>6”). Ex-1006, Table 3:

TABLE 3

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

160. The operations for this improved approach are shown in Table 8 of Karczewicz-II. Ex-1006, ¶105 (“Table 8, below demonstrates steps that can be taken for sixteen-bit implementation of interpolating {f,i,k,n}, which are the positions that use to interpolate the intermediate value ‘j1.’”), Table 8:

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

TABLE 8

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

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

161. Since Karczewicz-II teaches this optimization for averaging interpolated center- and half-pixels, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of an interpolated center pixel and an interpolated half-pixel in Scenario 2. For example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex-1005, ¶60):

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

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

162. For Scenario 2, the two predictions would be a center prediction (e.g., pred0(i,j)) and half-pixel prediction (e.g., pred1(i,j)). *See* Ex-1005, ¶60, ¶55. A

POSITA would have been motivated to apply Karczewicz-II's teachings by only partially rounding the center pixel by 5 bits rather than 10 bits to keep it at a higher precision, as Karczewicz-II teaches. *E.g.*, Ex-1006, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-103. Karczewicz-II uses the non-rounded half-pixel, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex-1006, ¶74, ¶93, ¶¶99-100. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶96, ¶99, Tables 1 and 3; Ex-1005, ¶60.

163. Scenario 3 (with both motion vectors pointing to half-pixel positions). Karczewicz-II explains how, for H.264, motion vectors point to pixels, including half-pixel positions. *E.g.*, Ex-1006, ¶93. When the two motion vectors in Karczewicz-I's bi-directional prediction point to two half-pixel positions, the default weighted prediction is calculated as an average of the two half-pixels. *See* Ex-1005, ¶60, ¶55.

164. Karczewicz-II teaches an improved calculation for averaging two half-pixel values. Ex-1006, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 3 because it averages two half pixels. As discussed above, Karczewicz-II explains

the conventional calculation for averaging two numbers: adding the half-pixel values (e.g., “b” and “h”) and a rounding offset 1, then dividing by 2 (using a right-shift “>>” operation that is mathematically equivalent to dividing by 2). Ex-1006, ¶97, Fig. 4D, Table 2:

TABLE 2

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

165. Karczewicz-II improves this conventional approach by “keep[ing] the highest possible precision through the intermediate steps[.]” Ex-1006, ¶99, ¶101. Karczewicz-II replaces the equations in Table 2 with those in Table 4, where the pixel values are combined at a higher precision. Ex-1006, ¶101. Instead of using a rounded 8-bit half-pixel (e.g., “b” or “h”), Karczewicz-II delays the rounding step and instead uses *non-rounded* half-pixels (e.g., “b1” and “h1”) that are 15 bits each. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (“>>6”). *Id.*, Table 4:

TABLE 4

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

166. The operations for this improved approach are shown in Table 6 of Karczewicz-II. Ex-1006, ¶103, Table 6:

TABLE 6

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

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

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

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

Bidirectional prediction: $\text{pred}(i,j)=(\text{pred0}(i,j)+\text{pred1}(i,j)+1)>>1$

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

168. For Scenario 3, the two predictions would be half-pixel predictions. *See* Ex-1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel predictions (pred0(i,j) and pred1(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex-1006, ¶¶97-101. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶97, ¶101, Tables 2 and 4; Ex-1005, ¶60.

169. Express Teaching Suggestion or Motivation (TSM) in the Art. Preserving higher-precision intermediate values in computations related to video coding was well known in the art as a method for improving the accuracy of computation; the prior art therefore provides express TSM to apply such teachings from Karczewicz-II to related techniques in Karczewicz-I to achieve a higher accuracy for Karczewicz-I's bi-directional prediction. *See, e.g.*, Ex-1016, Abstract

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

170. Karczewicz-I is directed to motion compensation. Preserving higher-precision intermediate values was known to benefit motion compensation. For example, U.S. Patent No. 9,344,744 ("Kirchhoffer-744") (Ex-1008) teaches performing "a prediction and a reconstruction for a block of a picture to be predicted … in a higher precision." Ex-1008, 7:4-19, 16:22-17:36. This approach was known to result in a more precise prediction and a smaller residual information. *Id*. ("An advantage of embodiments according to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."). Therefore, the prior art provides express TSM to apply Karczewicz-II's teachings to Karczewicz-I and thereby perform motion compensation at a higher precision. *See id.*

171. It was also known that preserving higher-precision intermediate values improves the accuracy of interpolation. *See, e.g.*, Ex-1009, 000005 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding

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

172. Compatible Teachings. The combination would not have changed the principle of operation of either reference, but merely includes the use of a known technique (e.g., Karczewicz-II's technique of calculating the average of two predictions with higher-precision intermediate values) to improve similar devices or methods (e.g., Karczewicz-I's system and method for encoding video using bi-directional prediction) in the same way. As explained above, Karczewicz-I and Karczewicz-II are filed by the same inventors from the same company and teach similar video encoding and decoding methods implemented on similar video encoding and decoding systems, both for averaging predicted pixel values in H.264. Ex-1005, Figs. 1-2, ¶¶29-50, ¶53; Ex-1006, Figs. 1-2, ¶¶40-53, ¶56. Given the similarities between Karczewicz-I and Karczewicz-II, a POSITA would have understood that Karczewicz-II's techniques are readily applicable to Karczewicz-I.

173. Moreover, the combination merely changes when rounding occurs for calculations that already included rounding. *See* Ex-1005, ¶60, ¶55; Ex-1006, ¶¶96-106, Tables 1-8. This minor implementation detail would not have changed the principle of operation of Karczewicz-I. Nor would it have changed the

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

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

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

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

4. Independent Claims 1, 7, and 13

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

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

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

Fig. 1. The video encoder "perform[s] … inter-coding of blocks within video frames" that includes motion estimation and compensation operations. Ex-1005, ¶¶51-76, Fig. 2.

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

179. Karczewicz-II "describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex-1006, Abstract, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."). Karczewicz-II teaches "method[s] of encoding video data." Ex-1006, ¶12. Karczewicz-II's methods of encoding video data include block-based inter-coding that includes motion estimation and compensation operations. Ex-1006, ¶4 ("Block based inter-coding is a very useful coding technique that relies on temporal prediction to reduce or remove temporal redundancy between video blocks of successive coded units of a video sequence. … For inter-coding, the video encoder performs motion estimation and motion compensation to track the

movement of corresponding video blocks of two or more adjacent coded units.”). Similar to Karczewicz-I, Karczewicz-II teaches a “video encoding and decoding system” that includes a “video encoder 22” that encodes video data. Ex-1006, ¶¶40-41, ¶43 (“In each case, the captured, pre-captured or computer-generated video may be encoded by video encoder 22.”), Fig. 1. The video encoder “perform[s] … inter-coding of blocks within video frames” that includes motion estimation and compensation operations. Ex-1006, ¶¶54-62, Fig. 2.

180. Similar to Karczewicz-I, Karczewicz-II encodes a block of pixels. *See, e.g.*, Ex-1006, ¶49 (“Video encoder 22 operates on video blocks within individual video frames in order to encode the video data. … Video blocks may comprise blocks of pixel data…”), ¶50 (“In general, macroblocks (MBs) and the various sub-blocks may be considered to be video blocks.”).

181. Karczewicz-II teaches interpolation methods for use in video encoding. *See, e.g.*, Ex-1006, ¶14 (“In another example, this disclosure provides a method of interpolating predictive video data for video coding.”), ¶35 (“This disclosure describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding.”). As explained above, a POSITA would have found it obvious to modify Karczewicz-I’s technique for obtaining a combined prediction, which is part of Karczewicz-I’s video encoding method, based on Karczewicz-II’s interpolation techniques. *Supra*

§V.B.3 (explaining the motivation to combine; that analysis is incorporated here). Thus, the combination of Karczewicz-I and Karczewicz-II teaches a method for encoding a block of pixels.

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

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

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

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

encoder for encoding a block of pixels that performs operations as described for limitations [7b]-[7g]. *Infra* §§V.B.4[7b-7g].

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

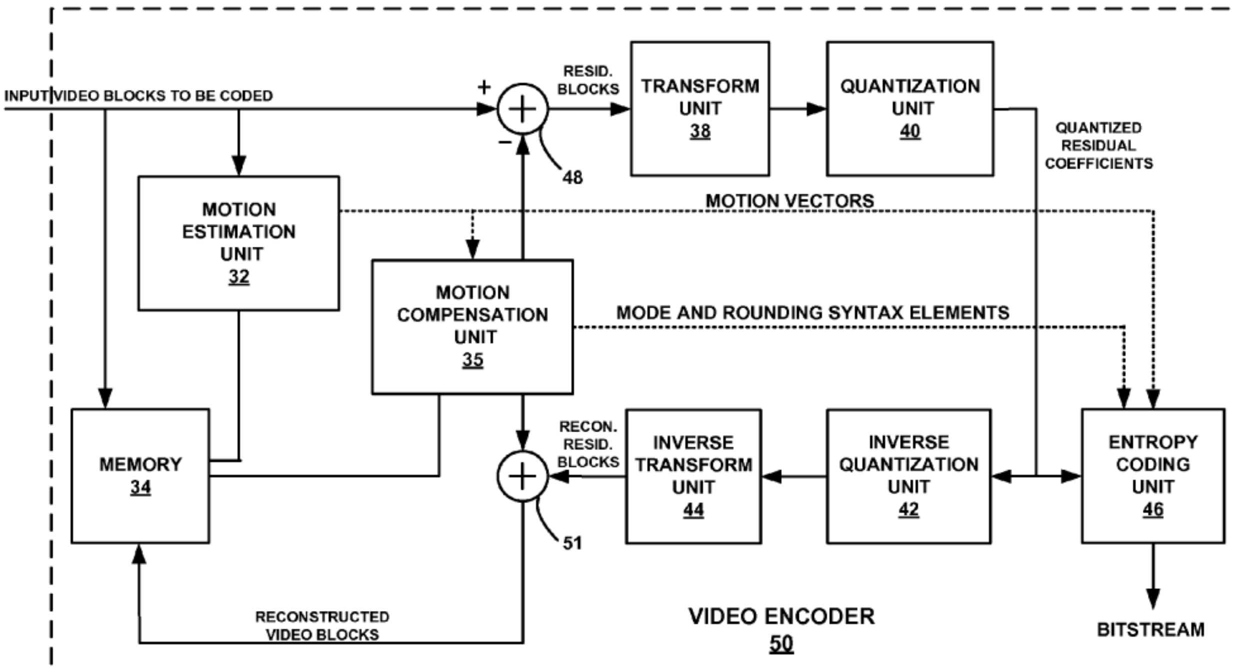


FIG. 2

Ex-1006, ¶¶40-41, ¶43, Fig. 1, ¶¶54-62, Fig. 2.

186. Karczewicz-I and Karczewicz-II teach that the video encoder is an apparatus for encoding video. Ex-1005, ¶10 ("[T]his disclosure describes a video encoder apparatus that encodes video data[.]"); Ex-1006, ¶15 ("[T]his disclosure provides an apparatus that encodes video data, the apparatus comprising a video encoder …"), ¶17, ¶21, ¶23. Karczewicz-I and Karczewicz-II teach the video

encoder encodes a block of pixels. *See, e.g.*, Ex-1005, ¶¶39-40; Ex-1006, ¶¶49-50. Karczewicz-I and Karczewicz-II teach implementing the video encoder in various types of devices. Ex-1005, ¶3 ("Digital multimedia capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless communication devices, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, video gaming devices, video game consoles, cellular or satellite radio telephones, digital media players, and the like."); Ex-1006, ¶3.

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

integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof."), ¶98; Ex-1006, ¶25.

188. Karczewicz-I and Karczewicz-II explain that the computer readable medium comprises well-known types of memories. Ex-1005, ¶99 ("The computer-readable storage medium may comprise random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like."); Ex-1006, ¶119. Karczewicz-I and Karczewicz-II further explain that its software includes computer program code. Ex-1005, ¶100 ("The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry."); Ex-1006, ¶26, ¶120.

189. Karczewicz-I and Karczewicz-II further teaches the processor executing computer program code in memory to cause the apparatus to carry out its functionalities. Ex-1005, ¶99 ("If implemented in software, the techniques may be realized at least in part by a computer-readable medium comprising instructions

that, when executed in a processor, performs one or more of the methods described above.”), ¶100 (“The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry.”); Ex-1006, ¶48, ¶¶119-120.

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

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

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

192. As explained above, Karczewicz-I and Karczewicz-II teach implementing the video encoder and performing operations for encoding a block of pixels using software stored in computer readable storage media and executed by a processor. *Supra* §V.B.4[7a]; Ex-1005, ¶12, ¶38, ¶¶98-100; Ex-1006, ¶¶25-26, ¶48, ¶¶119-120. A POSITA would have understood that the types of computer readable storage media disclosed by Karczewicz-I and Karczewicz-II, including random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, are forms of non-transitory computer readable storage medium, as opposed to transitory signals. Ex-1005, ¶99; Ex-1006, ¶119. Karczewicz-I and Karczewicz-II teach that the software, which is stored in computer readable storage medium and executable by a processor, includes computer executable program code portions that comprise program code instructions. Ex-1005, ¶100; Ex-1006, ¶26, ¶120.

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

comprise a computer-readable storage medium and may form part of a computer program product …"); Ex-1006, ¶119. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches a computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to perform the operations recited in claim 13.

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

194. It is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches limitations [1b], [7b], and [13b]. Karczewicz-I and Karczewicz-II teach using inter-coding to encode a block. Ex-1005, ¶4 ("Inter-coding relies on temporal prediction and transform coding to reduce or remove temporal redundancy between video blocks of successive video frames of a video sequence."), ¶51 ("Video encoder 50 may perform intra- and inter-coding of blocks within video frames … Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames of a video sequence."), Ex-1006, ¶4 ("Block based inter-coding is a very useful coding technique that relies on temporal prediction to reduce or remove temporal

redundancy between video blocks of successive coded units of a video sequence.”), ¶54 (“Video encoder 50 may perform intra- and inter-coding of blocks within video frames … Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames of a video sequence.”).

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

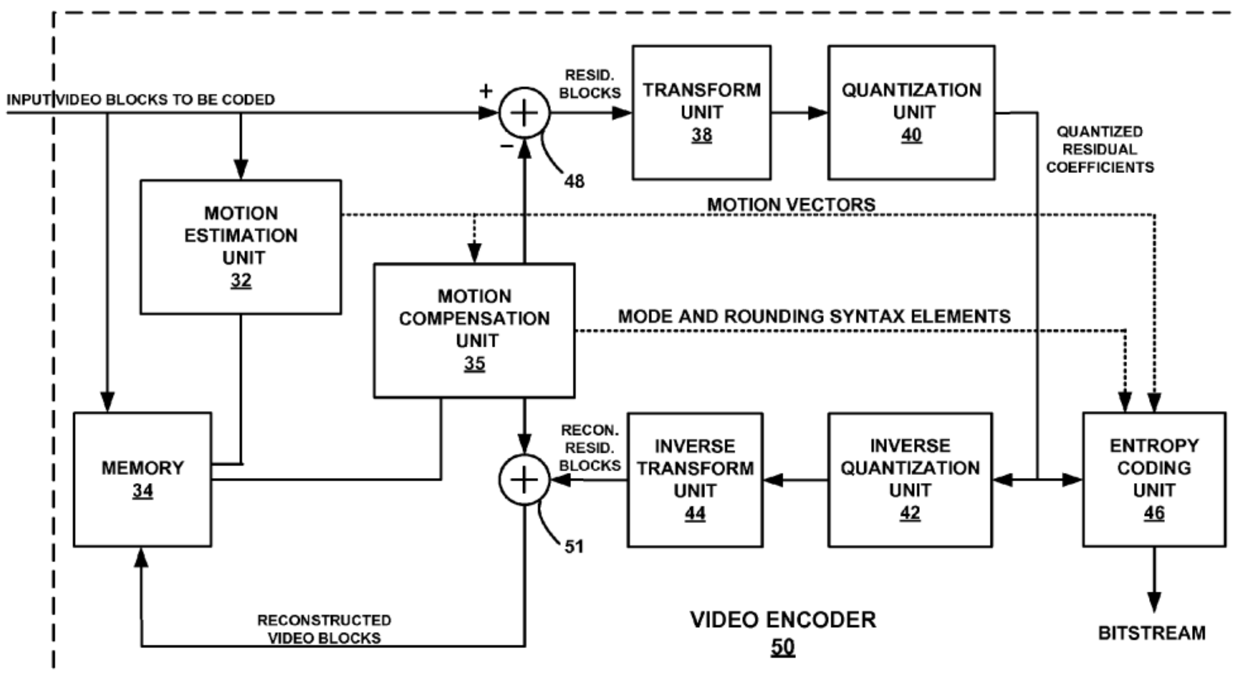


FIG. 2

Ex-1006, ¶4 (“For inter-coding, the video encoder performs motion estimation and motion compensation to track the movement of corresponding video blocks of two or more adjacent coded units.”), ¶6 (“Inter-coding based on motion estimation and motion compensation can achieve very good compression because successive video frames or other types of coded units are often very similar."), ¶56 ("During the encoding process, video encoder 50 receives a video block to be coded, and motion estimation unit 32 and motion compensation unit 35 perform inter-predictive coding."), Fig. 2.

196. Motion estimation generates motion vectors that point to reference blocks (e.g., predictive/prediction video blocks) **for a current block** and indicate the displacement between the reference blocks and the current block. Ex-1005, ¶7 ("For P- and B-video blocks, motion estimation generates motion vectors, which indicate the displacement of the video blocks relative to corresponding prediction video blocks in predictive reference frame(s) or other coded units."), ¶53 ("A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit)."), ¶54 ("The selected motion vector for any given list may point to a predictive video block that is most similar to the video block being coded, e.g., as defined by a metric such as sum of absolute difference (SAD) or sum of squared difference (SSD) of pixel values of the predictive block relative to pixel values of the block being coded."); Ex-1006, ¶4 ("Motion estimation generates motion vectors, which indicate the displacement of video blocks relative to corresponding prediction video blocks in one or more

reference frames or other coded units."), ¶56 ("Motion estimation is typically considered the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit).").

197. Motion compensation includes **determining reference blocks** (e.g., predictive video blocks) **based on motion vectors**. Ex-1005, ¶7 ("Motion compensation uses the motion vectors to generate prediction video blocks from the predictive reference frame (s) or other coded units."), ¶53 ("Motion compensation is typically considered the process of fetching or generating the predictive block based on the motion vector determined by motion estimation."); Ex-1006, ¶4 ("Motion compensation uses the motion vectors to generate prediction video blocks from the reference frame or other coded unit."), ¶56 ("Motion compensation is typically considered the process of fetching or generating the predictive block based on the motion vector determined by motion estimation."), ¶58 ("Once motion estimation unit 32 has selected the motion vector for the video block to be coded, motion compensation unit 35 generates the predictive video block associated with that motion vector."), ¶73 ("Then, the prediction video block is formed during motion compensation using the best motion vector."). By generating the predictive video block associated with a motion vector, Karczewicz-II

ascertains the values of the reference block that will be used for the current block, based on the associated motion vector. Alternatively, when Karczewicz-II generates a prediction video block from a reference frame, it determines a reference block from the reference frame that will be used for further calculations, e.g., interpolation. *See id.*

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

predicted from two lists of data from two previous frames, two lists of data from subsequent frames, or one list of data from a previous frame and one from a subsequent frame."), ¶54. Since Karczewicz-I teaches bi-prediction from two reference frames, it determines a first reference block (based on a first motion vector for one reference frame) and second reference block (based on a second motion vector for the other reference frame). *See id.*

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

200. Karczewicz-II further teaches **that the pixels of the current block, the first reference block, and the second reference block have values with a first precision** (e.g., 8 bits). As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Karczewicz-II's use of the term "precision" is consistent with this interpretation.

Ex-1006, ¶89 ("The average filter may also be quantized to a certain fixed-point precision (e.g., 13-bit precision).").

201. Karczewicz-II teaches that 8 bits are needed to represent possible pixel values for the two reference blocks. For example, Table 5 teaches operations for calculating pixel values using interpolation. Ex-1006, ¶103, Table 5:

TABLE 5

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

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

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

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

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

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

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

> According to the ITU-T H.264/AVC standard, three motion-compensated bi-predictive algorithms or modes may be used to predict a B-frame or portions thereof, such as video blocks, macroblocks or any other discreet and/or contiguous portion of a B-frame. A first motion-compensated bi-predictive algorithm or mode, which is commonly

> referred to as default weighted prediction, may involve applying default weights to each identified video block of the first frame of list 0 and the second frame of list 1. The default weights may be programmed according to the standard, and are often selected to be equal for default weighted prediction. The weighted blocks of the first and second frames are then added together and divided by the total number of frames used to predict the B-frame, e.g., two in this instance. Often, this division is accomplished by adding 1 to the addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit. The addition of 1 is a rounding adjustment.

See also ¶24, ¶44, ¶48.

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

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

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

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

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

209. <u>Scenario 1.</u> When Karczewicz-I's bi-directional prediction is calculated as an average of a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)), it would have been obvious to replace the first

prediction, which is a half-pixel prediction, with a non-rounded half-pixel prediction. *Supra* §V.B.3. Furthermore, it would have been obvious to replace the second prediction, which is an integer pixel prediction, with a left-shifted version of the integer pixel prediction. *Id*. Karczewicz-I's equation for calculating the bi-directional prediction would have been modified as shown below:

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

Id.

210. Karczewicz-II teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Karczewicz-II further teaches that the left-shifted integer pixel prediction (e.g., r1<<5) has possible values from 0 to 8160 and that a 13-bit unsigned number (i.e., "13u") is needed to represent these possible values. Ex-1006, ¶103, Table 5:

TABLE 5

positions {a, c, d, l} of FIGS. 4A-4D				
Operation	Comment	Min value	Max value	Register size
r1 = x	r1 is integer pixel x	0	255	8u
r1 = r1 << 5	r1 is 32 * x	0	8160	13u
r2 = y0	r2 is y0 (y0 is a one-dimensional (1-D) half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r1 = r1 + r2	r1 is 32 * x + y0	−2550	18870	16s
r1 = r1 + 32	r1 is 32 * x + y0 + 32	−2518	18902	16s
r1 = r1 >> 6	r1 is (32 * x + y0 + 32) >> 6	−39	295	11s
r1 = max (0, r1)	clip r1 on the low side	0	295	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

211. Therefore, 13 bits are needed to represent the possible values of each of the first prediction and the second prediction. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each need at least 13 bits to represent their possible values,[20] which is higher than the 8-bit

[20] Because the first prediction, which is a non-rounded half-pixel prediction, is represented by 15 bits, it needs at least 13 bits (along with 2 additional bits) to represent its possible values. Therefore, the first prediction attains the precision level of 13 bits and thus has 13 bits of precision.

precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4[1b]/[7b]/[13b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred0(i,j)))**, said first prediction having a second precision** (e.g., 13 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 1.

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

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

Id.

213. Karczewicz-II teaches that the partially-rounded center-pixel prediction (e.g., j1>>1)[21] has possible values from -4957 to 13116 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶105, Table 8. Karczewicz-II further teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶105, Table 8:

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

TABLE 8

positions {f, i, k, n} of FIGS. 4A-4D				
Operation	Comment	Min value	Max value	Register size
r1 = y0	r1 is y0 (1-D half-pixel such as b1, h1, ee1 and hh1 before shifting down)	−2550	10710	15s
r2 = j1	r2 is j1 (2-D half-pixel j1 before shifting down)	−9914	26232	16s
r2 = r2 >> 1	r2 is j1 >> 1	−4957	13116	15s
r1 = r1 + r2	r1 is y0 + (j1 >> 1)	−7507	23826	16s
r1 = r1 + 32	r1 is y0 + (j1 >> 1) + 32	−7491	23842	16s
r1 = r1 >> 6	r1 is (y0 + (j1 >> 1) + 32) >> 6	−235	745	11s
r1 = max (0, r1)	clip r1 on the low side	0	745	10u
r1 = min (255, r1)	clip r1 on the high side	0	255	8u

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

rounded(pred0(i,j))>>5)**, said first prediction having a second precision** (e.g., 15 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 2.

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

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

Id.

216. Karczewicz-II teaches that the non-rounded half-pixel predictions (e.g., b1, h1) each has values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶103, Table 6:

TABLE 6

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))+pred1(i,j)<<5+32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said second prediction (e.g., pred1(i,j)<<5), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶103, Table 5. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

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

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))>>5+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))>>5) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶105, Table 8. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

231. Scenario 3. When Karczewicz-I's bi-directional prediction is calculated as an average of two center-pixel prediction (e.g., pred0(i,j) and pred1(i,j)), it would have been obvious to modify Karczewicz-I's equation based on Karczewicz-II's teachings, as shown below:

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction and a rounding offset (e.g., non-rounded(pred0(i,j))+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶103, Table 6. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

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

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

233. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches the combined prediction as a sum of the first prediction, the second prediction, and a rounding offset. *Supra* §V.B.4[1e]/[7e]/[13e]. Both Karczewicz-I and Karczewicz-II teach the operation ">>" for shifting bits to the right. Ex-1005, ¶55 ("Often, this division is accomplished by adding 1 to the addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit."), ¶57 (">> is a right shift operation[.]"), ¶60; Ex-

1006, ¶94 ("In this disclosure, '>>' represents a right shift operation and '<<' represents a left shift operation."). The combination of Karczewicz-I and Karczewicz-II further teaches **decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right** for each of the three scenarios discussed above.

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

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

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

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

TABLE 5

positions {a, c, d, 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

Ex-1006, ¶103, Table 5. In this table, the operation “r1=r1+32” calculates the sum of a non-rounded half-pixel prediction (e.g., y0 which can take the value of b1), a left-shifted integer pixel prediction (e.g., 32*x), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -2518 and 18902 and that a 16-bit signed number (e.g., “16s”) is needed to represent these possible values. Next, the operation “r1=r1>>6” shifts the sum 6 bits to the right. The result has possible values between -39 and 295; an 11-bit signed number (e.g., “11s”) is needed to represent these possible values. As explained above, the term “precision” is at least satisfied by “a number of bits needed to represent possible values.”

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

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

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

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

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

TABLE 8

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

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

Ex-1006, ¶105, Table 8. In this table, the operation "r1=r1+32" calculates the sum of a partially-rounded center-pixel prediction (e.g., j1>>1), a non-rounded half-pixel prediction (e.g., y0, which may take the value of b1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -7491 and 23842 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values between -235 and 745; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent a value." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

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

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

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

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

TABLE 6

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

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

Ex-1006, ¶103, Table 6. In this table, the operation "r1=r1+32" calculates the sum of two non-rounded half-pixel predictions (e.g., y0 and y1, which may take the values of b1 and h1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values from -5068 to 21452 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values from -79 to 335; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

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

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

241. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches determining the combined prediction in the motion compensation process. *Supra* §§V.B.4[1b-1f]/[7b-7f]/[13b-13f]. Karczewicz-I teaches **determining residual data** (e.g., a residual video block) **based upon a difference between the combined prediction** (e.g., prediction data) **and the block of pixels** (e.g., original video block). Ex-1005, ¶73 ("Once the desired prediction data is identified by motion compensation unit 35, as described herein, video encoder 50 forms a residual video block by subtracting the prediction data from the original video block being coded."), ¶7 ("After motion compensation, a residual video block is formed by subtracting the prediction video block from the original video block to be coded."), Fig. 2. Here, the residual video block includes residual data. *See* Ex-1005, ¶84 ("residual data (e.g., a residual block)"), ¶89. Karczewicz-II includes similar teachings. Ex-1006, ¶4, ¶6, ¶35, ¶50, ¶58, ¶60, ¶73, Fig. 2.

242. Karczewicz-I further teaches **encoding** (e.g., transforming, quantizing, and entropy coding) **residual data in a bitstream**. Ex-1005, ¶73 ("Transform unit 38 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform block coefficients. Transform unit 38, for example, may perform other transforms, such as those defined by the H.264 standard, which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform unit 38 applies the transform to the residual block, producing a block of residual transform coefficients."), ¶74 ("Quantization unit 40 quantizes the residual transform coefficients to further reduce bit rate."), ¶75 ("The coded bitstream may include entropy coded residual blocks, motion vectors for such blocks, and other syntax such as the syntax described herein."), Fig. 2. Karczewicz-II similarly teaches encoding residual data in a bitstream. Ex-1006, ¶5, ¶35, ¶50, ¶¶60-61, ¶110, Fig. 2.

5. Dependent Claims 2, 8, and 14

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

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

244. Karczewicz-II teaches obtaining predictions by interpolation using pixel values of reference blocks when motion vectors point to subpixels. Ex-1006, ¶7 ("In this case, the predictive data generated during motion compensation, which is used to code a video block, may be interpolated from the pixels of video blocks of the video frame or other coded unit used in motion estimation. Interpolation is often performed to generate predictive half-pixel values (half-pel) and predictive quarter-pixel values (quarter-pel)."), ¶42 ("The interpolation techniques of this disclosure may be performed by any encoding device that supports motion compensated interpolation to sub-pixel resolution."), ¶66 ("Again, the techniques of this disclosure concern motion compensated interpolation in which pixel values

of predictive video blocks are interpolated to sub-pixel resolution.”), ¶¶68-72, Figs. 4A-4D.

245. As explained above, the combination of Karczewicz-I and Karczewicz-II contemplates at least three scenarios for determining a bi-directional prediction. Karczewicz-I and Karczewicz-II’s teachings with respect to each of the three scenarios satisfy the additional limitations of claims 2, 8, and 14.

246. Scenario 1. In this scenario, the first motion vector points to a half-pixel position. *Supra* §V.B.3. The half-pixel position refers to a subpixel. *See, e.g.*, Ex-1006, ¶74 (“For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample ‘C3’ and labeled ‘a’ through ‘o’ in FIGS. 4A-4D.”), Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

247. Karczewicz-II teaches **obtaining the first prediction** (e.g., the non-rounded half-pixel prediction) **by interpolation using pixel values of said first reference block**. Ex-1006, ¶93:

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

"C3" and "C4" being the closest, "C2" and "C5" being the next closest, and "C1" and "C6" being the next closest.

The interpolation is performed using pixel values of "six closest integer pixels that surround" the subpixel. Because the integer pixels are closest to the subpixel, they are located in the reference block that the first motion vector points to. *See* Ex-1006, Fig. 4B:

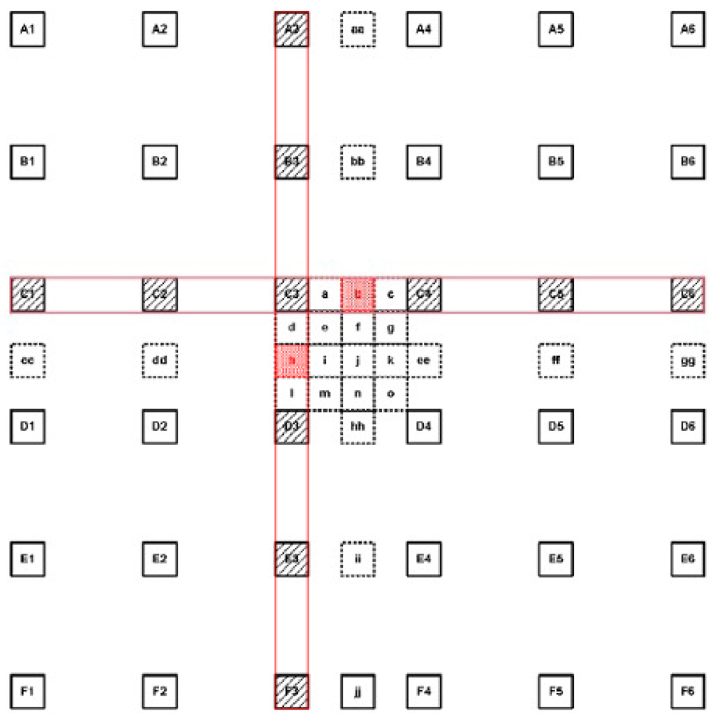


FIG. 4B

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

248. <u>Scenario 2.</u> In this scenario, the first motion vector points to a center-pixel position. *Supra* §V.B.3. The center-pixel position refers to a subpixel. *See,*

e.g., Ex-1006, ¶74, Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

249. Karczewicz-II teaches **obtaining the first prediction** (e.g., the partially-rounded center-pixel prediction) **by interpolation using pixel values of said first reference block**. Ex-1006, ¶95:

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

The interpolation is performed using non-rounded half-pixel predictions, which are in turn obtained by interpolation using integer pixel values. The half-pixel positions and their corresponding integer pixel positions are close to the center-pixel position and are located in the reference block that the first motion vector points to. *See* Ex-1006, Fig. 4C:

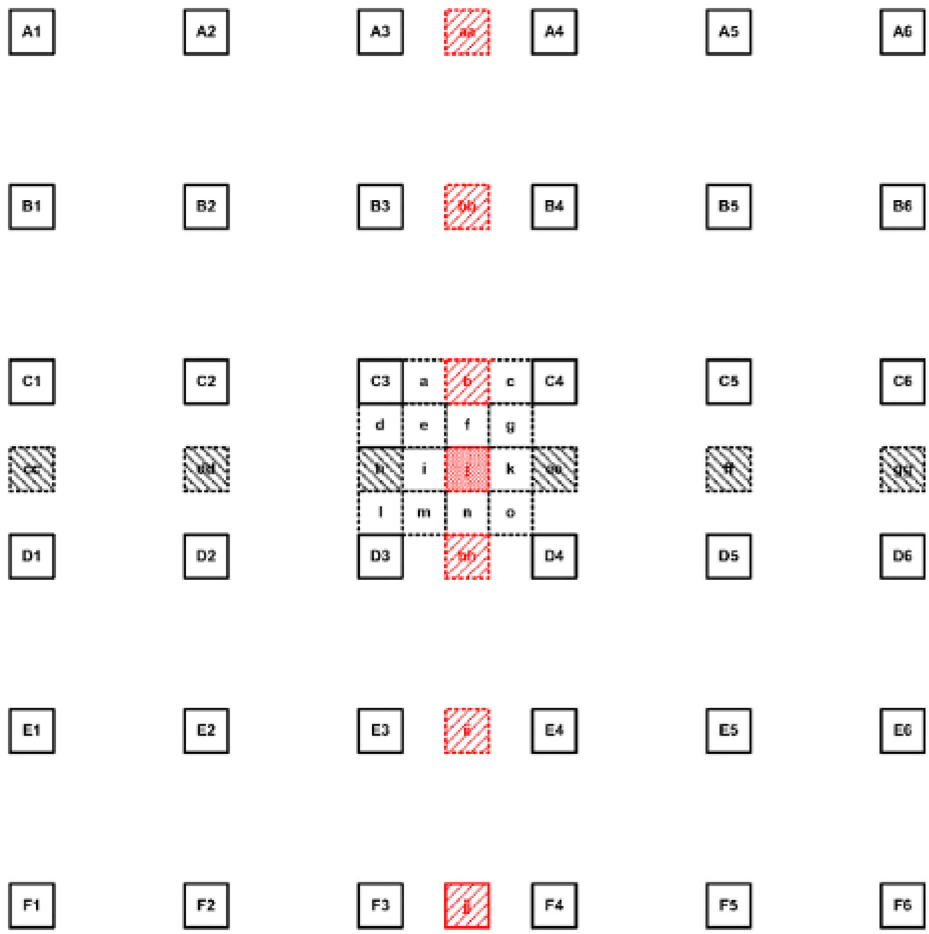


FIG. 4C

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

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

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

6. Dependent Claims 3, 9, and 15

3. The method according to claim 2,

9. The apparatus according to claim 8,

15. The computer program product according to claim 14,

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

252. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.B.5. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 3, 9, and 15 under Scenario 2.

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

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

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

254. Karczewicz-II teaches calculating the partially-rounded center-pixel prediction (e.g., j1>>5) by first determining a sum of a 6-tap filter (e.g., j1). Ex-1006, ¶95:

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

Supra §V.B.5. This sum is a non-rounded version of the center-pixel prediction.

255. Next, the non-rounded center-pixel prediction is shifted to the right by 5 bits to obtain the partially-rounded center-pixel prediction (j1>>5). *Supra* §V.B.3; *see, e.g.*, Ex-1006, 99, Table 3:

TABLE 3

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

Table 8 further teaches operations implementing the equations that include shifting bits to the right, consistent with Table 3. Ex-1006, ¶105, Table 8.

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

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

7. Dependent Claims 4, 10, and 16

4. The method according to claim 2,

10. The apparatus according to claim 8,

16. The computer program product according to claim 14,

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

258. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 2, the apparatus according to claim 8, and the computer program product according to claim 14. *Supra* §V.B.5. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 4, 10, and 16 under Scenario 1.

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

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

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

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

261. Karczewicz-II teaches calculating the left-shifted integer pixel prediction by shifting the pixel value to the left (e.g., C3<<5). *Supra* §V.B.3; *see, e.g.*, Ex-1006, ¶99, Table 3:

TABLE 3

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

Here, the value that is shifted to the left is the value of an integer pixel sample (e.g., C3). Ex-1006, ¶74 ("For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample 'C3' and labeled 'a' through 'o' in FIGS. 4A-4D."), ¶93 ("where 'C1,' 'C2,' 'C3,' 'C4,' 'C5' and 'C6' represent the six closest integer pixels that surround 'b' in the horizontal direction"). Because the second motion vector points to this integer pixel sample, the integer pixel sample is part of the second reference block; its value is a value of said second reference block.

262. The teachings of Karczewicz-I and Karczewicz-II above explain calculations for one pixel in a block. Karczewicz-I and Karczewicz-II teach that motion compensation is performed on a block basis. *See supra* §§V.B.4[1b, 1g]/[7b, 7g]/[13b, 13g]; Ex-1005, ¶7; Ex-1006, ¶4, ¶73. Therefore, the references teach performing the predictions operations on all the pixels of a block. This is further obvious because this is how block-based motion prediction has worked since the 1990s. As explained above, in motion estimation and compensation, a motion vector indicates the displacement of between a reference block and a current block of pixels. *Supra* §V.B.4[1b]/[7b]/[13b]; Ex-1005, ¶7, ¶¶53-54; Ex-1006, ¶4, ¶56. Therefore, Karczewicz-I and Karczewicz-II teach performing the above operations, including the left-shift operation, for each pixel of the current block based on corresponding pixels of the reference block. In Scenario 1, the left shifting is performed for multiple pixels.

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

8. Dependent Claims 5, 11, and 17

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

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

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

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

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

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

inserting a rounding offset (e.g., 32) **to the combined prediction**, as explained for limitations [1e], [7e], and [13e], **before said decreasing** of the precision, as explained for limitations [1f], [7f], and [13f].

266. The value added to the sum of the first and second predictions is a rounding offset. Karczewicz-I refers to the term that is added to the weighted sum of the first and second predictions as a rounding adjustment. Ex-1005, ¶63 ("Generally, a rounding adjustment of 2^{r-1} is commonly used prior to a right shift by r, where r represents a positive integer."), ¶55.[22] As was well known to those skilled in the art, "rounding adjustment" was used interchangeably with "rounding offset." The value (e.g., 32) is inserted to the combined prediction, increasing the value of the combined prediction, right before the rounding operation. A POSITA would have understood that this value is a rounding offset according to the plain meaning of the term.

267. Karczewicz-I and Karczewicz-II teach this rounding offset as part of its rounding process, which decreases the precision as recited by the claims. Ex-1005, ¶55, ¶63; Ex-1006, ¶¶96-101, Tables 1-4. Additionally, it was obvious for said decreasing to include the rounding offset (claim 5) because the insertion of the

---

[22] Karczewicz-I teaches a rounding adjustment of 2^{r-1} prior to a right shift by r. This is consistent with the modified equation for calculating the bi-directional prediction under each of the three scenarios as taught by the combination of Karczewicz-I and Karczewicz-II (*supra* §V.B.3), which teaches right shifting by 6 bits and a rounding offset of 32: $2^{6-1}=2^5=32$.

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

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

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

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

9. Dependent Claims 6, 12, and 18

6. The method according to claim 1,

12. The apparatus according to claim 7,

18. The computer program product according to claim 13,

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

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

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

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

VI. BACKGROUND AND QUALIFICATIONS

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

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

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

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

recognition of my work, I received National Aeronautics and Space Administration Group Achievement Awards in 1990 and 1992.

277. After June 1994, I began a six-month stint as a contract Software Engineer for the Federal National Mortgage Association in Washington D.C., for which I developed a graphical user interface to monitor and validate loan servicer input for a Loss Mitigation Project. I then served as an Independent Consultant to Optivision, Inc. for the next six months, where I researched and developed rate control algorithms and software based on the MPEG-2 Test Model 5 for the OPTIVideo™ MPEG-2 video encoder, as well as adaptive quantization algorithms based on the then-JPEG-3 draft standard. In this role, I researched and developed algorithms to improve the quality of gray scale image compression for the medical imaging DICOM Standard by providing a lossless hybrid algorithm encoding image residuals with a diagonal Golomb code based an Enhanced Universal Trellis Coded Quantization algorithm.

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

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

280. In October 1998, I began a six-month engagement with Rockwell Collins Inc., where I worked as a Lead Systems Engineer tasked with harmonizing requirements for an MPEG-2 in-flight entertainment system. I then worked for Sun Microsystems Inc. as a Software Engineer until November 1999. During my time at Sun Microsystems Inc., I developed a Distributed Component Object Model (DCOM) software interface between a TV control graphical user interface and a Microsoft broadcast application programming interface (API) with the goal of improving the visual quality of interactive TV displays derived from UDP/IP datagrams synchronized with MPEG-2 audio/video packet data.

281. For the next 22 months, from January 2000 to October 2002, I worked as the Chief Systems Engineer for Media Logic Systems Ltd. During my time at Media Logic Systems, I designed and developed a live interactive television system (iSeeTV) in which customers communicate with human sales agents in

video-enabled call centers. To create this system, I researched and developed tools and encoder systems to improve image quality at prescribed latency and bit rate for distributing live video and audio streams encoded via low latency methods. To perform the above, I was required to understand and implement video codec systems employing the MPEG-2 Simple Profile at Main Level (CATV), MPEG-4 Visual Profile with background sprite coding, and the H.263+ Standard (now known as H.264).

282. Since November 2002, I have been an engineering contractor, and more recently an independent consultant in mathematical modeling, for several companies, such as Cyra Technologies Inc. and Amgen Inc. I also served as a senior research fellow at Merck & Co., Inc., a manager at GlaxoSmithKline Inc., a director at Daiichi Sankyo, Inc., a senior director at Praxis Precision Medicines, and currently serve as a director at Takeda Pharmaceuticals. During this time, I have developed mathematical models and simulations related to various systems, signals, and images. Specifically, I have focused on analyzing, processing, storing, and deriving information from biomedical imaging and other data. Using the information derived from these data, I have created a variety of models related to biology and the effects of drugs on the human body. In recognition of my work, I have received GlaxoSmithKline R&D Recognition Awards in 2012, 2013, and

2016, a Daiichi Sankyo recognition award in 2021 and Takeda Pharmaceutical awards in 2022 and 2023.

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

284. I am also a Senior Member of the IEEE and serve as the current Philadelphia Chapter Chair of the Communications & Information Theory Societies as well as former Chair of the American Association of Pharmaceutical Scientists Pharmaco-Imaging Community. I also served as the 2019 Vice Chair of the IEEE P2673 Intelligence Augmentation for Medical Imaging Standards Working Group. I also have been registered to practice as a patent agent for the United States Patent and Trademark Office since 2002 (Reg. No. 51,704).

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

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

A. Compensation

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

B. Materials and Other Information Considered

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

VII. UNDERSTANDING OF THE LAW

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

VIII. RESERVATION OF RIGHTS

290. My opinions are based upon the information that I have considered to date. I am unaware of any evidence of secondary considerations with respect to the ’267 Patent that would render any of the challenged claims non-obvious. I reserve the right, however, to supplement my opinions in the future to respond to any arguments raised by the owner of the ’267 Patent and to take into account new information that becomes available to me.

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

Executed on March 14, 2024.

By:

I. Freedman

Immanuel Freedman

APPENDIX 1: CURRICULUM VITAE OF IMMANUEL FREEDMAN

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

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

SUMMARY OF EXPERIENCE

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

EDUCATION

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

LICENSES

Registered Patent Agent #51,704

EXPERIENCE

Takeda Pharmaceuticals **Cambridge, MA** **Mar '22-present**
Clinical Pharmacology Director

Dr. Freedman provides mathematical modeling of systems, signals, and images.

Freedman Patent **Harleysville, PA** **Oct '21-present**
Sole Proprietor

He provides consulting and expert witness services to industry and the legal profession.

State University of New York at Buffalo **Buffalo, NY** **Jun '17-present**
Volunteer Researcher

Praxis Precision Medicines **Boston, MA** **Aug '21-Sep '21**
Senior Director, Pharmacometrics

Dr. Freedman provided mathematical modeling of systems, signals, and images.

Daiichi Sankyo, Inc. **Basking Ridge, NJ** **Nov '20-Aug '21**
Director, Modeling and Simulation

Dr. Freedman provided mathematical modeling of systems, signals, and images.

Freedman Patent **Harleysville, PA** **Jun '16-Nov '20**
Sole Proprietor

He provided consulting and expert witness services to industry and the legal profession. In particular, he provided requirements analysis and design for a precision dosing system Graphical User Interface.

GlaxoSmithKline, Inc. **Upper Merion, PA** **Aug '07-Jun '16**
Manager, Clinical Pharmacology

Dr. Freedman provided mathematical modeling of systems, signals and images and participated in technical due diligence activities on demand.

Freedman Patent **Harleysville, PA** **Jul '05-Aug '07**
Sole Proprietor

He provided consulting and expert witness services to industry and the legal profession. In particular, he provided requirements analysis and design for a precision dosing system Graphical User Interface.

Merck & Co., Inc., **West Point, PA** **Nov '04-Jul '05**
Senior Research Fellow

Dr. Freedman provided mathematical models on demand.

Amgen, Inc. **Thousand Oaks, CA** **Apr '03-Mar '04**
Contractor

Dr. Freedman developed algorithms and software in MATLAB and FORTRAN for simulation and data modeling.

Cyra Technologies, Inc. **San Ramon, CA** **Nov '02-Apr '03**
Senior Hardware Engineer

Dr. Freedman designed, developed, and tested algorithms and software for calibrating a three-dimensional laser scanner. He calculated the statistical distribution of outcomes for an engineering tolerance stack by modeling and simulating the scanner response using a Jacobian sensitivity matrix to compare alternative placements of scanner calibration targets based on a D-matrix of scanner response.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Dr. Freedman evaluated frame grabber hardware for resolution and quality of time-integrated imagery and specified algorithms including cluster analysis and trending, further developing a user interface for a digital image processing system supporting gene-cloning science.

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

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

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

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

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

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

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

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

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

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

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

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

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

He provided technical consulting services to industry.

Anugraha **La Jolla, CA** **Jan '96-Jan '97**
Sole Proprietor

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

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

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

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

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

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

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

Optivision, Inc. **Davis, CA** **Mar ’95-Sep ’95**
Consultant

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

National Coal Board Nuneaton, England UK Nov '82-Sep '84
Scientist (Management Grade 7)

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

PUBLICATIONS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

PROFESSIONAL AWARDS
Takeda Pharmaceuticals Recognition Awards (2022, 2023)

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

PROFESSIONAL AFFILIATIONS
Institute of Electrical and Electronic Engineers (Senior Member; Chair, Communications & Information Theory Chapter, Philadelphia Section; 2018/2019 Vice-Chair, P2673 Standards Working Group; Trusted Analytic Exchange Sub-Group Chair, P2795 Standards Working Group; Member P1900.8 Standards Working Group; Former Secretary, Dynamic Spectrum Access Machine Learning Study Group)
Institute of Physics (Chartered Physicist)
American Association of Pharmaceutical Scientists (Former Chair, Pharmaco-Imaging Community; Former Chair, Predictive Modeling Community; 2017 Member, Predictive Modeling Task Force)
International Society of Pharmacometrics (Former Member Standards and Best Practices Committee Member–Model Evaluation Group)
Regulatory Affairs Professionals Society
International Biometric Society (Eastern North American Region)
International Association for Assyriology
National Coalition of Independent Scholars
Ronin Institute for Independent Scholarship (Research Scholar)
Royal Society of Medicine (Fellow Member)

December 22, 2023

APPENDIX 2: MATERIALS CONSIDERED IN THE PREPARATION OF THIS DECLARATION

Exhibit No.	Description
1001	U.S. Patent No. 11,805,267 to Ugur *et al.* (“the ’267 patent”)
1002	Prosecution History for the ’267 Patent
1004	U.S. Patent Application Publication No. 2005/0281334 (“Walker”)
1005	U.S. Patent Application Publication No. 2011/0007799 (“Karczewicz-I”)
1006	U.S. Patent Application Publication No. 2009/0257499 (“Karczewicz-II”)
1007	Prosecution History for U.S. Patent No. 9,432,693
1008	U.S. Patent No. 9,344,744 (“Kirchhoffer-744”)
1009	Srinivasan, An Overview of VC-1
1010	U.S. Patent Application Publication No. 2003/0112864 (“Karczewicz-864”)
1011	Wiegand, Overview of the H.264/AVC Video Coding Standard
1012	Richardson, The H.264 Advanced Video Compression Standard
1013	U.S. Patent Application Publication No. 2008/0198935 (“Srinivasan-935”)
1014	H.264 Advanced Video Coding for Generic Audiovisual Services (March 2009)
1015	U.S. Patent Application Publication No. 2013/0034158 (“Kirchhoffer-158”)
1016	U.S. Patent No. 8,594,188 (“Demos”)

APPENDIX 3: CHALLENGED CLAIMS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:

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

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

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

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

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

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

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

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

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

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

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

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

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

[13a]. A computer program product for encoding a block of pixels, the computer program product comprising at least one non-transitory computer readable

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

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

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

[13d] use said second reference block to obtain a second prediction, said second prediction having the second precision;

[13e] obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[13f] decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

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

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

15. The computer program product according to claim 14, wherein said first prediction is obtained by interpolation using values of said first reference block by:

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

16. The computer program product according to claim 14, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

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

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

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

APPENDIX 4: UNDERSTANDING OF THE LAW

I have applied the following legal principles provided to me by counsel in arriving at the opinions set forth in this report.

Legal Standard for Prior Art

I am not an attorney. I have been informed by attorneys of the relevant legal principles and have applied them to arrive at the opinions set forth in this declaration.

I understand that the petitioner for inter partes review may request the cancelation of one or more claims of a patent based on grounds available under 35 U.S.C. § 102 and 35 U.S.C. § 103 using prior art that consists of patents and printed publications.

Anticipation and Prior Art

I understand that § 102 specifies when a challenged claim is invalid for lacking novelty over the prior art, and that this concept is also known as "anticipation." I understand that a prior art reference anticipates a challenged claim, and thus renders it invalid by anticipation, if all elements of the challenged claim are disclosed in the prior art reference. I understand the disclosure in the prior art reference can be either explicit or inherent, meaning it is necessarily present or implied. I understand that the prior art reference does not have to use the same words as the challenged claim, but all of the requirements of the claim must be disclosed so that a person of ordinary skill in the art could make and use the claimed subject-matter.

In addition, I understand that § 102 also defines what is available for use as a prior art reference to a challenged claim.

Under § 102(a), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country before the challenged claim's date of invention.

Under § 102(b), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country more than one year prior to the challenged patent's filing date.

Under § 102(e), a challenged claim is anticipated if it was described in a published patent application that was filed by another in the United States before the challenged claim's date of invention, or was described in a patent granted to another that was filed in the United States before the challenged claim's date of invention.

I understand that a challenged claim's date of invention is presumed to be the challenged patent's filing date. I also understand that the patent owner may establish an earlier invention date and "swear behind" prior art defined by § 102(a) or § 102(e) by proving (with corroborated evidence) the actual date on which the named inventors conceived of the subject matter of the challenged claim and proving that the inventors were diligent in reducing the subject matter to practice.

I understand that the filing date of a patent is generally the filing date of the application filed in the United States that issued as the patent. However, I understand that a patent may be granted an earlier effective filing date if the patent owner properly claimed priority to an earlier patent application.

I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

I understand that when a challenged claim requires selection of an element from a list of alternatives, the prior art teaches the element if one of the alternatives is taught by the prior art.

Legal Standard for Obviousness

I understand that even if a challenged claim is not anticipated, it is still invalid if the differences between the claimed subject matter and the prior art are such that the claimed subject matter would have been obvious to a person of ordinary skill in the pertinent art at the time the alleged invention.

I understand that obviousness must be determined with respect to the challenged claim as a whole.

I understand that one cannot rely on hindsight in deciding whether a claim is obvious.

I also understand that an obviousness analysis includes the consideration of factors such as (1) the scope and content of the prior art, (2) the differences between the prior art and the challenged claim, (3) the level of ordinary skill in the pertinent art, and (4) "secondary" or "objective" evidence of non-obviousness.

Secondary or objective evidence of non-obviousness includes evidence of: (1) a long felt but unmet need in the prior art that was satisfied by the claimed invention; (2) commercial success or the lack of commercial success of the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise of the claimed invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the claimed invention; and (7) contemporaneous and independent invention by others. However, I understand that there must be a relationship between any secondary evidence of non-obviousness and the claimed invention.

I understand that a challenged claim can be invalid for obviousness over a combination of prior art references if a reason existed (at the time of the alleged invention) that would have prompted a person of ordinary skill in the art to combine elements of the prior art in the manner required by the challenged claim. I understand that this requirement is also referred to as a "motivation to combine," "suggestion to combine," or "reason to combine," and that there are several rationales that meet this requirement.

I understand that the prior art references themselves may provide a motivation to combine, but other times simple common sense can link two or more prior art references. I further understand that obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may come from market forces.

I understand obviousness to include, for instance, scenarios where known techniques are simply applied to other devices, systems, or processes to improve them in an expected or known way. I also understand that practical and common-sense considerations should be applied in a proper obviousness analysis. For instance, familiar items may have obvious uses beyond their primary purposes.

I understand that the combination of familiar elements according to known methods is obvious when it yields predictable results. For instance, obviousness bars patentability of a predictable variation of a technique even if the technique originated in another field of endeavor. This is because design incentives and other market forces can prompt variations of it, and predictable variations are not the product of innovation, but rather ordinary skill and common sense.

I understand that a particular combination may be obvious if it was obvious to try the combination. For example, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. This would result in something obvious because the result is the product not of innovation but of ordinary skill and common sense. However, I understand that it may not be obvious to try a combination when it involves unpredictable technologies.

It is further my understanding that a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill in the art, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

It is my understanding that the Manual of Patent Examining Procedure §2143 sets forth the following as exemplary rationales that support a conclusion of obviousness:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known technique to improve similar devices (methods, or products) in the same way;

- Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

A person of ordinary skill in the art looking to overcome a problem will often use the teachings of multiple publications together like pieces of a puzzle, even though the prior art does not necessarily fit perfectly together. Therefore, I understand that references for obviousness need not fit perfectly together like puzzle pieces. Instead, I understand that obviousness analysis takes into account inferences, creative steps, common sense, and practical logic and applications that a person of ordinary skill in the art would employ under the circumstances.

I understand that a claim can be obvious in light of a single reference, if the elements of the challenged claim that are not explicitly or inherently disclosed in the reference can be supplied by the common sense of one of skill in the art.

I understand that obviousness also bars the patentability of applying known or obvious design choices to the prior art. One cannot patent merely substituting one prior art element for another if the substitution can be made with predictable results. Likewise, combining prior art techniques that are interoperable with respect to one another is generally obvious and not patentable.

In order for a claim to be found invalid based upon a modification or combination of the prior art, there must be reasonable expectation that a person of ordinary skill would have successfully modified or combined the prior art to arrive at the claimed arrangement. This does not mean that it must be certain that a person of ordinary skill would have been successful – the law only requires that the person of ordinary skill in the art would have perceived a reasonable expectation of success in modifying or combining the prior art to arrive at the claimed invention.

In sum, my understanding is that obviousness invalidates claims that merely recite combinations of, or obvious variations of, prior art teachings using understanding and knowledge of one of skill in the art at the time and motivated by the general problem facing the inventor at the time. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor at the time of the invention, can provide a reason for combining the elements of or attempting obvious variations on prior art references in the claimed manner.

Legal Standard for Claim Construction

I understand that before any invalidity analysis can be properly performed, the scope and meaning of the challenged claims must be determined by claim construction.

I understand that a patent may include two types of claims, independent claims and dependent claims. I understand that an independent claim stands alone and includes only the limitations it recites. I understand that a dependent claim depends from an independent claim or another dependent claim. I understand that a dependent claim includes all the limitations that it recites in addition to the limitations recited in the claim (or claims) from which it depends.

In comparing the challenged claims to the prior art, I have carefully considered the patent and its file history in light of the understanding of a person of skill at the time of the alleged invention.

I understand that to determine how a person of ordinary skill would have understood a claim term, one should look to sources available at the time of the alleged invention that show what a person of skill in the art would have understood disputed claim language to mean. It is my understanding that this may include what is called "intrinsic" evidence as well as "extrinsic" evidence.

I understand that, in construing a claim term, one should primarily rely on intrinsic patent evidence, which includes the words of the claims themselves, the remainder of the patent specification, and the prosecution history. I understand that extrinsic evidence, which is evidence external to the patent and the prosecution history, may also be useful in interpreting patent claims when the intrinsic evidence itself is insufficient. I understand that extrinsic evidence may include principles, concepts, terms, and other resources available to those of skill in the art at the time of the invention.

I understand that words or terms should be given their ordinary and accepted meaning unless it appears that the inventors were using them to mean something else or something more specific. I understand that to determine whether a term has special meaning, the claims, the patent specification, and the prosecution history are particularly important, and may show that the inventor gave a term a particular definition or intentionally disclaimed, disavowed, or surrendered claim scope.

I understand that the claims of a patent define the scope of the rights conferred by the patent. I understand that because the claims point out and distinctly claim the subject matter which the inventors regard as their invention, claim construction analysis must begin with and is focused on the claim language itself. I understand that the context of the term within the claim as well as other claims of the patent can inform the meaning of a claim term. For example, because claim terms are normally used consistently throughout the patent, how a term is used in one claim can often inform the meaning of the same term in other claims. Differences among claims or claim terms can also be a useful guide in understanding the meaning of particular claim terms.

I understand that a claim term should be construed not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the entire specification. I understand that because the specification is a primary basis for construing the claims, a correct construction must align with the specification.

I understand that the prosecution history of the patent as well as art incorporated by reference or otherwise cited during the prosecution history are also highly relevant in construing claim terms. For instance, art cited by or incorporated by reference may indicate how the inventor and others of skill in the art at the time of the invention understood certain terms and concepts. Additionally, the prosecution history may show that the inventors disclaimed or disavowed claim scope, or further explained the meaning of a claim term.

With regard to extrinsic evidence, I understand that all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises, can also be considered. For example, technical dictionaries may indicate how one of skill in the art used or understood the claim terms. However, I understand that extrinsic evidence is considered to be less reliable than intrinsic evidence, and for that reason is generally given less weight than intrinsic evidence.

I understand that in general, a term or phrase found in the introductory words or preamble of the claim, should be construed as a limitation if it recites essential structure or steps, or is necessary to give meaning to the claim. For instance, I understand preamble language may limit claim scope: (i) if dependence on a preamble phrase for antecedent basis indicates a reliance on both the preamble and claim body to define the claimed invention; (ii) if reference to the preamble is necessary to understand limitations or terms in the claim body; or (iii) if the preamble recites additional structure or steps that the specification identifies as important.

On the other hand, I understand that a preamble term or phrase is not limiting where a challenged claim defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention. I understand that to make this determination, one should review the entire patent to gain an understanding of what the inventors claim they invented and intended to encompass in the claims.

I understand that 35 U.S.C. § 112 ¶ 6 created an exception to the general rule of claim construction called a "means plus function" limitation. These types of terms and limitations should be interpreted to cover only the corresponding structure described in the specification, and equivalents thereof. I also understand that a limitation is presumed to be a means plus function limitation if (a) the claim limitation uses the phrase “means for”; (b) the “means for” is modified by functional language; and (c) the phrase “means for” is not modified by sufficient structure for achieving the specified function.

I understand that a structure is considered structurally equivalent to the corresponding structure identified in the specification only if the differences between them are insubstantial. For instance, if the structure performs the same function in substantially the same way to achieve substantially the same result. I further understand that a structural equivalent must have been available at the time of the issuance of the claim.

Appendix C

IN THE 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 No. IPR2024-00627

U.S. Patent 11,805,267

Declaration of Immanuel Freedman

TABLE OF CONTENTS

Page

I, Immanuel Freedman declare as follows:

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

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

I. OVERVIEW OF THE TECHNOLOGY

A. Video Compression Basics

3. Video encoding, also referred to as video compression, exploited redundancies in video data to reduce the size of video. Since the early 1990s, major video coding standards such as MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263, and H.264 have applied the same model, where video encoders have a

motion estimation and compensation front end, a transform stage such as Discrete Cosine Transform ("DCT"), and an entropy encoder at the back end for generating the coded bitstream. At the decoder, the inverse process was used to decode the video. In 2003, the H.264 standard, also known as Advanced Video Coding (AVC), was introduced. This standard quickly became a prevalent and widely adopted video format. The model of a typical general video encoder is illustrated below. Ex-1005, Fig. 2. This fundamental model has been used by major video encoding standards since the 1990s.

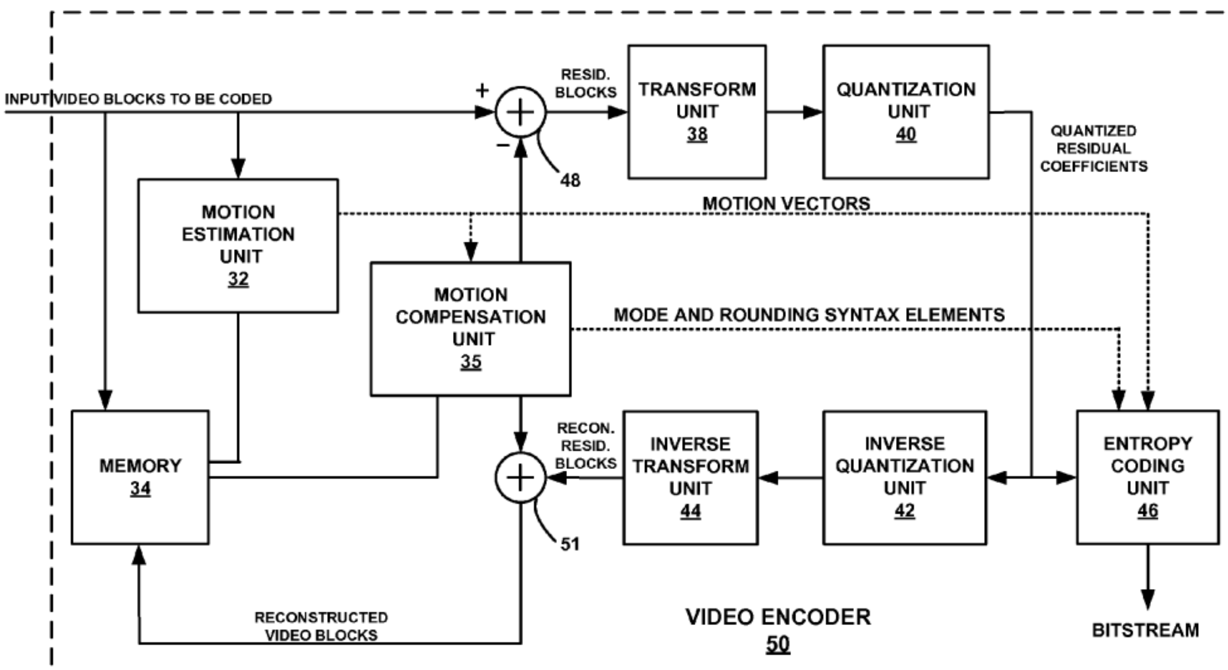


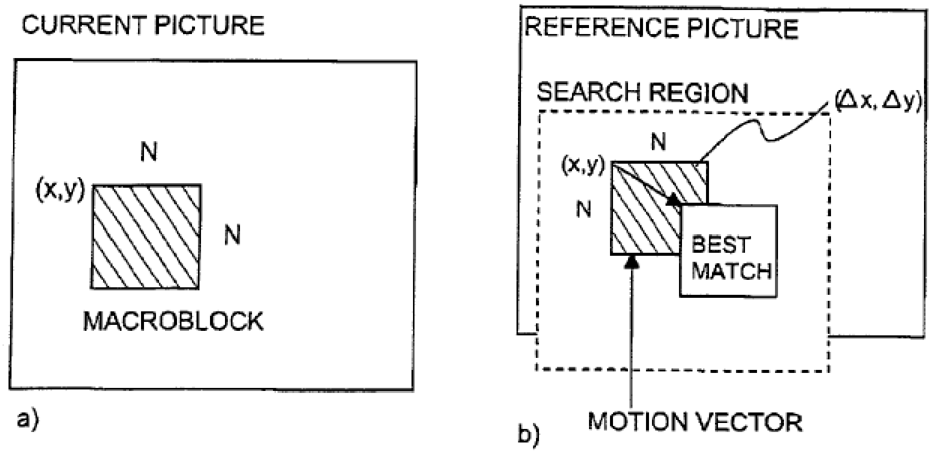
FIG. 2

4. Video was made up of a series of pictures known as frames. Each frame was segmented into blocks of pixels (e.g., referred to as video blocks,

macroblocks, sub-macroblocks, etc.). Each block contained a group of pixels, such as 16x16, 8x8, or 4x4 pixels.

B. Motion Estimation and Compensation

5. Video blocks were encoded with reference to each other. This was known as predictive coding. Two main types of predictive coding were intra-frame and inter-frame encoding. Intra-frame encoding used predictive coding within the same frame, where a block was encoded with reference to another block within the same frame. This took advantage of similarities within the same frame. Inter-frame coding allowed a block to be encoded with reference to blocks in other frames. This type of temporal prediction, called motion estimation and compensation, took advantage of similarities between different frames. For example, when an object appeared in successive frames, inter-frame prediction encoded and transmitted information for a first frame, and encoded subsequent frames by reference to reference blocks in the first frame. A motion vector indicated the displacement of a current block with respect to a reference block, for example indicating that a block moved to the right 5 pixels and moved down 3 pixels between frames. Ex-1010, ¶18, Fig. 4:



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

6. In many video standards, including H.264, blocks encoded with only intra-frame encoding were known as "I" blocks. Conversely, there were two types of inter-coded blocks: "P" and "B" blocks. "P" blocks allowed unidirectional prediction to other frames, while "B" blocks allowed bidirectional prediction, meaning blocks within the frame could be predicted in the forward *and* backwards directions. Ex-1011, 000002, 000007; Ex-1012, 000198-200.

7. Motion estimation and compensation involved identifying the movement of objects or regions between successive frames in a video sequence. In bidirectional prediction for a target block, the encoder searched for similar blocks in two reference frames, such as a past/previous reference frame and a future/subsequent reference frame, that best match the target block. This process

resulted in two motion vectors, each pointing to a different block in a different reference frame. Ex-1012, 000194-195, 000200, 000062, 000090.

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

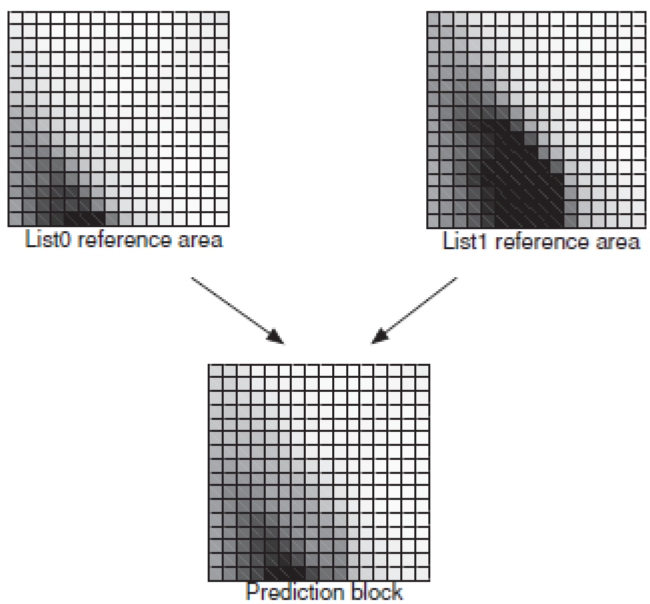


Figure 6.35 Biprediction example

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

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

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

11. The decoder then combined these blocks in the same manner as the encoder to reconstruct the bidirectional prediction of the target block, which included a prediction value for each pixel of the target block. Ex-1012, 000062, 000074. For example, when a weighted bidirectional prediction method was used at the encoder, the decoder used the same weights to combine the two reference

blocks. The decoder also extracted and decoded the residual block that was transmitted from the encoder. The original target block was reconstructed by adding the decoded residual block to the reconstructed bidirectional prediction. Ex-1012, 000059-60, 000062, 000074, 000123.

C. Subpixels and Interpolation

12. When a motion vector pointed to an integer pixel position, the values of the block at that position were used to generate a predicted block. But the H.264 video compression standard, along with multiple other standards, allowed motion vectors to have more granular subpixel resolution by pointing to fractional pixel positions (e.g., half-pixel or quarter-pixel positions), resulting in more accurate motion estimation and compensation. Ex-1011, 000002; Ex-1012, 000184. This situation arose when the best match for a target block in a reference frame was not located at an exact integer pixel position, for example where an object moved exactly one pixel between frames, but rather at a fractional (subpixel) position, e.g., where an object moved a half or quarter pixel between frames.

13. When a motion vector points to an integer pixel position, the values of the reference block are used to generate the predicted block. When a motion vector pointed to a subpixel position, the encoder/decoder used interpolation to generate the predicted block. Interpolation involved creating new pixel values at the subpixel positions based on the surrounding integer pixel values. For example, in

half-pixel interpolation, the value at a half-pixel position could be calculated as the average of six adjacent integer pixel values. Ex-1011, 000010; Ex-1012, 000187. In summary, bi-prediction used two motion vectors, which could point to integer or sub-pixel positions. Therefore, bi-prediction often involved different permutations of motion vectors that could point to integer or sub-pixel positions, with for example both vectors pointing to interpolated blocks, or one vector pointing to an interpolated block and the other pointing to an integer pixel position. *See* Ex-1011, 000010; Ex-1012, 000184-189:

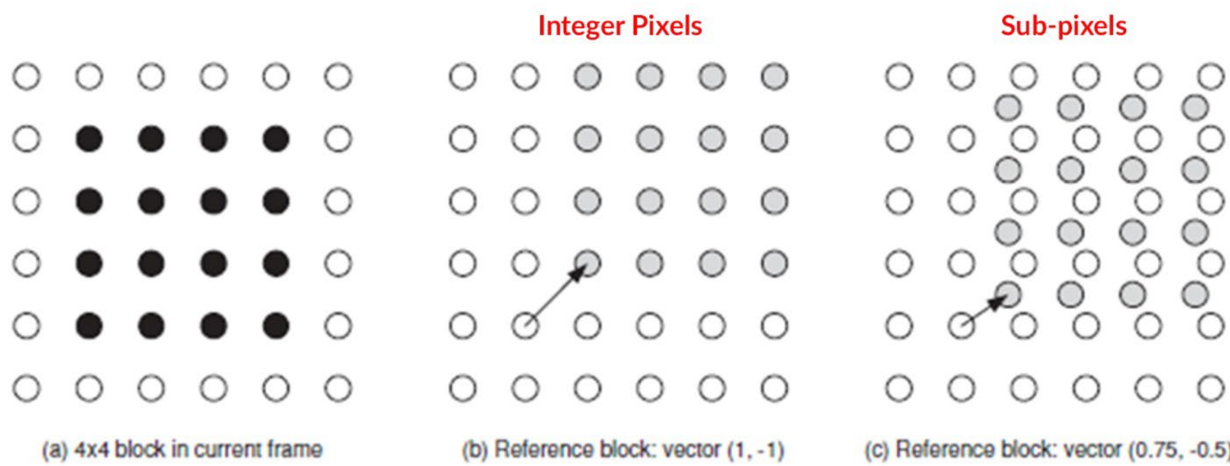


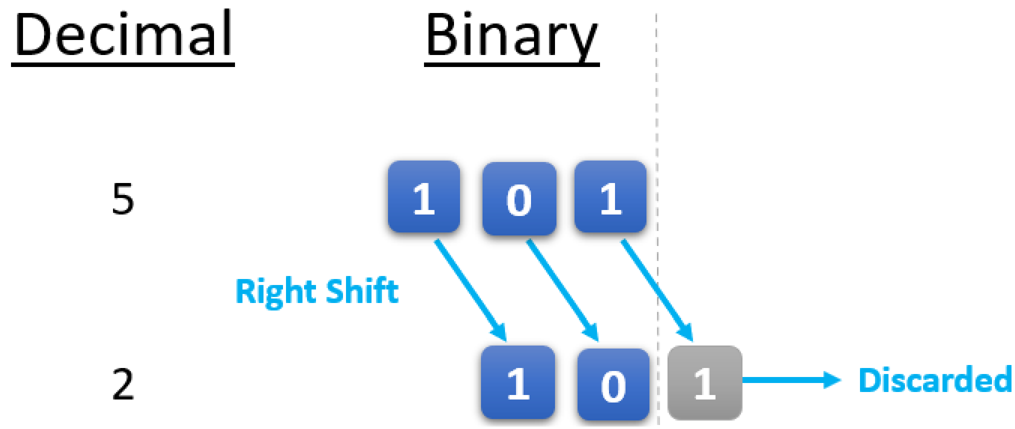
Figure 6.18 Example of integer and sub-pixel prediction

D. Precision and Bit Shifting

14. In computers, numeric values, such as pixel values (predicted or otherwise), were represented in registers or memories as binary numbers. An uncompressed binary number includes a series of positive or negative powers of 2. In a binary number, each digit is a 0 or 1. The precision of the binary number is the

number of terms in that series, which is also the number of bits needed to represent its value in this form. When a calculation may result in multiple possible values, the precision of the result of the calculation corresponds to the number of bits needed to represent the possible values in an uncompressed binary form, which could be changed by mathematical operations. For example, an uncompressed variable having values that range from 0 to 3 can be represented by a binary number of 2 bits. When multiplied by two, the result could range from 0 to 6 and thus required 3 bits of precision to represent the largest possible value 6 (110 in binary). Since multiplication and addition made numbers larger, they may have required more bits (higher precision) to represent the results. Conversely, since division made numbers smaller, it may have reduced the number of bits needed to represent the result. Higher precision allowed for a more accurate representation of a value but required more bits, while lower precision reduced the number of bits needed but might lead to a loss of accuracy. Ex-1009, 000005; Ex-1008, 7:4-19, 16:22-17:36, 20:42-47; Ex-1010, ¶¶61-62. In video, pixel values were often represented using 8 bits, which represented 2^8=256 possible values (from 0-255). A pixel on a two-color (black and white) display can be represented with a single bit having two possible values: 0 or 1. A pixel with 2 bits of precision would represent values from 0-3. A pixel with 3 bits of precision would represent values from 0-7, and so on. Motion prediction and sub-pixel interpolation involved mathematical calculations, including multiplication and addition, that increased the number of bits needed to represent the intermediate results of those calculations. For example, when two 8-bit integer pixel values were averaged together, they were first added together, resulting in a sum that may require 9 bits to represent. The sum is next divided by two, resulting in an 8-bit number. Division can cause a rounding error because, in integer arithmetic, any remainder from the division is discarded. I explain this in more detail below.

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



16. The inverse of truncation can be applied to a binary number by shifting its bits to the left by the desired number of positions. Ex-1008, 20:42-47; Ex-1013, ¶91, ¶116, ¶131. Left shifting effectively performs multiplication by 2 for each position shifted. Ex-1004, ¶46; Ex-1008, 20:42-47; Ex-1013, ¶91, ¶116, ¶131. The newly added least significant bits were filled with zeros. Each left shift effectively doubled the value and also increased the precision by one bit, as a new zero bit was added on the right.

17. Truncating a binary number is sometimes referred to as rounding the binary number. This rounding operation might cause a rounding error due to the difference between the original number and the rounded number. For example, truncating the binary number 101 (5 in decimal) by one bit results in 10 (2 in decimal). This is equivalent to dividing 5 by 2, which equals 2.5. Since the remainder is discarded, there is a rounding error of 0.5. This could occur when averaging 2 and 3, which normally would result in 2.5, but with binary numbers

results in $\frac{2+3}{2} = \frac{5}{2} = 2$. Rounding errors could accumulate in calculations, especially when many operations were performed sequentially. This could lead to significant discrepancies between the computed result and the true value. In this example, if the above average is calculated twice and then added together, the expected result would be 2.5 + 2.5 = 5. However, because two right-shift (division) operations occur during the average, the result is 2 + 2 = 4, meaning the rounding error is 1. To prevent this error, it was known in the art to maintain higher precision for the calculations, e.g., by delaying rounding (division/right-shift) operations. *E.g.*, Ex-1008, 7:4-19. Applying that concept to this example, using basic arithmetic, the division in the averaging operation would be delayed until the end, thereby reaching the expected result of 5 and preventing rounding error:

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

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

19. One well-known and commonly used approach to address accumulation of rounding errors was to maintain higher precision in intermediate steps of calculations. Ex-1008, 7:4-19 (“An advantage of embodiments according

to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."), 16:22-17:36 ("[A] higher bit-depth prediction[] and a higher bit-depth reconstruction residual information[] may lead to higher precise reconstructed samples[] of the block[], and therefore to a smaller needed residual information, as in systems, wherein a rounding of prediction samples and of reconstructed residual samples occurs before the prediction and residual reconstruction process."); Ex-1009, 000005 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding and clamping. It is possible to achieve additional accuracy by retaining a higher precision result after the first stage of filtering. … The two shifts are chosen so as to (a) add up to the required shift for normalizing the filters and (b) to allow for a 16 bit implementation - where the intermediate values in the second filtering operation are within 16 bits."); Ex-1010, ¶¶61-62 ("[T]runcation of the ¼ resolution sub-pixel values has a deleterious effect on the precision of some of the ¼ resolution sub-pixel values. Specifically, the ¼ resolution sub-pixel values are less precise than they would be if calculated from values that had not been truncated and clipped. … . In the encoder the interpolation method according to TML6 works like the previously described TML5 interpolation method, except that maximum precision is retained throughout. This is achieved by using intermediate values which are neither rounded nor clipped.”).

20. Since each rounding step had the potential to discard information and thereby introduce rounding errors, this practice delayed rounding and used more bits to represent numbers during the computation process than were used in the final output. By doing so, the accumulation of rounding errors was minimized, as intermediate operations had a finer granularity and could represent values more accurately. *Id*.

II. THE ’267 PATENT

A. Overview

21. The ’267 patent is directed to “[a]pparatuses, methods and computer programs … for utilizing motion prediction in video coding.” Ex-1001, Abstract. The ’267 patent discusses a process for generating a bi-directional prediction for a current block, including “us[ing] motion vector information to determine which block is used as a first reference block for the current block and which block is used as a second reference block for the current block,” “us[ing] some pixel values of the first reference block to obtain first prediction values and some pixel values

of the second reference block to obtain second prediction values," and combining "the two prediction values." *See, e.g.*, Ex-1001, 12:41-55, 13:43-55, Fig. 10:

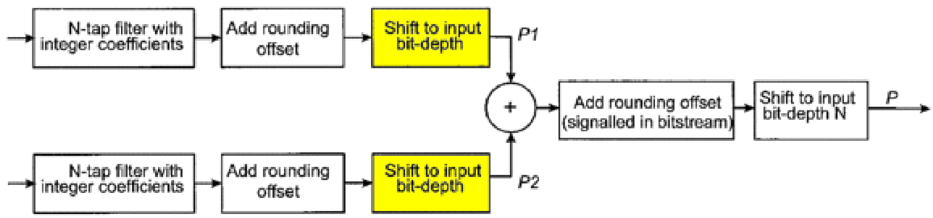


Fig. 10

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

22. The '267 patent does not purport to invent this conventional process of bi-directional prediction, admitting that video coding processes according to standards such as MPEG-2, H.263, and H.264 were known in the art. Ex-1001, 1:34-46 ("Background Information" section). The '267 patent states that "Background" art includes motion compensated prediction and specifically "bi-directional prediction" (e.g., Ex-1001, 2:35-59), where reference blocks are determined based on motion vectors (e.g., Ex-1001, 2:20-34, 3:12-18), predictions are determined based on reference blocks (e.g., Ex-1001, 1:34-46), a bi-directional prediction is obtained by combining two predictions based on two reference blocks

(e.g., Ex-1001, 3:49-55, 3:66-4:20), and residual data is calculated as a difference between the prediction and the current block, encoded, and later used to reconstruct the current block (e.g., Ex-1001, 1:52-59, 3:25-30, 2:1-12).

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

24. The purportedly inventive concept of the ’267 patent is to maintain prediction signals “in a higher precision during the prediction calculation” and reduce the precision “after the two or more prediction signals have been combined

with each other." Ex-1001, 4:29-43, 6:51-57, 12:41-13:55. By doing so, the '267 patent claims to "enable[] reducing the effect of rounding errors in bi-directional and multi-directional prediction." Ex-1001, 4:29-35, 6:51-57, Fig. 11:

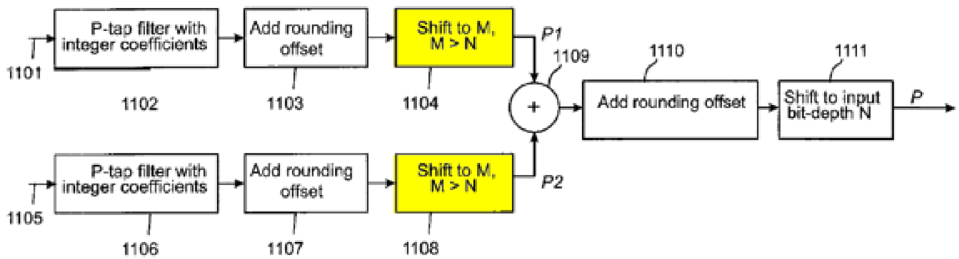


Fig. 11

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

B. Prosecution History

25. The '267 patent was allowed after one Office Action, which included an obviousness-type double patenting rejection, in response to which the Applicant submitted a terminal disclaimer. Ex-1002, 000123-130, 000155-156, 000159.

26. The application for U.S. Patent No. 9,432,693, a parent of the '267 patent, received Office Actions with substantive prior-art rejections. The prosecution history of the '693 patent includes three Office Actions, which present §103 rejections based on U.S. 2013/0142262 ("Ye"), U.S. 2009/0087111 ("Noda"), and U.S. 2010/0086027 ("Panchal"). Ex-1007, 000139-157, 000200-218, 000246-261. The original claims recited, among other limitations,

"determining a block of pixels of a video representation encoded in a bitstream, values of said pixels having a first precision" and "using said first reference pixel location to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision." Ex-1007, 000036.

27. The Examiner initially cited to Ye's teachings of integer pixel precision and fractional pixel precision as respectively teaching the recited "first precision" and "second precision." Ex-1007, 000146-148, 000203-204, 000207-208. In response, the Applicant distinguished the cited teachings of Ye by amending the claims to recite "wherein the first precision indicates the number of bits needed to represent values of said pixels" and "wherein the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction[.]" Ex-1007, 000236, 000243.

28. The Examiner cited Ye's weighted prediction teachings (e.g., the equation $P(x,y) = (w \cdot P_0(x,y)+(W-w) \cdot P_1(x,y)+ W/2) >> S)$ in the next Office Action, asserting that multiplying $P_0(x,y)$ with a weight w "increase[s] the precision of $P_0(x,y)$" and that this suggests the second precision. Ex-1007, 000249-251. However, the Examiner acknowledged that Ye does not have explicit teachings that "said first prediction having a second precision, which is <u>higher than said first precision</u>, wherein the second precision indicates <u>the number of bits needed to represent values of said first prediction and values of said second</u>

prediction[,]" among other limitations. Ex-1007, 000252 (emphasis in original). The Examiner cited to Noda's teaching of increasing pixel bit depth to address Ye's acknowledged deficiencies.

29. In response, the Applicant argued that neither reference teaches that "the reference blocks are at a first precision and the first and second predictions are at a second precision." Ex-1007, 000280-282. Regarding Ye, the Applicant did not dispute that the weighted prediction (e.g., $w \cdot P_0(x,y)$, $(W-w) \cdot P_1(x,y)$) teaches the claimed first or second prediction. The Applicant argued that "$w \cdot P_0(x,y)$ and (W-w)·P1(x,y) do not have increase[d] precision relative to $P_0(x,y)$ and P1(x,y), respectively, and, instead, simply serve to change the range of values," and that "[s]uch an increase in the range of values as in Ye does not teach or suggest that the precision increases such that Ye fails to teach or suggest any increase in precision from the reference blocks to the first and second predictions." Ex-1007, 000281.

30. Regarding Noda, the Applicant asserted that "Noda discloses increasing the bit depth of each pixel of an input image having an N bit depth to a reference image of (N+M) bit depth, and only then generating a prediction image of the (N+M) bit depth from the reference image of the (N+M) bit depth." Ex-1007, 000281-282. Based on this characterization of Noda, the Applicant argued that "Noda fails to teach or suggest any increase in precision from the reference

blocks to the first and second predictions. Instead, both the reference block from which the prediction image is generated as well as the prediction image itself have the same precision, that is, a (N+M) bit depth." *Id*. The Examiner allowed the application for the '693 patent after this response. Ex-1007, 000294-311.

C. Priority Date

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

D. Challenged Claims

32. I understand that Petitioner is challenging the validity of claims 19-36 of the '267 patent in the Petition for *Inter Partes* Review to which this declaration will be attached. Those claims are reproduced in Appendix 3. While the Petition and this declaration are directed to the challenged claims, I have considered all

claims 1-36 of the ’267 patent, as well as portions of the ’267 patent prosecution history in forming my opinions.

III. LEVEL OF ORDINARY SKILL IN THE ART

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

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

35. I also considered the sophistication of the technologies involved, and the educational background and experience of those actively working in the field at the time. I also considered the level of education that would be necessary to understand the ’267 patent. Finally, I placed myself back in the relevant period of time and considered the engineers and programmers that I have worked with and led in the field of video encoding/decoding.

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

IV. CLAIM CONSTRUCTION

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

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

A. “precision”

39. Based on my review of the claims and specification of the ’267 patent, it is my opinion that a POSITA would have understood “precision” is satisfied by,

but is not necessarily limited to, "a number of bits needed to represent possible values."

40. Claims 19, 25 and 31 recite that "the pixels of the current block, the first reference block, and the second reference block have values with a first precision," "said first prediction having a second precision," "said second prediction having the second precision," and "precision of said combined prediction" Ex-1001, cls. 19, 25, 31. Dependent claims 24, 30, and 36 further recite "wherein the first *precision indicates a number of bits needed to represent the values* of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction."[1] Ex-1001, cls. 24, 30, 36. Therefore, these claim limitations can be satisfied when precision indicates a number of bits needed to represent the possible values of binary data, including uncompressed representations of binary data.

41. This interpretation is confirmed by the specification of the '267 patent, which uses the term "precision" to refer to a number of bits representing possible values, with examples of the number of bits used to represent the possible pixel and prediction values. *See, e.g.*, Ex-1001, 12:41-13:18 ("The precision M is higher than the precision of the expected prediction value. For example, *pixel values and the prediction values may be represented by N bits* wherein M>N. In

---

[1]All annotations/emphasis added unless otherwise noted.

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

42. The interpretation is further confirmed by the Applicant's statement during prosecution of a parent application (U.S. Patent Application No. 13/344,893, issued as U.S. Patent No. 9,432,693) of the '267 patent. *See, e.g.*, Ex-1007, 000280 ("***The first precision indicates the number of bits needed to represent values of said pixels***. … The second precision indicates the number of bits needed to represent values of the first prediction and values of the second prediction."), 000243.

V. INVALIDITY GROUNDS

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

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

Grounds	Claims	Statutory Basis	Prior Art
1	19-36	§ 103	Walker
2	19-36	§ 103	Karczewicz-I in view of Karczewicz-II

A. Grounds 1: Claims 19-36 are Rendered Obvious by Walker

45. It is my opinion that a POSITA would have found Claims 19-36 obvious based on the teachings of Walker.

1. U.S. Patent Application Publication No. 2005/0281334 (“Walker”) (Ex-1004)

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

47. Walker was published December 22, 2005, and was filed as application 11/120,513 on May 2, 2005. Therefore, I understand Walker is prior art under at least pre-AIA §102(b) because it was published more than one year before the earliest possible filing date for the ’267 patent, its provisional application’s

filing date of January 7, 2011. I further understand Walker is prior art under at least pre-AIA §§102(a) and 102(e) because it was filed and published before January 7, 2011.

48. Walker is directed to "methods and apparatus for decoding compressed video data where various weighted prediction methods were used for encoding the video data." Ex-1004, ¶3. Walker "allows decoding of multiple weighted bi-directional encoding schemes with a single decoder[,]" including, for example, encoding schemes under MPEG-4 and H.264. Ex-1004, ¶8, ¶¶46-49, ¶¶58-72. For example, Walker teaches weighted prediction under H.264, where a combined prediction is obtained based upon a weighted combination of predictions of two reference blocks. Ex-1004, ¶¶58-70. On the decoding side, Walker proposes "a universal formula that is used by embedded hardware, … to decode weighted prediction frames" encoded in the various implementations described therein. Ex-1004, ¶72, ¶92. Moreover, Walker teaches obtaining predictions for subpixels via interpolation and that "[p]ixel interpolation can be used to improve the performance of motion compensated predictive coding." Ex-1004, ¶111, ¶114, Figs. 8-9.

49. Walker includes multiple figures and corresponding teachings directed to aspects of video encoding and decoding. The teachings and figures of Walker's embodiments, as relied on by this Declaration, are explained as aspects

of video encoding/decoding that are used in conjunction with each other. For example, Figure 1 shows "a general communications system for encoding and decoding streaming pictures" that includes "multiple types of encoder devices … and a decoder device[.]" Ex-1004, Fig. 1, ¶25. Walker then explains interrelated aspects of those video encoders and decoders.

50. Walker teaches that the encoder devices of Figure 1 each "performs weighted bi-directional prediction by one of a plurality of methods." Ex-1004, ¶25; Walker further teaches a decoder that "can receive the various encoded data" and includes "predictive decoding modules" that "decode the received … inter-coded data[.]" Ex-1004, ¶26. Walker's "[p]redictive decoding module 165 includes logic to decode all of the various types of weighted prediction encoded by weighted prediction modules 135, 140 and 145." Walker teaches, instead of having separate sections of code to decode the various types of weighted prediction, its decoder utilizes a single method of decoding to decode all types of weighted prediction described therein. Ex-1004, ¶26 ("Instead of having separate sections of code to decode the various types of weighted prediction, decoder device 155 utilizes a pre-processor module 170 to manipulate the weighted prediction parameters such that a single method of decoding can be used to decode all of the types of weighted prediction."); *see also* ¶45 ("A versatile decoder, such as decoder device 155 depicted in FIG. 1, should be able to decode video that was encoded by multiple

implementations with various encoded bit configurations and various types of weighted/non-weighted prediction methods.”). Therefore, Walker teaches at least one decoder device of Figure 1 performs weighed bi-directional prediction.

51. Figure 5 illustrates “an example of a weighted B Frame construction process[,]” which is carried out in both encoding and decoding processes. Ex-1004, Fig. 5, ¶¶34-35. Walker teaches the process of Figure 5 being performed by the encoder devices and decoder device of Figure 1. Ex-1004, ¶29 (“FIGS. 3, 4, 5 and 7 illustrate various inter-coding processes including those used for constructing P Frames, B Frames, weighted B Frames and H.264 predicted frames. The encoder devices 105, 110 and 115 and the decoder device 155 depicted in FIG. 1 can perform these processes in whole or in part.”). Because Walker teaches at least one decoder device of Figure 1 performing weighed bi-directional prediction, which is a technique for constructing B frames, Walker’s disclosure with respect to Figure 5 encompasses this technique.

52. Figure 8 illustrates “an example of a decoder process for decoding multiple encoder implementation of weighted bi-directional predicted video data.” Ex-1004, Fig. 8, ¶110. Walker teaches the process of Figure 8 being carried out by the decoder device of Figure 1. Ex-1004, ¶110 (“Process 800 could be carried out with a device such as decoder device 155 depicted in FIG. 1.”). A POSITA would have understood that, when the decoder device of Figure 1 carries out the process

of Figure 8, the process is used to decode weighted bi-directional predicted video data generated by the encoder devices of Figure 1.

53. Walker teaches using pixel interpolation as part of the process of Figure 8. Ex-1004, ¶111. Walker provides further details about the pixel interpolation process in Figure 9, which illustrates "an example of half-pixel interpolation for use in motion compensation." Ex-1004, ¶114, Fig. 9.

54. In short, the teachings and figures of Walker's embodiments are used in conjunction with each other. A POSITA would have been motivated to combine Walker's teachings, as explained for its embodiments, because Walker presents those teachings as complementary aspects of video encoders and decoders that are meant to be used together. Ex-1004, ¶25, ¶29, ¶¶34-35, ¶¶45-46, ¶72, ¶¶110-111, ¶114, Figs. 1, 5, 8-9. Moreover, it was known in the art that weighted bi-directional prediction and sub-pixel interpolation were used together because widespread industry standards including the ubiquitous H.264 standard utilized those concepts together. *See, e.g.*, Ex-1006, ¶¶46-47, ¶51, ¶¶68-69, ¶74, ¶86, ¶88, ¶93; Ex-1014, 000188-193.

55. Walker is in the same field of endeavor as the '267 patent because it is directed to video encoding/decoding, and in particular motion prediction. *See supra* §III; Ex-1004, ¶3 ("This invention relates to methods and apparatus for decoding compressed video data where various weighted prediction methods were

used for encoding the video data."). Walker teaches and improves upon the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex-1004, ¶¶46-49, ¶¶58-72, ¶112 *with* Ex-1001, 2:60-3:11, 9:26-41.

56. Walker teaches limitations that the Examiner found missing in Ye. As an example, Walker includes explicit teachings about the numbers of bits needed to represent its pixel values, weighted predictions, and combined predictions. *Infra* §§V.A.2[19b-19d, 19f]/[25b-25d, 25f]/[31b-31d, 31f]. This type of explicit teaching was acknowledged by the Examiner to be missing from Ye. *Supra* §II.B; Ex-1007, 000252. As explained below, Walker teaches the limitations of claims 19-36 at least partly based on the teachings that Ye lacks.

2. Independent Claims 19, 25, and 31

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

57. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that Walker teaches the preamble.

58. Walker "relates to *methods* and apparatus for *decoding* compressed video data where various weighted prediction methods were used for encoding the video data." Ex-1004, ¶3. Walker teaches "a general communications system for encoding and decoding streaming pictures." Ex-1004, ¶25, Fig. 1:

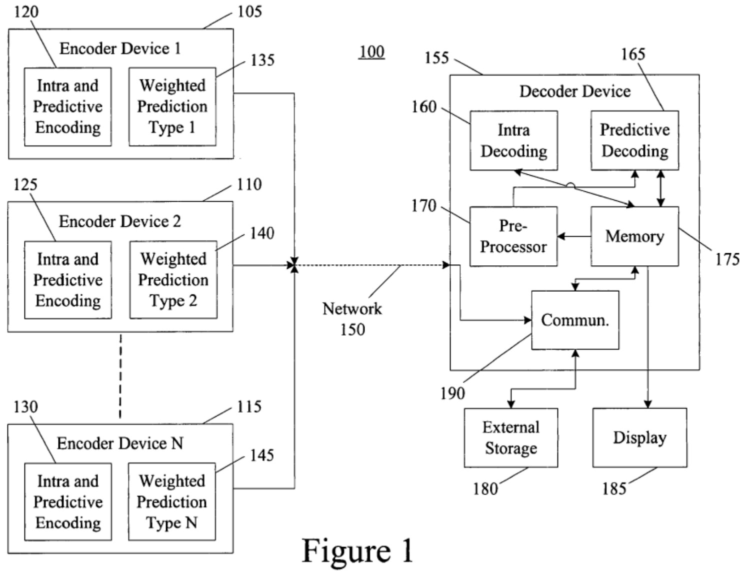


Figure 1

Walker teaches "predictive decoding modules" in a decoder device that "decode the received inter-coded data[.]" Ex-1004, ¶26. Walker's teachings include a "[p]redictive decoding module 165" that "decode[s] all of the various types of weighted prediction encoded by weighted prediction modules 135, 140 and 145." Ex-1004, ¶26; *see also* ¶111, ¶17, Fig. 8:

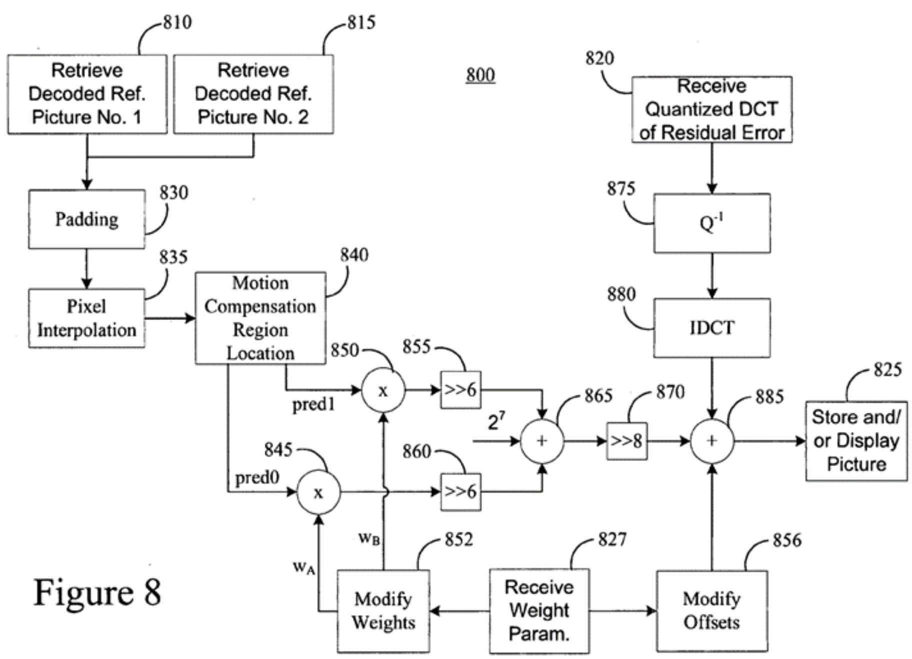


Figure 8

59. Walker decodes a block of pixels, including with bi-directional motion prediction on macroblocks. Ex-1004, ¶35 ("The encoded motion vectors 525 and 585 are decoded and used to locate the already reconstructed best matching macroblock 565 in previous reference picture 510, and to locate the already reconstructed best matching macroblock 590 in subsequent reference picture 575."), ¶8, ¶21, ¶26, ¶110, ¶111, Fig. 8. A macroblock is a block of pixels. Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels."). Therefore, Walker teaches a method for decoding a block of pixels.

60. As explained below, operations related to weighted bi-directional prediction, as taught by Walker, teach the limitations of claim 19. *Infra* §§V.A.2[19b-19g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1. Therefore, Walker teaches a method for decoding a block of pixels, comprising the operations explained below for limitations [19b]-[19g].

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

61. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble

states a limitation, it is my opinion that Walker teaches the preamble and any additional limitations of element [25a].

62. As explained above for [19a], Walker teaches a method for decoding a block of pixels, including operations as described for limitations [19b]-[19g]. *Supra* §V.A.2[19a]. As explained below, Walker teaches a decoder device for decoding a block of pixels that performs operations as described for limitations [25b]-[25g]. *Infra* §§V.A.2[25b-25g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1.

63. Walker teaches implementing its methods using a processor and memory that includes computer program code (e.g., software module). Ex-1004, ¶123 ("The steps of a method or algorithm described in connection with the examples disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art."). Because Walker's decoder device performs its method of encoding a block of pixels, Walker teaches implementing the decoder device as an apparatus that comprises such a processor and memory.

64. A POSITA would have been knowledgeable about basic computer architecture and understood that, in a conventional computing device, the processor executes computer program code in the memory to cause the device to carry out its functionalities. Therefore, Walker's decoder device teaches an apparatus for decoding a block of pixels, the apparatus comprising: at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to perform operations as described for limitations [25b]-[25g].

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

65. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble states a limitation, it is my opinion that Walker teaches the preamble.

66. Walker teaches implementing its methods using program code (e.g., software module) residing on computer readable storage media, including memory types, hard disks and CD-ROMs that are executable by a processor. Ex-1004, ¶123 ("The steps of a method or algorithm described in connection with the examples

disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art."). A POSITA would have understood that the types of storage media disclosed by Walker, including RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, are forms of non-transitory computer readable storage medium, as opposed to transitory signals. Ex-1004, ¶123. A POSITA would have found it obvious that the software module, which is stored in non-transitory computer readable storage medium and executable by a processor, includes computer executable program code portions that comprise program code instructions.

67. As explained above, Walker teaches an apparatus with program code that implements its teachings (*supra* §V.A.2[25a]), stored on non-transitory medium. Therefore, Walker applies its teachings to a computer-program product. The operations taught by Walker, which are performed by the software module, teach the limitations of claim 31. *Infra* §§V.A.2[31b-31g]. Walker teaches implementing its systems, processes, and techniques in conjunction with each other. *Supra* §V.A.1. Additionally, a POSITA would have found it obvious to have

a computer program product that includes a storage medium storing the software module, since computer program products have almost universally stored their software program code in non-transitory mediums (e.g., hard disk, memory, or CD-ROM) for decades. *See* Ex-1004, ¶123. For example, software program code instructions saved on a CD-ROM had been a very common form of computer program product since the 1990s. Therefore, Walker teaches a computer program product for decoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to perform the operations recited in claim 31.

[19b]/257b]/[31b] [determining/determine], for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector, wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

68. Walker teaches limitations [19b], [25b], and [31b]. First, Walker teaches **determining, for a current block** (e.g., Walker's "current macroblock"), **a first reference block** (e.g., "the already reconstructed best matching macroblock 565 in previous reference picture 510") **based on a first motion vector** (e.g., motion vector 525) **and a second reference block** (e.g., "the already reconstructed best matching macroblock 590 in subsequent reference picture 575") **based on a**

second motion vector (e.g., motion vector 585). Walker's Figure 8 teaches "a flow chart of an example of a decoder process for decoding multiple encoder implementations of weighted bi-directional predicted video data" Ex-1004, ¶17, Fig.8:

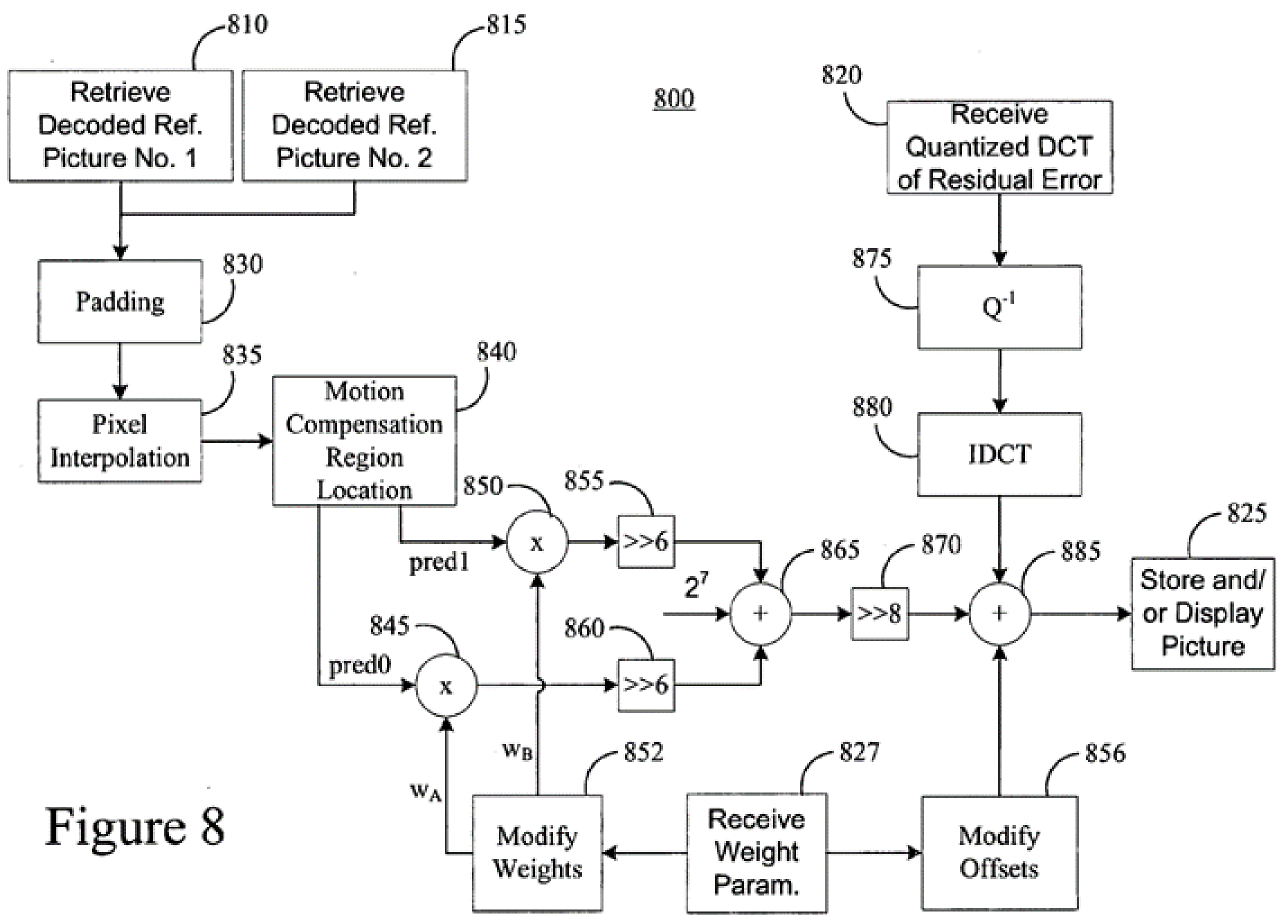


Figure 8

Walker teaches the decoder carrying out the process of Figure 8. Ex-1004, ¶110 ("FIG. 8 is a flow chart of an example of a decoder process for decoding multiple encoder implementations of weighted bi-directional predicted video data. Process 800 could be carried out with a device such as decoder device 155 depicted in FIG. 1. The process is carried out by three main components including predictive decoding component 165, pre-processor component 170 such as a DSP that is external to the predictive decoding component 165 and at least one memory module 175 to store various data.”).

69. Prior to reconstructing a B frame, Walker reconstructs backward reference frames and forward reference frames to obtain reconstructed reference frames. Ex-1004, ¶35 (“Reconstruction of a B Frame section can be started after both the backward reference frame (or a portion of a picture or frame that is being referenced) and the forward reference frame (or a portion of a picture or frame that is being referenced) are reconstructed.”), ¶111 (“Decoding of an inter-coded picture can start when the reference picture or pictures are already decoded and stored in memory such as memory module 175 in FIG. 1. … When doing bi-directional predicted decoding, both past and future reference pictures are stored in memory. Retrieving steps 810 and 815 access the first reference picture (a past reference picture, for example) and the second reference picture (a future reference picture, for example), respectively, from memory.”). For each macroblock of a B frame, Walker teaches determining two “reconstructed best matching” reference blocks based on two motion vectors: forward and backward. Ex-1004, ¶35; *see also* ¶111. Each B frame combines forward and backward motion vectors that reference blocks in I or P frames. Ex-1004, ¶28 (“Each B frame can combine

forward and backward motion vectors and residual errors referenced to I frame 22A or predicted P frames 24[.]").

70. Walker determines a *first*, reconstructed best-matching reference block for the current macroblock based on a first, *backward*-pointing motion vector. The backward motion vector points to a previous frame (reference picture). For the backwards motion vector, "[t]he encoded motion vector[] 525 … [is] decoded and used to locate *the already reconstructed best matching macroblock 565 in previous reference picture 510*[.]" Ex-1004, ¶35 ("The encoded motion vectors 525 and 585 are decoded and used to locate the already reconstructed best matching macroblock 565 in previous reference picture 510, and to locate the already reconstructed best matching macroblock 590 in subsequent reference picture 575."); *see also* ¶111 ("In the case of bi-directional prediction, there are two motion vectors. … The decoder uses the motion vectors to perform motion compensation region location, step 840, to locate the best matching regions among the interpolated pixels."), Fig. 8.

71. Walker determines a *second*, reconstructed best-matching reference block for the current macroblock based on a second, *forward*-pointing motion vector. For the forward-pointing motion vector, Walker applies a similar approach to the one explained above for the backwards vector: "[t]he encoded motion vector[] … 585 [is] decoded and used to locate … *the already reconstructed best*

matching macroblock 590 in subsequent reference picture 575." *Id.*; *see also* ¶111, Fig. 8.

72. Walker determines both reference blocks (e.g., reconstructed best matching macroblocks) for the current macroblock by locating the reference blocks based on their corresponding motion vectors.[2]

73. Second, Walker teaches that **the pixels of the current block, the first reference block, and the second reference block have values with a first precision** (e.g., 8 bits). Walker teaches that the pixels in its macroblocks have 8-bit luminance and chrominance values. *See, e.g.*, Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels. Pixels can be defined by an 8-bit luminance value (Y) and two 8-bit chrominance values (Cr and Cb)."), ¶93, Table 4 (showing that pred0 and pred1 of the already reconstructed best matching reference macroblocks are values of 8-bit, where w_A and w_B each has 15 bits), ¶49 ("[W]here $pred1_{ij}$ and $pred2_{ij}$, are 8-bit luminance and chrominance samples from prediction blocks from two reference frames (one past, one future)[.]"), ¶52 ("[W]here $pred1_{ij}$ and $pred2_{ij}$, are 8-bit luminance and chrominance samples from prediction blocks from the two

[2] The '267 patent admits that determining reference blocks based on motion vectors were known in the art by describing it in the "Background Information" section. *See* Ex-1001, 2:20-34, 3:12-18 ("Each of these motion vectors represents the displacement of the image block in the picture to be coded (in the encoder) or decoded (at the decoder) and the prediction source block in one of the previously coded or decoded images (or pictures).").

reference frames (one past, one future)[.]”), ¶57 (“WMV9 implements B Frames similarly to MPEG-4 with (5) above, where $pred1_{ij}$ and $pred2_{ij}$, are the 8-bit luminance and chrominance samples from prediction blocks from the two reference frames (one past, one future)[.]”), ¶60 (“where pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future)[.]”). Therefore, Walker teaches that 8 bits are needed to represent the possible values of the pixels of the current block, the first reference block, and the second reference block.

74. As explained above, the term “precision” is at least satisfied by “a number of bits needed to represent possible values.” *Supra* §IV.A. Because 8 bits are needed to represent the possible luminance and chrominance values of the pixels of the current block, the first reference block, and the second reference block, these pixels all have values with a same “first precision.”

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

75. It is my opinion that Walker teaches limitations [19c], [25c], and [31c].

76. As explained above, Walker teaches at least one decoder device of Figure 1 performing weighted bi-directional prediction, carrying out the process

depicted in Figure 8. *Supra* §V.A.1.As explained above, Walker "allows decoding of multiple weighted bi-directional encoding schemes with a single decoder." Ex-1004, ¶8; *supra* §V.A.2[19a]. Walker teaches that its decoder "identifies weighting factors used in the particular encoding scheme and modifies the weight factors to conform to a universal bit configuration for decoding with a universal formula." Ex-1004, ¶8. Walker teaches four widely used implementations for video compression, including MPEG-4, RV9, WMV9, and H.264, each of which implements weighted prediction in its own way. *See, e.g.*, Ex-1004, ¶¶47-49 (discussing MPEG-4 implementation), ¶¶50-55 (discussing RV9 implementation), ¶¶56-57 (discussing WMV9 implementation), ¶¶58-70 (discussing H.264 implementation). Walker teaches a "universal formula" Equation 18 that is used by its decoder to "decode weighted prediction frames encoded in any of the four implementations":

$$pred_{ij} = ((((pred0)w_A)>>6 + ((pred1)w_B)>>6) + 2^7)>>8 + Offset \qquad (18)$$

(Ex-1004, ¶72, ¶92), "where w_A, w_B and Offset can be calculated … based on what implementation was used to encode the weight parameters[.]" Ex-1004, ¶93.

77. Walker teaches **using said first reference block** (e.g., reconstructed best matching macroblock in previous reference picture) **to obtain a first prediction** (e.g., "$((pred0)w_A)>>6$"). Here, the value "$((pred0)w_A)>>6$" is the weighted prediction from the first reference block: in Equation 18, "pred0 and

pred1" represent "[t]he luma and chroma values of the two best matching prediction regions." Ex-1004, ¶111, ¶60.[3]

78. Here, "the two best matching prediction regions" refer to the reconstructed best matching macroblocks in previous and subsequent reference pictures mentioned elsewhere in Walker, which satisfy the recited "first reference block" and "second reference block." *Supra* §V.A.2[19b]/[25b]/[31b]; Ex-1004, ¶35, ¶111. This is clear from the context of Walker and consistent with how bi-predicted blocks had worked since the 1990s.

79. Walker teaches executing a series of computational operations based on Equation 18. Ex-1004, ¶¶92-93. The operations are outlined in Table 4:

[3] Walker teaches that Equation 14 is a simplified equation that represents Equations 9, 10, and 13 of H.264 implementation (Ex-1004, ¶74). Walker teaches that Equation 16 is derived from Equation 14, and that Equation 18 is further derived from Equation 16. *See* Ex-1004, ¶¶90-91 (explaining that Equation 18 is derived from Equation 16), ¶¶85-86 (explaining that Equation 16 is derived from Equation 14). Therefore, the definitions of variables in these earlier equations are applicable to Equation 18.

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

Ex-1004, ¶93

80. Operation No 1 computes the values "$(pred0)w_A$" and "$(pred1)w_B$," each of which is a product of a sample value (e.g., pred0, pred1) and a weight (e.g., w_A, w_B). Operation No. 2 shifts the values to the right by 6 bits to obtain weighted predictions "$((pred0)w_A)>>6$" and "$((pred1)w_B)>>6$." Ex-1004, ¶¶92-93. The weighted prediction value "$((pred0)w_A)>>6$" is calculated based on the sample value pred0 and is thus obtained using the first reference block. Ex-1004, ¶35, ¶60, ¶93, ¶111, ¶116, Fig. 10, Fig. 8.

81. Walker's "$((pred0)w_A)>>6$" constitutes a first prediction because it is a value calculated by multiplying a pixel value (e.g., pred0) from a first reference

block with a scaling factor (e.g., weight w_A) and shifting the product to the right by 6 bits, which is used for motion prediction of a current block.[4] Ex-1004, ¶93, Fig. 8, ¶111 ("The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights w_A and w_B respectively (weight modification is discussed below). After applying the weights, both weighted regions are right bit-shifted 6 bits, steps 855 and 860, and added, step 865, to the rounding factor [2^7] and then right bit shifted 8

[4] This is consistent with the usage of "prediction" in the '267 patent specification, which includes embodiments where predictions are calculated by performing mathematical operations on pixel values in reference blocks, such as applying a scaling factor (e.g., weights or other coefficients) on the pixel values and shifting bits of the result. For example, the '267 patent describes "us[ing] some pixel values of the first reference block to obtain first prediction values and some pixel values of the second reference block to obtain second prediction values." Ex-1001, 12:41-13:18. "[I]f a first motion vector points to a fraction of a pixel, … a P-tap filter such as a six-tap filter in which P pixel values of the reference block are *used to calculate the prediction value*" by, e.g., multiplying each pixel value with a weight (e.g., 1, -5, 20). *Id*. ("Hence, the filter 1102 would receive 1101 the pixel values of pixels E, F, G, H, I and J and filter these values by the equation P1=(E1−5*F1+20*G1+20*H1−5*I1+J1)."); *see also* Ex-1001, 13:19-42, 14:11-22. After applying the P-tap filter, "the sum may be shifted … to the right so that the precision of the sum becomes M bits." Ex-1001, 12:41-13:18. The '267 patent further states that "if a motion vector of one of the prediction directions point to an integer sample, the bit-depth of prediction samples with integer accuracy may be increased by shifting the samples to the left[.]" Ex-1001, 14:4-10. Shifting a binary number to the left is mathematically equivalent to multiplying the binary number by a scaling factor. *Supra* §I.D.; *infra* §V.A.5. For example, shifting a binary number to the left by 5 bits is equivalent to multiplying the number by 2^5 or 32. *Id*. The '267 patent thus describes multiplying the value of a pixel sample from a reference block by a scaling factor and performing bit shifting to obtain a prediction. During prosecution, the Applicant did not dispute that weighted predictions teach first and second predictions. Ex-1007, 000280-281; *supra* §II.B.

bits, step 870, to form the combined weighted prediction."),[5] ¶35 ("The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock."), ¶93 ("[O]perations of equation (18) us[e] 15 bit weights w_A and w_B and 8 bit offsets that can accommodate all of the examples of weighted prediction described above."), ¶60, ¶116, Fig. 10.

82. Walker teaches that **said first prediction** (e.g., the value "((pred0)w_A)>>6") **having a second precision** (e.g., 17 bits)**, which is higher than said first precision** (e.g., 8 bits). In Walker's Table 4, the column named "Bitwidths Involved" includes the number of bits needed to store the possible values of terms used in calculations. Ex-1004, ¶93. The row corresponding to Operation No. 2 indicates that 17 bits are needed to represent the possible values of "((pred0)w_A)>>6" and "((pred1)w_B)>>6" and 7 bits are needed to represent the rounding factor 2^7.

83. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because 17 bits are needed to represent the possible values of the first prediction (e.g.,

[5] I note that "27" in ¶111 of Walker appears to be a typographical error. The correct number should be 2^7, which is evident from Equation 18 and Table 4. Ex-1004, ¶¶92-93.

"((pred0)w_A)>>6"), the first prediction has a second precision of 17 bits, which is higher than the first precision of 8 bits. *Supra* §V.A.2[19b]/[25b]/[31b].

84. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶35. "The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock." *Id.* "The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights w_A and w_B respectively (weight modification is discussed below)." Ex-1004, ¶111.

[19d]/[25d]/[31d] [using/use] said second reference block to obtain a second prediction, said second prediction having the second precision;

85. It is my opinion that Walker teaches limitations [19d], [25d], and [31d]. Walker teaches **using said second reference block** (e.g., the reconstructed best matching macroblock in subsequent reference picture) **to obtain a second prediction** (e.g., "((pred1)w_B)>>6 ")**, said second prediction having the second precision** (e.g., 17 bits).

86. As explained with respect to limitations [19c], [25c], and [31c], Walker's Equation 18 governs weighted predictions. Ex-1004, ¶¶92-93, Table 4; *supra* §V.A.2[19c]/[25c]/[31c]. Operation No. 2 in the process of executing this

equation computes the values "((pred0)w_A)>>6" and "((pred1)w_B)>>6." Ex-1004, ¶¶92-93, Table 4. The value "((pred1)w_B)>>6" is the product of a sample value (e.g., pred1) from a second reference block multiplied by a scaling factor (e.g., weight w_B) and shifted to the right by 6 bits, and is thus obtained using said second reference block. Ex-1004, ¶35, ¶60, ¶93, ¶111, ¶116, Fig. 10, Fig. 8; *supra* §V.A.2[19c]/[25c]/[31c]. For the same reasons as explained above for why "((pred0)w_A)>>6" is a first prediction, the value "((pred1)w_B)>>6" is a second prediction. *Supra* §V.A.2[19c]/[25c]/[31c]. Furthermore, the second prediction "((pred1)w_B)>>6" has a precision of 17 bits, which is the second precision. Ex-1004, ¶93, Table 4; *supra* §V.A.2[19c]/[25c]/[31c].

87. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶35. "The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock." *Id.* "The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights w_A and w_B respectively (weight modification is discussed below)." Ex-1004, ¶111.

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

88. It is my opinion that Walker teaches limitations [19e], [25e], and [31e]. As explained above, Equation 18 governs weighted predictions:

$$pred_{ij} = ((((pred0)w_A)\gg 6 + ((pred1)w_B)\gg 6) + 2^7)\gg 8 + \text{Offset} \quad (18)$$

Ex-1004, ¶92. Walker teaches executing a series of computational operations outlined in Table 4 to determine a prediction based on Equation 18:

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

Ex-1004, ¶93; *supra* §§V.A.2[19c-19d]/[25c-25d]/[31c-31d].

89. Walker teaches **obtaining a combined prediction based at least partly upon said first prediction** (e.g., "((pred0)w_A)>>6") **and said second**

prediction (e.g., "((pred1)w_B)>>6"). Operation No. 2 calculates a sum by adding up the first prediction (e.g., "((pred0)w_A>>6"), the second prediction (e.g., "((pred1)w_B)>>6"), and a rounding offset (e.g., 2^7). Ex-1004, ¶93, Table 4; *see supra* §§V.A.2[19c-19d]/[25c-25d]/[31c-31d]; *infra* §V.A.6.

90. This sum of Operation No. 2 (e.g., "(((pred0)w_A)>>6 + ((pred1)w_B)>>6)+ 2^7") is a combined prediction because it includes a combination of the first and second predictions. The combined prediction is calculated based on said first prediction (e.g., ((pred0)w_A)>>6) and said second prediction (e.g., ((pred1)w_B)>>6), as well as the rounding offset (e.g., 2^7). This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

91. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶35. "The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock." *Id.* "The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights w_A and w_B respectively (weight modification is discussed below)." Ex-1004, ¶111.

These weights are applied for each pixel of the block (e.g., in Operations 1 and 2), before obtaining a combined prediction for each of the pixels of the block. *Id.*

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

92. It is my opinion that Walker teaches limitations [19f], [25f], and [31f].

93. As explained above, Equation 18 governs weighted predictions:

$$pred_{ij} = ((((pred0)w_A) \gg 6 + ((pred1)w_B) \gg 6) + 2^7) \gg 8 + \text{Offset} \quad (18)$$

Ex-1004, ¶92. Walker teaches executing a series of computational operations outlined in Table 4 to determine a prediction based on Equation 18:

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

Ex-1004, ¶93; *supra* §§V.A.2[19c-19e]/[25c-25e]/[31c-31e]. Walker teaches obtaining a combined prediction based on its calculation of the sum

$((((pred0)w_A)>>6 + ((pred1)w_B)>>6)+ 2^7$ in Operation No. 2. *Supra* §V.A.2[19e]/[25e]/[31e].

94. Walker teaches **shifting bits of the combined prediction to the right** (e.g., “>>8”). Operation No. 3 teaches shifting bits of the sum to the right as emphasized above. Ex-1004, ¶¶92-93, Table 4, ¶111 (“After applying the weights, both weighted regions are right bit-shifted 6 bits, steps 855 and 860, and added, step 865, to the rounding factor [2^7] and then right bit shifted 8 bits …”). In this expression, the symbol >> means right bit-shift. *See, e.g.*, Ex-1004, ¶46 (“Digital signal processing functional symbols such as left bit shift (<<) and right bit shift (>>) will be used extensively in this discussion. Such symbols are well known in the art.”). 8 indicates the number of bits shifted.

95. Walker teaches that shifting bits to the right **decreases a precision of said combined prediction**[6] (e.g., from 19 bits to 11 bits). The column named

---

[6] Walker teaches performing additional mathematical operations on the combined prediction. Ex-1004, ¶¶92-93. For example, Operation No. 4 computes an additive offset (o_0+o_1+1). Ex-1004, ¶¶92-93. Operation No. 5 adds the additive offset to the combined prediction and uses a “Clip1” function to further limit the bitwidth of the combined prediction. Ex-1004, ¶¶92-93. The “Clip1” function limits the result of Equation 18 to 0-255. Ex-1004, ¶66. These additional operations are allowed by the claims 19, 25, and 31 because each claim recites the transitional term “comprising,” which I have been informed indicates that the claims are open-ended and do not exclude additional, unrecited elements or steps. This understanding is confirmed by dependent claims 23, 29, and 35, which like Walker, recite an additional step of “[inserting/insert] a rounding offset to the combined prediction before said decreasing.” This additional step is not excluded by the independent

"Bitwidth of Operation Result" in Table 4 includes the number of bits needed to store the result of each operation based on the possible values of the result. Ex-1004, ¶93, Table 4. After Operation No. 2, the number of bits needed to represent possible values of the combined prediction is 19. Ex-1004, ¶93, Table 4. After the right bit shift of Operation No. 3, the number of bits needed to represent possible values of the combined prediction becomes 11. *Id*. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent the combined prediction is decreased from 19 to 11, Walker teaches decreasing a precision of said combined prediction.

96. Walker applies the above-described teachings to all the pixels of the block. *E.g.*, Ex-1004, ¶35. "The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock." *Id.* "The luma and chroma values of the two best matching prediction regions, pred0 and pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights

---

claims and does not change the nature of the combined prediction—regardless of whether additional operations are performed (e.g., per claim 23), the combined prediction remains the combined prediction. Therefore, the additional operations with respect to the combined prediction, as taught by Walker, are not excluded by claims 19, 25, and 31 and do not affect Walker's teaching of relevant limitations.

w_A and w_B respectively (weight modification is discussed below)." Ex-1004, ¶111. These weights are applied for each pixel of the block (e.g., in Operations 1 and 2), before obtaining a combined prediction for each of the pixels of the block. *Id.*

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

97. It is my opinion that Walker teaches limitations [19g], [25g], and [31g].

98. Walker teaches **reconstructing the block of pixels** (e.g., reconstructed macroblock) **based on the combined prediction** (e.g., "combined weighted prediction"). As explained above, Walker teaches obtaining a combined prediction and decreasing a precision of the combined prediction using a weighted bi-directional prediction method. *Supra* §§V.A.2[19e-19f]/[25e-25f]/[31e-31f].

99. The value of the combined prediction is obtained for each pixel of the current block, which includes multiple pixels, based on pixel values of the reference blocks. *See* Ex-1004, ¶30 ("A macroblock is made up of 16×16 pixels. Pixels can be defined by an 8-bit luminance value (Y) and two 8-bit chrominance values (Cr and Cb)."), ¶60 ("where pred0 and pred1, are 8-bit luminance and chrominance (also known as luma and chroma) samples from prediction blocks from the two reference frames (one past, one future)[.]"), ¶35.

100. As was well known to those skilled in the art, when the value of the combined prediction is determined for each pixel of the current macroblock using a weighted bi-directional prediction method based on two reference blocks, the combined prediction for the current block is determined. *Supra* §I.B. The collective combined prediction values are referred to as a "combined weighted prediction macroblock" in Walker. Ex-1004, ¶35 ("The encoded weight w_1 is decoded and applied to reconstructed best matching previous macroblock 565 and the encoded weight w_2 is decoded and applied to reconstructed best matching subsequent macroblock 590 to form a combined weighted prediction macroblock."). Walker teaches that a "reconstructed residual error 560 is … added to the combined weighted prediction macroblock to form reconstructed macroblock 570." *Id.*; *see also* ¶111 ("Performing inverse quantization, step 875, and performing the Inverse Transform, step 880, (such as, for example, an inverse DCT or an inverse wavelet transform) results in the decoded residual error, which is added, step 885, to the combined weighted prediction and the preprocessor modified offset (offset modification is discussed below) to form an output picture."). This teaches reconstructing the block of pixels based on the combined prediction.

3. Dependent Claims 20, 26, and 32

20. The method according to claim 19,

26. The apparatus according to claim 25,

32. The computer program product according to claim 21,

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

101. Walker teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 20, 26, and 32.

102. First, Walker teaches **an instance in which said first motion vector points to a subpixel** (e.g., an "interpolated pixel"). Walker teaches an encoder using a motion vector to point to either a "pixel or interpolated pixel." Ex-1004, ¶111 ("An encoder can perform pixel interpolation to locate the best matching reference macroblock (or any size section) and point to the pixel or interpolated pixel with a motion vector."), Fig. 8. A "decoder uses the motion vectors to perform motion compensation region location, step 840, to locate the best matching regions among the interpolated pixels." Ex-1004, ¶111, Fig. 8. As further explained below, Walker teaches that these interpolated pixels are subpixels positioned between integer pixels, e.g., half pixels, quarter pixels, or eighth pixels. *See e.g.*, Ex-1004, ¶114, Fig. 9. These subpixels are interpolated from integer

pixels, which is why Walker refers to them as interpolated pixels. I note that integer pixels are not interpolated. *Supra* §I.C. Thus, Walker's teachings encompass the motion vector pointing to a subpixel.

103. Second, Walker teaches that in an instance in which said first motion vector points to a subpixel, **said first prediction is obtained by interpolation using pixel values of said first reference block**. Walker teaches that "[p]ixel interpolation can be used to improve the performance of motion compensated predictive coding." Ex-1004, ¶114, ¶111 ("Pixel interpolation, step 835, is used to achieve better matching reference regions for motion compensation.").

104. Walker teaches obtaining a prediction for a subpixel by interpolation (e.g., half pixel, quarter pixel, eighth pixel). Ex-1004, ¶114 ("FIG. 9 is an illustration of an example of half-pixel interpolation for use in motion compensation. The example shown is half pixel interpolation where one interpolated pixel is located between each of the original integer pixels. Integer pixels 910 are depicted as circles labeled upper case 'A' to 'I' and the interpolated or half-pixels 920 are depicted as squares labeled lower case 'a' to 'o'. … Other orders of pixel interpolation are supported by various standards. H.264 supports quarter pixel interpolation as well as eighth pixel interpolation. Those of ordinary skill in the art would understand these other pixel interpolation methods and they are not discussed in greater detail herein."), Fig. 9.

105. The half pixel, quarter pixel, and eighth pixel taught by Walker are "subpixels," which refer to fractions of pixels located between full pixels. Ex-1004, ¶114 ("Integer pixels 910 are depicted as circles labeled upper case 'A' to 'I' and the interpolated or half-pixels 920 are depicted as squares labeled lower case 'a' to 'o'."), Fig. 9. I note this is consistent with the use of the term "subpixels" in the '267 patent, which likewise refers to a fraction of a pixel or a pixel position that is between two full pixels. *Compare* Ex-1004, Fig. 9 *with* Ex-1001, 12:56-63, Fig. 12.

<u>Walker</u>

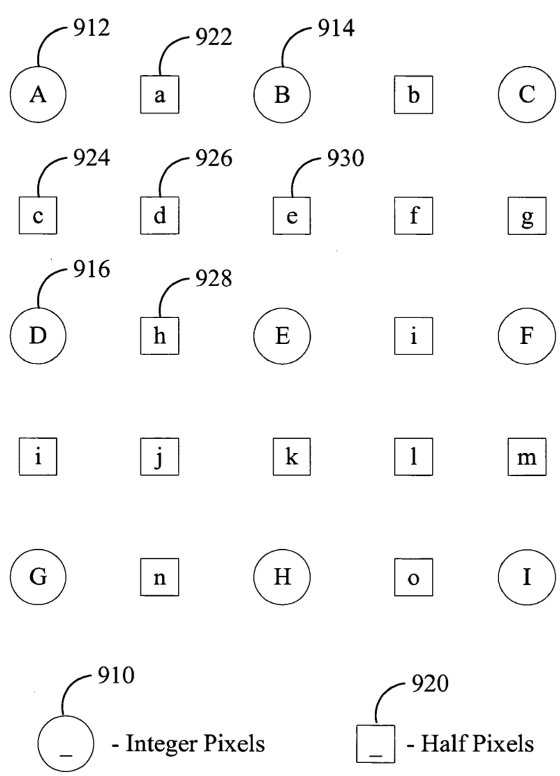


Figure 9

<u>'267 patent</u>

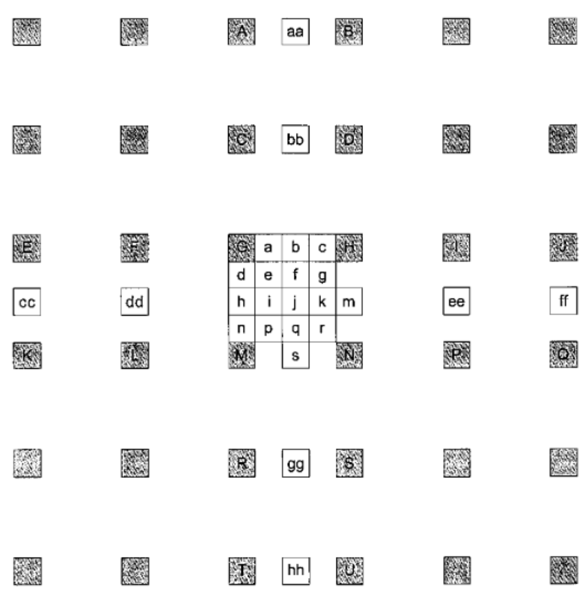


Fig. 12

106. Walker teaches an example of performing interpolation with a 2-tap FIR filter using pixel values of integer pixels neighboring a half pixel. Ex-1004, ¶114 ("Half pixel interpolation can be carried out with a bilinear filter such as, for example, a 2-tap FIR filter with weights [0.5 0.5]. For example, interpolated pixel 922 can be calculated as the average of integer pixel 912 and integer pixel 914, interpolated pixel 924 can be the average of integer pixel 912 and integer pixel 916, and interpolated pixel 926 can be the average of two interpolated pixels (for example, 922 and 928 or 924 and 930)."), Fig. 9. As indicated in Walker's figures, the integer pixels are the pixels in the reference block. *Id.*, Fig. 9. They are the input to the FIR filter that is used for interpolating sub-pixels, including half pixels. Thus, Walker teaches obtaining said first prediction by interpolation using pixel values of said first reference block.

107. In addition, as explained for claim 19, Walker teaches using the first motion vector and the first prediction in a bi-directional weighted motion prediction process (*supra* §V.A.2). Walker further teaches "improv[ing] the performance of motion compensated predictive coding" using sub-pixel interpolation. Ex-1004, ¶114. Walker teaches that the interpolation results are used in determining a weighted prediction for motion prediction. Ex-1004, ¶111 ("The luma and chroma values of the two best matching prediction regions, pred0 and

pred1, output at step 840, are multiplied, steps 845 and 850, by modified weights w_A and w_B respectively[.]"), Fig. 8 (showing "Pixel Interpolation" step 835 used for generating sample values pred0 and pred1, which are used to calculate the first and second predictions ((pred0)w_A)>>6, ((pred1)w_B)>>6):

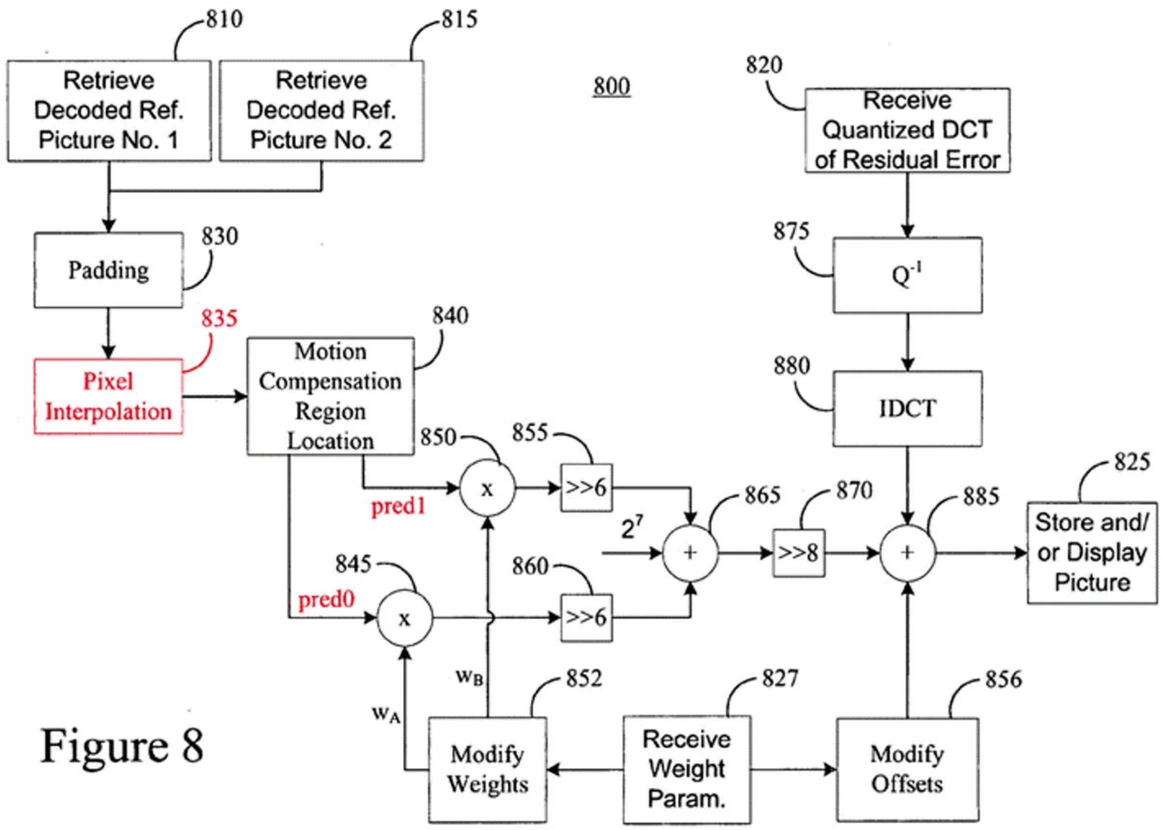


Figure 8

This teaches the instance where the first motion vector points to a subpixel and the first prediction is obtained by interpolation.[7]

[7] Moreover, the '267 patent admits that the limitations of claims 20, 26, and 32 were known in the prior art. Ex-1001, 2:60-3:11. The '267 patent states that, in the "Background" of the patent, including for MPEG-2 and H.264, there were instances where a motion vector points to a subpixel, e.g., "fractional-pixel positions" (Ex-1001, 2:60-65), and in such an instance, the prediction was obtained

4. Dependent Claims 21, 27, and 33

21. The method according to claim 20, **27. The apparatus according to claim 26,** **33. The computer program product according to claim 32,** **wherein said first prediction is obtained by interpolation using values of said first reference block by: right shifting a sum of a P-tap filter using values of said first reference block.**

108. Walker teaches the method according to claim 20, the apparatus according to claim 26, and the computer program product according to claim 32. *Supra* §V.A.3. Walker further teaches that said first prediction is obtained by interpolation using values of said first reference block. *Id*. As explained below, it is my opinion that Walker further teaches the additional limitation of claims 21, 27, and 33.

---

by interpolation using pixel values of the reference block, e.g., neighboring samples at full-pixel locations. Ex-1001, 2:65-3:11 ("In order to obtain samples at fractional-pixel locations, interpolation filters may be used in the MCP [Motion Compensated Prediction] process. Conventional video coding standards describe how a decoder can obtain samples at fractional-pixel accuracy by defining an interpolation filter. In MPEG-2, for example, motion vectors can have at most, half-pixel accuracy, where the samples at half-pixel locations are obtained by a simple averaging of neighboring samples at full-pixel locations. The H.264/AVC video coding standard supports motion vectors with up to quarter-pixel accuracy. Furthermore, in the H.264/AVC video coding standard, half-pixel samples are obtained through the use of symmetric and separable 6-tap filters, while quarter-pixel samples are obtained by averaging the nearest half or full-pixel samples.").

109. Walker teaches performing interpolation for a half pixel using a “2-tap FIR filter,” which computes the average of two pixel values with equal weights. Ex-1004, ¶114, Fig. 9. Walker’s 2-tap FIR filter is a P-tap filter, where P is 2.[8] And because Walker teaches performing the interpolation using values of said first reference block (*supra* §V.A.3), it teaches a P-tap filter using values of said first reference block. As was known in the art, the popular H.264 standard used 6-tap filters for interpolation, where P is 6.[9] *See* Ex-1004, ¶114.

110. Walker teaches one mathematical implementation where this interpolation uses equal filter weights of 0.5 and 0.5 to average nearby integer pixels. Ex-1004, ¶114 (“Half pixel interpolation can be carried out with a bilinear filter such as, for example, a 2-tap FIR filter with weights [0.5 0.5]. For example, interpolated pixel 922 can be calculated as the average of integer pixel 912 and integer pixel 914, interpolated pixel 924 can be the average of integer pixel 912 and integer pixel 916, and interpolated pixel 926 can be the average of two interpolated pixels (for example, 922 and 928 or 924 and 930).”), Fig. 9. This

---

[8] A POSITA would have understood that the term “P-tap filter” was commonly used to describe a filter with multiple taps, where P is a variable that can take an integer value. *See, e.g.*, Ex-1001, 12:60-63 (“a P-tap filter such as a six-tap filter”), 16:25-29.

[9] This was admitted by the ’267 patent. Ex-1001, 2:60-3:11 (“Furthermore, in the H.264/AVC video coding standard, half-pixel samples are obtained through the use of symmetric and separable 6-tap filters, while quarter-pixel samples are obtained by averaging the nearest half or full-pixel samples.”).

performs the averaging function using weights of 0.5 to effectively divide values by two (multiply by 1/2) when averaging them. *See id.*

111. Walker further teaches an optimization where bit-shifting to the right is used instead of division. Ex-1004, ¶37 ("Another way of normalizing without a division operation is by use of bit shifting. The weights can be derived with a common denominator and the division can be represented by a right shift of the combined weighted prediction a number of bits based on the base 2 logarithm of the denominator. For example, w1 could be equal to 12 and w2 could be equal to 4 and the denominator could be 16. The denominator of 16 would translate to a right shift of 4 bits. A right shift of 4 bits is equivalent to dividing by 16, thus w1 would translate to a normalized weight of 0.75 and w2 would translate to a normalized weight of 0.25."), ¶46. When applied to Walker's interpolation teachings (Ex-1004, ¶114), it would have been obvious to avoid multiplication and division by using an FIR filter with equal weights [1 1] and right-shifting the results by 1 bit (e.g., dividing the sum by 2). This calculates the average of the two pixels as Walker teaches (Ex-1004, ¶114) while reducing computational complexity and avoiding multiplication and division as Walker also teaches (Ex-1004, ¶37).

112. A POSITA would have been motivated to apply Walker's teachings in this manner because, in computing, multiplication and division were known to be far more computationally complex than addition and bit-shifting, which is why

calculations were often optimized in the art by bit-shifting rather than multiplying or dividing values. *See id.*

5. Dependent Claims 22, 28, and 34

22. The method according to claim 20,

28. The apparatus according to claim 26,

34. The computer program product according to claim 32,

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

113. Walker teaches the method according to claim 20, the apparatus according to claim 26, and the computer program product according to claim 32. *Supra* §V.A.3. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 22, 28, and 34.

114. Walker teaches **instances in which said second motion vector points to an integer sample**. As explained above, Walker teaches that the motion vector may point to an integer pixel or a subpixel and therefore Walker's teachings encompass instances in which said second motion vector points to an integer sample. *Supra* §V.A.3; Ex-1004, ¶111, ¶114, Figs. 8-9. The use of integer pixels was well-known in the art.[10]

---

[10] The '267 patent admits that integer ("full") pixels were known in the art. Ex-1001, 2:60-3:11 ("The motion vectors are not limited to having full-pixel accuracy, but could have fractional-pixel accuracy as well.").

115. Walker further teaches, in an instance in which said second motion vector points to an integer sample, **said second prediction is obtained by shifting values of said second reference block to the left**. For both integer and sub-pixels, Walker teaches obtaining the second prediction (e.g., "((pred1)w_B)>>6 ") as the product of a sample value (e.g., pred1) from a second reference block (e.g., from a future frame) multiplied by a weight (e.g., "w_B"). *Supra* §V.A.2[19d]/[25d]/[31d]. This calculation is applied to all the pixels of the block. Ex-1004, ¶35, ¶111; *supra* §V.A.2[19d]/[25d]/[31d]. Walker teaches examples where the weight takes a value that is a power of 2 (e.g., 4). Ex-1004, ¶37 ("w2 could be equal to 4"). Walker further teaches that multiplication and bit-shifting are equivalent mathematical operations. Ex-1004, ¶46 ("Those of ordinary skill in the art would understand that the bit shifting operations could be accomplished by other methods such as, for example, applying a scaling factor through multiplication or division."). Therefore, it would have been obvious to implement Walker's teachings, where the sample pixel value from the reference block is multiplied by 4, by left-shifting the pixel value to the left by 2 bits. A POSITA would have found it obvious and been motivated to do so because, in computing terms, left-shifting is a simpler operation than multiplication and can be performed without the need for a multiplication unit.

116. In fact, left-shifting to multiply a value by a coefficient and increase the precision of intermediate values in calculations was well known in the art. For example, in U.S. Patent Application Publication No. 2008/0198935 ("Srinivasan-935") (Ex-1013), "the input is pre multiplied by 8 (i.e. left shifted by 3 bits)" "[f]or the sake of improved coding performance by the reduction of rounding errors[.]" Ex-1013, ¶91; *see also Id.*, ¶116, ¶124, ¶129 ("[T]o minimize the damage of truncation errors and thus maximize transform performance, input data to a transform needs to be left shifted several bits."), ¶131 ("One way to reduce the damage of truncation errors is to left shift the input data[.]"). As another example, U.S. Patent Application Publication No. 2013/0034158 ("Kirchhoffer-158") (Ex-1015) discusses increasing the bit-depth representation of a value by left-shifting the value with a predetermined number of bits. Ex-1015, ¶84 ("A precision may also be called a bit-depth representation, wherein a higher precision corresponds to a higher bit-depth representation. A bit-depth representation of a value may be increased by left-shifting the value with a predetermined number of bits. A left-shift corresponds to a multiplication with 2."); *see also Id.*, ¶85 ("[A]n increase in the precision of the reconstructed reference image samples[] may be obtained by left-shifting each value of these reconstructed reference image samples[.]"), ¶67, ¶103. The explicit teachings of Srinivasan-935 and Kirchhoffer-158 further

confirm that it would have been obvious to perform left-shifting in calculating the weighted prediction.

6. Dependent Claims 23, 29, and 35

23. The method according to claim 19, wherein said decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprises: **29. The apparatus according to claim 25, wherein the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, by:** **35. The computer program product according to claim 31, wherein the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right, further comprise program code instructions configured to:** **[inserting/insert] a rounding offset to the combined prediction before said decreasing.**

117. Walker teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 23, 29, and 35.

118. Walker teaches **inserting a rounding offset** (e.g., 2^7) **to the combined prediction** as part of Operation No. 2 **before said decreasing** of the precision in Operation No. 3. As explained above, Walker teaches obtaining a combined prediction via its disclosure of calculating a sum by adding up the first

prediction (e.g., "((pred0)w_A)>>6"), the second prediction (e.g., "((pred1)w_B)>>6"), and the rounding offset (e.g., 2^7) in Operation No. 2 of Table 4. *Supra* §V.A.2[19e]/[25e]/[31e]; Ex-1004, ¶¶92-93, Table 4. Therefore, Walker inserts a rounding offset to the combined prediction before its next step, in Operation No. 3 of Table 4, where Walker decreases a precision of said combined prediction by shifting bits of the combined prediction to the right. *Supra* §V.A.2[19f]/[25f]/[31f]; Ex-1004, ¶¶92-93:

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

119. Walker explains that 2^7 is a **rounding offset**. Ex-1004, ¶111 ("After applying the weights, both weighted regions are right bit-shifted 6 bits, steps 855 and 860, and added, step 865, to the rounding factor [2^7] and then

right bit shifted 8 bits, step 870, to form the combined weighted prediction.”). A POSITA would have understood that rounding factor 2^7 is a rounding offset as the term “rounding offset” was often used interchangeably with “rounding adjustment.” Moreover, the rounding factor 2^7 is inserted into the combined prediction before a right-shift operation, which causes values to round to the closest integer after the right shift, rather than always rounding down which would be the result absent the offset. In light of this effect on the rounding operation, a POSITA would have understood that the rounding adjustment 2^7 is a rounding offset.

120. Walker teaches this rounding offset as part of its rounding process, which decreases the precision as recited by the claims. Ex-1004, ¶¶92-93, ¶111. Additionally, it was obvious for said decreasing to include the rounding offset (claim 23) because the insertion of the rounding offset is performed right before and in conjunction with the right-shifting to affect the direction of the rounding. Walker includes a rounding offset to control rounding error resulting from the right-shift operation that decreases precision. This was common in the art. *Supra* §I.D.

121. Regarding claim 29, as explained above, Walker teaches at least one memory and computer program code are configured to cause the apparatus to perform the operations recited in limitations [25e]-[25f]. *Supra* §V.A.2[25a]. Since

Walker applies its teachings to a computer implementation, Walker also applies the above-described teachings, for decreasing said precision of said combined prediction by shifting bits of the combined prediction to the right, to a computer implementation. Therefore, it would have been obvious that **the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right by** inserting a rounding offset to the combined prediction before said decreasing (claim 19).

122. Likewise, for claim 35, as explained above, Walker applies its teachings to a computer implementation, including for program code instructions that are configured to perform the operations recited in claim 31, including limitations [31e]-[31f]. *Supra* §V.A.2[31a]. Therefore, for reasons explained above, Walker teaches that **the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right further comprise program code instructions configured to** inserting a rounding offset to the combined prediction before said decreasing (claim 35).

7. Dependent Claims 24, 30, and 36

24. The method according to claim 19,

30. The apparatus according to claim 25,

36. The computer program product according to claim 31,

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

123. Walker teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §V.A.2. As explained below, it is my opinion that Walker further teaches the additional limitations of claims 24, 30, and 36.

124. As explained above, Walker teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision because 8 bits are needed to represent the possible pixel values of these blocks. *Supra* §V.A.2[19b]/[25b]/[31b]. Here, **the first precision indicates a number of bits needed to represent the values of the pixels**.

125. As explained above, Walker teaches said first prediction having a second precision, which is higher than said first precision, and said second prediction having the second precision because 17 bits are needed to represent the possible values of the first prediction and the second prediction. *Supra* §§V.A.2[19c-19d]/[25c-25d]/[31c-31d]. Here, **the second precision indicates the**

number of bits needed to represent values of said first prediction and values of said second prediction.

B. Ground 2: Claims 19-36 are Rendered Obvious in View of Karczewicz-I and Karczewicz-II

126. It is my opinion that a POSITA would have found Claims 19-36 obvious based on the combination of Karczewicz-I and Karczewicz-II.

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

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

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

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

130. Karczewicz-I is in the same field of endeavor as the ’267 patent (video encoding/decoding). *See supra* §III. Karczewicz-I is directed to “video encoding and decoding techniques applicable to bi-directional prediction.” Ex-1005, ¶8. Similar to the ’267 patent, Karczewicz-I is directed to motion prediction techniques. *See, e.g.*, Ex-1005, ¶89. Karczewicz-I teaches using the same known standards (e.g., H.264) as the ’267 patent for video encoding and decoding. *Compare* Ex-1005, ¶¶35-37 *with* Ex-1001, 2:60-3:11, 9:26-41.

2. U.S. Patent Application Publication No. 2009/0257499 Karczewicz (“Karczewicz-II”)

131. I have reviewed the Karczewicz-II reference. I understand that Karczewicz-II is cited on the face of the ’267 patent. Karczewicz-II was cited in an Information Disclosure Statement filed June 7, 2021 along with 76 other references. Ex-1007, 000074-78. Other than acknowledging the Information Disclosure Statement, the Examiner did not discuss Karczewicz-II during the prosecution history or use Karczewicz-II in any rejections.

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

133. Karczewicz-II teaches block-based techniques for motion prediction. *E.g.*, Ex-1006, ¶8, ¶¶35-36, ¶46, ¶54. Karczewicz-II teaches improved calculations

for predictions involving interpolated fractional pixel positions. Ex-1006, ¶2, ¶10, ¶¶93-106. Karczewicz-II teaches that prediction values for quarter-pixel positions can be calculated as an average of two adjacent integer or half-pixel positions. Ex-1006, ¶¶96-102. Karczewicz-II teaches such calculation for multiple different scenarios. For example, Karczewicz-II teaches a quarter-pixel position between an integer pixel and a half-pixel position is calculated as an average of the predictions of the integer pixel and the half-pixel position. Ex-1006, ¶96, ¶99, ¶103, Tables 1, 3, 5. In addition, Karczewicz-II teaches a quarter-pixel position between a center-pixel position (i.e., the position that is at the center of four integer pixels) and a half-pixel position is calculated as an average of predictions for the two positions. Ex-1006, ¶96, ¶99, ¶108, Tables 1, 3, 8. Moreover, Karczewicz-II teaches a quarter-pixel position between two half-pixel positions is calculated as an average of predictions for the two positions. Ex-1006, ¶97, ¶101, ¶103, Tables 2, 4, 6.

134. Karczewicz-II teaches an improved method for averaging interpolated pixel values. Specifically, Karczewicz-II discloses maintaining higher precision for intermediate values (e.g., integer or half-pixel predictions) during calculation and delaying rounding until later in the process in order to reduce rounding inaccuracies. *See, e.g.*, Ex-1006, ¶10, ¶39, ¶53, ¶59, ¶¶99-106.

135. Karczewicz-II is in the same field of endeavor as the ’267 patent (video encoding/decoding). *See supra* §III. Karczewicz-II is directed to “digital

video coding" and "interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex-1006, ¶2, Abstract. Karczewicz-II uses and improves the same known standards (e.g., H.264) as the '267 patent for video encoding and decoding. *Compare* Ex-1006, ¶¶46-47, ¶82, ¶88, ¶¶93-0106 *with* Ex-1001, 2:60-3:11, 9:26-41.

3. Motivation to Combine and Reasonable Expectation of Success

136. Karczewicz-I and Karczewicz-II are Qualcomm patent applications by the same inventors Marta Karczewicz, Peisong Chen, and Yan Ye. Both are directed to video coding and apply their teachings to similar architectures. Ex-1005, ¶2, Fig. 1; Ex-1006, ¶2, Fig. 1

Karczewicz-I **Karczewicz-II**

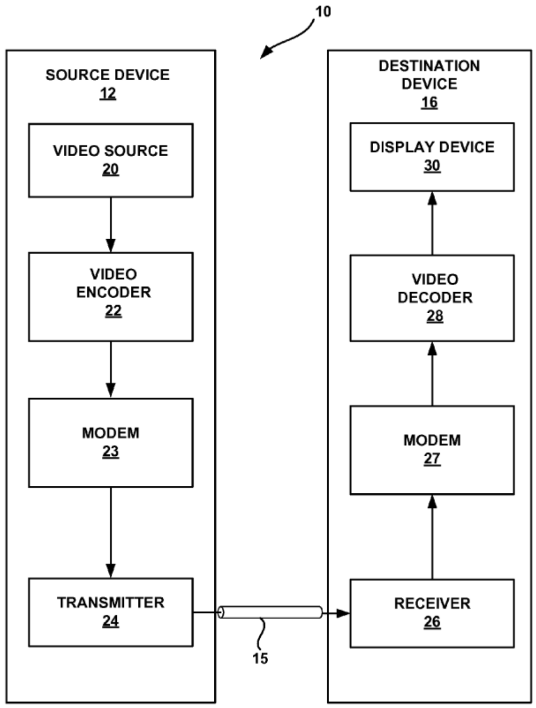


FIG. 1

Ex-1005, Fig. 1, ¶¶29-50.

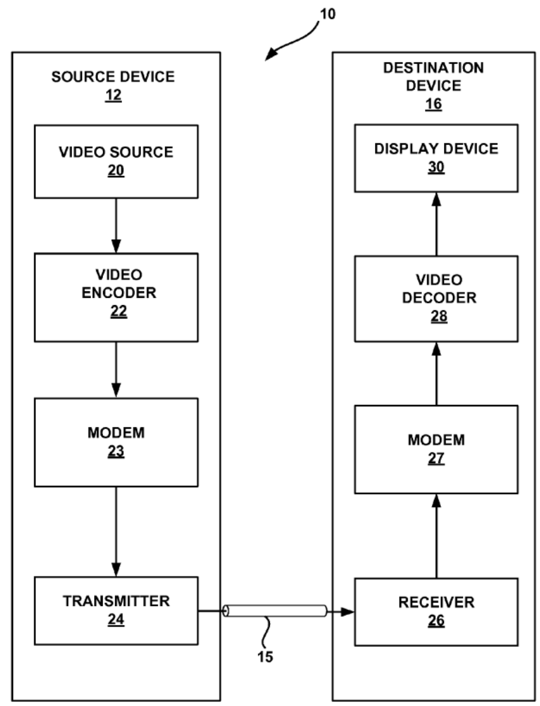


FIG. 1

Ex-1006, Fig. 1, ¶¶40-53.

137. Karczewicz-I and Karczewicz-II are both directed to block-based, e.g., H.264, techniques for motion prediction. Ex-1005, ¶35, ¶44, ¶¶55-60; Ex-1006, ¶8, ¶¶35-36, ¶46, ¶54. Both include teachings for a video decoder that performs motion compensation for block-based decoding:

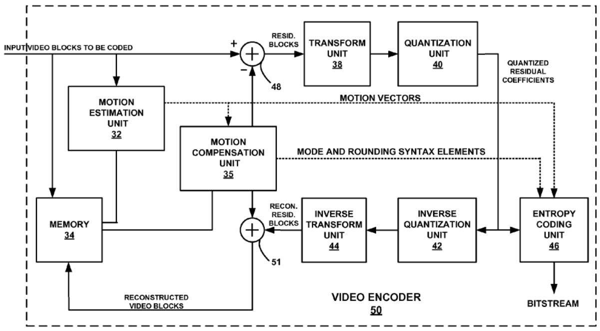


FIG. 2

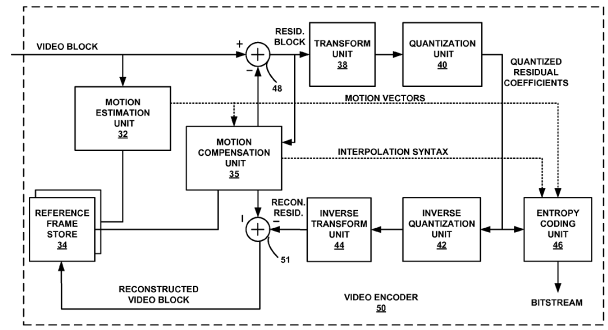


FIG. 2

Ex-1005, Fig. 4, ¶89 ("Prediction unit 75 invokes motion compensation unit 86 for block based predictive decoding.").

Ex-1006, Fig. 3, ¶63 ("Video decoder 60 includes a motion compensation unit 55 that performs the interpolation techniques of this disclosure for decoding.").

138. Since Karczewicz-I and Karczewicz-II are from the same inventors of the same company and are directed to performing video coding using the same architecture, a POSITA would have found it obvious the combine the teachings of Karczewicz-I and Karczewicz-II, including those teachings described above. *Supra* §V.B.1 (Karczewicz-I teachings), §V.B.2 (Karczewicz-II teachings). The similarities of Karczewicz-I and Karczewicz-II's architecture would have suggested to a POSITA to implement techniques taught by Karczewicz-I and techniques taught by Karczewicz-II using that common architecture.

139. This would have been a combination of prior art elements according to known methods. Karczewicz-I teaches bi-prediction techniques for the H.264 standard with two motion vectors pointing to two blocks of pixels that are averaged together. Ex-1005, ¶35, ¶44, ¶60. For interpolated sub-pixels, this calculation averages the interpolated values together. Beyond integer pixels, Karczewicz-II teaches that, for H.264, motion vectors can also point to fractional sub-pixels (Ex-1006, ¶¶56-58, ¶¶93-102),[11] and Karczewicz-II teaches an improved calculation for averaging interpolated pixel values, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate calculations (E.g., Ex-1006, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-106). Therefore, Karczewicz-II provides complementary teachings that improve Karczewicz-I's teachings for averaging predicted pixel values.

140. As Karczewicz-II explains, its use of higher precision for intermediate steps eliminates the propagation of rounding inaccuracies and improves the accuracy of the average. *See, e.g.*, Ex-1006, ¶102 ("By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate."), ¶10

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

(“In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values.”), ¶39, ¶53, ¶59. Therefore, a POSITA who understood fundamental computer logic concepts (e.g., binary arithmetic, bit shifting, precision control, rounding and offsetting, and error analysis) would have been motivated to apply Karczewicz-II’s improved calculations to Karczewicz-I’s teachings.

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

improved implementation of the mathematical calculations taught by Karczewicz-I, a POSITA would have been motivated to use the known technique of Karczewicz-II to improve similar devices/methods, as taught by Karczewicz-I, in the same way, using higher precision when combining interpolated pixel values, as Karczewicz-II teaches. This would have achieved the same benefits taught by Karczewicz-II, e.g., eliminating the propagation of rounding inaccuracies and improving prediction accuracy. *E.g.*, Ex-1006, ¶102, ¶10. And it would have had predictable results because it applies teachings from Karczewicz-II to corresponding mathematical calculations used for Karczewicz-I, using similar video decoding architectures as explained above. A POSITA would have recognized the applicability of Karczewicz-II to the corresponding calculations in Karczewicz-I, which was a simple matter given the level of ordinary skill. Notably, a POSITA would have understood fundamental computer logic concepts (e.g., binary arithmetic, bit shifting, precision control, rounding and offsetting, and error analysis) and mathematical calculations for known motion estimation and compensation techniques (e.g., bi-directional prediction, determination and use of motion vectors, interpolation for fractional pixels) because they are integral to working with video codecs.

142. <u>Karczewicz-I and Karczewicz-II provide complementary teachings</u>. Karczewicz-I is directed to weighted bi-directional prediction that averages two

prediction values together. *See, e.g.*, Ex-1005, ¶8. A weighted bi-directional prediction is determined based on two reference blocks pointed to by two motion vectors. Ex-1005, ¶53, ¶85. Karczewicz-I teaches weighted bi-directional prediction that averages two predictions based on two reference blocks (e.g., pred0(i,j), pred1(i,j)) with an offset (e.g., +1), including a default mode (Ex-1005, ¶55) with equal weights that performs a simple average. Ex-1005, ¶60:

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

143. Karczewicz-I's discussion focuses on methods for combining two predictions, relying on known techniques, such as the techniques under the H.264 standard, for determining the two predictions to be combined. *See, e.g.*, Ex-1005, ¶35 ("Video encoder 22 and video decoder 28 may operate according to a video compression standard, such as the ITU-T H.264 standard[.]"). The H.264 standard provides that, when a motion vector points to a fractional pixel/subpixel, interpolation is used to calculate a prediction. *Supra* §I.C; Ex-1006, ¶¶68-69, ¶74, ¶93. Consistent with H.264, Karczewicz-I teaches that the inter-predictive coding process, where motion vectors point to different frames, includes interpolation, and therefore provides express teaching, suggestion, and motivation ("TSM") to

combine with known teachings for interpolation. Ex-1005, ¶41 ("Following inter-based predictive encoding (which includes interpolation and the techniques of this disclosure to efficiently select a prediction algorithm or mode by which to predict a coded unit) …").

144. Karczewicz-II teaches improved calculations for predictions involving interpolated fractional pixel positions in H.264, where rounding is delayed until later in the process, thereby maintaining higher precision for intermediate values. *See, e.g.*, Ex-1006, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."), ¶99, ¶10. Karczewicz-II explains that, according to the H.264 standard, predictions for half-pixel positions are calculated using 6-tap interpolation filters that interpolate the sub-pixel based on nearby pixels in the same row (e.g., half-pixel "b") or column (e.g., half-pixel "h"). Ex-1006, ¶74, ¶¶93-94, Fig. 4A-4B:

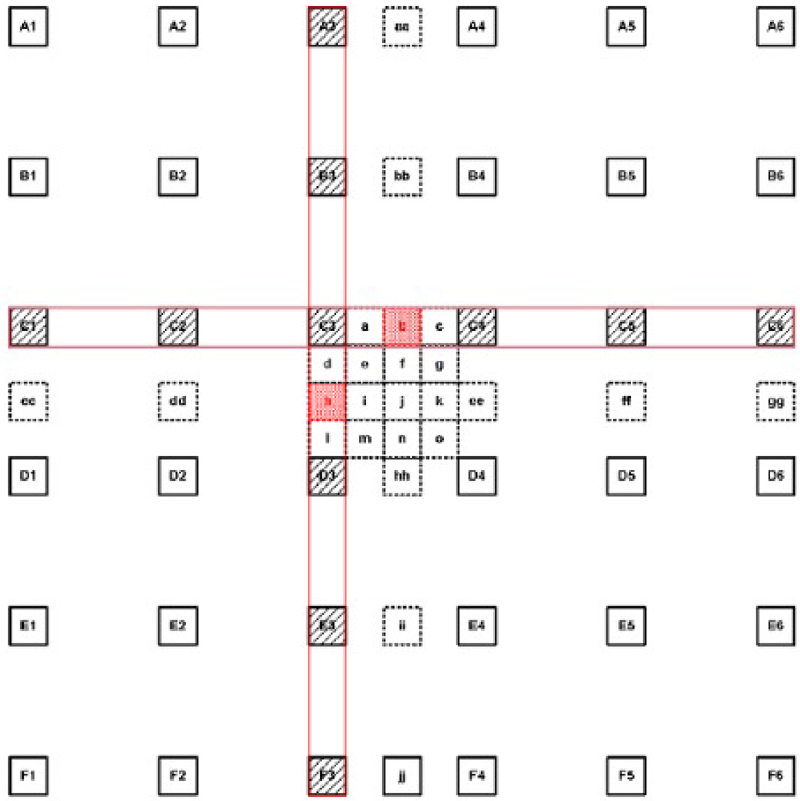


FIG. 4B

145. This process involves two steps. First, a 6-tap filter multiplies the six pixels in the row (or column) by the filter values (1, −5, 20, 20, −5, 1), adds the products together to produce a non-rounded prediction (e.g., b1). Second, the result is rounded down, e.g., using a right-shift operation (">>") that decreases the

number of bits needed to represent the prediction back down to the original precision.[12] *Supra* §I.D. Here is the process for interpolating sub-pixel "b":

b1=C1−5*C2+20*C3+20*C4−5*C5+C6

…

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

Ex-1006, ¶¶93-94, Fig. 4B. This produces a rounded prediction (e.g., b).

146. For half-pixel positions that are in the center of four integer pixels on two dimensions (e.g., j) (referred to as a "center-pixel position" hereinafter), the interpolation process is applied in two rounds: one to interpolate a middle row of half pixels, and another to interpolate that middle row into the center pixel. In other words, the 6-tap interpolation filter is applied on values of six half-pixel predictions. Ex-1006, ¶95, Fig. 4C.

147. Karczewicz-II teaches calculating quarter-pixel predictions by averaging the predictions of the two nearest integer or half-pixel positions as explained above. Ex-1006, ¶¶96-97. Karczewicz-II teaches an improvement to reduce coding inefficiencies and increase the accuracy of pixel calculations by "keep[ing] the highest possible precision through the intermediate steps" and

---

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

avoiding "any shifting, rounding and clipping operations[] until the very last step of the interpolation process." Ex-1006, ¶99. This prevents the propagation of rounding errors. Ex-1006, ¶102 ("By preserving the full precision of the intermediate values, the interpolated sub-pixels will be more accurate."), ¶10 ("In addition, this disclosure also recognizes coding inefficiencies due to conventional rounding of half-pixel values, and provides techniques that may improve interpolation by reducing or eliminating intermediate rounding. … Quarter-pixel values, however, which may be generated based on one or more of the interpolated half-pixel values, may rely on non-rounded versions of the half-pixel values. This can eliminate propagation of rounding inaccuracies from the half-pixel values to the quarter-pixel values."), ¶39, ¶53, ¶59.

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

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

$$\text{pred}(i,j)=(\text{pred0}(i,j)+\text{pred1}(i,j)+1)>>1$$

Ex-1005, ¶¶58-60. As explained above, H.264 allows motion vectors to point to integer pixels or fractional pixels, such as half pixels, and uses interpolation to calculate predictions for fractional pixels. *See e.g.*, Ex-1005, ¶41; Ex-1006 ¶¶93-102 ("A sub-pixel motion vector refers to a sub-pixel position in a reference picture which needs to be interpolated."), ¶¶56-58; *supra* §I.C.[13] Therefore, the predictions pred0(i,j) and pred1(i,j) encompass scenarios that include integer pixel prediction, a half-pixel prediction, or a center-pixel prediction, and Karczewicz-I calculates an average of these pixel values.

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

151. <u>Scenario 1</u> (with the first motion vector pointing to a half-pixel position and the second motion vector pointing to an integer pixel position). Beyond integer pixels, Karczewicz-II explains how, for H.264, motion vectors can also point to half-pixel positions. *E.g.*, Ex-1006, ¶¶93-102, ¶¶56-58. When the two motion vectors in Karczewicz-I's bi-directional prediction point to a half-pixel

---

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

position and an integer pixel position, the default weighted prediction is calculated as an average of the half-pixel and the integer pixel. *See* Ex-1005, ¶60, ¶55.

152. Karczewicz-II teaches an improved calculation for averaging a half-pixel value with an integer pixel value. While Karczewicz-II uses this calculation to interpolate a quarter-pixel position, it is the same calculation as Scenario 1 because it averages an integer and half pixel. *See* Ex-1006, ¶96, Fig. 4D. Karczewicz-II explains the conventional calculation for averaging two numbers: adding the half-pixel value (e.g., “b”) with the integer pixel value (e.g., C3) and a rounding offset 1, then dividing by 2 (using a right-shift “>>” operation that is mathematically equivalent to dividing by 2). Ex-1006, Fig. 4D, Table 1:

TABLE 1

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

153. Karczewicz-II improves this conventional approach by “keep[ing] the highest possible precision through the intermediate steps[.]” Ex-1006, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex-1006, 99. The integer pixel value (e.g., C3) is multiplied by 32 (by bit-shifting to the left 5 bits, which is

mathematically equivalent), taking its precision from 8 to 13 bits. Ex-1006, Table 5. Instead of using a rounded 8-bit half-pixel (e.g., b), Karczewicz-II delays the rounding step and instead uses a *non-rounded* half-pixel prediction (e.g., b1) that is 15 bits. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (">>6"). *Id.*, Table 3:

TABLE 3

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

154. The operations for this improved approach are shown in Table 5 of Karczewicz-II. Ex-1006, ¶103 ("The following Tables show the interpolation process for other sub-pixels in sixteen bit storage elements. In the Tables below, the operations defined in each column are performed sequentially through the respective table."), Table 5:

TABLE 5

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

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

155. Since Karczewicz-II teaches this optimization for averaging an interpolated half-pixel and an integer pixel, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of an integer pixel and an interpolated half-pixel in Scenario 1. For example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex-1005, ¶60):

Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

156. For Scenario 1, the two predictions would be a half-pixel prediction (e.g., pred0(i,j)) and an integer pixel prediction (e.g., pred1(i,j)). *See* Ex-1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel prediction (pred0(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied (Ex-1006, ¶¶93-94). To match this higher precision, the integer pixel (pred1(i,j)) is left-shifted, as Karczewicz-II teaches. Ex-1006, ¶99, Table 3. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶96, ¶99, Tables 1 and 3; Ex-1005, ¶60.

157. <u>Scenario 2</u> (with the first motion vector pointing to a center-pixel position and the second motion vector pointing to a half-pixel position). Karczewicz-II explains how, for H.264, motion vectors point to center- and half-pixel positions. *E.g.*, Ex-1006, ¶¶93-95. When the two motion vectors in Karczewicz-I's bi-directional prediction point to a center- and half-pixel position,

the default weighted prediction is calculated as an average of the center- and half-pixels. *See* Ex-1005, ¶60, ¶55.

158. Karczewicz-II teaches an improved calculation for averaging a center- and half-pixel. Ex-1006, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 2 because it averages a center and half pixel. As discussed above, Karczewicz-II explains the conventional calculation for averaging two numbers: adding the center-pixel value (e.g., “j”) with the half-pixel value (e.g., “b”) and a rounding offset 1, then dividing by 2 (using a right-shift “>>” operation that is mathematically equivalent to dividing by 2). Ex-1006, ¶96, Fig. 4D, Table 1:

TABLE 1

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

159. Karczewicz-II improves this conventional approach by “keep[ing] the highest possible precision through the intermediate steps[.]” Ex-1006, ¶99. Karczewicz-II replaces the equations in Table 1 with those in Table 3, where the pixel values are combined at a higher precision. Ex-1006, ¶99. Whereas Table 1 used a fully-rounded center-pixel (e.g., “j”), Table 3 uses a partially-rounded pixel

(e.g., “j1>>5”) that remains 5 bits longer than the rounded value in Table 1.[14] *Infra* §V.B.6. Likewise, Table 3 replaces the rounded half-pixel (e.g., “b”) with a non-rounded half-pixel (e.g., “b1”) that is 15 bits. These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (“>>6”). Ex-1006, Table 3:

TABLE 3

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

160. The operations for this improved approach are shown in Table 8 of Karczewicz-II. Ex-1006, ¶105 (“Table 8, below demonstrates steps that can be taken for sixteen-bit implementation of interpolating {f,i,k,n}, which are the positions that use to interpolate the intermediate value ‘j1.’”), Table 8:

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

TABLE 8

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

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

161. Since Karczewicz-II teaches this optimization for averaging interpolated center- and half-pixels, a POSITA would have been motivated to apply that improved calculation to Karczewicz-I, which likewise calculates the average of an interpolated center pixel and an interpolated half-pixel in Scenario 2. For example, Karczewicz-I calculates the bi-directional prediction (e.g., pred(i,j)) as an average of two predictions (Ex-1005, ¶60):

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

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

162. For Scenario 2, the two predictions would be a center prediction (e.g., pred0(i,j)) and half-pixel prediction (e.g., pred1(i,j)). *See* Ex-1005, ¶60, ¶55. A

POSITA would have been motivated to apply Karczewicz-II's teachings by only partially rounding the center pixel by 5 bits rather than 10 bits to keep it at a higher precision, as Karczewicz-II teaches. *E.g.*, Ex-1006, ¶10, ¶20, ¶¶23-24, ¶39, ¶¶99-103. Karczewicz-II uses the non-rounded half-pixel, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex-1006, ¶74, ¶93, ¶¶99-100. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶96, ¶99, Tables 1 and 3; Ex-1005, ¶60.

163. Scenario 3 (with both motion vectors pointing to half-pixel positions). Karczewicz-II explains how, for H.264, motion vectors point to pixels, including half-pixel positions. *E.g.*, Ex-1006, ¶93. When the two motion vectors in Karczewicz-I's bi-directional prediction point to two half-pixel positions, the default weighted prediction is calculated as an average of the two half-pixels. *See* Ex-1005, ¶60, ¶55.

164. Karczewicz-II teaches an improved calculation for averaging two half-pixel values. Ex-1006, Fig. 4D. While Karczewicz-II uses this calculation for to interpolate a quarter-pixel position, it is the same calculation as Scenario 3 because it averages two half pixels. As discussed above, Karczewicz-II explains

the conventional calculation for averaging two numbers: adding the half-pixel values (e.g., “b” and “h”) and a rounding offset 1, then dividing by 2 (using a right-shift “>>” operation that is mathematically equivalent to dividing by 2). Ex-1006, ¶97, Fig. 4D, Table 2:

TABLE 2

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

165. Karczewicz-II improves this conventional approach by “keep[ing] the highest possible precision through the intermediate steps[.]” Ex-1006, ¶99, ¶101. Karczewicz-II replaces the equations in Table 2 with those in Table 4, where the pixel values are combined at a higher precision. Ex-1006, ¶101. Instead of using a rounded 8-bit half-pixel (e.g., “b” or “h”), Karczewicz-II delays the rounding step and instead uses *non-rounded* half-pixels (e.g., “b1” and “h1”) that are 15 bits each. *Id.* These pixel values are combined, along with a rounding offset of 32, before rounding to reduce the precision at the end, e.g., shifting 6 bits to the right (“>>6”). *Id.*, Table 4:

TABLE 4

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

166. The operations for this improved approach are shown in Table 6 of Karczewicz-II. Ex-1006, ¶103, Table 6:

TABLE 6

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

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

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

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

Bidirectional prediction: pred(*i,j*)=(pred0(*i,j*)+pred1(*i,j*)+1)>>1

where pred0(i,j) and pred1(i,j) are prediction data from list 0 and list 1.

168. For Scenario 3, the two predictions would be half-pixel predictions. *See* Ex-1005, ¶60, ¶55. A POSITA would have been motivated to apply Karczewicz-II's teachings by keeping the half-pixel predictions (pred0(i,j) and pred1(i,j)) at a higher, non-rounded precision, i.e., the weighted sum of the 6-tap interpolation filter without any rounding applied. Ex-1006, ¶¶97-101. As Karczewicz-II teaches, these values are combined with a rounding offset of 32 and then right-shifted 6 bits to reduce the precision. This combination results in the following equation:

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

See Ex-1006, ¶97, ¶101, Tables 2 and 4; Ex-1005, ¶60.

169. Express Teaching Suggestion or Motivation (TSM) in the Art. Preserving higher-precision intermediate values in computations related to video coding was well known in the art as a method for improving the accuracy of computation; the prior art therefore provides express TSM to apply such teachings from Karczewicz-II to related techniques in Karczewicz-I to achieve a higher accuracy for Karczewicz-I's bi-directional prediction. *See, e.g.*, Ex-1016, Abstract

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

170. Karczewicz-I is directed to motion compensation. Preserving higher-precision intermediate values was known to benefit motion compensation. For example, U.S. Patent No. 9,344,744 ("Kirchhoffer-744") (Ex-1008) teaches performing "a prediction and a reconstruction for a block of a picture to be predicted … in a higher precision." Ex-1008, 7:4-19, 16:22-17:36. This approach was known to result in a more precise prediction and a smaller residual information. *Id*. ("An advantage of embodiments according to the second aspect of the present invention is that by predicting and reconstructing in a higher precision than the picture is defined, a more precise prediction and reconstruction can be obtained, leading to a smaller residual information for the block."). Therefore, the prior art provides express TSM to apply Karczewicz-II's teachings to Karczewicz-I and thereby perform motion compensation at a higher precision. *See id.*

171. It was also known that preserving higher-precision intermediate values improves the accuracy of interpolation. *See, e.g.*, Ex-1009, 000005 ("In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding

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

172. Compatible Teachings. The combination would not have changed the principle of operation of either reference, but merely includes the use of a known technique (e.g., Karczewicz-II's technique of calculating the average of two predictions with higher-precision intermediate values) to improve similar devices or methods (e.g., Karczewicz-I's system and method for decoding video using bi-directional prediction) in the same way. As explained above, Karczewicz-I and Karczewicz-II are filed by the same inventors from the same company and teach similar video encoding and decoding methods implemented on similar video encoding and decoding systems, both for averaging predicted pixel values in H.264. Ex-1005, Figs. 1-2, ¶¶29-50, ¶53; Ex-1006, Figs. 1-2, ¶¶40-53, ¶56. Given the similarities between Karczewicz-I and Karczewicz-II, a POSITA would have understood that Karczewicz-II's techniques are readily applicable to Karczewicz-I.

173. Moreover, the combination merely changes when rounding occurs for calculations that already included rounding. *See* Ex-1005, ¶60, ¶55; Ex-1006, ¶¶96-106, Tables 1-8. This minor implementation detail would not have changed the principle of operation of Karczewicz-I. Nor would it have changed the

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

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

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

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

4. Independent Claims 19, 25, and 31

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

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

177. Karczewicz-I "describes video encoding and decoding techniques applicable to bi-directional prediction." Ex-1005, ¶8, ¶22 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction."). Karczewicz-I teaches a "video encoding and decoding system" that includes a "video encoder 22" that encodes video data and a "video decoder 28" that decodes video data. Ex-1005, ¶¶29-30, ¶33, Fig. 1. The video encoder "perform[s] … inter-coding of blocks within video frames" that includes motion estimation and compensation operations. Ex-1005, ¶¶51-76, Fig. 2. The video decoder "perform[s] the reciprocal decoding techniques to the encoding

techniques." Ex-1005, ¶83, Fig. 4.[15] The operations carried out by the video decoder include "generating the predictive data" and "combin[ing] the prediction data … with the residual block … to create a reconstructed video block[.]" Ex-1005, ¶¶85-86. The operations of the video decoder teach a method for decoding video data, including bi-directional prediction techniques taught in Karczewicz-I. *See, e.g.*, Ex-1005, ¶¶55-60.

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

179. Karczewicz-II "describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding." Ex-1006, Abstract, ¶2 ("This disclosure relates to digital video coding and, more particularly, fractional interpolations of predictive data used in video coding."). Karczewicz-II teaches "method[s] of decoding video data." Ex-1006,

---

[15] Karczewicz-I explains that its teachings regarding bi-directional prediction (e.g., ¶¶55-60) are applicable to video decoding. Ex-1005, ¶8 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction."), ¶35; *supra* I.A.

¶13. Karczewicz-II's methods of decoding video data include block-based inter-coding that includes motion estimation and compensation operations. Ex-1006, ¶6 ("[G]iven a set of residual blocks and a set of motion vectors (and possibly some additional syntax), the decoder may be able to reconstruct a video frame that was originally encoded. Inter-coding based on motion estimation and motion compensation can achieve very good compression because successive video frames or other types of coded units are often very similar."). Similar to Karczewicz-I, Karczewicz-II teaches a "video encoding and decoding system" that includes a "video encoder 22" that encodes video data and a "video decoder 28" that decodes video data. Ex-1006, ¶¶40-41 ("video decoder 28 of destination device 16 may be configured to apply one or more of the interpolation techniques of this disclosure as part of a video decoding process."), 44 ("The video decoding process performed by video decoder 28 may also perform interpolation during its motion compensation stage of the decoding process."), Fig. 1. The video decoder "includes a motion compensation unit 55 that performs the interpolation techniques of this disclosure for decoding." Ex-1006, ¶¶63-66, Fig. 3.

180. Similar to Karczewicz-I, Karczewicz-II decodes a block of pixels. *See, e.g.*, Ex-1006, ¶64 ("Motion compensation unit 55 produces motion compensated blocks in the manner described herein, e.g., including interpolation based on a set of interpolation filter coefficients identified by the syntax element

(i.e., the interpolation syntax).”), ¶66 (“Again, the techniques of this disclosure concern motion compensated interpolation in which pixel values of predictive video blocks are interpolated to sub-pixel resolution.”), ¶50 (“In general, macroblocks (MBs) and the various sub-blocks may be considered to be video blocks.”).

181. Karczewicz-II teaches interpolation methods for use in video decoding. *See, e.g.*, Ex-1006, ¶14 (“In another example, this disclosure provides a method of interpolating predictive video data for video coding.”), ¶35 (“This disclosure describes various interpolation techniques performed by an encoder and a decoder during the motion compensation process of video coding.”). As explained above, a POSITA would have found it obvious to modify Karczewicz-I’s technique for obtaining a combined prediction, which is part of Karczewicz-I’s video decoding method, based on Karczewicz-II’s interpolation techniques. *Supra* §V.B.3 (explaining the motivation to combine; that analysis is incorporated here). Thus, the combination of Karczewicz-I and Karczewicz-II teaches a method for decoding a block of pixels.

182. As explained below, Karczewicz-I’s bi-directional prediction operations, as modified based on Karczewicz-II’s interpolation techniques, teach the limitations of claim 1. *Infra* §§V.B.4[19b-19g]. Therefore, the combination of

Karczewicz-I and Karczewicz-II teaches a method for encoding a block of pixels, comprising the operations explained below for limitations [19b]-[19g].

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

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

184. As explained above for [19a], the combination of Karczewicz-I and Karczewicz-II teaches a method for decoding a block of pixels, including operations as described for limitations [19b]-[19g]. *Supra* §V.B.4[19a]. As explained below, the combination of Karczewicz-I and Karczewicz-II teaches a video decoder for encoding a block of pixels that performs operations as described for limitations [25b]-[25g]. *Infra* §§V.B.4[25b-25g].

185. Karczewicz-I and Karczewicz-II teach a video decoder performing video decoding operations. *Supra* §V.B.4[19a]; Ex-1005, ¶¶29-30, ¶33, Fig. 1, ¶¶83-86, Fig. 4:

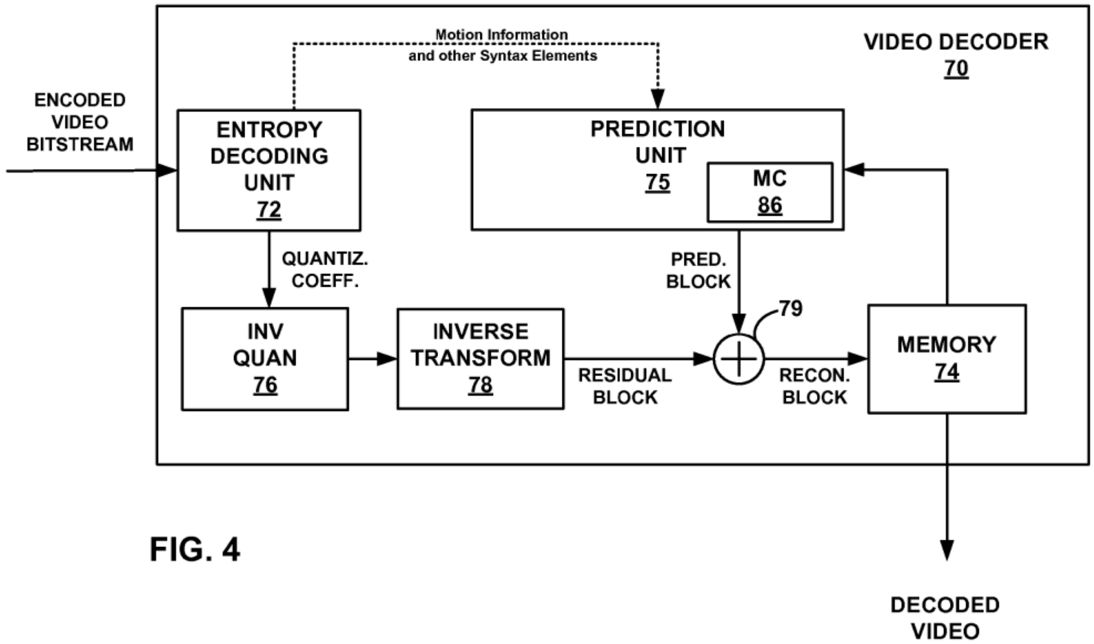


FIG. 4

Ex-1006, ¶¶40-41, ¶44, Fig. 1, ¶¶63-66, Fig. 3:

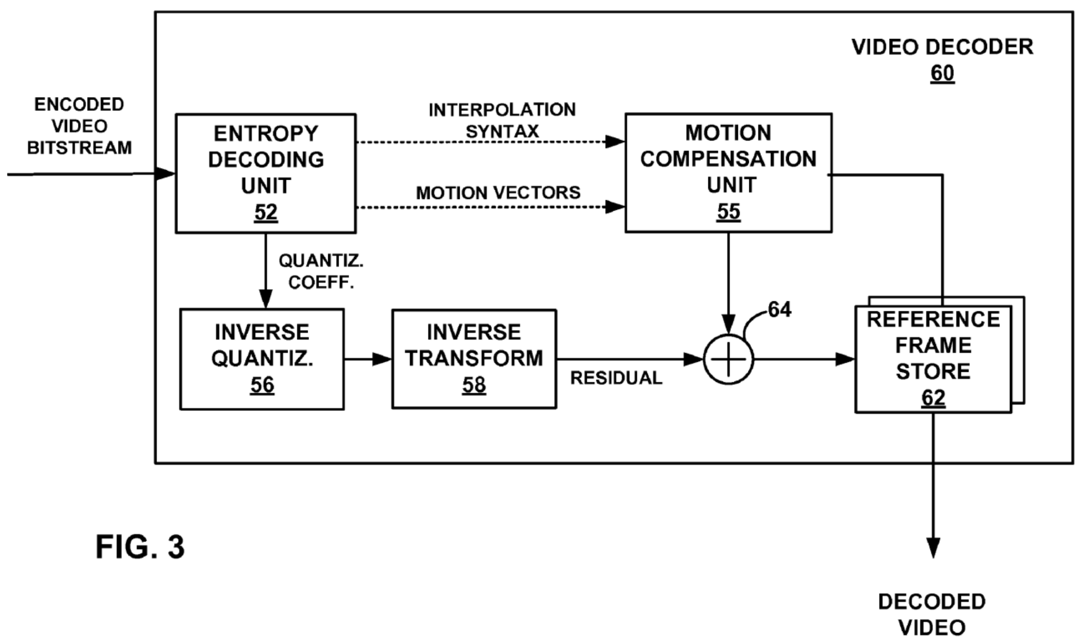


FIG. 3

186. Karczewicz-II, for example, teaches that the video decoder is an apparatus for decoding video. Ex-1006, ¶15 ("[T]his disclosure provides an apparatus that decodes video data …"), ¶17, ¶22, ¶23. Karczewicz-I and

Karczewicz-II teach the video decoder decodes a block of pixels. *See, e.g.*, Ex-1005, ¶¶39-40; Ex-1006, ¶50, ¶64, ¶66. Karczewicz-I and Karczewicz-II teach implementing the video decoder in various types of devices. Ex-1005, ¶3 ("Digital multimedia capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless communication devices, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, video gaming devices, video game consoles, cellular or satellite radio telephones, digital media players, and the like."); Ex-1006, ¶3.

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

more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof."), ¶98 ("The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses …"); Ex-1006, ¶25, ¶118.

188. Karczewicz-I and Karczewicz-II explain that the computer readable medium comprises well-known types of memories. Ex-1005, ¶99 ("The computer-readable storage medium may comprise random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like."); Ex-1006, ¶119. Karczewicz-I and Karczewicz-II further explain that its software includes computer program code. Ex-1005, ¶100 ("The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry."); Ex-1006, ¶27, ¶120.

189. Karczewicz-I and Karczewicz-II further teach the processor executing computer program code in memory to cause the apparatus to carry out its

functionalities. Ex-1005, ¶99 ("If implemented in software, the techniques may be realized at least in part by a computer-readable medium comprising instructions that, when executed in a processor, performs one or more of the methods described above."), ¶100 ("The code or instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry."); Ex-1006, ¶48, ¶¶119-120.

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

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

191. I understand that a preamble generally does not state a claim limitation. However, to the extent that Patent Owner argues that the preamble

states a limitation, it is my opinion that the combination of Karczewicz-I and Karczewicz-II teaches the preamble.

192. As explained above, Karczewicz-I and Karczewicz-II teach implementing the video decoder and performing operations for decoding a block of pixels using software stored in computer readable storage media and executed by a processor. *Supra* §V.B.4[25a]; Ex-1005, ¶12, ¶38, ¶¶98-100; Ex-1006, ¶¶25-26, ¶48, ¶¶119-120. A POSITA would have understood that the types of computer readable storage media disclosed by Karczewicz-I and Karczewicz-II, including random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, are forms of non-transitory computer readable storage medium, as opposed to transitory signals. Ex-1005, ¶99; Ex-1006, ¶119. Karczewicz-I and Karczewicz-II teach that the software, which is stored in computer readable storage medium and executable by a processor, includes computer executable program code portions that comprise program code instructions. Ex-1005, ¶100; Ex-1006, ¶27, ¶120.

193. The operations taught by the combination of Karczewicz-I and Karczewicz-II, which are performed by the software residing on the non-transitory computer-readable storage medium, teach the limitations of claim 31. *Infra* §V.B.4[31b-31g]. Karczewicz-I and Karczewicz-II further teach forming a

computer program product using the non-transitory computer readable storage media that stores software. Ex-1005, ¶99 ("The computer-readable medium may comprise a computer-readable storage medium and may form part of a computer program product …"); Ex-1006, ¶119. Therefore, the combination of Karczewicz-I and Karczewicz-II teaches a computer program product for decoding a block of pixels, the computer program product comprising at least one non-transitory computer readable storage medium having computer executable program code portions stored therein, the computer executable program code portions comprising program code instructions configured to perform the operations recited in claim 31.

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

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

195. Karczewicz-I and Karczewicz-II teach generating motion vectors, in the encoding process, that point to **reference blocks** (e.g., predictive/prediction video blocks) **for a current block** and indicate the displacement between the reference blocks and the current block. Ex-1005, ¶7 ("For P- and B-video blocks, motion estimation generates motion vectors, which indicate the displacement of the video blocks relative to corresponding prediction video blocks in predictive

reference frame(s) or other coded units."), ¶53 ("A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit)."), ¶54 ("The selected motion vector for any given list may point to a predictive video block that is most similar to the video block being coded, e.g., as defined by a metric such as sum of absolute difference (SAD) or sum of squared difference (SSD) of pixel values of the predictive block relative to pixel values of the block being coded."); Ex-1006, ¶4 ("Motion estimation generates motion vectors, which indicate the displacement of video blocks relative to corresponding prediction video blocks in one or more reference frames or other coded units."), ¶56 ("Motion estimation is typically considered the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a predictive block within a predictive frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit).").

196. The decoder performs the inverse process (*supra* §I.B; Ex-1005, ¶83), which includes motion compensation, to reconstruct the current block based on data received from the encoder. Ex-1005, ¶85 ("The motion information (e.g., motion vectors) and other syntax are forwarded to prediction unit 75 for use in generating the predictive data. Prediction unit 75 performs bidirectional prediction

consistent with this disclosure, avoiding rounding adjustments in some cases, and possibly implementing default, implicit or explicit weighted prediction according to the received syntax elements."), ¶88 ("Entropy decoding unit 72 may output syntax elements to prediction unit, which includes the one or more syntax elements that indicate whether a rounding adjustment was used to encode the video data, motion vectors and possibly other syntax."), ¶89 ("Prediction unit 75 invokes motion compensation unit 86 for block based predictive decoding."), Figs. 4, 6.

197. As part of motion compensation, the decoder **determines reference blocks** (e.g., predictive blocks) **based on motion vectors** and reconstructs the target block based on the determined reference blocks. Ex-1006, ¶6 ("A coded video block may be represented by prediction information that can be used to create or identify a predictive block, and a residual block of data indicative of differences between the block being coded and the predictive block. The prediction information may comprise the one or more motion vectors that are used to identify the predictive block of data. Given the motion vectors, the decoder is able to reconstruct the predictive blocks that were used to code the residual."), ¶63 ("Specifically, motion compensation unit 55 may generate the prediction data based on motion vectors received from entropy decoding unit 52 and the interpolations as defined by syntax element (labeled interpolation syntax in FIG.

3). Based on this interpolated prediction data, the video data (e.g., a reconstructed residual video block) can be decoded."), Fig. 3.

198. Karczewicz-I teach video decoding using bi-directional prediction based on two lists of reference data. Ex-1005, ¶8 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction. In bi-directional prediction, a video block may be predictively encoded and decoded based on two different lists of predictive reference data."), ¶22 ("This disclosure describes video encoding and decoding techniques applicable to bi-directional prediction. In bi-directional prediction, a video block is predictively encoded and decoded based on two different lists of predictive reference data."), ¶42 ("The techniques of this disclosure are specifically applicable to weighted bi-directional prediction. As mentioned above, bi-directional prediction is prediction of so-called 'B-video blocks' based on two different lists of data. B-video blocks may be predicted from two lists of data from two previous frames, two lists of data from subsequent frames, or one list of data from a previous frame and one from a subsequent frame."), ¶¶5-6.

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

motion vectors, two reference blocks are determined based on those two motion vectors. Therefore, the combination teaches **determining, for a current block, a first reference block based on a first motion vector and second reference block based on a second motion vector**.

200. Karczewicz-II further teaches **that the pixels of the current block, the first reference block, and the second reference block have values with a first precision** (e.g., 8 bits). As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Karczewicz-II's use of the term "precision" is consistent with this interpretation. Ex-1006, ¶89 ("The average filter may also be quantized to a certain fixed-point precision (e.g., 13-bit precision).").

201. Karczewicz-II teaches that 8 bits are needed to represent possible pixel values for the two reference blocks. For example, Table 5 teaches operations for calculating pixel values using interpolation. Ex-1006, ¶103, Table 5:

TABLE 5

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

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

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

202. Karczewicz-I teaches that bi-directional inter-prediction is performed based on reference blocks that are I- or P-blocks for a current B-block. *See, e.g.*, Ex-1005, ¶7 ("I-and P-units are commonly used to define reference blocks for the inter-coding of P- and B-units."). Given Karczewicz-II's teaching that 8 bits are

needed to represent possible integer pixel values, the combination teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision (e.g., 8 bits).

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

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

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

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

See also ¶24, ¶44, ¶48.

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

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

206. Here, pred0(i,j) is a prediction based on a "motion compensated reference area[] … obtained from list 0 … reference picture." Ex-1005, ¶58. Because Karczewicz-I teaches that the inter-prediction process for a current block includes determining reference blocks based on motion vectors (*supra* §V.B.4[19b]/[25b]/[31b]; Ex-1005, ¶7, ¶¶53-54), the motion compensated reference area refers to a reference block.

207. As explained above, a POSITA would have found it obvious, based on Karczewicz-II's teachings of using higher-precision intermediate values, to modify Karczewicz-I's calculation of the bi-directional prediction to use higher-precision predictions as intermediate values. *Supra* §V.B.3 (explaining how and why a POSITA would have combined teachings from Karczewicz-I and Karczewicz-II; that analysis is incorporated here). This modification is directed to calculation bit depth and order; it therefore does not change the reference blocks on

which Karczewicz-I's predictions are based. Thus, the combination of Karczewicz-I and Karczewicz-II teaches obtaining said first prediction using said first reference block.

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

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

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

Id.

210. Karczewicz-II teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., “15s”) is needed to represent these possible values. Karczewicz-II further teaches that the left-shifted integer pixel prediction (e.g., r1<<5) has possible values from 0 to 8160 and that a 13-bit unsigned number (i.e., “13u”) is needed to represent these possible values. Ex-1006, ¶103, Table 5:

TABLE 5

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

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

211. Therefore, 13 bits are needed to represent the possible values of each of the first prediction and the second prediction. As explained above, the term “precision” is at least satisfied by “a number of bits needed to represent possible

values." *Supra* §IV.A. The first prediction and the second prediction each need at least 13 bits to represent their possible values,[16] which is higher than the 8-bit precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4[19b]/[25b]/[31b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred0(i,j))), **said first prediction having a second precision** (e.g., 13 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 1.

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

---

[16] Because the first prediction, which is a non-rounded half-pixel prediction, is represented by 15 bits, it needs at least 13 bits (along with 2 additional bits) to represent its possible values. Therefore, the first prediction attains the precision level of 13 bits and thus has 13 bits of precision.

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

Id.

213. Karczewicz-II teaches that the partially-rounded center-pixel prediction (e.g., j1>>1)[17] has possible values from -4957 to 13116 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶105, Table 8. Karczewicz-II further teaches that the non-rounded half-pixel prediction (e.g., b1) has possible values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶105, Table 8:

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

TABLE 8

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

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

214. Therefore, 15 bits are required to represent the possible values of each of the first prediction and the second prediction. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each has a precision of 15 bits, which is higher than the 8-bit precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4[19b]/[25b]/[31b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., partially-rounded center-pixel prediction, i.e., non-

rounded(pred0(i,j))>>5)**, said first prediction having a second precision** (e.g., 15 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 2.

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

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

Id.

216. Karczewicz-II teaches that the non-rounded half-pixel predictions (e.g., b1, h1) each has values from -2550 to 10710 and that a 15-bit signed number (i.e., "15s") is needed to represent these possible values. Ex-1006, ¶103, Table 6:

TABLE 6

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

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

217. Therefore, 15 bits are required to represent the possible values of each of the first prediction and the second prediction. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. The first prediction and the second prediction each has a precision of 15 bits, which is higher than the 8-bit precision for the pixels of the current block, the first reference block, and the second reference block. *Supra* §V.B.4[19b]/[25b]/[31b]. Thus, the combination of Karczewicz-I and Karczewicz-II teach **using said first reference block** (e.g., block from list 0) **to obtain a first prediction** (e.g., non-rounded half-pixel prediction, i.e., non-rounded(pred0(i,j))), **said first prediction having a second precision** (e.g., 15 bits), **which is higher than said first precision** (e.g., 8 bits) under Scenario 3.

[19d]/[25d]/[31d] [using/use] said second reference block to obtain a second prediction, said second prediction having the second precision;

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

219. As explained above, Karczewicz-I teaches calculating a bi-directional prediction as an average of two predictions (e.g., pred0(i,j) and pred1(i,j)). *Supra* §V.B.4[19c]/[25c]/[31c]. Here, pred1(i,j) is a prediction based on a "motion compensated reference area[] … obtained from … list 1 reference picture." Ex-1005, ¶58. Because Karczewicz-I teaches that the inter-prediction process for a current block includes determining reference blocks based on motion vectors (*supra* §V.B.4[19b]/[25b]/[31b]; Ex-1005, ¶7, ¶¶53-54), the motion compensated reference area refers to a reference block.

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

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

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

222. As explained with respect to limitations [19c], [25c], and [31c], in Scenario 1, the combination of Karczewicz-I and Karczewicz-II teach **using said second reference block** (e.g., block from list 1) **to obtain a second prediction** (e.g., left-shifted integer pixel prediction, i.e., pred1(i,j)<<5)**, said second prediction having the second precision** (e.g., 13 bits). *Supra* §V.B.4[19c]/[25c]/[31c].

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

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

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

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

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

226. As explained above, Karczewicz-I's pred0(i,j) and pred1(i,j) respectively satisfy said first prediction and said second prediction. *Supra* §§V.B.4[19c-19d]/[25c-25d]/[31c-31d]. Karczewicz-I teaches obtaining a bi-directional prediction by averaging the first and second predictions. Ex-1005, ¶60:

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

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

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

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

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

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction, and a rounding offset (e.g., non-rounded(pred0(i,j))+pred1(i,j)<<5+32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said second prediction (e.g., pred1(i,j)<<5), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶103, Table 5. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

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

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction, and a rounding offset (e.g., non-rounded(pred0(i,j))>>5+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))>>5) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶105, Table 8. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

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

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

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

Supra §V.B.3. The combination teaches calculating a sum of the first prediction, the second prediction, and a rounding offset (e.g., non-rounded(pred0(i,j))+non-rounded(pred1(i,j)) +32). This sum is a combined prediction because it combines the first and second predictions. The combined prediction is obtained based on said first prediction (e.g., non-rounded(pred0(i,j))) and said second prediction (e.g., non-rounded(pred1(i,j))), as well as the rounding offset (e.g., 32). *See* Ex-1006, ¶103, Table 6. This teaches that the combined prediction is obtained based *at least partly* upon said first prediction and said second prediction.

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

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

233. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches the combined prediction as a sum of the first prediction, the second prediction, and a rounding offset. *Supra* §V.B.4[19e]/[25e]/[31e]. Both Karczewicz-I and Karczewicz-II teach the operation ">>" for shifting bits to the right. Ex-1005, ¶55 ("Often, this division is accomplished by adding 1 to the

addition of the weighted blocks of the first and second frames and then shifting the result to the right by one bit.”), ¶57 (“>> is a right shift operation[.]”), ¶60; Ex-1006, ¶94 (“In this disclosure, ‘>>’ represents a right shift operation and ‘<<’ represents a left shift operation.”). The combination of Karczewicz-I and Karczewicz-II further teaches **decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right** for each of the three scenarios discussed above.

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

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

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

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

TABLE 5

positions {a, c, d, 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

Ex-1006, ¶103, Table 5. In this table, the operation "r1=r1+32" calculates the sum of a non-rounded half-pixel prediction (e.g., y0 which can take the value of b1), a left-shifted integer pixel prediction (e.g., 32*x), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -2518 and 18902 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values between -39 and 295; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values."

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

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

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

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

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

TABLE 8

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

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

Ex-1006, ¶105, Table 8. In this table, the operation "r1=r1+32" calculates the sum of a partially-rounded center-pixel prediction (e.g., j1>>1), a non-rounded half-pixel prediction (e.g., y0, which may take the value of b1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values between -7491 and 23842 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values between -235 and 745; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent a value." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

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

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

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

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

TABLE 6

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

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

Ex-1006, ¶103, Table 6. In this table, the operation "r1=r1+32" calculates the sum of two non-rounded half-pixel predictions (e.g., y0 and y1, which may take the values of b1 and h1), and a rounding offset (e.g., 32). Karczewicz-II teaches that this sum has possible values from -5068 to 21452 and that a 16-bit signed number (e.g., "16s") is needed to represent these possible values. Next, the operation "r1=r1>>6" shifts the sum 6 bits to the right. The result has possible values from -79 to 335; an 11-bit signed number (e.g., "11s") is needed to represent these possible values. As explained above, the term "precision" is at least satisfied by "a number of bits needed to represent possible values." *Supra* §IV.A. Because the number of bits needed to represent possible values of the combined prediction is decreased from 16 to 11, the combination teaches decreasing a precision of said combined prediction.

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

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

241. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches determining the combined prediction in the motion compensation process. *Supra* §§V.B.4[19b-19f]/[25b-25f]/[31b-31f]. Karczewicz-I teaches **reconstructing the block of pixels** (e.g., reconstructed video block) **based on the combined prediction** (e.g., prediction block). Ex-1005, ¶86 ("Adder 79 combines the prediction data (e.g., a prediction block) generated by prediction unit 75 with the residual block from inverse transform unit 78 to create a reconstructed video block, which may be stored in memory 74 and/or output from video decoder 70 as decoded video output."), ¶89 ("[V]ideo decoder 70 may invoke adder 79 to combine weighted prediction data (e.g., a prediction block) with residual video data (e.g., a residual block) in order to generate a reconstruction of the video data (e.g., a reconstructed video block)."); Figs. 4, 6. Karczewicz-II includes similar teachings. Ex-1006, ¶6, ¶65, ¶111, Figs. 3, 6.

5. Dependent Claims 20, 26, and 32

20. The method according to claim 19,

26. The apparatus according to claim 25,

32. The computer program product according to claim 31,

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

242. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §§V.B.3-4. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitations of claims 20, 26, and 32.

243. Karczewicz-II teaches obtaining predictions by interpolation using pixel values of reference blocks when motion vectors point to subpixels. Ex-1006, ¶7 ("In this case, the predictive data generated during motion compensation, which is used to code a video block, may be interpolated from the pixels of video blocks of the video frame or other coded unit used in motion estimation. Interpolation is often performed to generate predictive half-pixel values (half-pel) and predictive quarter-pixel values (quarter-pel)."), ¶42 ("The interpolation techniques of this disclosure may be performed by any encoding device that supports motion compensated interpolation to sub-pixel resolution."), ¶66 ("Again, the techniques of this disclosure concern motion compensated interpolation in which pixel values

of predictive video blocks are interpolated to sub-pixel resolution.”), ¶¶68-72, Figs. 4A-4D.

244. As explained above, the combination of Karczewicz-I and Karczewicz-II contemplates at least three scenarios for determining a bi-directional prediction. Karczewicz-I and Karczewicz-II’s teachings with respect to each of the three scenarios satisfy the additional limitations of claims 20, 26, and 32.

245. Scenario 1. In this scenario, the first motion vector points to a half-pixel position. *Supra* §V.B.3. The half-pixel position refers to a subpixel. *See, e.g.*, Ex-1006, ¶74 (“For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample ‘C3’ and labeled ‘a’ through ‘o’ in FIGS. 4A-4D.”), Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

246. Karczewicz-II teaches **obtaining the first prediction** (e.g., the non-rounded half-pixel prediction) **by interpolation using pixel values of said first reference block**. Ex-1006, ¶93:

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

"C3" and "C4" being the closest, "C2" and "C5" being the next closest, and "C1" and "C6" being the next closest.

The interpolation is performed using pixel values of "six closest integer pixels that surround" the subpixel. Because the integer pixels are closest to the subpixel, they are located in the reference block that the first motion vector points to. *See* Ex-1006, Fig. 4B:

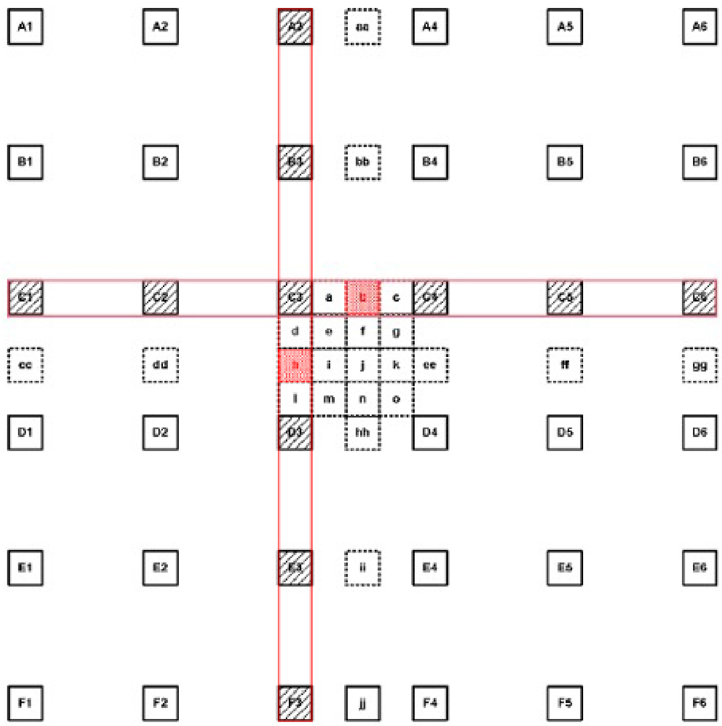


FIG. 4B

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

247. Scenario 2. In this scenario, the first motion vector points to a center-pixel position. *Supra* §V.B.3. The center-pixel position refers to a subpixel. *See,*

e.g., Ex-1006, ¶74, Figs. 4A-4D, ¶10. This is **an instance in which said first motion vector points to a subpixel**.

248. Karczewicz-II teaches **obtaining the first prediction** (e.g., the partially-rounded center-pixel prediction) **by interpolation using pixel values of said first reference block**. Ex-1006, ¶95:

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

The interpolation is performed using non-rounded half-pixel predictions, which are in turn obtained by interpolation using integer pixel values. The half-pixel positions and their corresponding integer pixel positions are close to the center-pixel position and are located in the reference block that the first motion vector points to. *See* Ex-1006, Fig. 4C:

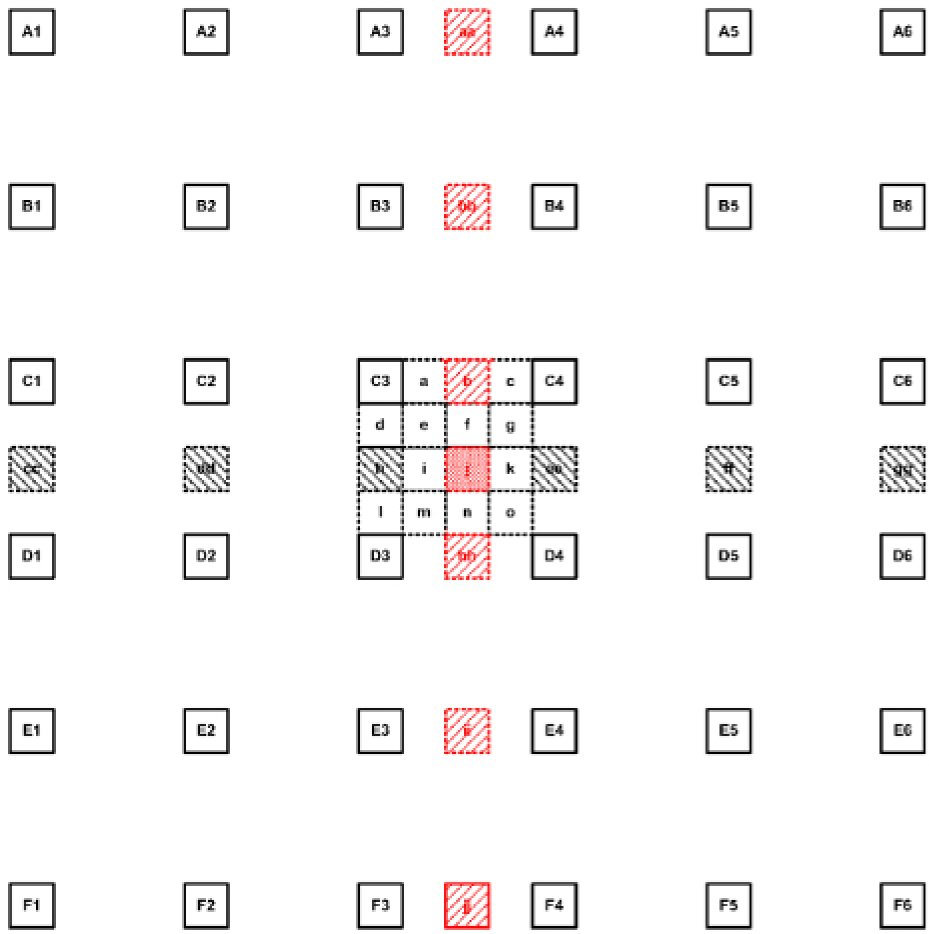


FIG. 4C

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

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

250. Moreover, the ’267 patent admits that the limitations of claims 20, 26, and 32 were known in the prior art by describing it in the “Background Information” section. Ex-1001, 2:60-3:11.

6. Dependent Claims 21, 27, and 33

21. The method according to claim 20,

27. The apparatus according to claim 26,

33. The computer program product according to claim 32,

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

251. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 20, the apparatus according to claim 26, and the computer program product according to claim 32. *Supra* §V.B.5. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 21, 27, and 33 under Scenario 2.

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

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

Supra §V.B.3. In this equation, the first prediction "non-rounded(pred0(i,j))>>5" is a partially-rounded center-pixel prediction. *Supra* §V.B.3, §V.B.4[19c]/[25c]/[31c]. As explained for claims 20, 26, and 32, the partially-rounded center-pixel prediction is determined by interpolation using values of said first reference block. §V.B.5.

253. Karczewicz-II teaches calculating the partially-rounded center-pixel prediction (e.g., j1>>5) by first determining a sum of a 6-tap filter (e.g., j1). Ex-1006, ¶95:

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

Supra §V.B.5. This sum is a non-rounded version of the center-pixel prediction. Next, the non-rounded center-pixel prediction is shifted to the right by 5 bits to obtain the partially-rounded center-pixel prediction (j1>>5). *Supra* §V.B.3; *see, e.g.*, Ex-1006, 99, Table 3:

TABLE 3

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

Table 8 further teaches operations implementing the equations that include shifting bits to the right, consistent with Table 3. Ex-1006, ¶105, Table 8.

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

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

7. Dependent Claims 22, 28, and 34

22. The method according to claim 20,

28. The apparatus according to claim 26,

34. The computer program product according to claim 32,

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

256. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 20, the apparatus according to claim 26, and the computer program product according to claim 32. *Supra* §V.B.5. The combination of Karczewicz-I and Karczewicz-II further teaches that said first prediction is obtained by interpolation using values of said first reference block based on each of Scenarios 1, 2, and 3. *Id*. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitation of claims 22, 28, and 34 under Scenario 1.

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

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

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

Supra §V.B.3. In this equation, the second prediction "pred1(i,j)<<5" is a left-shifted integer pixel prediction. *Supra* §V.B.3, §§V.B.4[19c-19d]/[25c-25d]/[31c-31d].

259. Karczewicz-II teaches calculating the left-shifted integer pixel prediction by shifting the pixel value to the left (e.g., C3<<5). *Supra* §V.B.3; *see, e.g.*, Ex-1006, ¶99, Table 3:

TABLE 3

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

Here, the value that is shifted to the left is the value of an integer pixel sample (e.g., C3). Ex-1006, ¶74 ("For any given integer-pixel sample, there are altogether 15 sub-pixel positions, which are shown for integer-pixel sample "C3" and labeled 'a' through 'o' in FIGS. 4A-4D."), ¶93 ("where 'C1,' 'C2,' 'C3,' 'C4,' 'C5' and 'C6' represent the six closest integer pixels that surround 'b' in the horizontal direction"). Because the second motion vector points to this integer pixel sample, the integer pixel sample is part of the second reference block; its value is a value of said second reference block.

260. The teachings of Karczewicz-I and Karczewicz-II above explain calculations for one pixel in a block. Karczewicz-I and Karczewicz-II teach that motion compensation is performed on a block basis. *See supra* §§V.B.4[19b, 19g]/[25b, 25g]/[31b, 31g]; Ex-1005, ¶7; Ex-1006, ¶4, ¶73. Therefore, the references teach performing the predictions operations on all the pixels of a block. This is further obvious because this is how block-based motion prediction has worked since the 1990s. As explained above, in motion estimation and compensation, a motion vector indicates the displacement of between a reference block and a current block of pixels. *Supra* §V.B.4[19b]/[25b]/[31b]; Ex-1005, ¶7, ¶¶53-54; Ex-1006, ¶4, ¶56. Therefore, Karczewicz-I and Karczewicz-II teach performing the above operations, including the left-shift operation, for each pixel of the current block based on corresponding pixels of the reference block. In Scenario 1, the left shifting is performed for multiple pixels.

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

8. Dependent Claims 23, 29, and 35

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

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

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

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

262. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §V.B.4. As explained below, it is my opinion that the combination of Karczewicz-I and Karczewicz-II further teaches the additional limitations of claims 23, 29, and 35.

263. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches obtaining a combined prediction via their disclosure of calculating a sum by adding up the first prediction, the second prediction, and the rounding offset (e.g., 32). *Supra* §V.B.4[19e]/[25e]/[31e]. The combination teaches decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right. *Supra* §V.B.4[19f]/[25f]/[31f]. The combination thus

teaches **inserting a rounding offset** (e.g., 32) **to the combined prediction**, as explained for limitations [19e], [25e], and [31e], **before said decreasing** of the precision, as explained for limitations [19f], [25f], and [31f].

264. The value added to the sum of the first and second predictions is a rounding offset. Karczewicz-I refers to the term that is added to the weighted sum of the first and second predictions as a rounding adjustment. Ex-1005, ¶63 ("Generally, a rounding adjustment of 2^{r-1} is commonly used prior to a right shift by r, where r represents a positive integer."), ¶55.[18] As was well known to those skilled in the art, "rounding adjustment" was used interchangeably with "rounding offset." The value (e.g., 32) is inserted to the combined prediction, increasing the value of the combined prediction, right before the rounding operation. A POSITA would have understood that this value is a rounding offset according to the plain meaning of the term.

265. Karczewicz-I and Karczewicz-II teach this rounding offset as part of its rounding process, which decreases the precision as recited by the claims. Ex-1005, ¶55, ¶63; Ex-1006, ¶¶96-101, Tables 1-4. Additionally, it was obvious for said decreasing to include the rounding offset (claim 23) because the insertion of

[18] Karczewicz-I teaches a rounding adjustment of 2^{r-1} prior to a right shift by r. This is consistent with the modified equation for calculating the bi-directional prediction under each of the three scenarios as taught by the combination of Karczewicz-I and Karczewicz-II (*supra* §V.B.3), which teaches right shifting by 6 bits and a rounding offset of 32: $2^{6-1}=2^5=32$.

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

266. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the at least one memory and computer program code are configured to cause the apparatus to perform operations that render obvious limitation [25e] and [25f]. *Supra* §V.B.4[25a]. Because it would have been obvious for decreasing said precision to comprise inserting a rounding offset, it would have been obvious that **the at least one memory and computer code are configured to cause the apparatus to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right by** inserting a rounding offset to the combined prediction before said decreasing (claim 19).

267. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the program code instructions are configured to perform the operations recited in claim 31, including limitations [31e] and [31f]. *Supra* §V.B.4[31a]. Because it would have been obvious for decreasing said precision to comprise inserting a rounding offset, it would have been obvious that **the program code instructions configured to decrease said precision of said combined prediction by shifting bits of the combined prediction to the right further**

comprise program code instructions configured to inserting a rounding offset to the combined prediction before said decreasing (claim 35).

9. Dependent Claims 24, 30, and 36

24. The method according to claim 19,

30. The apparatus according to claim 25,

36. The computer program product according to claim 31,

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

268. The combination of Karczewicz-I and Karczewicz-II teaches the method according to claim 19, the apparatus according to claim 25, and the computer program product according to claim 31. *Supra* §V.B.4. As explained below, it is my opinion that the combination further teaches the additional limitations of claims 24, 30, and 36.

269. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches that the pixels of the current block, the first reference block, and the second reference block have values with a first precision because 8 bits are needed to represent the possible pixel values of these blocks. *Supra* §V.B.4[19b]/[25b]/[31b]. Here, **the first precision indicates a number of bits needed to represent the values of the pixels**.

270. As explained above, the combination of Karczewicz-I and Karczewicz-II teaches said first prediction and second prediction having a second precision that is higher than said first precision because more bits are needed to represent the possible values of the predictions under each of Scenarios 1, 2, and 3. *Supra* §§V.B.4[19c-19d]/[25c-25d]/[31c-31d]. In Scenario 1, 13 bits are needed to represent the possible values of the first prediction and the second prediction. *Id*. In Scenarios 2 and 3, 15 bits are needed to represent the possible values of the first prediction and the second prediction. *Id*. Here, **the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction**.

VI. BACKGROUND AND QUALIFICATIONS

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

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

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

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

recognition of my work, I received National Aeronautics and Space Administration Group Achievement Awards in 1990 and 1992.

275. After June 1994, I began a six-month stint as a contract Software Engineer for the Federal National Mortgage Association in Washington D.C., for which I developed a graphical user interface to monitor and validate loan servicer input for a Loss Mitigation Project. I then served as an Independent Consultant to Optivision, Inc. for the next six months, where I researched and developed rate control algorithms and software based on the MPEG-2 Test Model 5 for the OPTIVideo™ MPEG-2 video encoder, as well as adaptive quantization algorithms based on the then-JPEG-3 draft standard. In this role, I researched and developed algorithms to improve the quality of gray scale image compression for the medical imaging DICOM Standard by providing a lossless hybrid algorithm encoding image residuals with a diagonal Golomb code based an Enhanced Universal Trellis Coded Quantization algorithm.

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

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

278. In October 1998, I began a six-month engagement with Rockwell Collins Inc., where I worked as a Lead Systems Engineer tasked with harmonizing requirements for an MPEG-2 in-flight entertainment system. I then worked for Sun Microsystems Inc. as a Software Engineer until November 1999. During my time at Sun Microsystems Inc., I developed a Distributed Component Object Model (DCOM) software interface between a TV control graphical user interface and a Microsoft broadcast application programming interface (API) with the goal of improving the visual quality of interactive TV displays derived from UDP/IP datagrams synchronized with MPEG-2 audio/video packet data.

279. For the next 22 months, from January 2000 to October 2002, I worked as the Chief Systems Engineer for Media Logic Systems Ltd. During my time at Media Logic Systems, I designed and developed a live interactive television system (iSeeTV) in which customers communicate with human sales agents in

video-enabled call centers. To create this system, I researched and developed tools and encoder systems to improve image quality at prescribed latency and bit rate for distributing live video and audio streams encoded via low latency methods. To perform the above, I was required to understand and implement video codec systems employing the MPEG-2 Simple Profile at Main Level (CATV), MPEG-4 Visual Profile with background sprite coding, and the H.263+ Standard (now known as H.264).

280. Since November 2002, I have been an engineering contractor, and more recently an independent consultant in mathematical modeling, for several companies, such as Cyra Technologies Inc. and Amgen Inc. I also served as a senior research fellow at Merck & Co., Inc., a manager at GlaxoSmithKline Inc., a director at Daiichi Sankyo, Inc., a senior director at Praxis Precision Medicines, and currently serve as a director at Takeda Pharmaceuticals. During this time, I have developed mathematical models and simulations related to various systems, signals, and images. Specifically, I have focused on analyzing, processing, storing, and deriving information from biomedical imaging and other data. Using the information derived from these data, I have created a variety of models related to biology and the effects of drugs on the human body. In recognition of my work, I have received GlaxoSmithKline R&D Recognition Awards in 2012, 2013, and

2016, a Daiichi Sankyo recognition award in 2021 and Takeda Pharmaceutical awards in 2022 and 2023.

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

282. I am also a Senior Member of the IEEE and serve as the current Philadelphia Chapter Chair of the Communications & Information Theory Societies as well as former Chair of the American Association of Pharmaceutical Scientists Pharmaco-Imaging Community. I also served as the 2019 Vice Chair of the IEEE P2673 Intelligence Augmentation for Medical Imaging Standards Working Group. I also have been registered to practice as a patent agent for the United States Patent and Trademark Office since 2002 (Reg. No. 51,704).

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

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

A. Compensation

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

B. Materials and Other Information Considered

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

VII. UNDERSTANDING OF THE LAW

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

VIII. RESERVATION OF RIGHTS

288. My opinions are based upon the information that I have considered to date. I am unaware of any evidence of secondary considerations with respect to the ’267 Patent that would render any of the challenged claims non-obvious. I reserve the right, however, to supplement my opinions in the future to respond to any arguments raised by the owner of the ’267 Patent and to take into account new information that becomes available to me.

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

Executed on March 14, 2024.

By:

I. Freedman

Immanuel Freedman

APPENDIX 1: CURRICULUM VITAE OF IMMANUEL FREEDMAN

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

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

SUMMARY OF EXPERIENCE

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

EDUCATION

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

LICENSES

Registered Patent Agent #51,704

EXPERIENCE

Takeda Pharmaceuticals **Cambridge, MA** **Mar '22-present**
Clinical Pharmacology Director

Dr. Freedman provides mathematical modeling of systems, signals, and images.

Freedman Patent **Harleysville, PA** **Oct '21-present**
Sole Proprietor

He provides consulting and expert witness services to industry and the legal profession.

State University of New York at Buffalo **Buffalo, NY** **Jun '17-present**
Volunteer Researcher

Praxis Precision Medicines **Boston, MA** **Aug '21-Sep '21**
Senior Director, Pharmacometrics

Dr. Freedman provided mathematical modeling of systems, signals, and images.

Daiichi Sankyo, Inc. **Basking Ridge, NJ** **Nov '20-Aug '21**
Director, Modeling and Simulation

Dr. Freedman provided mathematical modeling of systems, signals, and images.

Freedman Patent **Harleysville, PA** **Jun '16-Nov '20**
Sole Proprietor

He provided consulting and expert witness services to industry and the legal profession. In particular, he provided requirements analysis and design for a precision dosing system Graphical User Interface.

GlaxoSmithKline, Inc. Upper Merion, PA Aug '07-Jun '16
Manager, Clinical Pharmacology

Dr. Freedman provided mathematical modeling of systems, signals and images and participated in technical due diligence activities on demand.

Freedman Patent Harleysville, PA Jul '05-Aug '07
Sole Proprietor

He provided consulting and expert witness services to industry and the legal profession. In particular, he provided requirements analysis and design for a precision dosing system Graphical User Interface.

Merck & Co., Inc., West Point, PA Nov '04-Jul '05
Senior Research Fellow

Dr. Freedman provided mathematical models on demand.

Amgen, Inc. Thousand Oaks, CA Apr '03-Mar '04
Contractor

Dr. Freedman developed algorithms and software in MATLAB and FORTRAN for simulation and data modeling.

Cyra Technologies, Inc. San Ramon, CA Nov '02-Apr '03
Senior Hardware Engineer

Dr. Freedman designed, developed, and tested algorithms and software for calibrating a three-dimensional laser scanner. He calculated the statistical distribution of outcomes for an engineering tolerance stack by modeling and simulating the scanner response using a Jacobian sensitivity matrix to compare alternative placements of scanner calibration targets based on a D-matrix of scanner response.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Dr. Freedman evaluated frame grabber hardware for resolution and quality of time-integrated imagery and specified algorithms including cluster analysis and trending, further developing a user interface for a digital image processing system supporting gene-cloning science.

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

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

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

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

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

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

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

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

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

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

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

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

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

He provided technical consulting services to industry.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

National Coal Board Nuneaton, England UK Nov ’82-Sep ’84
Scientist (Management Grade 7)

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

PUBLICATIONS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Freedman, I. (2014), "On horses and bitumen in mulAPIN," NABU 2014-1 Nr. 33

Freedman, I. (2014), "Pharmaco-Imaging in Translational Science and Research," in Pharmaco-Imaging in Drug and Biologics Development (Moyer, B.R., Cheruvu, N.P.S. & Hu, T. C.-C., eds.), Springer, Inc.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

PROFESSIONAL AWARDS
Takeda Pharmaceuticals Recognition Awards (2022, 2023)

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

PROFESSIONAL AFFILIATIONS

Institute of Electrical and Electronic Engineers (Senior Member; Chair, Communications & Information Theory Chapter, Philadelphia Section; 2018/2019 Vice-Chair, P2673 Standards Working Group; Trusted Analytic Exchange Sub-Group Chair, P2795 Standards Working Group; Member P1900.8 Standards Working Group; Former Secretary, Dynamic Spectrum Access Machine Learning Study Group)
Institute of Physics (Chartered Physicist)
American Association of Pharmaceutical Scientists (Former Chair, Pharmaco-Imaging Community; Former Chair, Predictive Modeling Community; 2017 Member, Predictive Modeling Task Force)
International Society of Pharmacometrics (Former Member Standards and Best Practices Committee Member–Model Evaluation Group)
Regulatory Affairs Professionals Society
International Biometric Society (Eastern North American Region)
International Association for Assyriology
National Coalition of Independent Scholars
Ronin Institute for Independent Scholarship (Research Scholar)
Royal Society of Medicine (Fellow Member)

December 22, 2023

APPENDIX 2: MATERIALS CONSIDERED IN THE PREPARATION OF THIS DECLARATION

Exhibit No.	Description
1001	U.S. Patent No. 11,805,267 to Ugur *et al.* ("the '267 patent")
1002	Prosecution History for the '267 Patent
1004	U.S. Patent Application Publication No. 2005/0281334 ("Walker")
1005	U.S. Patent Application Publication No. 2011/0007799 ("Karczewicz-I")
1006	U.S. Patent Application Publication No. 2009/0257499 ("Karczewicz-II")
1007	Prosecution History for U.S. Patent No. 9,432,693
1008	U.S. Patent No. 9,344,744 ("Kirchhoffer-744")
1009	Srinivasan, An Overview of VC-1
1010	U.S. Patent Application Publication No. 2003/0112864 ("Karczewicz-864")
1011	Wiegand, Overview of the H.264/AVC Video Coding Standard
1012	Richardson, The H.264 Advanced Video Compression Standard
1013	U.S. Patent Application Publication No. 2008/0198935 ("Srinivasan-935")
1014	H.264 Advanced Video Coding for Generic Audiovisual Services (March 2009)
1015	U.S. Patent Application Publication No. 2013/0034158 ("Kirchhoffer-158")
1016	U.S. Patent No. 8,594,188 ("Demos")

APPENDIX 3: CHALLENGED CLAIMS

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

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

[19c] using said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[19d] using said second reference block to obtain a second prediction, said second prediction having the second precision;

[19e] obtaining a combined prediction based at least partly upon said first prediction and said second prediction;

[19f] decreasing a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[19g] reconstructing the block of pixels based on the combined prediction.

20. The method according to claim 19, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

21. The method according to claim 20, wherein said first prediction is obtained by interpolation using values of said first reference block by:

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

22. The method according to claim 20, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

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

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

24. The method according to claim 19, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

[25a] An apparatus for decoding a block of pixels, the apparatus comprising:

at least one processor and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the apparatus to:

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

[25c] use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[25d] use said second reference block to obtain a second prediction, said second prediction having the second precision;

[25e] obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[25f] decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[25g] reconstruct the block of pixels based on the combined prediction.

26. The apparatus according to claim 25, wherein in an instance in which said first motion vector points to a subpixel, said first prediction is obtained by interpolation using pixel values of said first reference block.

27. The apparatus according to claim 26, wherein said first prediction is obtained by interpolation using values of said first reference block by:

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

28. The apparatus according to claim 26, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

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

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

30. The apparatus according to claim 25, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

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

[31b] determine, for a current block, a first reference block based on a first motion vector and a second reference block based on a second motion vector,

wherein the pixels of the current block, the first reference block, and the second reference block have values with a first precision;

[31c] use said first reference block to obtain a first prediction, said first prediction having a second precision, which is higher than said first precision;

[31d] use said second reference block to obtain a second prediction, said second prediction having the second precision;

[31e] obtain a combined prediction based at least partly upon said first prediction and said second prediction;

[31f] decrease a precision of said combined prediction by shifting bits of the combined prediction to the right; and

[31g] reconstruct the block of pixels based on the combined prediction.

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

33. The computer program product according to claim 32, wherein said first prediction is obtained by interpolation using values of said first reference block by:

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

34. The computer program product according to claim 32, wherein in an instance in which said second motion vector points to an integer sample, said second prediction is obtained by shifting values of said second reference block to the left.

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

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

36. The computer program product according to claim 31, wherein the first precision indicates a number of bits needed to represent the values of the pixels, and the second precision indicates the number of bits needed to represent values of said first prediction and values of said second prediction.

APPENDIX 4: UNDERSTANDING OF THE LAW

I have applied the following legal principles provided to me by counsel in arriving at the opinions set forth in this report.

Legal Standard for Prior Art

I am not an attorney. I have been informed by attorneys of the relevant legal principles and have applied them to arrive at the opinions set forth in this declaration.

I understand that the petitioner for inter partes review may request the cancelation of one or more claims of a patent based on grounds available under 35 U.S.C. § 102 and 35 U.S.C. § 103 using prior art that consists of patents and printed publications.

Anticipation and Prior Art

I understand that § 102 specifies when a challenged claim is invalid for lacking novelty over the prior art, and that this concept is also known as "anticipation." I understand that a prior art reference anticipates a challenged claim, and thus renders it invalid by anticipation, if all elements of the challenged claim are disclosed in the prior art reference. I understand the disclosure in the prior art reference can be either explicit or inherent, meaning it is necessarily present or implied. I understand that the prior art reference does not have to use the same words as the challenged claim, but all of the requirements of the claim must be disclosed so that a person of ordinary skill in the art could make and use the claimed subject-matter.

In addition, I understand that § 102 also defines what is available for use as a prior art reference to a challenged claim.

Under § 102(a), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country before the challenged claim's date of invention.

Under § 102(b), a challenged claim is anticipated if it was patented or described in a printed publication in the United States or a foreign country more than one year prior to the challenged patent's filing date.

Under § 102(e), a challenged claim is anticipated if it was described in a published patent application that was filed by another in the United States before the challenged claim's date of invention, or was described in a patent granted to another that was filed in the United States before the challenged claim's date of invention.

I understand that a challenged claim's date of invention is presumed to be the challenged patent's filing date. I also understand that the patent owner may establish an earlier invention date and "swear behind" prior art defined by § 102(a) or § 102(e) by proving (with corroborated evidence) the actual date on which the named inventors conceived of the subject matter of the challenged claim and proving that the inventors were diligent in reducing the subject matter to practice.

I understand that the filing date of a patent is generally the filing date of the application filed in the United States that issued as the patent. However, I understand that a patent may be granted an earlier effective filing date if the patent owner properly claimed priority to an earlier patent application.

I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

I understand that when a challenged claim requires selection of an element from a list of alternatives, the prior art teaches the element if one of the alternatives is taught by the prior art.

Legal Standard for Obviousness

I understand that even if a challenged claim is not anticipated, it is still invalid if the differences between the claimed subject matter and the prior art are such that the claimed subject matter would have been obvious to a person of ordinary skill in the pertinent art at the time the alleged invention.

I understand that obviousness must be determined with respect to the challenged claim as a whole.

I understand that one cannot rely on hindsight in deciding whether a claim is obvious.

I also understand that an obviousness analysis includes the consideration of factors such as (1) the scope and content of the prior art, (2) the differences between the prior art and the challenged claim, (3) the level of ordinary skill in the pertinent art, and (4) "secondary" or "objective" evidence of non-obviousness.

Secondary or objective evidence of non-obviousness includes evidence of: (1) a long felt but unmet need in the prior art that was satisfied by the claimed invention; (2) commercial success or the lack of commercial success of the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise of the claimed invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the claimed invention; and (7) contemporaneous and independent invention by others. However, I understand that there must be a relationship between any secondary evidence of non-obviousness and the claimed invention.

I understand that a challenged claim can be invalid for obviousness over a combination of prior art references if a reason existed (at the time of the alleged invention) that would have prompted a person of ordinary skill in the art to combine elements of the prior art in the manner required by the challenged claim. I understand that this requirement is also referred to as a "motivation to combine," "suggestion to combine," or "reason to combine," and that there are several rationales that meet this requirement.

I understand that the prior art references themselves may provide a motivation to combine, but other times simple common sense can link two or more prior art references. I further understand that obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may come from market forces.

I understand obviousness to include, for instance, scenarios where known techniques are simply applied to other devices, systems, or processes to improve them in an expected or known way. I also understand that practical and common-sense considerations should be applied in a proper obviousness analysis. For instance, familiar items may have obvious uses beyond their primary purposes.

I understand that the combination of familiar elements according to known methods is obvious when it yields predictable results. For instance, obviousness bars patentability of a predictable variation of a technique even if the technique originated in another field of endeavor. This is because design incentives and other market forces can prompt variations of it, and predictable variations are not the product of innovation, but rather ordinary skill and common sense.

I understand that a particular combination may be obvious if it was obvious to try the combination. For example, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. This would result in something obvious because the result is the product not of innovation but of ordinary skill and common sense. However, I understand that it may not be obvious to try a combination when it involves unpredictable technologies.

It is further my understanding that a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill in the art, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

It is my understanding that the Manual of Patent Examining Procedure §2143 sets forth the following as exemplary rationales that support a conclusion of obviousness:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known technique to improve similar devices (methods, or products) in the same way;

- Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;
- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

A person of ordinary skill in the art looking to overcome a problem will often use the teachings of multiple publications together like pieces of a puzzle, even though the prior art does not necessarily fit perfectly together. Therefore, I understand that references for obviousness need not fit perfectly together like puzzle pieces. Instead, I understand that obviousness analysis takes into account inferences, creative steps, common sense, and practical logic and applications that a person of ordinary skill in the art would employ under the circumstances.

I understand that a claim can be obvious in light of a single reference, if the elements of the challenged claim that are not explicitly or inherently disclosed in the reference can be supplied by the common sense of one of skill in the art.

I understand that obviousness also bars the patentability of applying known or obvious design choices to the prior art. One cannot patent merely substituting one prior art element for another if the substitution can be made with predictable results. Likewise, combining prior art techniques that are interoperable with respect to one another is generally obvious and not patentable.

In order for a claim to be found invalid based upon a modification or combination of the prior art, there must be reasonable expectation that a person of ordinary skill would have successfully modified or combined the prior art to arrive at the claimed arrangement. This does not mean that it must be certain that a person of ordinary skill would have been successful – the law only requires that the person of ordinary skill in the art would have perceived a reasonable expectation of success in modifying or combining the prior art to arrive at the claimed invention.

In sum, my understanding is that obviousness invalidates claims that merely recite combinations of, or obvious variations of, prior art teachings using understanding and knowledge of one of skill in the art at the time and motivated by the general problem facing the inventor at the time. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor at the time of the invention, can provide a reason for combining the elements of or attempting obvious variations on prior art references in the claimed manner.

<u>Legal Standard for Claim Construction</u>

I understand that before any invalidity analysis can be properly performed, the scope and meaning of the challenged claims must be determined by claim construction.

I understand that a patent may include two types of claims, independent claims and dependent claims. I understand that an independent claim stands alone and includes only the limitations it recites. I understand that a dependent claim depends from an independent claim or another dependent claim. I understand that a dependent claim includes all the limitations that it recites in addition to the limitations recited in the claim (or claims) from which it depends.

In comparing the challenged claims to the prior art, I have carefully considered the patent and its file history in light of the understanding of a person of skill at the time of the alleged invention.

I understand that to determine how a person of ordinary skill would have understood a claim term, one should look to sources available at the time of the alleged invention that show what a person of skill in the art would have understood disputed claim language to mean. It is my understanding that this may include what is called "intrinsic" evidence as well as "extrinsic" evidence.

I understand that, in construing a claim term, one should primarily rely on intrinsic patent evidence, which includes the words of the claims themselves, the remainder of the patent specification, and the prosecution history. I understand that extrinsic evidence, which is evidence external to the patent and the prosecution history, may also be useful in interpreting patent claims when the intrinsic evidence itself is insufficient. I understand that extrinsic evidence may include principles, concepts, terms, and other resources available to those of skill in the art at the time of the invention.

I understand that words or terms should be given their ordinary and accepted meaning unless it appears that the inventors were using them to mean something else or something more specific. I understand that to determine whether a term has special meaning, the claims, the patent specification, and the prosecution history are particularly important, and may show that the inventor gave a term a particular definition or intentionally disclaimed, disavowed, or surrendered claim scope.

I understand that the claims of a patent define the scope of the rights conferred by the patent. I understand that because the claims point out and distinctly claim the subject matter which the inventors regard as their invention, claim construction analysis must begin with and is focused on the claim language itself. I understand that the context of the term within the claim as well as other claims of the patent can inform the meaning of a claim term. For example, because claim terms are normally used consistently throughout the patent, how a term is used in one claim can often inform the meaning of the same term in other claims. Differences among claims or claim terms can also be a useful guide in understanding the meaning of particular claim terms.

I understand that a claim term should be construed not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the entire specification. I understand that because the specification is a primary basis for construing the claims, a correct construction must align with the specification.

I understand that the prosecution history of the patent as well as art incorporated by reference or otherwise cited during the prosecution history are also highly relevant in construing claim terms. For instance, art cited by or incorporated by reference may indicate how the inventor and others of skill in the art at the time of the invention understood certain terms and concepts. Additionally, the prosecution history may show that the inventors disclaimed or disavowed claim scope, or further explained the meaning of a claim term.

With regard to extrinsic evidence, I understand that all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises, can also be considered. For example, technical dictionaries may indicate how one of skill in the art used or understood the claim terms. However, I understand that extrinsic evidence is considered to be less reliable than intrinsic evidence, and for that reason is generally given less weight than intrinsic evidence.

I understand that in general, a term or phrase found in the introductory words or preamble of the claim, should be construed as a limitation if it recites essential structure or steps, or is necessary to give meaning to the claim. For instance, I understand preamble language may limit claim scope: (i) if dependence on a preamble phrase for antecedent basis indicates a reliance on both the preamble and claim body to define the claimed invention; (ii) if reference to the preamble is necessary to understand limitations or terms in the claim body; or (iii) if the preamble recites additional structure or steps that the specification identifies as important.

On the other hand, I understand that a preamble term or phrase is not limiting where a challenged claim defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention. I understand that to make this determination, one should review the entire patent to gain an understanding of what the inventors claim they invented and intended to encompass in the claims.

I understand that 35 U.S.C. § 112 ¶ 6 created an exception to the general rule of claim construction called a "means plus function" limitation. These types of terms and limitations should be interpreted to cover only the corresponding structure described in the specification, and equivalents thereof. I also understand that a limitation is presumed to be a means plus function limitation if (a) the claim limitation uses the phrase "means for"; (b) the "means for" is modified by functional language; and (c) the phrase "means for" is not modified by sufficient structure for achieving the specified function.

I understand that a structure is considered structurally equivalent to the corresponding structure identified in the specification only if the differences between them are insubstantial. For instance, if the structure performs the same function in substantially the same way to achieve substantially the same result. I further understand that a structural equivalent must have been available at the time of the issuance of the claim.