

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

Petitioner,

v.

NOKIA TECHNOLOGIES OY,

Patent Owner.

Case IPR2024-01141
U.S. Patent No. 8,036,273

EXHIBIT 2001

DECLARATION OF DOUGLAS A. CHRISSAN, Ph.D.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND AND QUALIFICATIONS	2
III.	BASES OF OPINIONS	4
IV.	APPLICABLE LEGAL STANDARDS	5
	A. Ordinary Skill in the Art	5
	B. Claim Construction	6
	C. Obviousness	7
V.	OVERVIEW OF THE '273 Patent	10
	A. Challenged claims 1, 4–17, 19–26, 28–29, 31–33, 36–49, 51–59, 61–62, and 64–83	16
VI.	CLAIM CONSTRUCTION OF '273 PATENT CLAIM TERMS	46
VII.	THE ASSERTED PRIOR ART	46
	A. TML6 (Ex. 1005)	46
	B. Fandrianto (Ex. 1006)	49
	C. TML6 in Combination with Fandrianto Does Not Teach or Disclose the Challenged Claims	56
	1. Claim 1[a]: “interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates 1/2, 1/2, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates 1/2, 1/2 and the nearest neighbouring pixel”	57
	2. Claims 4–17, 19–26, and 28–29	58
	3. Independent claims 31, 32, 33, 64, 68, 71, 73, 75, 76, 79, 82, and 83 and their dependent claims 36–49, 51–59, 61–62, 65–67, 69–70, 72, 74, 77–78, and 80–81	58

D.	A POSITA Would Not Have Been Motivated to Combine TML6 and Fandrianto	58
VIII.	OPINIONS.....	61

I, Douglas A. Chrissan, declare as follows:

I. INTRODUCTION

1. I have been retained as a technical expert on behalf of Patent Owner Nokia Technologies Oy (“Nokia”) for the above-captioned inter-partes review (“IPR”). I have been asked to provide my opinions on the technology disclosed in U.S. Patent No. 8,036,273 (“the ’273 Patent”) (Ex. 1001) and on the unpatentability arguments presented in Amazon’s (“Petitioner”) Petition (“Pet.”) alleging that claims 1, 4–17, 19–26, 28–29, 31–33, 36–49, 51–59, 61–62, and 64–83 of the ’273 Patent (the “Challenged Claims”) are invalid under 35 U.S.C. § 103.

2. I make this declaration based on my personal knowledge, consideration of the materials I discuss herein, and my expert opinion.

3. I am being paid for my work on this matter on an hourly basis at \$300/hour. My compensation is not contingent on reaching any particular findings or conclusions, or on any particular outcome in this matter. I have no financial interest in the outcome of this matter.

4. In preparing this Declaration, I have reviewed the ’273 Patent, its prosecution history, and considered each of the documents cited herein. I have also read and considered the IPR Petition concerning the ’273 Patent and the exhibits and evidence cited therein as grounds for invalidity, including but not limited to Exhibits 1001–24. Additional documents reviewed are discussed in Section III below.

5. In formulating my opinions, I have also relied upon my experience, education, knowledge and expertise in the area of video processing, including video compression and decompression—also referred to as video encoding and decoding. In reviewing the material relevant to these proceedings, I have considered the viewpoint of a person of ordinary skill in the art in the September 2001 timeframe.

II. BACKGROUND AND QUALIFICATIONS

6. The following sections recite the most relevant aspects of my educational, professional, and technical background. A copy of my curriculum vitae, which describes my qualifications in more detail, is attached to this report as Exhibit 2002.

7. I am presently a technical consultant in the areas of communications systems, multimedia systems, computer systems, and digital signal processing. My past work experience includes at least four years of designing, programming and directly working with semiconductors and systems for video encoding and decoding.

8. I earned a B.S. and M.S. in Electrical Engineering from the University of Southern California in 1988 and 1990, respectively, and a Ph.D. in Electrical Engineering from Stanford University in 1998.

9. From 1993 to 1997, I was a full-time doctoral student at Stanford University. My Ph.D. dissertation and related publications were in the fields of statistical signal processing and communication systems, and more specifically in

the area of impulsive noise modeling for low-frequency communication systems. In 1995, I was the instructor for the graduate Statistical Signal Processing class (EE278) in the Electrical Engineering department at Stanford University. Prior to teaching this class, I was a teaching assistant for ten different classes in signal processing and radio frequency electronics at Stanford.

10. I was a Masters Fellow and Member of the Technical Staff at Hughes Aircraft Company in El Segundo, California, from 1988–1993. While at Hughes Aircraft, I designed and developed wireless communication systems for commercial and military spacecraft, including for the MILSTAR satellite program.

11. Between 1992 and 1993, while at Hughes Aircraft Company, I designed and built a state-of-the-art, 800 megabit-per-second (Mbps) telecommunications modem for the NASA Lewis Research Center.

12. From 1997–2003, I worked at 8x8, Inc., starting as a DSP software engineer in 1997, becoming a manager in 1998, a director in 1999, and Vice President of Engineering in 2000 (managing a team of approx. 60 engineers in the company's microelectronics group). I played a key role in developing several semiconductor products used worldwide in multimedia and communications devices, mainly for video conferencing systems and Internet Protocol ("IP") telephones. Some of these semiconductor products were in production more than ten years.

13. From 2003–2007, I was a Systems Architect and Engineering Program Manager at Texas Instruments in the Digital Subscriber Line (“DSL”) product business unit. At Texas Instruments, I was directly involved in the architecture, design, development and production of multicarrier DSL modem products. My work specifically included architecting a multicarrier DSL semiconductor and software product and managing all aspects of its development from inception to production. I also gained further experience with IEEE 802.11 (WLAN/WiFi) and WiMAX systems.

14. From 2009–2011, I was an Engineering Director at Maxim Integrated Products. At Maxim, I was directly involved in the architecture, design, development and production of semiconductors and systems for video encoding and decoding.

15. I have developed, and managed the development of, several successful semiconductor, software and systems products in the communications and multimedia fields. These products are listed in the attached curriculum vitae.

III. BASES OF OPINIONS

16. In the course of performing my analysis and forming my opinions, I have reviewed the Petition for this IPR2024-01141 and its Exhibits 1001–24. Exhibits 1001–24 include:

- U.S. Patent No. 8,036,273 (“the ’273 Patent”) (Ex. 1001) and its file history (Ex. 1002)

- Declaration of Dr. Kannan Ramachandran (Ex. 1003)
- TML6 (Test Model Long 6) test source code (Ex. 1005)
- U.S. Patent No. 5,594,813 (“Fandrianto”) (Ex. 1006)

17. I have also reviewed Exhibits 2003-2009 and the Rebuttal Expert Report of Dr. Alan Bovik re Validity of U.S. Patent Nos. 7,532,808 and 6,950,469 submitted for ITC Inv. No. 337-TA-1208. However, I did not use the Bovik report to formulate my opinions herein.

IV. APPLICABLE LEGAL STANDARDS

A. Ordinary Skill in the Art

18. My opinions in this declaration are based on the understandings of a person of ordinary skill in the art—sometimes referred to as an “ordinary artisan” or by the acronyms “POSA” or “POSITA”—as of the time of the invention, which I understand to be Sept. 17, 2001, the filing date of the non-provisional application leading to the U.S. Patent No. 6,950,469. I understand that the person of ordinary skill in the art is a hypothetical person who is presumed to have known the art relevant to the Challenged Claims of the ’273 Patent at the time of the invention.

19. I understand that, in assessing the level of skill of a person of ordinary skill in the art, one should consider the type of problems encountered in the art, the prior solutions to those problems found in the prior art references, the rapidity with which innovations are made, the sophistication of the technology, the level of

education of active workers in the field, and one's own experience working with those of skill in the art at the time of the invention.

20. Petitioner submits that a POSITA at the time of the '273 Patent would have had (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. I agree that this definition of a POSITA is sufficient for this IPR proceeding.

21. I was at the time of invention, and am, one of more than ordinary skill in the art through my education, research experience and work experience. As of the date of the invention, I was familiar with the types of problems encountered in mobile communications systems and the rapidity at which innovations were made. I was and am familiar with people having this level of skill in the area of mobile communication systems.

B. Claim Construction

22. I understand that the Challenged Claims of the '273 Patent in this IPR are generally interpreted according to their ordinary and customary meaning taking into consideration the so-called "intrinsic evidence" of the patent consisting of (1) the claim language; (2) the specification; and (3) the prosecution history.

23. I understand that claim terms may be explicitly defined in the patent specification or they may be implicitly defined through consistent usage in the specification. I also understand that the scope of claim terms may be limited by statements in the specification or prosecution history where the applicant clearly disavows or disclaims subject matter.

24. I understand that claim construction is a matter of law. As of this time, my understanding is that no claim terms have been proposed for construction.

C. Obviousness

25. Counsel has informed me, and I understand, that an issued patent claim is invalid as obvious if it can be shown that the differences between the patented subject matter and the prior art are such that the subject matter as a whole would have been obvious, at the time the invention was made, to a person having ordinary skill in the art. Relevant considerations include the level of ordinary skill in the art; the scope and content of the prior art; differences between the prior art and the claims at issue; and the objective indicia, or secondary considerations, of nonobviousness.

26. In this regard, I am informed and I understand that the secondary considerations of nonobviousness could include factors such as commercial success, long felt but unresolved needs, the failure of others, skepticism by experts, praise by others, award and accolades, and copying.

27. Counsel has informed me, and I understand, that in order to evaluate

the obviousness of any claim of the '273 Patent over a given prior art combination, I should analyze whether the prior art references, included collectively in the combination, disclose each and every element of the allegedly invalid claim as those references are read by the person of ordinary skill in the art at the time of the invention.

28. Counsel has informed me, and I understand, that even where all of the claim limitations are expressly disclosed in the prior art references, there must be some showing that a person of ordinary skill in the art would have been motivated to combine such prior art references, and explanation of how the combination would be achieved, and that there would have been a reasonable expectation of successfully achieving the claimed invention from such combination.

29. Counsel has informed me, and I understand, that absent some additional reasoning, mere routine optimization for a person of ordinary skill in the art to arrive at the claimed invention through combination of prior art references is insufficient to support a conclusion of obviousness.

30. Counsel has informed me, and I understand, that to rely on a reference it must be analogous prior art. Counsel has informed me and I understand that a reference is considered analogous prior art: (1) if the reference is from the same field of endeavor as the claimed subject matter, or (2) if the reference “is reasonably pertinent to the particular problem with which the inventor is involved,” even though

the reference is not within the field of the inventor's endeavor.

31. Counsel has informed me, and I understand, in considering the obviousness of a claimed invention, one should not view the invention and the prior art with the benefit of hindsight. In an obviousness analysis, one must be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. Counsel has instructed me and I understand that when considering obviousness, I should not consider what is known today or what was learned from the asserted patents. Instead, obviousness is assessed by the person of ordinary skill in the art at the time the invention was made.

32. In this regard, I am informed and I understand that the '273 Patent may not be used as a guide to select and understand the prior art, because such an approach, at a minimum, discounts the value of combining various existing features or principles in a new way so as to achieve a new result.

33. Counsel has informed me, and I understand, that obviousness cannot be predicated on what was unknown at the time of the invention, even if the inherency of a certain feature is later established. Counsel has also informed me, and I understand, that unknown properties of the prior art may not be relied upon to provide the rationale for modifying or combining the prior art to reach the claimed subject matter.

34. Counsel has informed me, and I understand, that a reference may be

said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by applicant of the '273 Patent.

V. OVERVIEW OF THE '273 Patent

35. The '273 Patent provides a detailed “Background of the Invention” section, which I reproduce or paraphrase for my overview description herein. *See* Ex. 1001, 1:18–13:27.

36. Consecutive frames in a video sequence tend to be similar and to contain a considerable amount of redundant information. For example, a typical scene may comprise some stationary elements, such as background scenery, and some moving areas, which may take many different forms. Alternatively, the camera recording the scene may itself be moving, in which case all elements of the image have the same kind of motion. In many cases, this means that the overall change between one video frame and the next is often small. *See id.*, 1:21–35.

37. Each frame of a raw (uncompressed) digital video sequence comprises a very large amount of image information. Each frame of an uncompressed digital video sequence is formed from an array of image pixels. In turn, each pixel is represented by a certain number of bits, which carry information about the luminance and/or color content of the region of the image corresponding to the pixel. Commonly, a so-called YUV color model is used to represent the luminance and

chrominance content of the image. The luminance, or Y, component represents the intensity (brightness) of the image, while the color content of the image is represented by two chrominance components, labeled U and V. *See id.*, 1:41–56.

38. Chrominance components often have a lower spatial resolution than corresponding luminance components. The lower spatial resolution of the chrominance components is usually attained by sub-sampling, e.g. of an input video that originally has a chrominance resolution that is the same as the luminance resolution. Typically, a block of 16×16 image pixels is represented by one block of 16×16 pixels comprising luminance information and the corresponding chrominance components are each represented by one block of 8×8 pixels representing an area of the image equivalent to that of the 16×16 pixels of the luminance component. The chrominance components are thus spatially sub-sampled by a factor of 2 in the x (horizontal) and y (vertical) directions. The resulting assembly of one 16×16 pixel luminance block and two 8×8 pixel chrominance blocks is commonly referred to as a YUV macroblock, or macroblock, for short. *See id.*, 2:1-12.

39. A significant reduction in the amount of information used to represent video data must be achieved in order to enable transmission of digital video sequences over low bandwidth communication networks. For this reason, video compression techniques have been developed which reduce the amount of information transmitted while retaining an acceptable image quality. *See id.*, 2:35–

41.

40. Video compression methods are based on reducing the redundant and perceptually irrelevant parts of video sequences. The redundancy in video sequences can be categorized into spatial, temporal and spectral redundancy. “Spatial redundancy” is the term used to describe the correlation between neighboring pixels within a frame. The term “temporal redundancy” expresses the fact that the objects appearing in one frame of a sequence are likely to appear in subsequent frames, while “spectral redundancy” refers to the correlation between different color components of the same image. *See id.*, 2:42–52.

41. Video compression standards¹ at the time of the '273 Patent, such as ITU-T recommendations H.261, H.263(+)(++), H.26L and the Motion Picture Experts Group recommendation MPEG-4 made use of “motion compensated temporal prediction.” This is a form of temporal redundancy reduction in which the content of some (often many) frames in a video sequence is “predicted” from other frames in the sequence by tracing the motion of objects or regions of an image between frames. *See id.*, 2:62–3:2.

¹ The '273 patent uses the term “video compression standards” (*see, e.g.*, Ex. 1001, 2:62). A POSITA understands that a video compression standard or a video codec (i.e., coding and decoding) standard in fact typically only specifies the decoding operation and the encoded bitstream format, thereby leaving significant freedom in how any given encoder may be implemented.

42. Compressed images which do not make use of temporal redundancy reduction are usually called INTRA-coded or I-frames, whereas temporally predicted images are called INTER-coded or P-frames. In the case of INTER frames, the predicted (motion-compensated) image is rarely precise enough to represent the image content with sufficient quality, and therefore a spatially compressed prediction error (PE) frame is also associated with each INTER frame. *See id.*, 3:3–10.

43. The operating principle of video coders using motion compensation is to minimize the amount of information in a prediction error frame, which is the difference between a current frame being coded and a prediction frame. *See id.*, 3:40–45.

44. The prediction frame (denoted $P_n(x,y)$) is built using pixel values of a reference frame (denoted $R_n(x,y)$), which is generally one of the previously coded and transmitted frames, for example the frame immediately preceding the current frame, and is available from the frame memory. More specifically, the prediction frame $P_n(x,y)$ is constructed by finding so-called ‘prediction pixels’ in the reference frame $R_n(x,y)$ which correspond substantially with pixels in the current frame. Motion information, describing the relationship (e.g. relative location, rotation, scale etc.) between pixels in the current frame and their corresponding prediction pixels in the reference frame is derived and the prediction frame is constructed by moving

the prediction pixels according to the motion information. In this way, the prediction frame is constructed as an approximate representation of the current frame, using pixel values in the reference frame. The prediction error frame referred to above therefore represents the difference between the approximate representation of the current frame provided by the prediction frame and the current frame itself. *See id.*, 3:47–66.

45. The basic advantage provided by video encoders that use motion compensated prediction arises from the fact that a comparatively compact description of the current frame can be obtained by representing it in terms of the motion information required to form its prediction together with the associated prediction error information in the prediction error frame. *See id.*, 3:66–4:5.

46. The motion information usually takes the form of motion vectors $[\Delta x(x,y), \Delta y(x,y)]$. The pair of numbers $\Delta x(x,y)$ and $\Delta y(x,y)$ represents the horizontal and vertical displacements of a pixel at location (x,y) in the current frame $I_n(x,y)$ with respect to a pixel in the reference frame $R_n(x,y)$. *See id.*, 4:18–22.

47. Typically, the location of a macroblock in a current video frame is specified by the (x,y) co-ordinate of its upper left-hand corner. Thus, in a video coding scheme in which motion information is associated with each macroblock of a frame, each motion vector describes the horizontal and vertical displacement $\Delta x(x,y)$ and $\Delta y(x,y)$ of a pixel representing the upper left-hand corner of a

macroblock in the current frame $I_n(x,y)$ with respect to a pixel in the upper left-hand corner of a substantially corresponding block of prediction pixels in the reference frame $R_n(x,y)$. *See id.*, 4:27–36, Fig. 4b.

48. Modeling of motion between video frames by using only full pixel resolution for motion vectors is not sufficiently accurate to allow efficient minimization of the prediction error (PE) information associated with each macroblock/frame. For example, if the motion from one frame to the next moves 2.5 pixels to the right, but the choice of the motion vector horizontal component is limited to full pixel resolution, then the closest possible choices for the motion vector would be either 2 pixels or 3 pixels, and the accuracy of the motion modeling would correspondingly be limited. Therefore, to enable more accurate modelling of real motion and to help reduce the amount of prediction error information that must be transmitted from encoder to decoder, many video coding standards, such as H.263(+)(++) and H.26L, allow motion vectors to point “in between” image pixels. In other words, the motion vectors can have ‘sub-pixel’ resolution. Allowing motion vectors to have sub-pixel resolution adds to the complexity of the encoding and decoding operations that must be performed, so it is still advantageous to limit the degree of spatial resolution a motion vector may have. Thus, video compression standards, such as those previously mentioned, typically only allow motion vectors to have full-, half- or quarter-pixel resolution. *See id.*, 6:20–47.

49. Sub-pixel values do not in fact exist, and must be estimated. This estimation must take place in both the encoder and decoder, and should be performed in the same way in both in order for the encode and decoder to remain synchronized. In practice, the estimation of a sub-pixel value in the reference frame is performed by interpolating the value of the sub-pixel from surrounding pixel values.

50. The '273 Patent, entitled "Method for sub-pixel value interpolation" and issued to Marta Karczewicz and Antti Olli Hallapuro, claims novel ways to perform sub-pixel interpolation.

A. Challenged claims 1, 4–17, 19–26, 28–29, 31–33, 36–49, 51–59, 61–62, and 64–83

51. Challenged claims 1, 4–17, 19–26, 28–29, 31–33, 36–49, 51–59, 61–62, and 64–83 recite as follows, using the indexing of the Petition:

Claim 1[pre]: A method for sub-pixel value interpolation to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the method comprising causing an apparatus to:

1[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

1[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

1[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of subpixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 4: A method according to claim 1, comprising causing the apparatus to use a first and a second weight in the weighted average chosen to interpolate the sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, wherein when the chosen weighted average is that involving the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L,

causing the apparatus to select the relative magnitudes of the first and second weights to be inversely proportional to the respective straight-line diagonal distances of the sub-pixels having co-ordinates with even values of K and L to the sub-pixel having co-ordinates with odd values of K and L whose value is being interpolated.

Claim 5: A method according to claim 4, comprising causing the apparatus to use first and second weights with equal values when the sub-pixels having co-ordinates with even values of K and L are equidistant from the sub-pixel having co-ordinates with odd values of K and L whose value is being interpolated.

Claim 6: A method according to claim 1, comprising causing the apparatus to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions when the value of a sub-pixel having co-ordinates with K equal to an even value and L equal to an odd value is also required.

Claim 7: A method according to claim 1, comprising causing the apparatus to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions when the value of a sub-pixel having co-ordinates with K equal to an odd value and L equal to an even value is also required.

Claim 8: A method according to claim 1, comprising causing the apparatus to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an odd value O and L equal to an even value E, not including zero or 2^N , by taking an average of the value of a first sub-pixel having co-ordinates with K equal to the even value O-1 and L equal to E and a second sub-pixel having co-ordinates with K equal to the even value O+1 and L equal to E.

Claim 9: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 1 and L equal to zero, by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2 and L equal to zero.

Claim 10: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N-1 and L equal to zero, by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2^N-2 and L equal to zero.

Claim 11: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 1 and L equal to 2^N , by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2 and L equal to 2^N .

Claim 12: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N-1 and L equal to 2^N , by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2^N-2 and L equal to 2^N .

Claim 13: A method according to claim 1, comprising causing the apparatus to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value E, not including zero or 2^N , and L equal to an odd value O, by taking an average of the value of a first sub-pixel having co-ordinates with K equal to E and L equal to the even value O-1 and a second sub-pixel having co-ordinates with K equal to E and L equal to the even value O+1.

Claim 14: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to zero and L equal to 1, by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to zero, and a sub-pixel having co-ordinates with K equal to zero and L equal to 2.

Claim 15: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N and L equal to 1, by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2.

Claim 16: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to zero and L equal to 2^N-1 , by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to zero and L equal to 2^N-2 .

Claim 17: A method according to claim 1, comprising causing the apparatus to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2^N-1 , by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L

equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2^{N-2} .

Claim 19: A method according to claim 1, in which N is one of the values 2, 3, and 4.

Claim 20: A method according to claim 1, wherein the pixels have a predetermined dynamic range, said predetermined dynamic range corresponding to the range of values said pixels can take.

Claim 21: A method according to claim 20, comprising interpolating sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and for sub-pixels having co-ordinates with K equal to zero and L equal to an even value, using weighted sums of the values of pixels located in rows and columns respectively, to produce a first intermediate value, said first intermediate value having a dynamic range equal to the predetermined dynamic range of the pixels used in said weighted sum multiplied by a value equal to the sum of the respective weights used in said weighted sum.

Claim 22: A method according to claim 21, wherein the dynamic range of said first intermediate value is defined by the number of bits used to represent the respective pixels used in said weighted sum plus the number of bits required to represent the sum of the respective weights used in said weighted sum

Claim 23: A method according to claim 21, comprising truncating the first intermediate value by mathematically dividing the first intermediate value by a first scale factor, said first scale factor having a value equal to the sum of the respective weights used in said weighted sum, thereby forming a sub-pixel value with a dynamic range equal to the predetermined dynamic range of said pixels.

Claim 24: A method according to claim 23, comprising using a truncated first intermediate sub-pixel value when interpolating a value for a sub-pixel having co-ordinates with odd values of K and L.

Claim 25: A method according to claim 21, comprising interpolating sub-pixel values for sub-pixels having co-ordinates with even values of K and L, using a weighted sum comprising first intermediate values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and corresponding first intermediate values for sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, to produce a second intermediate value, said second intermediate value having a dynamic range equal to the dynamic range of the first intermediate values used in said weighted sum, multiplied by a value equal to the sum of the respective weights used in said weighted sum.

Claim 26: A method according to claim 25, comprising truncating the second intermediate value by mathematically dividing the second intermediate value by a second scale factor, said second scale factor having a value equal to the sum of the respective weights used in the weighted sum of claim 21 multiplied by the sum of the respective weights used in the weighted sum of claim 25, thereby forming a sub-pixel value with a dynamic range equal to the predetermined dynamic range of said pixels.

Claim 28: A method according to claim 21, comprising interpolating sub-pixel values for sub-pixels having co-ordinates with even values of K and L, using a weighted sum comprising first intermediate values for sub-pixels having co-ordinates with K equal to zero and L equal to an even value and corresponding first intermediate values for sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions to produce a second intermediate value, said second intermediate value having a dynamic range equal to the dynamic range of the first intermediate values used in said weighted sum, multiplied

by a value equal to the sum of the respective weights used in said weighted sum.

Claim 29: A method according to claim 28, comprising truncating the second intermediate value by mathematically dividing the second intermediate value by a second scale factor, said second scale factor having a value equal to the sum of the respective weights used in the weighted sum of claim 21 multiplied by the sum of the respective weights used in the weighted sum of claim 28, thereby forming a sub-pixel value with a dynamic range equal to the dynamic range of said pixels.

Claim 31[pre]: A method for quarter resolution sub-pixel value interpolation to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/4$, $L/4$, K and L being positive integers having respective values between zero and 4, the method comprising causing an apparatus to:

Claim 31[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with K and L equal to 1 or 3, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $2/4$, $2/4$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with K and L equal to zero, 2 or 4, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $2/4$, $2/4$ and the nearest neighbouring pixel;

Claim 31[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to 2 and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to 2, used in the

interpolation of the sub-pixels having co-ordinates with K and L equal to 1 or 3, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 31[c]: interpolate a sub-pixel value for the sub-pixel having co-ordinates with K and L equal to 2, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with K and L equal to 1 or 3, using a predetermined choice of either a weighted sum of the values of the sub-pixel having co-ordinates with K equal to 2 and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the value of the sub-pixel having co-ordinates with K equal to zero and L equal to 2 and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 32[pre]: A method for eighth resolution sub-pixel value interpolation to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically with the rectangular bounded region using the co-ordinate notation $K/8$, $L/8$, K and L being positive integers having respective values between zero and 8, the method comprising causing an apparatus to:

Claim 32[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with K and L equal to 1, 3, 5 or 7, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $4/8$, $4/8$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with K and L equal to zero, 2, 4, 6 or 8, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $4/8$, $4/8$ and the nearest neighbouring pixel;

Claim 32[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to 2, 4 or 6 and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to 2, 4 or 6, used in the interpolation of the sub-pixels having co-ordinates with K and L equal to 1, 3, 5 or 7, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 32[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with K and L equal to 2, 4 or 6, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with K and L equal to 1, 3, 5 or 7, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to 2, 4 or 6 and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to 2, 4 or 6 and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 33[pre]: An interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2N$, $L/2N$, K and L being positive integers having respective values between zero and $2N$, N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 33[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-

ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 33[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 33[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 36: An interpolator according to claim 33, wherein the interpolator is configured to use a first and a second weight in the weighted average chosen to interpolate the sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, wherein when the chosen weighted average is that involving the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, the interpolator is configured to select the relative magnitudes of the first and second weights to be inversely proportional to the respective straight-line diagonal distances of the sub-pixels having co-

ordinates with even values of K and L to the sub-pixel having co-ordinates with odd values of K and L whose value is being interpolated.

Claim 37: An interpolator according to claim 36, wherein the interpolator is configured to use first and second weights with equal values when the sub-pixels having co-ordinates with even values of K and L are equidistant from the sub-pixel having co-ordinates with odd values of K and L whose value is being interpolated.

Claim 38: An interpolator according to claim 33, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions when the value of a sub-pixel having co-ordinates with K equal to an even value and L equal to an odd value is also required.

Claim 39: An interpolator according to claim 33, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions when the value of a sub-pixel having co-ordinates with K equal to an odd value and L equal to an even value is also required.

Claim 40: An interpolator according to claim 33, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an odd value O and L equal to an even value E, not including zero or 2^N , by taking an average of the value

of a first sub-pixel having co-ordinates with K equal to the even value $O-1$ and L equal to E and a second sub-pixel having co-ordinates with K equal to the even value $O+1$ and L equal to E.

Claim 41: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 1 and L equal to zero, by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2 and L equal to zero.

Claim 42: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N-1 and L equal to zero, by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2^N-2 and L equal to zero.

Claim 43: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 1 and L equal to 2^N , by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2 and L equal to 2^N .

Claim 44: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N-1 and L equal to 2^N , by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2^N-2 and L equal to 2^N .

Claim 45: An interpolator according to claim 33, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value E, not including zero

or 2^N , and L equal to an odd value O, by taking an average of the value of a first sub-pixel having co-ordinates with K equal to E and L equal to the even value O-1 and a second sub-pixel having co-ordinates with K equal to E and L equal to the even value O+1.

Claim 46: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to zero and L equal to 1, by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to zero, and a sub-pixel having co-ordinates with K equal to zero and L equal to 2.

Claim 47: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N and L equal to 1, by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to zero, and a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2.

Claim 48: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to zero and L equal to 2^N-1 , by taking an average of the value of a pixel having co-ordinates with K equal to zero and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to zero and L equal to 2^N-2

Claim 49: An interpolator according to claim 33, wherein the interpolator is configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2^N-1 , by taking an average of the value of a pixel having co-ordinates with K equal to 2^N and L equal to 2^N , and a sub-pixel having co-ordinates with K equal to 2^N and L equal to 2^N-2 .

Claim 51: An interpolator according to claim 33, wherein N is set equal to 2.

Claim 52: An interpolator according to claim 33, wherein N is set equal to 3.

Claim 53: An interpolator according to claim 33, wherein the pixels have a predetermined dynamic range, said predetermined dynamic range corresponding to the range of values said pixels can take.

Claim 54: An interpolator according to claim 53, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and for sub-pixels having co-ordinates with K equal to zero and L equal to an even value, using weighted sums of the values of pixels located in rows and columns respectively, to produce a first intermediate value, said first intermediate value having a dynamic range equal to the predetermined dynamic range of the pixels used in said weighted sum multiplied by a value equal to the sum of the respective weights used in said weighted sum.

Claim 55: An interpolator according to claim 54, wherein the dynamic range of said first intermediate value is defined by the number of bits used to represent the respective pixels used in said weighted sum plus the number of bits required to represent the sum of the respective weights used in said weighted sum.

Claim 56: An interpolator according to claim 54, wherein the interpolator is configured to truncate the first intermediate value by mathematically dividing the first intermediate value by a first scale factor, said first scale factor having a value equal to the sum of the respective weights used in said weighted sum, thereby forming a sub-pixel value with a dynamic range equal to the predetermined dynamic range of said pixels.

Claim 57: An interpolator according to claim 56, wherein the interpolator is configured to use a truncated first intermediate sub-pixel

value when interpolating a value for a sub-pixel having co-ordinates with odd values of K and L.

Claim 58: An interpolator according to claim 54, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, using a weighted sum comprising first intermediate values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and corresponding first intermediate values for sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, to produce a second intermediate value, said second intermediate value having a dynamic range equal to the dynamic range of the first intermediate values used in said weighted sum, multiplied by a value equal to the sum of the respective weights used in said weighted sum.

Claim 59: An interpolator according to claim 58, wherein the interpolator is configured to truncate the second intermediate value by mathematically dividing the second intermediate value by a second scale factor, said second scale factor having a value equal to the sum of the respective weights used in the weighted sum of claim 54 multiplied by the sum of the respective weights used in the weighted sum of claim 58, thereby forming a sub-pixel value with a dynamic range equal to the predetermined dynamic range of said pixels.

Claim 61: An interpolator according to claim 54, wherein the interpolator is configured to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, using a weighted sum comprising first intermediate values for sub-pixels having co-ordinates with K equal to zero and L equal to an even value and corresponding first intermediate values for sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions to produce a second intermediate value, said second intermediate value having a dynamic range equal to the dynamic range of the first intermediate values used in said weighted sum, multiplied by a value equal to the sum of the respective weights used in said weighted sum.

Claim 62: An interpolator according to claim 61, wherein the interpolator is configured to truncate the second intermediate value by mathematically dividing the second intermediate value by a second scale factor, said second scale factor having a value equal to the sum of the respective weights used in the weighted sum of claim 54 multiplied by the sum of the respective weights used in the weighted sum of claim 61, thereby forming a sub-pixel value with a dynamic range equal to the predetermined dynamic range of said pixels.

Claim 64[pre]: A video encoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 64[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L , according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L , including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 64[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even

value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 64[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 65[pre]: A codec comprising a video encoder according to claim 64 and a video decoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 65[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of

K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 65[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 65[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 66: A communications terminal comprising a video encoder according to claim 64.

Claim 67: A communications terminal according to claim 66, wherein the communications terminal is a wireless terminal.

Claim 68[pre]: A still image encoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows

and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 68[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L , according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L , including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 68[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L , using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 68[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L , used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L , using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 69[pre]: A codec comprising a still image encoder according to claim 68 and a still image decoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 69[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L , according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L , including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 69[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L , using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 69[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L , used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L , using a predetermined choice of either a weighted sum of the

values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 70: A communications terminal comprising a still image encoder according to claim 68.

Claim 71[pre]: A video decoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 71[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 71[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-

pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 71[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 72: A communications terminal comprising a video decoder according to claim 71.

Claim 73[pre]: A still image decoder comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 73[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined

choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 73[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 73[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 74: A communications terminal comprising a still image decoder according to claim 73.

Claim 75[pre]: An interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and

sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator comprising:

Claim 75[a]: circuitry configured to interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L , according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L , including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 75[b]: circuitry configured to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L , using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 75[c]: circuitry configured to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L , used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L , using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 76[pre]: An interpolator for quarter resolution sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, region using the co-ordinate notation $K/4$, $L/4$, K and L being positive integers having respective values between zero and 4, the interpolator being configured to:

Claim 76[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with K and L equal to 1 or 3, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $2/4$, $2/4$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with K and L equal to zero, 2 or 4, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $2/4$, $2/4$ and the nearest neighbouring pixel;

Claim 76[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to 2 and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to 2, used in the interpolation of the sub-pixels having co-ordinates with K and L equal to 1 or 3, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 76[c]: interpolate a sub-pixel value for the sub-pixel having co-ordinates with K and L equal to 2, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with K and L equal to 1 or 3, using a predetermined choice of either a weighted sum of the values of the sub-pixel having co-ordinates with K equal to 2 and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the value of the sub-pixel having co-ordinates with K

equal to zero and L equal to 2 and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 77: A video encoder or still image encoder comprising an interpolator according to claim 76.

Claim 78: A communications terminal comprising a video encoder or still image encoder according to claim 77.

Claim 79[pre]: An interpolator for eighth resolution sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically with the rectangular bounded region using the co-ordinate notation $K/8, L/8$, K and L being positive integers having respective values between zero and 8, the interpolator being configured to:

Claim 79[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with K and L equal to 1, 3, 5 or 7, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $4/8, 4/8$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with K and L equal to zero, 2, 4, 6 or 8, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $4/8, 4/8$ and the nearest neighbouring pixel;

Claim 79[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to 2, 4 or 6 and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to 2, 4 or 6, used in the interpolation of the sub-pixels having co-ordinates with K and L

equal to 1, 3, 5 or 7, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 79[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with K and L equal to 2, 4 or 6, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with K and L equal to 1, 3, 5 or 7, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to 2, 4 or 6 and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to 2, 4 or 6 and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 80: A video decoder or still image decoder comprising an interpolator according to claim 79.

Claim 81: A communications terminal comprising a video decoder or still image decoder according to claim 80.

Claim 82[pre]: A computer program for sub-pixel value interpolation, embodied on a computer readable storage medium, the computer program being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the computer program comprising:

Claim 82[a]: computer program code for interpolating a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 82[b]: computer program code for interpolating sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L, using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 82[c]: computer program code for interpolating sub-pixel values for sub-pixels having co-ordinates with even values of K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L, using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

Claim 83[pre]: A network apparatus comprising an interpolator for sub-pixel value interpolation, the interpolator being configured to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being configured for display in rows and columns, pixel and sub-pixel locations in the rows and columns

being representable mathematically within the rectangular bounded region using the co-ordinate notation $K/2^N$, $L/2^N$, K and L being positive integers having respective values between zero and 2^N , N being a positive integer greater than one and representing a particular degree of sub-pixel value interpolation, the interpolator being configured to:

Claim 83[a]: interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L , according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $\frac{1}{2}$, $\frac{1}{2}$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L , including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $\frac{1}{2}$, $\frac{1}{2}$ and the nearest neighbouring pixel;

Claim 83[b]: interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal to zero and sub-pixels having co-ordinates with K equal to zero and L equal to an even value, used in the interpolation of the sub-pixels having co-ordinates with odd values of K and L , using weighted sums of the values of pixels located in rows and columns respectively; and

Claim 83[c]: interpolate sub-pixel values for sub-pixels having co-ordinates with even values of K and L , used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of K and L , using a predetermined choice of either a weighted sum of the values of sub-pixels having co-ordinates with K equal to an even value and L equal to zero and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions, or a weighted sum of the values of sub-pixels having co-ordinates with K equal to zero and L equal to an even value and the values of sub-pixels having corresponding co-ordinates in immediately adjacent rectangular bounded regions.

VI. CLAIM CONSTRUCTION OF '273 PATENT CLAIM TERMS

52. My understanding is that no claim terms of the '273 Patent have been proposed for construction by Petitioner. It is also my understanding that no claim terms of the '273 Patent have been construed in any prior legal proceeding.

VII. THE ASSERTED PRIOR ART

A. TML6 (Ex. 1005)

53. Based on my understanding of the Petition and its exhibits, there is no dispute that the description of TML6 in the '273 Patent is accurate. Therefore, I reproduce and cite that description.

54. The '273 Patent describes “Two interpolation schemes have been developed as part of the work ongoing in the ITU-Telecommunications Standardization Sector, Study Group 16, Video Coding Experts Group (VCEG), Questions 6 and 15. These approaches were proposed for incorporation into ITU-T recommendation H.26L and have been implemented in test models (TML) for the purposes of evaluation and further development. The test model corresponding to Question 15 is referred to as Test Model 5 (TML5), while that resulting from Question 6 is known as Test Model 6 (TML6).” Ex. 1001, 8:38–47. The TML6 prior art reference in this IPR lists the software of Test Model 6. Ex. 1005. The '273 Patent further describes:

The sub-pixel value interpolation scheme according to TML6, referred to herein as direct interpolation, will now be described. 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. A step-by-step description of interpolation method according to TML6 as applied in the encoder is given below with reference to FIGS. 13 *a*, 13 *b* and 13 *c*.

1. The value for the sub-pixel at half unit horizontal and unit vertical location, that is $\frac{1}{2}$ resolution sub-pixel *b* in FIG. 13 *a*, is obtained by first calculating an intermediate value *b* using a six tap filter. The filter calculates *b* based upon the values of the 6 pixels (A_1 to A_6) situated in a row at unit horizontal locations and unit vertical locations symmetrically about *b*, as shown in FIG. 13 *b*, according to the formula $b=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$. The final value of *b* is then calculated as $b=(b+16)/32$ and is clipped to lie in the range $[0, 2^n-1]$. As before, the operator / denotes division with truncation.

2. Values for the $\frac{1}{2}$ resolution sub-pixels labelled *c* are obtained by first calculating intermediate values *c*. Referring to FIG. 13 *c*, an intermediate value *c* for the $\frac{1}{2}$ resolution sub-pixel *c* located at unit horizontal and half vertical location is calculated based upon the values of the 6 pixels (A_1 to A_6) situated in a column at unit horizontal locations and unit vertical locations symmetrically about *c*, according to the formula $c=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$. The final value for the $\frac{1}{2}$ resolution sub-pixel *c* located at unit horizontal and half vertical location is then calculated according to $c=(c+16)/32$. Similarly, an intermediate value *c* for the $\frac{1}{2}$ resolution sub-pixel *c* at half horizontal and half vertical location is calculated according to $c=(b_1-5b_2+20b_3+20b_4-5b_5+b_6)$. A final value for this $\frac{1}{2}$ resolution sub-pixel is then calculated according to $(c+512)/1024$. Again, the operator / denotes division with truncation and the values calculated for $\frac{1}{2}$ resolution sub-pixels *c* are further clipped to lie in the range $[0, 2^n-1]$.

3. Values for the $\frac{1}{4}$ resolution sub-pixels labelled *d* are calculated as follows. Values for $\frac{1}{4}$ resolution sub-pixels *d*

located at quarter horizontal and unit vertical locations, are calculated from the value of the immediately neighbouring pixel at unit horizontal and unit vertical location (pixel A) and the intermediate value b calculated in step (1) for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixel at half horizontal and unit vertical location ($\frac{1}{2}$ resolution sub-pixel b), according to $d=(32A+b+32)/64$. Values for $\frac{1}{4}$ resolution sub-pixels d located at quarter horizontal and half vertical locations, are interpolated using the intermediate values c calculated for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixels c which lie at unit horizontal and half vertical location and half horizontal and half vertical locations respectively, according to $d=(32c_1+c_2+1024)/2048$. Again operator / indicates division with truncation and the finally obtained $\frac{1}{4}$ resolution sub-pixel values d are clipped to lie in the range $[0, 2^n-1]$.

4. Values for the $\frac{1}{4}$ resolution sub-pixels labelled e are calculated as follows. Values for $\frac{1}{4}$ resolution sub-pixels e located at unit horizontal and quarter vertical locations are calculated from the value of the immediately neighbouring pixel at unit horizontal and unit vertical location (pixel A) and the intermediate value c calculated in step (2) for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixel at unit horizontal and half vertical location, according to $e=(32A+c+32)/64$. Values for $\frac{1}{4}$ resolution sub-pixels e located at half horizontal and quarter vertical locations are calculated from the intermediate value b calculated in step (1) for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixel at half horizontal and unit vertical location and the intermediate value c calculated in step (2) for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixel at half horizontal and half vertical location, according to $e=(32b+c+1024)/2048$. Once more, operator / indicates division with truncation and the finally obtained $\frac{1}{4}$ resolution sub-pixel values e are clipped to lie in the range $[0, 2^n-1]$.

5. Values for $\frac{1}{4}$ resolution sub-pixels labelled g are computed using the value of the nearest original pixel A and the intermediate values of the three nearest neighbouring $\frac{1}{2}$ resolution sub-pixels, according to

$g = (1024A + 32b + 32c_1 + c_2 + 2048) / 4096$. As before, operator / indicates division with truncation and the finally obtained for $\frac{1}{4}$ resolution sub-pixel values g are clipped to lie in the range $[0, 2^n - 1]$.

6. The value for $\frac{1}{4}$ resolution sub-pixel f is interpolated by averaging the values of the 4 closest pixels at unit horizontal and vertical locations, according to $f = (A_1 + A_2 + A_3 + A_4 + 2) / 4$, where the locations of pixels A_1 , A_2 , A_3 and A_4 are the four nearest original pixels.

Ex. 1001, 11:25–12:45. The '273 Patent describes the drawbacks of the TML6 scheme, e.g., that high precision arithmetic is required both in the encoder and in the decoder. High precision interpolation requires more silicon area in ASICs and requires more computations in some CPUs. Furthermore, implementation of direct interpolation as specified in TML6 in an on-demand fashion has a high memory requirement. *Id.*, 13:11–18.

B. Fandrianto (Ex. 1006)

55. U.S. Patent 5,594,813 to Fandrianto et al. has a priority date of February 19, 1992, the filing date of its non-provisional application number 838,380. It issued on January 14, 1997. As such, it is much older than TML6 and the video standard resulting after the TML6 effort (H.264). Fandrianto is primarily a hardware architecture patent for an application specific integrated circuit (ASIC) that accelerates video encoding and decoding. As such, it is not a patent about video encoding and decoding algorithms. To the extent it discusses video encoding and decoding algorithms, it does so in the context of how certain video encoding and

decoding operations would execute on its architecture. Even given that architecture's innovative design, it is still severely memory constrained, primarily due to the technology limitations of integrated circuits in 1992.

56. Fandrianto very specifically describes how quarter-pixel interpolation would be performed, using its Figure 5 (reproduced below) as a reference:

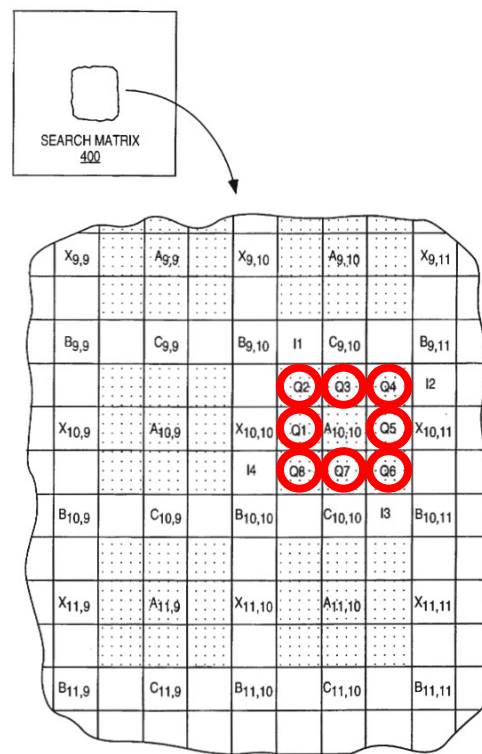


FIG. 5

Ex. 1006, Fig. 5 (annotated with red circles to show pixels Q1–Q8 that need to be quarter-pixel interpolated). Fandrianto explains:

First, the best match block is moved from A block 72 of the DPCM memory 130 into the D block 82 of the DP memory 124. In conformance with FIG. 5, however, the pixels of the best match block now stored in the D block 82 are referred to as $A_{r,c}$. This frees up the A block 72 to hold the current quarter pixel

interpolated block, the pixels of which for convenience are referred to as $Q_{r,c}$.

The first quarter pixel estimation search block of Q1 pixels is generated from a horizontal interpolation of the data in the X block 70 and the D block 82, and stored in the A block 72 for the absolute displaced block difference calculation, in accordance with the following expression.

$$Q1_{r,c} = (X_{r,c} + A_{r,c})/2 \quad (3)$$

Consider, for example, the interpolation of the Q1 pixel to the left of pixel $A_{10,10}$. The pixel group $A_{10,8} - A_{10,15}$ in row ten of the D block 82 is addressed on, say, port A of the DP memory 124 and presented through the multiplexer 152 to the A port of the ALU 154. At about the same time, a collection of pixels containing the pixel group $X_{10,8} - X_{10,15}$ in row ten of the X block 70 is addressed on, say, port A of the DPCM memory 130 and the pixel group $X_{10,8} - X_{10,15}$ is selected by shifter 140 and presented through the multiplexer 152 to the B port of the ALU 154. The ALU 154 sums the pixel groups, divides by two, and rounds the result. An eight pixel result at the output of the ALU 154 is routed back to the DPCM memory 130, where it is stored as pixels $Q1_{10,8} - Q1_{10,15}$ in the tenth row of the A block 72.

The second quarter pixel estimation search block of Q2 pixels is generated from a vertical interpolation of the Q1 pixels with the results of a horizontal interpolation of the data in the B block 74 and the C block 76, with the results being stored in the A block 72 (the Q2 pixels progressively overwriting the Q1 pixels) for the absolute displaced block difference calculation, in accordance with the following expression.

$$Q2_{r,c} = (Q1_{r,c} + I1_{r,c})/2 \quad (4)$$

where

$$I_{r,c} = (B_{r-1,c} + C_{r-1,c})/2 \quad (5)$$

Consider, for example, the interpolation of the Q2 pixel to the upper left of pixel $A_{10,10}$. A collection of pixels containing the pixel group $B_{9,8} - B_{9,15}$ in row nine of the B block 74 is addressed on, say, port A of the DPCM memory 130 and the pixel group $B_{9,8} - B_{9,15}$ is selected by shifter 140 and presented through the multiplexer 152 to the A port of the ALU 154. At about the same time, a collection of pixels containing the pixel group $C_{9,8} - C_{9,15}$ in row nine of the C block 76 is addressed on, say, port B of the DPCM memory 130 and the pixel group $C_{9,8} - C_{9,15}$ is selected by shifter 144 and presented through the multiplexer 152 to the B port of the ALU 154. The ALU 154 sums the pixel groups, divides by two, and rounds the result. An eight pixel result at the output of the ALU 154 is routed to the DP memory 124, where it is stored as pixels $I_{9,8} - I_{9,15}$ in the ninth row of I block 84, a free 16×16 block of memory in the DP memory 124. Next, the pixel group $I_{9,8} - I_{9,15}$ in row nine of the I block 84 is addressed on, say, port A of the DP memory 124 and presented through the multiplexer 152 to the A port of the ALU 154. At about the same time, a collection of pixels containing the pixel group $Q_{10,8} - Q_{10,15}$ in row ten of the A block 72 is addressed on, say, port A of the DPCM memory 130 and the pixel group $Q_{10,8} - Q_{10,15}$ is selected by shifter 140 and presented through the multiplexer 152 to the B port of the ALU 154. The ALU 154 sums the pixel groups, divides by two, and rounds the result. An eight pixel result at the output of the ALU 154 is routed back to the DPCM memory, where it is stored as pixels $Q_{2,10,8} - Q_{2,10,15}$ in the tenth row of the A block 72.

The third quarter pixel estimation search block of Q3 pixels is generated from a vertical interpolation of the data in the C block 76 and the D block 82 (which contains data previously copied from the A block 72), and stored in the A block 72 for the absolute displaced block difference calculation, in accordance with the following expression.

$$Q3_{r,c} = (C_{r-1,c} + A_{r,c})/2 \quad (6)$$

The interpolation of the Q3 pixel above pixel $A_{10,10}$ is done essentially as described above with respect to the Q1 pixel, but in accordance with expression (6).

The fourth quarter pixel estimation search block of Q4 pixels is generated from a horizontal interpolation of the Q3 pixels with the results of a vertical interpolation of the data in the B block 74 and the X block 70, with the results being stored in the A block 72 (the Q4 pixels progressively overwriting the Q3 pixels) for the absolute displaced block difference calculation, in accordance with the following expression.

$$Q4_{r,c} = (Q3_{r,c} + I2_{r,c})/2 \quad (7)$$

where

$$I2_{r,c} = (B_{r-1,c+1} + X_{r,c+1})/2 \quad (8)$$

The interpolation of the Q4 pixel to the upper right of the pixel $A_{10,10}$ is done essentially as described above with respect to the Q2 pixel, but in accordance with expression (7).

The fifth quarter pixel estimation search block of Q5 pixels is generated from a horizontal interpolation of the data in the X block 70 and the D block 82, and stored in the A block 72 for the absolute displaced block difference calculation, in accordance with the following expression.

$$Q5_{r,c} = (X_{r,c+1} + A_{r,c})/2 \quad (9)$$

The interpolation of the Q5 pixel to the right of the pixel $A_{10,10}$ is done essentially as described above with respect to the Q1 pixel, but in accordance with expression (9).

....

Ex. 1006, 8:28–10:5. Fandrianto continues on to similarly discuss the derivation of pixels Q6, Q7 and Q8. *Id.*, 10:6–50. From this description, it is clear that memory management and the need to continually overwrite previous results with new results imposes tight constraints on an efficient processing procedure.

57. Fandrianto goes on to state “It will be appreciated that a great many permutations of the order of derivation of pixels Q1–Q8 may be practiced, if desired. For example, the Q2 pixels may be derived from a horizontal interpolation of the Q3 pixels with the results of a vertical interpolation of the data in the B block 74 and the X block 70. As a further example, the Q2 pixels may be derived from a diagonal interpolation of the data in the A block 72 and the B block 74.” *Id.*, 10:51–58. Petitioner and Petitioner’s expert use this statement as their basis for expanding their interpretation of Fandrianto beyond what Fandrianto describes. Ex. 1003, ¶¶ 32, 117. However, the first sentence above is a broad stroke statement about the “order of derivation” of pixels Q1–Q8, not the *values* of pixels Q1–Q8 themselves. It’s true that a POSITA would see there are many possible orders of derivation, even given limitations imposed by reuse of Fandrianto’s limited memory. For example, there are $4 * 3 * 2 = 24$ possible orders of derivation for Q1, Q3, Q5 and Q7 alone, not counting how the derivation of Q2, Q4, Q6 and Q8 fits in as well. Even in light of this, Fandrianto’s limited memory architecture places significant restrictions on how a POSITA would implement the quarter pixel interpolation, as evidenced by

Fandrianto's detailed description above of the derivation of the Q1–Q8 interpolated quarter pixels.

58. Fandrianto gives the example of “Q2 pixels may be derived from a diagonal interpolation of the data in the A block 72 and the B block 74.” Applying this to Fandrianto's described algorithm, a POSITA would refer to the above description: “An eight pixel result at the output of the ALU 154 is routed back to the DPCM memory 130, where it is stored as pixels $Q1_{10,8}$ – $Q1_{10,15}$ in the tenth row of the A block 72... The second quarter pixel estimation search block of Q2 pixels is generated from a vertical interpolation of the Q1 pixels with the results of a horizontal interpolation of the data in the B block 74 and the C block 76, with the results being stored in the A block 72 (the Q2 pixels progressively overwriting the Q1 pixels).” Ex. 1006, 8:54–63. The POSITA would then modify the second quarter pixel estimation described above to use just the data in the A block 72 and the B block 74, rather than the data in the A block 72 (which contains the Q1 pixels), the B block 74 and the C block 76.

59. Since the data in the A block 72 contains the values $Q1_{x,y}$ (see Fandrianto Figure 5 above) and the B block 74 contains the values $B_{x,y}$, the value of Q2 would be the average of the Q1 and B values, resulting in a “partial diagonal” interpolation as shown below.

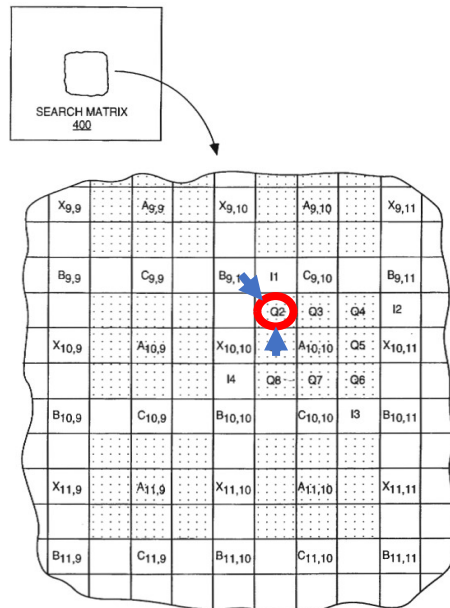


FIG. 5

60. Note that if Fandrianto had intended Q2 to be interpolated from B_{9,10} and A_{10,10} in this embodiment, it would have specified using D block 82 of the DP memory 124 (see Ex. 1001, 8:29–31) instead of the data in the A block 72. Furthermore, a POSITA would see that it is impractical to perform the described Q2 computation first among the Q1–Q8 pixels, because then the A data in A block 72—which is needed for later computations—would be overwritten, and therefore unavailable for computing the remaining Q1 and Q3–Q8 pixels.

C. TML6 in Combination with Fandrianto Does Not Teach or Disclose the Challenged Claims

61. In the two asserted ground, Petitioner asserts that TML6 in Combination with Fandrianto and in further view of the knowledge of a POSITA teaches or discloses all of the challenged claims 1, 4–17, 19–26, 28–29, 31–33, 36–

49, 51–59, 61–62, and 64–83. I disagree, for the following reasons.

1. **Claim 1[a]: “interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of K and L, according to a predetermined choice of either a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $1/2, 1/2$, or a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $1/2, 1/2$ and the nearest neighbouring pixel”**

62. Petitioner’s grounds do not teach this limitation. TML6 does not teach diagonal interpolation, and Fandrianto’s teaching of “diagonal” interpolation is different than that claimed by the ‘273 Patent’s claim element 1[a].

63. Refer to Section VII.B above, reproduced by reference herein. TML6 does not teach diagonal interpolation at all. Fandrianto teaches quarter-pixel interpolation that does not satisfy the claim language. Fandrianto’s quarter-pixel interpolation uses neither a) a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $1/2, 1/2$, nor b) a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of K and L, including zero, situated within a quadrant of the rectangular bounded region, the quadrant being defined by the sub-pixel having co-ordinates $1/2, 1/2$ and the nearest neighbouring pixel. Therefore, Fandrianto does not teach this claim element 1[a], nor does the combination of

TML6 and Fandrianto.

2. Claims 4–17, 19–26, and 28–29

64. Claims 4–17, 19–26, and 28–29 depend from claim 1. Because Petitioner’s ground does not render obvious claim 1, it does not render obvious any of claims 4–17, 19–26, and 28–29.

3. Independent claims 31, 32, 33, 64, 68, 71, 73, 75, 76, 79, 82, and 83 and their dependent claims 36–49, 51–59, 61–62, 65–67, 69–70, 72, 74, 77–78, and 80–81

The combination of TML6 and Fandrianto does not teach or disclose claim elements 31[a], 32[a], 33[a], 64[a], 68[a], 71[a], 73[a], 75[a], 76[a], 79[a], 82[a], and 83[a] for the same reasons it does not teach or disclose claim element 1[a]. Therefore, Petitioner’s grounds do not teach or disclose claims 31, 32, 33, 64, 68, 71, 73, 75, 76, 79, 82, and 83. Since claims 36–49, 51–59, 61–62, 65–67, 69–70, 72, 74, 77–78, and 80–81 are each dependent claims from one of independent claims 31, 32, 33, 64, 68, 71, 73, 75, 76, 79, 82, and 83, Petitioner’s grounds also do not teach or disclose claims 31, 32, 33, 64, 68, 71, 73, 75, 76, 79, 82, and 83.

D. A POSITA Would Not Have Been Motivated to Combine TML6 and Fandrianto

65. Petitioner’s expert opines that a POSITA would have been motivated to combine TML6 with Fandrianto. He reasons that “the two references are in the same field (*i.e.*, motion prediction video coding), address the same problem (*i.e.*, objects in video data moving from frame to frame by distances that do not match to

the pixel locations precisely), and purport to solve this problem by interpolating subpixels at fractional locations (*i.e.*, $\frac{1}{2}$ and $\frac{1}{4}$ sub-pixels) to improve motion prediction accuracy.” Ex 1003, ¶ 119. I disagree, for the following reasons.

66. The TML6 reference describes a video coding system in the process of being standardized in 2001. Fandrianto describes a hardware architecture for efficient video compression and motion search, under the constraints of very limited memory, which was designed in 1992 and was obsolete by 2001. For example, Fandrianto is incapable of performing the weighted interpolations described in TML6. *See*, e.g., Ex. 1001, 10:1–25, 11:35–63, Figs. 12(b), 12(c), 13(b) and 13(c) (describing weighted interpolations with sets of filter coefficients such as “ $b = (A_1 - 5 A_2 + 20 A_3 + 20 A_4 - 5 A_5 + A_6 + 16)/32$ ”). Fandrianto’s architecture does not even have a multiplier. Ex. 1006, *passim*. A POSITA seeking information on how to improve TML6 to develop a next-generation video standard would not consider a document describing an obsolete hardware architecture (with respect to that next-generation standard) to be relevant.

67. Furthermore, combining TML6 with Fandrianto undermines the design improvement that TML6 offers over TML5. For example, one key disadvantage of TML5 is its computational complexity. Ex. 1001, 11:6–24. The complexity arises from the fact that TML5 employs an approach where the interpolation of $\frac{1}{4}$ resolution sub-pixel values is dependent on the interpolation of $\frac{1}{2}$ resolution pixel

values. *Id.*, 11:7–10. This dependency results in truncation of the $\frac{1}{4}$ resolution sub-pixel values, a deleterious effect on the precision of some of the $\frac{1}{4}$ resolution sub-pixel values. *Id.*, 11:13–17. In contrast to TML5, TML6 preserves maximum precision throughout the interpolation process that is achieved by using intermediate values that are neither rounded nor clipped. *Id.*, 11:27–31. However, if TML6 were modified and combine with Fandrianto, the 6-tap filter in TML6 would be replaced with the 2-tap filter, and the precision of the intermediate values would once again decrease due to the reintroduction of the modification step. Since TML6 is more advanced than Fandrianto, a POSITA would not combine TML6 with Fandrianto to improve TML6.

68. I also disagree that the relevant substance of TML6 is not disclosed in the '273 Patent's specification. The fact that specific source code functions' names are not disclosed in the specification is immaterial. Columns 11 and 12 describe TML6's functionality in relevant detail. Ex. 1001, Cols. 11–12. Thus, the inclusion of the names of the functions is not necessary.

69. I also disagree that Fandrianto is directed to the problem of objects in video data moving from frame to frame by distances that do not match to the pixel locations precisely. Fandrianto's purpose is not to design a video coding algorithm; it is to disclose a hardware architecture for implementing video coding algorithms. Therefore, even if Fandrianto were contemporary with TML6, which it is not, its

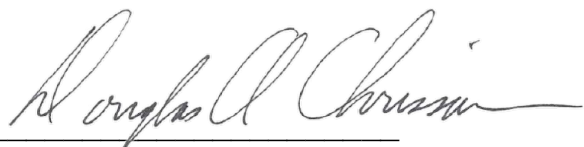
solution of a novel hardware architecture for video processing is still different than a solution to designing the best next-generation video codec algorithm.

VIII. OPINIONS

70. For all the reasons described above, it is my opinion that Petitioner's Ground of TML6 in combination with Fandrianto does not render obvious the Challenged Claims 1, 4–17, 19–26, 28–29, 31–33, 36–49, 51–59, 61–62, and 64–83 of the '273 Patent.

71. The reasons described above do not necessarily represent the entirety of my opinions regarding Petitioner's Ground of TML6 in combination with Fandrianto. I reserve the right to provide further analysis in the event this IPR is instituted. I declare under penalty of perjury that the foregoing is true and correct. I acknowledge that willful false statements and the like are punishable by fine and/or imprisonment, that all statements made of my own knowledge are true, and that all statements made on information and belief are believed to be true.

Executed November 27, 2024, in Los Gatos, CA.

By: 
Douglas A. Chrissan, Ph. D.