

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.

IPR2024-01140

U.S. Patent No. 7,280,599

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,280,599**

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

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING.....	1
III.	IDENTIFICATION OF CHALLENGE	1
	A. Prior Art.....	1
	B. Grounds	1
IV.	'599 PATENT OVERVIEW	2
	A. Technology Background	2
	B. The Alleged Problem in the Prior Art	7
	C. The Alleged Invention of the '599 Patent.....	9
V.	LEVEL OF ORDINARY SKILL	10
VI.	CLAIM CONSTRUCTION	11
VII.	PRIOR ART OVERVIEW	12
	A. TML6.....	12
	B. Fandrianto.....	17
VIII.	DETAILED EXPLANATION OF GROUNDS.....	21
	A. Ground 1: Obviousness over TML6 and Fandrianto (Claims 1- 20, 22-42, and 44-51)	21
	1. Motivation to Combine	21
	2. Claims 1-19.....	29
	3. Independent Claim 20	112
	4. Claims 22-42 and 44-51	112
	B. Ground 2: Claims 21 and 43 Are Obvious Over TML6 in View of Fandrianto and the Knowledge of a POSITA.....	120
	1. Independent Claim 21	120
	2. Claim 43	122
IX.	NO BASIS FOR DISCRETIONARY INSTITUTION DENIAL	123
	A. <i>General Plastic</i>	123
	B. <i>Fintiv</i>	125

C.	§325(d)	125
X.	CONCLUSION.....	127
XI.	MANDATORY NOTICES	127
A.	Real Party in Interest	127
B.	Related Matters.....	127
C.	Notice of Counsel and Service Information.....	128

EXHIBIT LIST

Exhibit	Reference
1001	U.S. Patent No. 7,280,599 (“the ’599 patent”)
1002	File History of the ’599 patent
1003	Declaration of Dr. Kannan Ramchandran (“Ramchandran”)
1004	Curriculum Vitae of Dr. Kannan Ramchandran
1005	H.26L Test Model Long Term Number 6 (TML-6) draft0 Reference Code (“TML6”)
1006	U.S. Patent No. 5,594,813 (“Fandrianto”)
1007	VCEG-L45 H.26L Test Model Long Term Number 6 (TML-6) draft0 (“VCEG-L45”)
1008	Declaration of Dr. Stephan Wenger (“Wenger”)
1009	U.S. Patent No. 6,295,376 (“Nakaya”)
1010	Original Complaint, <i>Nokia Techs. Oy v. Amazon.com, Inc.</i> , No. 1:23-cv-01236 (D. Del. Oct. 31, 2023), ECF No. 1.
1011	Martyn J. Riley and Iain E.G. Richardson, <i>Digital Video Communications</i> , Artech House, Inc. (1997) (“Riley”)
1012	Gary J. Sullivan, Thomas Wiegand, and Thomas Stockhammer, <i>Using the Draft H.26L Video Coding Standard for Mobile Applications</i> , IEEE, pp. 573-576 (2001) (“Sullivan”)
1013	Ushashree, <i>Video Compression Based on RMB</i> , International Journal of Electronics Communication and Computer Engineering (2013) (“Ushashree”)
1014	Iain Richardson, <i>Video Codec Design: Developing Image and Video Compression Systems</i> (2002) (“Richardson”)
1015	Markus Flierl, Thomas Wiegand, and Bernd Girod, <i>Multihypothesis Pictures for H.26L</i> , Proceedings of the IEEE International Conference on Image Processing, Vol. 3, pp. 526-529 (2001) (“Flierl”)
1016	“Transfer of ftp://standards.pictel.com to ITU,” https://www.itu.int/wftp3/av-

Exhibit	Reference
	arch/Transfer%20of%20standards.pictel.com%20to%20ITU.htm (2003)
1017	ITU, https://www.itu.int/wftp3/av-arch/video-site/h26L/older_tml/
1018	Electrical Engineering Dictionary CRCnetBASE 2000, <i>codec</i> (2000) (“CRC-Dictionary”)
1019	Original Complaint, <i>Nokia Techs. Oy v. Lenovo (Shanghai) Elecs. Tech. Co.</i> , No. 5:19-cv-00427 (E.D.N.C. Sept. 25, 2019), ECF No. 1
1020	Joint Stipulation to Dismissal Without Prejudice, <i>Nokia Techs. Oy v. Lenovo (Shanghai) Elecs. Tech. Co.</i> , No. 5:19-cv-00427 (E.D.N.C. Apr. 12, 2021), ECF No. 126
1021	Gisle Bjontegaard et al., A simplified 1/3 pixel prediction method (Mar. 21, 2001) (“VCEG-M28”)
1022	<i>Interpolation</i> , The New Encyclopædia Britannica (1986) (“Britannica”)
1023	VCEG-N10 H.26L Test Model Long Term Number 8 (TML-8) draft0 (“VCEG-N10”)
1024	William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, <i>Numerical Recipes in C: The Art of Scientific Computing</i> , Cambridge University Press (2 nd ed. 1992) (reprinted 1996) (“Press”)

LISTING OF CHALLENGED CLAIMS

Claim 1	
1[pre][a]	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 arranged in rows and columns,
1[pre][b]	the pixel and sub-pixel locations 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:
1[a]	interpolating a sub-pixel value for a sub-pixel having co-ordinates with odd values of both K and L, according to a predetermined choice of a weighted average of the value of a nearest-neighbouring pixel and the value of the sub-pixel situated at co-ordinates $1/2$, $1/2$, and a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of both K and L, including zero, situated within a quadrant of the rectangular bounded region defined by corner pixels having co-ordinates $1/2$, $1/2$ and the nearest neighbouring pixel;
1[b]	interpolating sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal 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 both K and L, using weighted sums of the values of pixels located in rows and columns respectively; and
1[c]	interpolating sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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 bounded rectangular regions.
Claim 2	
2	A method according to claim 1, comprising using 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 both K and L, wherein when the chosen weighted average is that involving the value of a nearest neighbouring pixel and the value of the sub-pixel situated at co-ordinates 1/2, 1/2, selecting the relative magnitudes of the first and second weights to be inversely proportional to the respective straight-line diagonal distances of the nearest-neighbouring pixel and the sub-pixel situated at co-ordinates 1/2, 1/2 to the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 3	
3	A method according to claim 2, comprising using first and second weights with equal values when the nearest-neighbouring pixel and sub-pixel situated at co-ordinates 1/2, 1/2 are equidistant from the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 4	
4	A method according to claim 1, comprising using 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 both 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 both K and L, selecting 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 both K and L to the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 5	
5	A method according to claim 4, comprising using first and second weights with equal values when the sub-pixels having co-ordinates with even values of both K and L are equidistant from the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 6	
6	A method according to claim 1, comprising interpolating sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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	
7	A method according to claim 1, comprising interpolating sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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	
8	A method according to claim 1, comprising interpolating 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	
9	A method according to claim 1, comprising interpolating 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	
10	A method according to claim 1, comprising interpolating 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	
11	A method according to claim 1, comprising interpolating 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	
12	A method according to claim 1, comprising interpolating 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	
13	A method according to claim 1, comprising interpolating 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	
14	A method according to claim 1, comprising interpolating 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	
15	A method according to claim 1, comprising interpolating 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 zero and L equal to 2.
Claim 16	
16	A method according to claim 1, comprising interpolating a sub-pixel value for a sub-pixel having co-ordinates with K equal to 0 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	
17	A method according to claim 1, comprising interpolating 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 18	
18	A method according to claim 1, comprising interpolating a sub-pixel value for the sub-pixel having co-ordinates with K equal to 2^N-1 and L equal to 2^N-1 , by taking a weighted average of the values of the four corner pixels that define the rectangular bounded region.
Claim 19	
19	A method according to claim 1, comprising selecting a value for N from a list consisting of the values 2, 3, and 4.
Claim 20	
20 <pre>[a]</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 arranged in rows and columns,
20 <pre>[b]</pre>	the pixel and sub-pixel locations 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:
20[a]	interpolating a sub-pixel value for a sub-pixel having co-ordinates with both K and L equal to 1 or 3, according to a predetermined choice of 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, and a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with both K

	and L equal to zero, 2 or 4, situated within a quadrant of the rectangular bounded region defined by corner pixels having co-ordinates 2/4, 2/4 and the nearest neighbouring pixel;
20[b]	interpolating sub-pixel values for sub-pixels having co-ordinates with K equal to 2 and L equal 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 both K and L equal to 1 or 3, using weighted sums of the values of pixels located in rows and columns respectively; and
20[c]	interpolating a sub-pixel value for the sub-pixel having co-ordinates with both K and L equal to 2, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with both 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 21	
21[pre][a]	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 arranged in rows and columns,
21[pre][b]	the pixel and sub-pixel locations 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:
21[a]	interpolating a sub-pixel value for a sub-pixel having co-ordinates with both K and L equal to 1, 3, 5 or 7, according to a predetermined choice of 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, and a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with both K and L equal to zero, 2, 4, 6 or 8, situated within a quadrant of the rectangular bounded region defined by corner pixels having co-ordinates 4/8, 4/8 and the nearest neighbouring pixel;
21[b]	interpolating sub-pixel values for sub-pixels having co-ordinates with K equal to 2, 4 or 6 and L equal 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 both 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
21[c]	interpolating sub-pixel values for sub-pixels having co-ordinates with both 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 both 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 22	
22[pre][a]	An apparatus for sub-pixel value interpolation, the apparatus being operable 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 arranged in rows and columns,
22[pre][b]	the pixel and sub-pixel locations 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 apparatus comprising:

22[a]	circuitry operable to interpolate a sub-pixel value for a sub-pixel having co-ordinates with odd values of both K and L, according to a predetermined choice of 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, and a weighted average of the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of both K and L, including zero, situated within a quadrant of the rectangular bounded region defined by corner pixels having co-ordinates 1/2, 1/2 and the nearest neighbouring pixel;
22[b]	circuitry operable to interpolate sub-pixel values for sub-pixels having co-ordinates with K equal to an even value and L equal 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 both K and L, using weighted sums of the values of pixels located in rows and columns respectively; and
22[c]	circuitry operable to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in the interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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 bounded rectangular regions.
Claim 23	
23	An apparatus according to claim 22, wherein the apparatus is operable 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 both K and L, wherein when the chosen weighted average is that involving the value of a nearest neighbouring pixel and the value of the sub-pixel situated

	at co-ordinates $1/2, 1/2$, the apparatus is operable to select the relative magnitudes of the first and second weights to be inversely proportional to the respective straight-line diagonal distances of the nearest-neighbouring pixel and the sub-pixel situated at co-ordinates $1/2, 1/2$ to the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 24	
24	An apparatus according to claim 23, wherein the apparatus is operable to use first and second weights with equal values when the nearest-neighbouring pixel and sub-pixel situated at co-ordinates $1/2, 1/2$ are equidistant from the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 25	
25	An apparatus according to claim 23, wherein the apparatus is operable to use first and second weights with equal values when the sub-pixels having co-ordinates with even values of both K and L are equidistant from the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.
Claim 26	
26	An apparatus according to claim 22, wherein the apparatus is operable 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 both 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 both K and L, the apparatus is operable 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 both K and L to the sub-pixel having co-ordinates with odd values of both K and L whose value is being interpolated.

Claim 27	
27	A communications terminal comprising a video encoder according to claim 44.
Claim 28	
28	A communications terminal comprising a video decoder according to claim 46.
Claim 29	
29	An apparatus according to claim 22, wherein the apparatus is operable to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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 30	
30	An apparatus according to claim 22, wherein the apparatus is operable to interpolate sub-pixel values for sub-pixels having co-ordinates with even values of both K and L, used in interpolation of sub-pixel values for the sub-pixels having co-ordinates with odd values of both 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 31	
31	An apparatus according to claim 22, wherein the apparatus is operable 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 32	
32	An apparatus according to claim 22, wherein the apparatus is operable 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 33	
33	An apparatus according to claim 22, wherein the apparatus is operable 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 34	
34	An apparatus according to claim 22, wherein the apparatus is operable 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 35	
35	An apparatus according to claim 22, wherein the apparatus is operable 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 36	
36	An apparatus according to claim 22, wherein the apparatus is operable 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 37	
37	An apparatus according to claim 22, wherein the apparatus is operable 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 38	
38	An apparatus according to claim 22, wherein the apparatus is operable 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 zero and L equal to 2.
Claim 39	
39	An apparatus according to claim 22, wherein the apparatus is operable to interpolate a sub-pixel value for a sub-pixel having co-ordinates with K equal to 0 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 40	
40	An apparatus according to claim 22, wherein the apparatus is operable 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 41	
41	An apparatus according to claim 22, wherein the apparatus is operable to interpolate a sub-pixel value for the sub-pixel having co-ordinates with K equal to 2^{N-1} and L equal to 2^{N-1} , by taking a weighted average of the values of the four corner pixels that define the rectangular bounded region.
Claim 42	
42	An apparatus according to claim 22, wherein N is set equal to 2.
Claim 43	
43	An apparatus according to claim 22, wherein N is set equal to 3.
Claim 44	
44	A video encoder comprising an apparatus for sub-pixel value interpolation according to claim 22.
Claim 45	
45	A still image encoder comprising an apparatus for sub-pixel value interpolation according to claim 22.
Claim 46	
46	A video decoder comprising an apparatus for sub-pixel value interpolation according to claim 22.
Claim 47	
47	A communications terminal comprising a still image encoder according to claim 45.
Claim 48	
48	A still image decoder comprising an apparatus for sub-pixel value interpolation according to claim 22.

Claim 49	
49	A codec comprising a video encoder according to claim 44 and a video decoder according to claim 46.
Claim 50	
50	A communications terminal comprising a still image decoder according to claim 48.
Claim 51	
51	A codec comprising a still image encoder according to claim 45 and a still image decoder according to claim 48.

I. INTRODUCTION

Amazon.com, Inc. and Amazon.com Services LLC (“Amazon” or “Petitioner”) request *inter partes* review (“IPR”) of all claims (1-51; the “Challenged Claims”) of U.S. Patent No. 7,280,599 (“’599 patent”; EX-1001).

II. GROUNDS FOR STANDING

Petitioner certifies that the ’599 patent is available for IPR, and Petitioner is not barred or estopped from requesting IPR on the grounds herein. Petitioner files this Petition within one year of service on Amazon of Patent Owner’s district court complaint alleging infringement of the ’599 patent. EX-1020.

III. IDENTIFICATION OF CHALLENGE

A. Prior Art

Prior Art
TML6 (H.26L Test Model Long Term Number 6) , publicly available by March 28, 2001, prior art at least under §102(a) ¹ (EX-1005)
Fandrianto (U.S. Patent No. 5,594,813) , issued on January 14, 1997, prior art at least under §102(b) (EX-1006)

B. Grounds

Petitioner respectfully requests review and cancellation under 35 U.S.C. §311 of the Challenged Claims in view of:

¹ All references to 35 U.S.C. §§102, 103 are to the pre-AIA statutory framework.

Ground	Claims	References
1 (§103)	1-20, 22-42, 44-51	TML6, Fandrianto
2 (§103)	21, 43	TML6, Fandrianto, knowledge of a POSITA

IV. '599 PATENT OVERVIEW

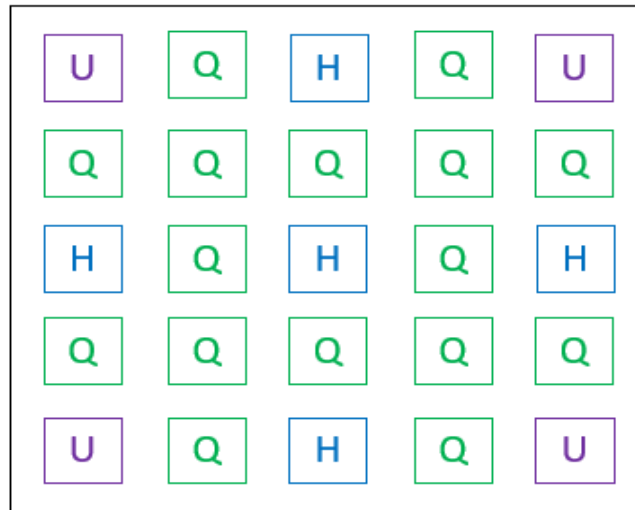
A. Technology Background

Video data consist of sequences of still images, or “frames,” comprising rows and columns of “pixels.” EX-1001, 1:41-43; Ramchandran, ¶¶21-23. “Motion compensated temporal prediction” is a video coding technique that exploits the redundancy typically present between successive frames in a video sequence. EX-1001, 2:62-3:3. Motion-predicted video frames are represented by “motion vectors” that indicate the displacement of pixels in a current frame relative to their position in another (“reference”) frame. EX-1001, 4:20-28; Ramchandran, ¶¶24-25. A motion vector is determined by comparing pixels in the current frame with pixels within a “search region” of the reference frame. EX-1001, 4:58-64, Figure 4(c).

Objects—i.e., pixel groups/blocks—may move between frames by distances that are not exact pixel locations but are more closely represented by fractions of the distance between pixels. Ramchandran, ¶26. In such cases, the values of virtual “sub-pixels” must be “interpolated” using actual pixel values. Ramchandran, ¶27. The benefits of sub-pixel interpolation were well-known by the '599 patent's earliest

possible priority date. *See* VCEG-L45, 15; Fandrianto, 5:26-28; Ramchandran, ¶¶29-33, 43.

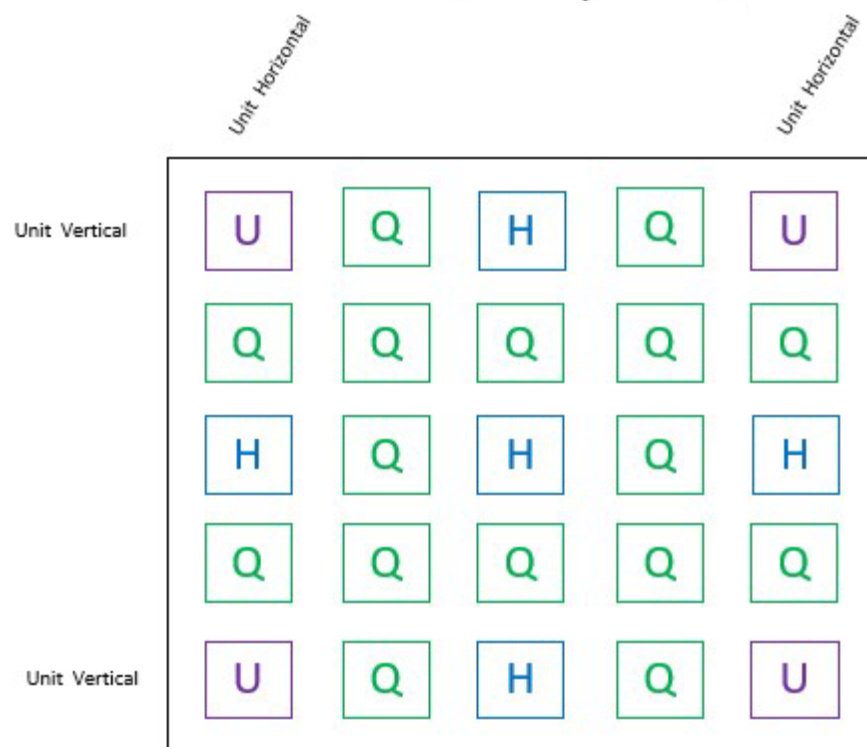
For clarity, certain terminology is addressed below. The figure below² shows four actual pixels, referred to as **unit pixels U** (purple). **Half (1/2) sub-pixels H** (blue) are located halfway (horizontally, vertically, or both) between unit pixels. **Quarter (1/4) sub-pixels Q** (green) are located quarter-way between unit pixels.



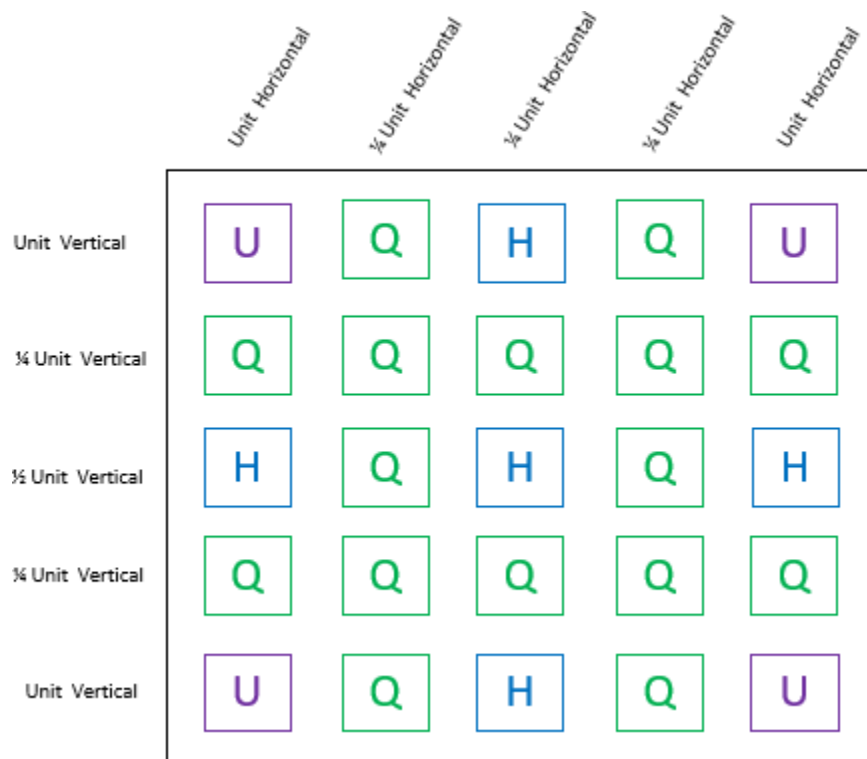
Ramchandran, ¶28. As shown below, **unit horizontal** location refers to a pixel/sub-pixel situated in a **column** that contains unit pixels (purple). Similarly, **unit vertical**

² Figures like the one below that include a citation to Dr. Ramchandran's declaration (EX-1003) have been created to graphically illustrate certain concepts in this Petition.

location refers to a pixel/sub-pixel situated in a **row** that contains unit pixels. Thus, a unit pixel is situated at both unit horizontal and unit vertical locations.

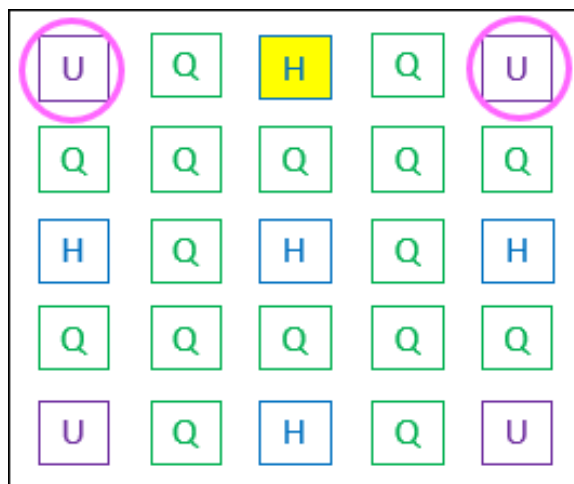


Ramchandran, ¶28-29. **Half (1/2) unit horizontal location** refers to a position halfway between two unit horizontal locations and **half (1/2) unit vertical location** refers to a position halfway between two unit vertical locations. **Quarter (1/4) unit horizontal location** refers to a position quarter-way between two unit horizontal locations and **quarter (1/4) unit vertical location** refers to a position quarter-way between two unit vertical locations.



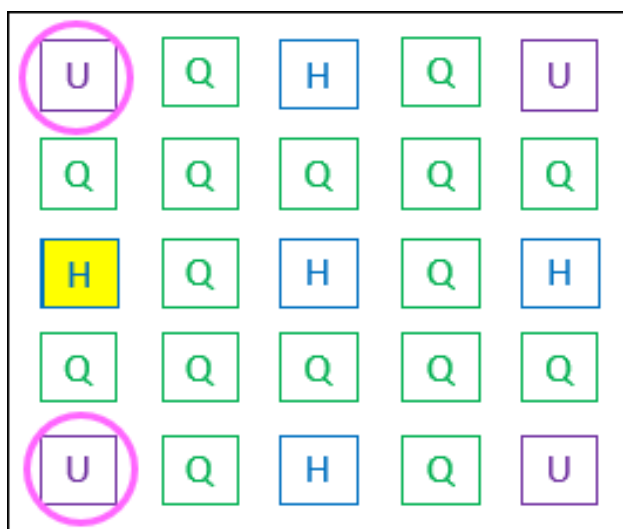
Ramchandran, ¶28-29.

Sub-pixel values can be interpolated using pixel/sub-pixel values along a horizontal, vertical, or diagonal line. As shown below, highlighted half sub-pixel H (blue)—situated at the half unit horizontal, unit vertical location—can be **horizontally interpolated** using two unit pixels (purple; circled) in the **same row**.



$\frac{1}{2}$ unit horizontal
unit vertical

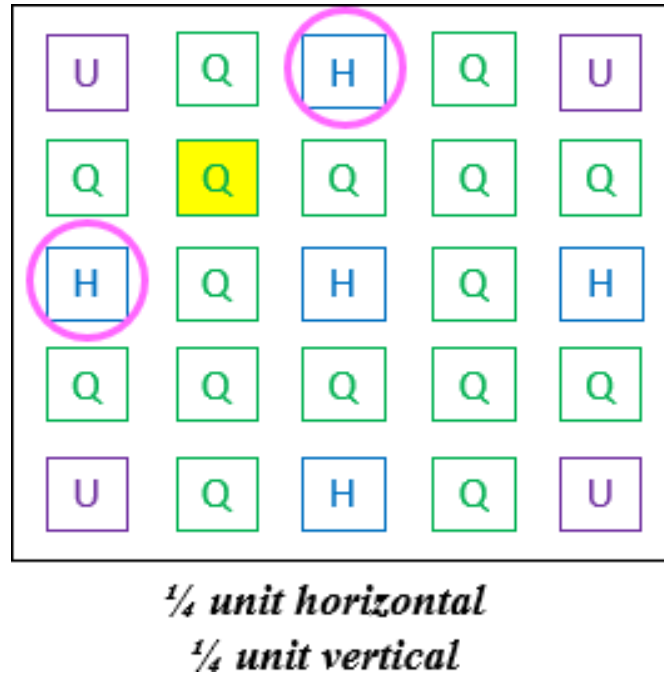
Ramchandran, ¶30. Likewise, as shown below, highlighted half sub-pixel H (blue)—situated at the unit horizontal, half unit vertical location—can be **vertically interpolated** using the two unit pixels (purple; circled) in the **same column**.



unit horizontal
 $\frac{1}{2}$ unit vertical

Ramchandran, ¶31. Finally, as shown below, highlighted quarter sub-pixel Q (green)—situated at the quarter unit horizontal, quarter unit vertical (or “quarter-

quarter”)³ location—can be **diagonally interpolated** using previously interpolated half sub-pixels H (blue; circled).



Ramchandran, ¶32.

B. The Alleged Problem in the Prior Art

The '599 patent admits that interpolating half and quarter sub-pixels was known. EX-1001, 6:40-46, 7:7-27, 8:50-9:16, 10:9-13:22; Ramchandran, ¶34. For example, the Background describes TML6's sub-pixel interpolation method in

³ This Petition uses the shorthand “quarter-quarter sub-pixels” to refer to sub-pixels situated at the quarter unit horizontal and quarter unit vertical locations, and similarly uses “half-half sub-pixels” to refer to sub-pixels situated at the half unit horizontal and half unit vertical locations.

conjunction with Figure 13(a), below, and explains that values of the quarter sub-pixels “d” and “e” (green) are calculated using horizontal or vertical interpolation.

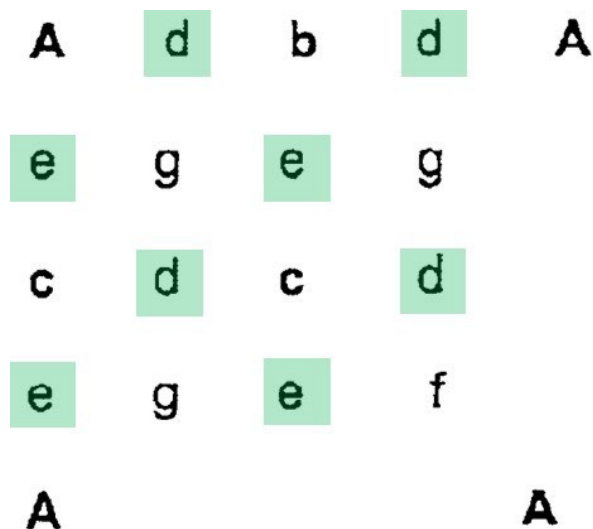


Fig. 13(a)

EX-1001, Figure 13(a), 12:18-57, 22:17-18.

According to the '599 patent, TML6 does not calculate highlighted quarter sub-pixel “g” (green) (below) using horizontal or vertical interpolation. Instead, sub-pixel “g” is interpolated based on pixel “A,” sub-pixel “b,” and sub-pixels “c” (circled).

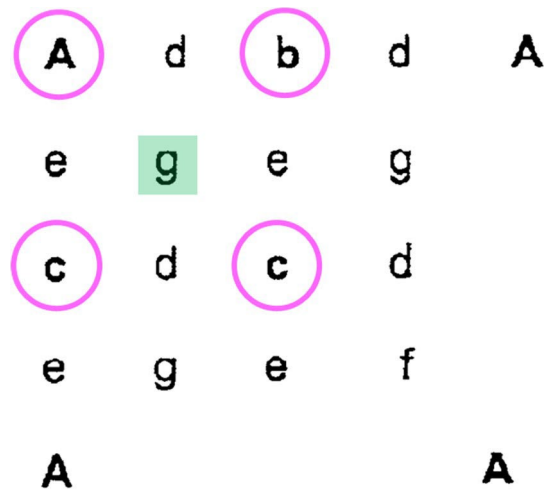


Fig. 13(a)

EX-1001, Figure 13(a), 8:43-53, 12:45-13:9; Ramchandran, ¶¶35-38. The '599 patent asserts, however, interpolating certain sub-pixels based on four other pixels/sub-pixels is too computationally intensive. *See* EX-1001, 38:17-38:41.

C. The Alleged Invention of the '599 Patent

To address this purported problem, as illustrated in Figure 14(a) below, the '599 patent claims as its allegedly novel concept using two diagonally opposite half sub-pixels “b” (blue) to interpolate quarter sub-pixel “h” (green).

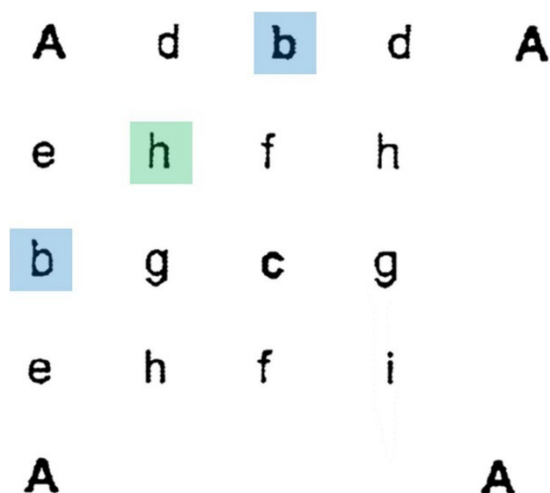


Fig. 14(a)

EX-1001, Figure 14(a), 22:23-24, 36:55-37:10; Ramchandran, ¶39. This technique allegedly improves on TML6 because it “does not require high precision arithmetic to be used in the calculation of all sub-pixels.” EX-1001, 38:17-41; Ramchandran, ¶40.

V. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) at the time of the alleged invention of the ’599 patent would have had a (1) Bachelor’s degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience

with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa.⁴ Ramchandran, ¶41.

VI. CLAIM CONSTRUCTION

Claims should be construed “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. §42.100(b); *see also Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). Petitioner is unaware of any “prior claim construction determination” related to the ’599 patent. 37 C.F.R. §42.100(b).

Petitioner has not proposed any terms for construction because no constructions are necessary to resolve any disputes identified in this Petition.⁵ *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

⁴ Although the definition of a POSITA proposed in the three prior IPRs challenging the ’599 patent (IPR2021-00037, IPR2021-00038, and IPR2021-00039) differs slightly from the definition proposed here, the Challenged Claims are invalid under both definitions. Ramchandran, ¶42.

⁵ If trial is instituted, Petitioner reserves the right to address construction of any terms raised at trial, or in the related district court action.

VII. PRIOR ART OVERVIEW

A. TML6

TML6 (EX-1005) comprises excerpts of the “Test Model Long Term Number 6” source code authored by the Video Coding Experts Group (“VCEG”) of the International Telecommunication Union Telecommunication Standardization Section (“ITU-T”). EX-1001, 8:50-60; Wenger, ¶¶15, 21; Ramchandran, ¶44. Dr. Stephan Wenger, a member of VCEG who authored portions of TML6, explains TML6 comprises a subset of a group of files that together make up the fully functional source code of a working video encoder and decoder. Wenger, ¶¶19-21. These files were packaged in a single compressed file named “*tml6pre.zip*” and made publicly available via the VCEG FTP server by March 28, 2001. Wenger, ¶¶18-38.

As Dr. Ramchandran explains, except for “readme.txt,” the files that comprise EX-1005 originate from subfolders in “*tml6pre.zip*” named “*lencod*,” “*ldecod*,” and “*bin*” containing files with executable code. Ramchandran, ¶46. Each page of EX-1005 indicates the file name and location within the file structure of each file at the top left corner. They are listed below with corresponding page ranges:

File	Pages in TML6 (EX-1005)
/tml6pre.zip/readme.txt	Pages 1-2
/tml6pre.zip/lencod/inc/image.h	Page 3
/tml6pre.zip/lencod/src/block.c	Pages 4-34

File	Pages in TML6 (EX-1005)
/tml6pre.zip/lencod/inc/global.h	Pages 35-47
/tml6pre.zip/lencod/src/image.c	Pages 48-68
/tml6pre.zip/lencod/src/lencod.c	Pages 69-100
/tml6pre.zip/lencod/src/macroblock.c	Pages 101-128
/tml6pre.zip/ldecod/inc/global.h	Pages 129-134
/tml6pre.zip/ldecod/inc/image.h	Pages 135-136
/tml6pre.zip/ldecod/src/image.c	Pages 137-154
/tml6pre.zip/ldecod/src/ldecod.c	Pages 155-163
/tml6pre.zip/ldecod/src/macroblock.c	Pages 164-183

TML6 thus excerpts the source code that was packaged and made publicly available in “*tml6pre.zip*” by VCEG. Wenger, ¶¶15, 18-38; Ramchandran, ¶¶45-46. Consequently, for purposes of this Petition, treating TML6 as a single prior art reference is justified and appropriate. *Avi Networks, Inc. v. Citrix Sys., Inc.*, IPR2019-00844, Paper 19, 13 (Oct. 1, 2019) (finding “four source code files...[which] were ‘distributed together in a single compressed file’...constitute a single prior art printed publication under [§102(b)]”); see *MacSports, Inc. v. Idea Nuevo, Inc.*, IPR2018-01006, Paper 6, 14 (Nov. 13, 2018) (two webpages and the 24 images appearing thereon deemed a single publication because they “would have been accessible at the same website as of [the same date]”).

A reference is considered publicly accessible if “persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can

locate it.” *Acceleration Bay, LLC v. Activision Blizzard Inc.*, 908 F.3d 765, 772 (Fed. Cir. 2018) (citation omitted). TML6 was made available on a public VCEG FTP server by March 28, 2001. Wenger, ¶¶18-38. Subscribers to the VCEG email reflector, including participants, students, and the public, were commonly notified by email when files were uploaded to the FTP folder. Wenger, ¶¶26-27. Files were frequently downloaded from the publicly accessible folder. Wenger, ¶¶20-22, 27-28; Ramchandran, ¶¶44-45. Persons interested in tracking the developments of the latest video coding standard regularly visited the VCEG sites to ensure that products and services they were developing were consistent with the standards under development. Wenger, ¶¶37-38. A POSITA would have known test models discussed at VCEG meetings such as TML6 included details of technologies with the potential to make their way into the standard under development. Wenger, ¶37. Dr. Wenger personally shared VCEG work with colleagues and students during the relevant time. Wenger, ¶35. TML6 is thus §102(a) prior art to the ’599 patent.

TML6 implements methods of interpolating half and quarter sub-pixels. TML6, 57-59 (lines 538-632); Ramchandran, ¶¶47-108. As the ’599 patent explains in its Background section, the TML6 sub-pixel interpolation scheme uses “direct interpolation” in which “maximum precision is retained throughout. This is achieved by using intermediate values which are neither rounded nor clipped.” EX-1001, 11:44-50. The ’599 patent describes the “interpolation method according to

TML6 as applied in the encoder...with reference to FIGS. 13[a-c].” EX-1001, 11:50-13:53.

First, the '599 patent explains that in TML6's sub-pixel interpolation method, “[t]he value for...½ resolution sub-pixel b [blue] in FIG. 13a, 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 (A₁ to A₆) [purple] situated in a row...symmetrically about b, as shown in FIG. 13b.”

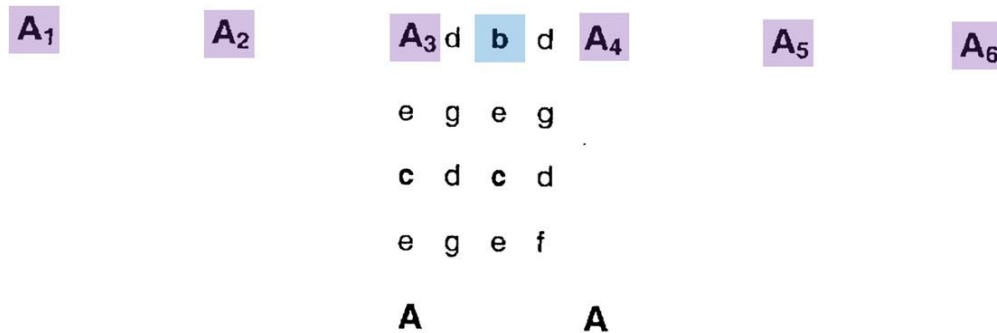
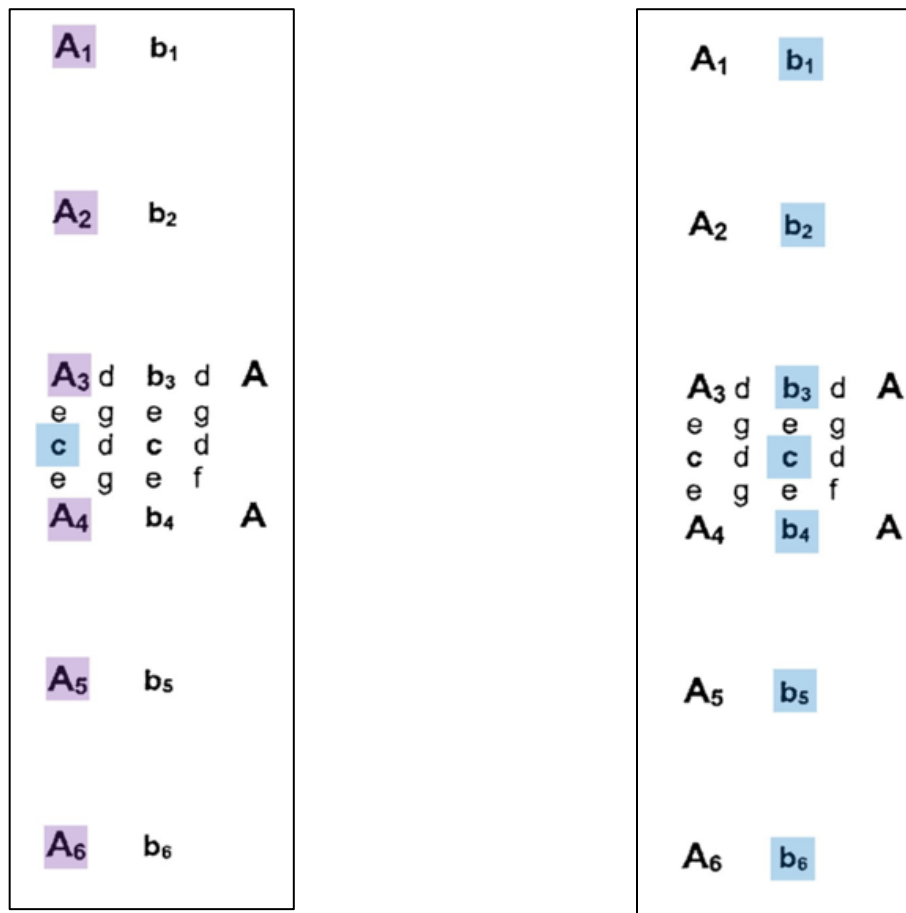


Figure13(b)

EX-1001, Figure 13(b), 11:54-65. As explained for 1[a], the TML6 code confirms this interpolation method. TML6, 57-58 (lines 563-575, 588); Ramchandran, ¶¶48, 65-69.

Second, the '599 patent explains in TML6, “[v]alues for the ½ resolution sub-pixels labelled c [blue] are obtained by first calculating intermediate values c. Referring to FIG. 13(c) [below], an intermediate value c for the ½ resolution sub-

pixel c...is calculated based upon the values of the 6 pixels (A_1 to A_6) [purple on the left] situated in a column...symmetrically about c....Similarly, an intermediate value c for the $\frac{1}{2}$ resolution sub-pixel c...is calculated according to [values of the 6 previously-interpolated sub-pixels b_1 - b_6 (blue on the right)].”



EX-1001, Figure 13(c), 11:66-12:17. As described for 1[a]-1[b], the TML6 code confirms this interpolation method. TML6, 57-58 (lines 576-592); Ramchandran, ¶¶49, 70-79.

Third, the '599 patent explains in TML6, “[v]alues for $\frac{1}{4}$ resolution sub-pixels labelled g [(green) in Figure 13(a) (below)] are computed using the value of the nearest original pixel A [purple] and the intermediate values of the three nearest neighbouring $\frac{1}{2}$ resolution sub-pixels [blue].”

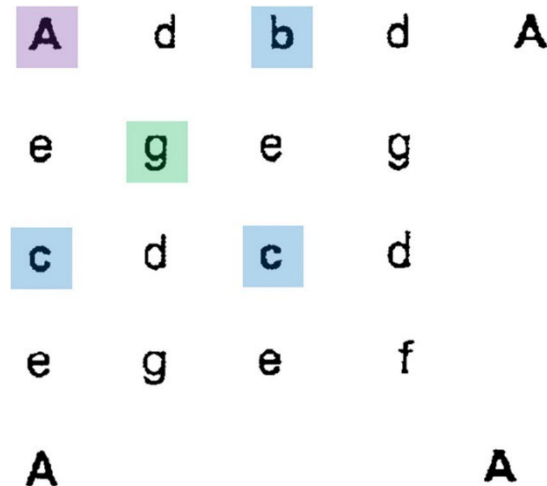


Fig. 13(a)

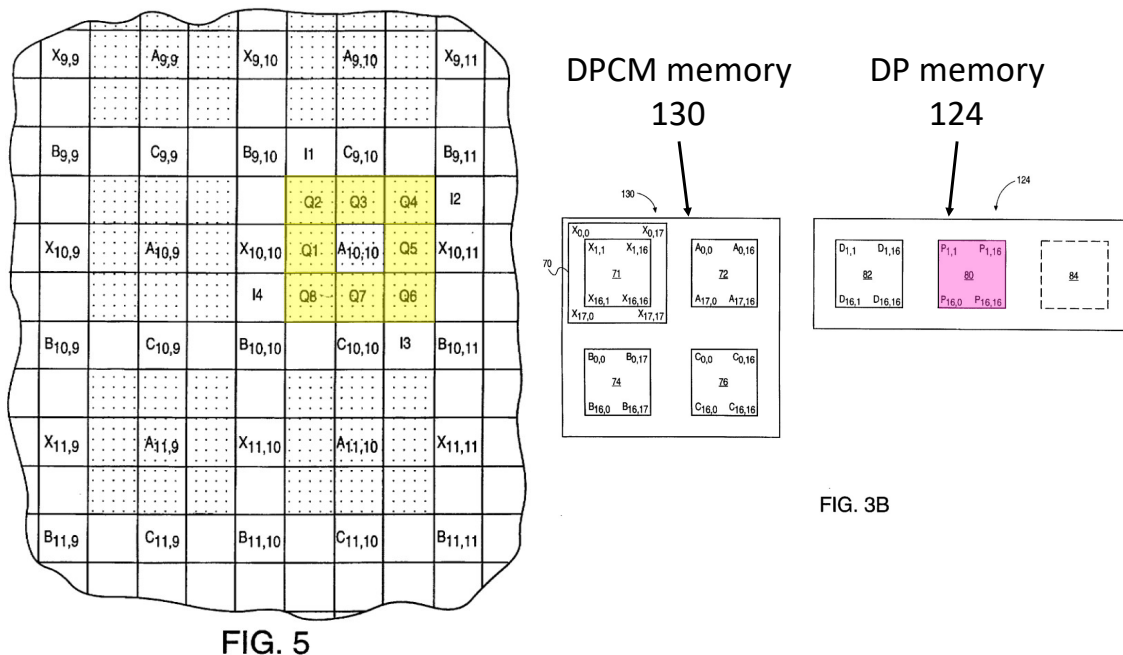
EX-1001, Figure 13(a), 12:58-65. As described for 1[c], the TML6 code confirms this interpolation method. TML6, 58 (lines 593-625); Ramchandran, ¶¶50, 96-101.

B. Fandrianto

Fandrianto is directed to a method and apparatus for “rapid and efficient motion vector searching” in a digital video compression system. Fandrianto, Abstract, 1:7-10, 1:43-67, 3:55-62; Ramchandran, ¶¶109-112. Fandrianto specifically teaches using “diagonal interpolation” to perform “motion vector

searching with...quarter pixel accuracy.” *Id.*, 5:26-32, 7:53-55, Figures 3B, 4-5; Ramchandran, ¶¶113-117.

With reference to Figure 5 (below left), Fandrianto describes deriving highlighted quarter pixels Q1-Q8 surrounding sub-pixel $A_{10,10}$ in sequential order using the two-memory apparatus of Figure 3B (below right) comprising DPCM memory 130 and DP memory 124. DPCM memory 130 is a “search memory” in which “[t]he search window...[for] a previously processed frame” is stored. DP memory 124 is “used...to store...temporarily needed blocks,” e.g., “an input or image block” in P block 80 (pink) for “successively compar[ing]...with the search blocks in the search window” stored in DPCM memory 130.



Fandrianto, Figure 5, Figure 3B, Abstract, 3:16-43, 3:57-66, 4:18-5:5, 5:26-32, 8:16-10:58, 12:9-15. Though Fandrianto focuses for simplicity’s sake on quarter pixels

Q1-Q8 around sub-pixel $A_{10,10}$, its technique is block-based and calculates all the “highlighted areas about the elements A” in Figure 5, as described in detail below.

See id., 8:24-28; Ramchandran, ¶156.

First, pixel block $X_{0,0}$ - $X_{17,17}$ (purple)⁶ is stored in “X” block 70 (purple) of DPCM memory 130.

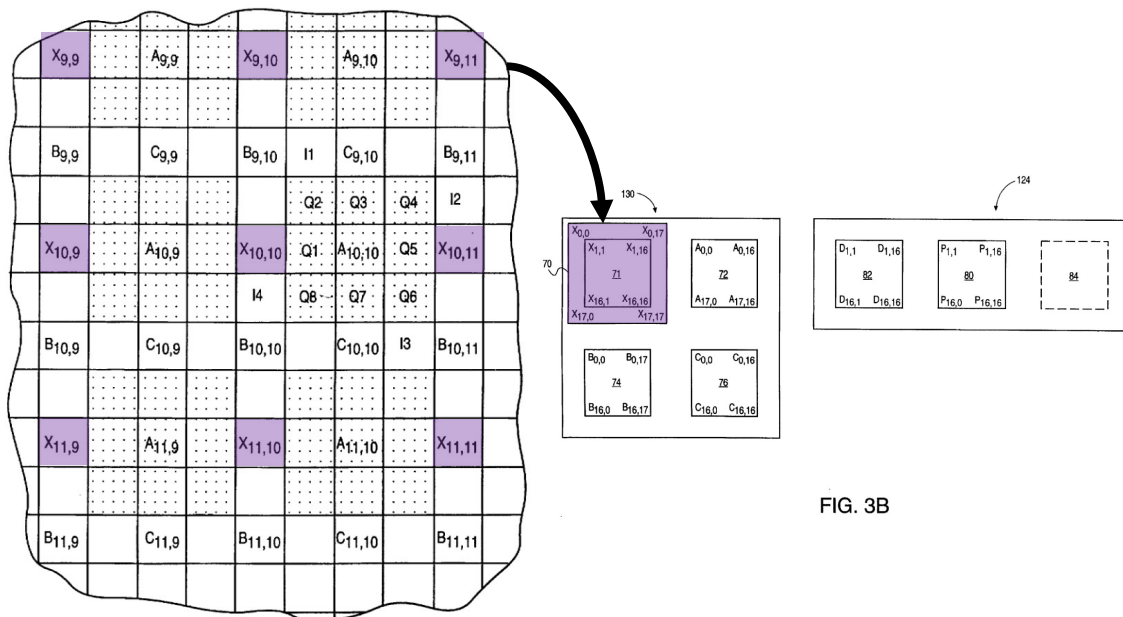


FIG. 3B

Fandrianto, Figure 5, Figure 3B, Abstract, 5:37-41. Next, “‘A’ block 72 [red], ‘B’ block 74 [blue], and ‘C’ block 76 [green] are interpolated from...X block 70[.]” The interpolated sub-pixels are annotated using like colors below.

⁶ In Fandrianto, the subscript notation “ $X_{r,c}$ ” indicates a range of pixel positions within a two-dimensional array “X” where the first subscript “r” denotes the row and the second subscript “c” denotes the column. *See* Fandrianto, 5:51-55.

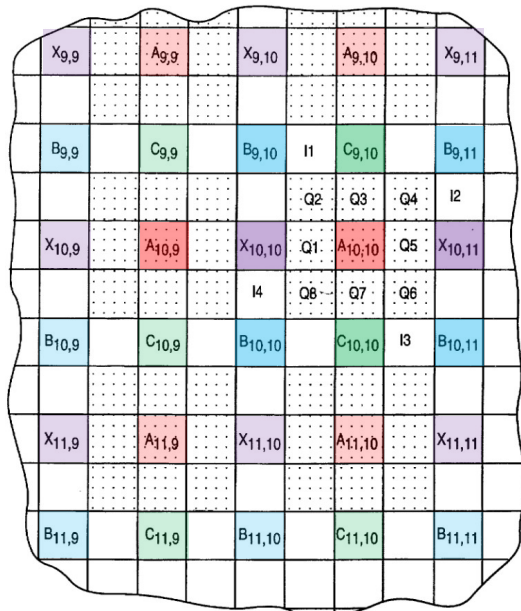


FIG. 5

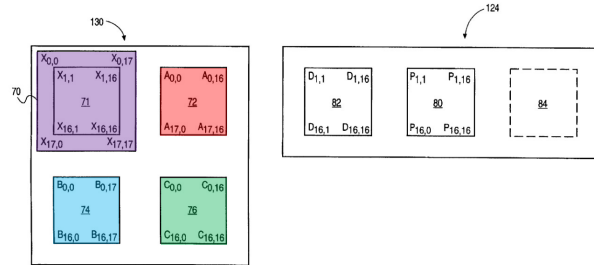


FIG. 3B

Fandrianto, Figure 5, Figure 3B, 5:41-44, 6:18-7:2. Fandrianto then describes an iterative process for generating quarter pixels Q1-Q8 in sequential order. *Id.*, 8:29-10:50. Each quarter pixel is calculated based on the data of the preceding quarter pixel and is stored in A block 72, progressively overwriting the preceding quarter pixel data. *Id.*; Ramchandran, ¶157.

Fandrianto notes “that a great many permutations of the order of derivation of pixels Q1-Q8 may be practiced[.]” Fandrianto, 10:51-53. Accordingly, “the order in which [Q1-Q8] are generated and compared, and the selection of obsolete full and half pixel estimation search blocks to be overwritten are...discretionary” such that “[t]he order shown in FIG. 5” is only illustrative. *Id.*, 8:15-21; Ramchandran, ¶158.

As illustrated below, Fandrianto teaches “the Q2 pixels [yellow] may be derived from a **diagonal interpolation** of the data in the A block 72 [red] and the B

block 74 [blue].” Because Fandrianto previously established both the order in which Q1-Q8 are generated and compared and the selection of obsolete full and half pixel estimation search blocks to be overwritten are discretionary, A block 72 and B block 74 contain values for A pixels [red] and B pixels [blue], respectively. Ramchandran, ¶159.

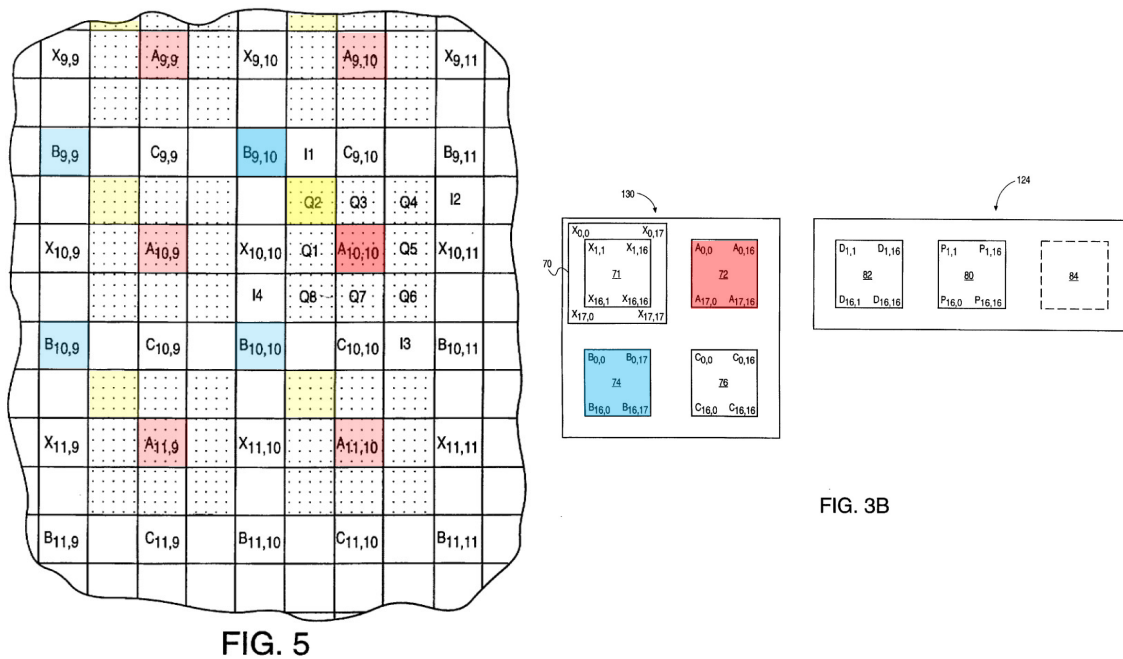


FIG. 3B

Fandrianto, Figures 5, 3B, 10:51-58.

VIII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Obviousness over TML6 and Fandrianto (Claims 1-20, 22-42, and 44-51)

1. Motivation to Combine

A POSITA would have been motivated to incorporate the teachings of Fandrianto (interpolating quarter-quarter sub-pixels) into TML6 (video coder with half and quarter sub-pixel interpolation) by implementing Fandrianto’s sub-pixel

interpolation method in the TML6 source code. Ramchandran, ¶¶119-121. TML6 and Fandrianto are in the same field (motion prediction video coding), address the same problem (video objects moving from frame to frame by sub-pixel distances), and solve that problem by interpolating half and quarter sub-pixels. *See* Fandrianto, 5:26-28; TML6, 16-24 (lines 734-1180 (“*motion_search*” function)), 115-116 (lines 769-842 (“Motion vector search”)), 57-59 (lines 538-632 (sub-pixel interpolation)); EX-1001, 8:43-9:9, 11:33-13:29; Figures 13(a)-(c).

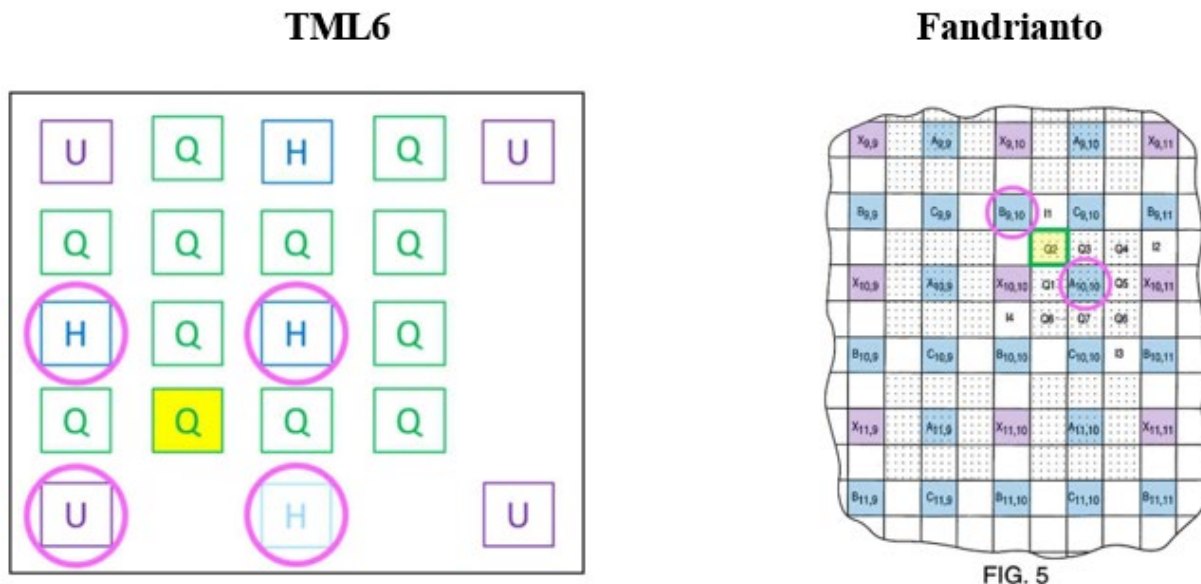
Fandrianto provides an express teaching that would have led a POSITA to combine its method of interpolating quarter-quarter sub-pixels with TML6’s digital video compression functionality. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418-419 (2007). Fandrianto, 1:61-67, Figure 2; VCEG-L45, 15. Fandrianto states: “Various interpolation techniques may be used in the derivation of a given quarter pixel estimation block, depending on various factors such as the amount of memory available[.]” Fandrianto, 7:32-35; *see id.*, 2:4-7, 3:55-4:6, 5:26-32, Figure 2. With that in mind, Fandrianto teaches a quarter-pixel interpolation technique suitable for a “limited memory” architecture. *Id.*, 7:36-37, 8:16-21, Figure 2. A POSITA would have been motivated to implement Fandrianto’s “[r]apid and efficient motion vector searching” technique for interpolating quarter-quarter sub-pixels in TML6 to realize Fandrianto’s stated benefit of “improved prediction accuracy” “while avoiding the need for additional memory.” *Id.*, 3:57-62, 5:1-5, 5:26-28.

Fandrianto discloses a “limited memory embodiment” capable of interpolating quarter sub-pixels using averages of values for pixels (“X-block”) and/or already interpolated half sub-pixels (“A-block,” “B-block,” and “C-block”). *Id.*, Figure 5, 5:26-32, 7:27-10:58. A POSITA would have been motivated to replace the quarter sub-pixel interpolation in the TML6 *oneforthpix()* function (TML6, 58 (lines 593-625)) with Fandrianto’s efficient technique of leveraging dedicated and optimized hardware for its quarter sub-pixel interpolation technique rather than relying on a CPU to execute the TML6 code. Fandrianto, Figures 2, 5, 2:4-7, 3:55-4:6, 5:26-32, 7:27-10:58; Ramchandran, ¶132.

Furthermore, a POSITA would have recognized that Fandrianto’s alternative method of interpolating quarter-quarter sub-pixels reflects a design choice between finite options with well-understood tradeoffs of improved accuracy and computational efficiency. *E.g.*, *Uber Techs., Inc. v. X One, Inc.*, 957 F.3d 1334, 1341 (Fed. Cir. 2020). Although TML6’s interpolation method relying on four values is arguably more accurate because it considers more pixels/sub-pixels near the quarter-quarter sub-pixel being interpolated, it is more computationally complex. TML6, 58 (lines 593-625). Conversely, Fandrianto’s interpolation method is less computationally intensive because it uses two values rather than four. Fandrianto, Figure 5, 10:51-58. Fandrianto discusses optimizing interpolation methods based on accuracy and efficiency considerations such as available memory, number of

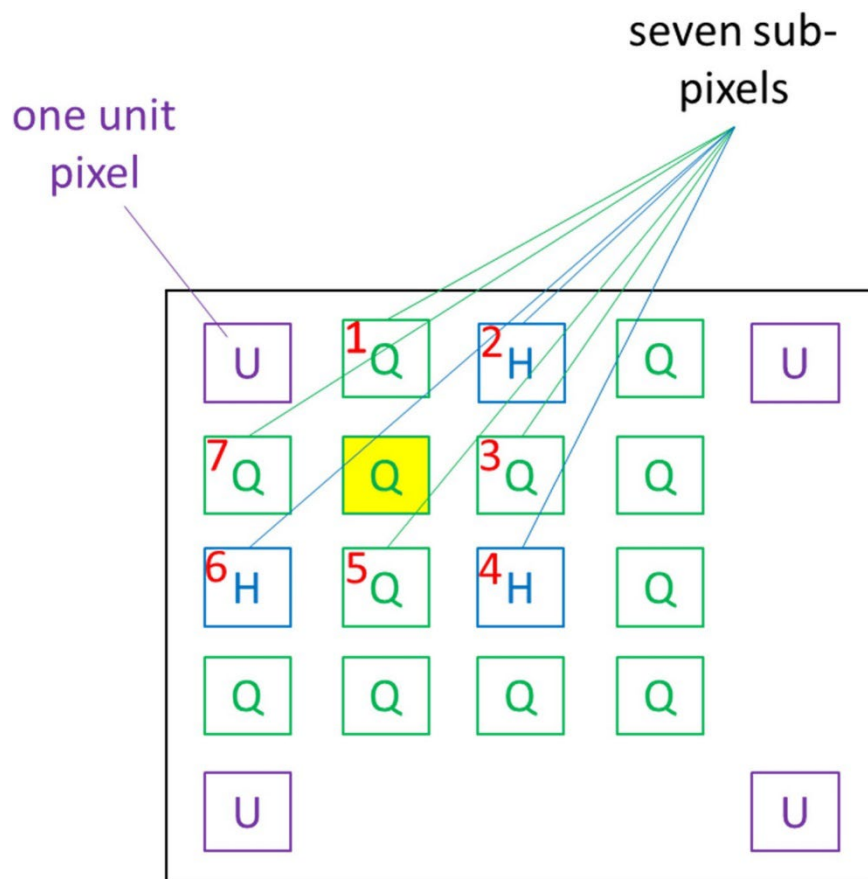
ports, and data buses. *Id.*, 7:32-37. Therefore, a POSITA would have been motivated to use Fandrianto's interpolation method as a design choice to increase computational efficiency. Ramchandran, ¶¶127-128.

The TML6-Fandrianto combination would have yielded predictable results because the combination is a simple substitution of Fandrianto's quarter-quarter sub-pixel interpolation method with TML6's quarter-quarter sub-pixel interpolation method to obtain a predictable result. As illustrated below, TML6 interpolates highlighted quarter-quarter sub-pixel based on the closest unit pixel (purple) and the three closest half sub-pixels (blue). TML6, 58 (lines 593-625). Fandrianto interpolates highlighted quarter-quarter sub-pixel based on the two closest diagonally opposed half sub-pixels (blue). Fandrianto, Figure 5, 10:51-58.



A POSITA would have substituted Fandrianto's known interpolation method in TML6 to produce the predictable result of interpolating quarter-quarter sub-pixels. Ramchandran, ¶¶123-124.

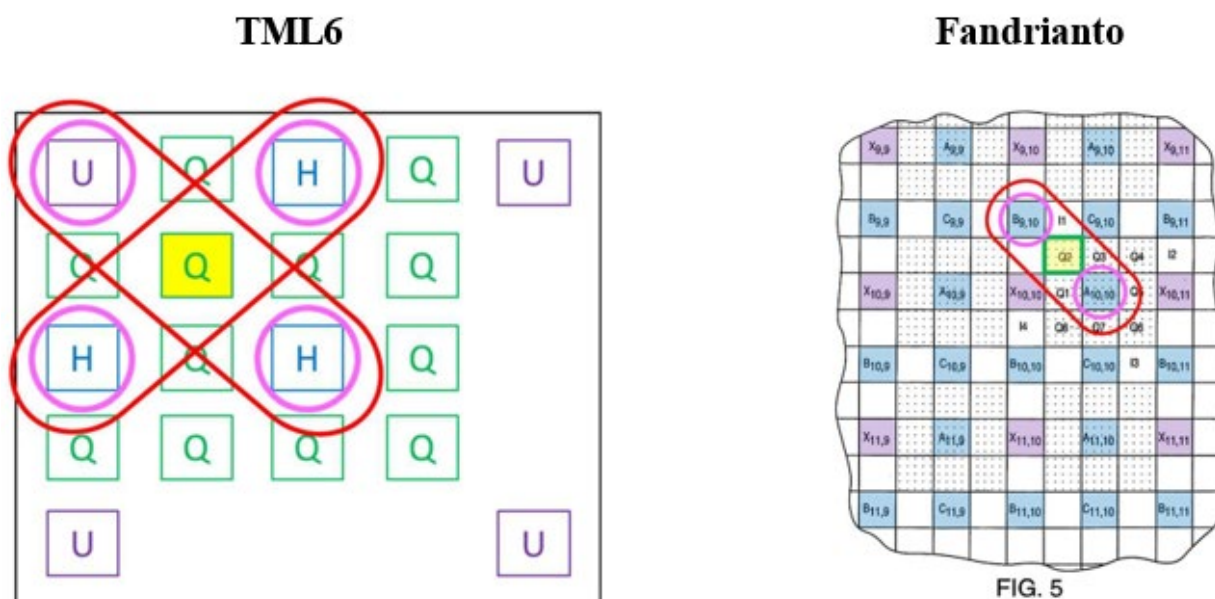
It would have been obvious to try Fandrianto's method to interpolate the quarter-quarter sub-pixels in TML6 because there are a finite number of identifiable and predictable ways to do so. *KSR*, 550 U.S. at 421. Interpolation refers generally to estimating new data points based on assumptions regarding surrounding known data points. *E.g.*, Britannica, 356 (“**interpolation**, in mathematics, the determination or estimation of the value of a function $f(x)$ from certain known values of the function.”). Sub-pixel interpolation is based on the corresponding assumption that values for sub-pixels at fractional locations are typically more similar to values of nearby pixels and sub-pixels than to those farther away. *See, e.g.*, VCEG-M28 (describing 2-tap, 4-tap, and 6-tap filters). Thus, a POSITA would have recognized there are a finite number of pixels and sub-pixels near the highlighted quarter-quarter sub-pixel (below) to reasonably use for interpolation—*e.g.*, one adjacent unit pixel, and seven adjacent sub-pixels.



Ramchandran, ¶¶124-125. Accordingly, it would have been obvious to try a well-known one of these finite options—Fandrianto’s interpolation method for quarter-quarter sub-pixels—as an alternative to TML6’s method. Ramchandran, ¶¶122-126.

Implementing Fandrianto’s method for interpolating quarter-quarter sub-pixels in TML6’s source code would have been a predictable result with a reasonable expectation of success. TML6 and Fandrianto disclose methods for interpolating quarter-quarter sub-pixels using the same inputs and basic arithmetic. For any quarter-quarter sub-pixel, Fandrianto’s interpolation method would achieve the same result in Fandrianto’s system as in the TML6 code. Ramchandran, ¶129.

Furthermore, making the combination would have been trivial and thus predictable. The only difference between the TML6 and Fandrianto methods is that TML6 discloses averaging both sets of two diagonally opposed sub-pixels (red), whereas Fandrianto discloses averaging only one set of diagonally opposed sub-pixels (also red).



In mathematical terms, Dr. Ramchandran explains that the single diagonal interpolation of Fandrianto can be written as:

$$Q = \frac{H+H}{2}.$$

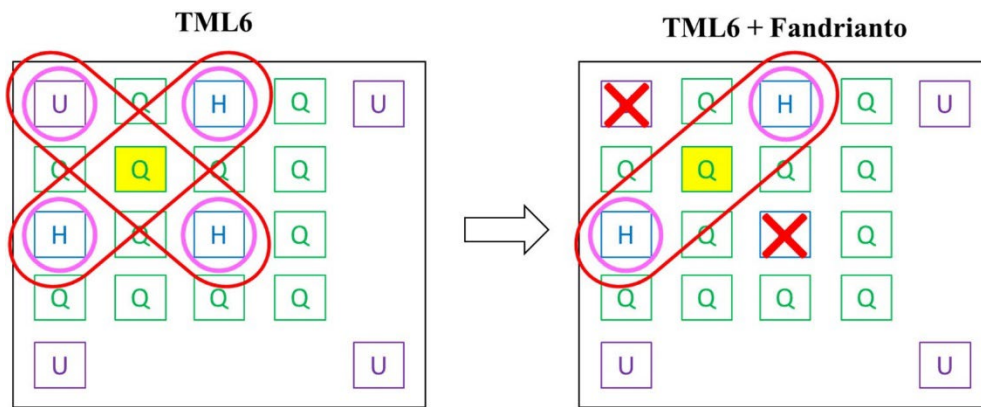
The average of TML6 can be written as:

$$Q = \frac{H+H+H+U}{4},$$

which is mathematically equivalent to averaging two diagonal interpolations:

$$Q = \frac{\left(\frac{H+H}{2}\right) + \left(\frac{H+U}{2}\right)}{2}.$$

Ramchandran, ¶130. TML6 therefore mathematically includes the average taught by Fandrianto, highlighted in yellow above, showing that TML6 teaches all of the required input values for such a calculation—i.e., values for the two half sub-pixels H. Accordingly, a POSITA would have easily combined Fandrianto and TML6 by either (1) using the interpolation hardware disclosed in Fandrianto’s Figure 2 (for which TML6 calculates all required inputs), or (2) removing one of the two diagonal interpolation calculations (shown above) for the quarter-quarter sub-pixels in the TML6 code.



TML6, 58 (lines 593-625); Fandrianto, 10:51-58; Ramchandran, ¶¶131-132.

A POSITA would have expected predictable results and a reasonable likelihood of success because both TML6 and Fandrianto result in interpolated quarter sub-pixels, and as discussed above, TML6 discloses that the values required to interpolate the quarter sub-pixels of Fandrianto—i.e., unit pixels and half sub-

pixels—are stored in the *imgY* and *mref* data structures. Moreover, implementing the quarter sub-pixel interpolation functionality in Fandrianto’s hardware, rather than TML6’s software, would increase the speed with which quarter sub-pixels can be interpolated. Ramchandran, ¶132.

2. Claims 1-19

a. Independent Claim 1

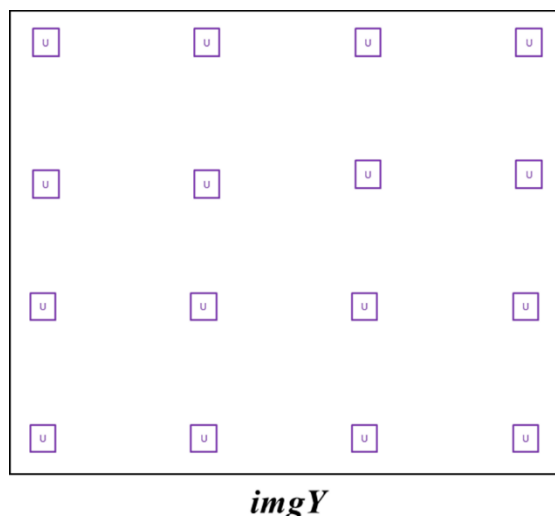
- (i) **1[pre][a]: A method for sub-pixel value interpolation to determine values for sub-pixels situated within a rectangular bounded region defined by four corner pixels with no intermediate pixels between the corners, the pixels and sub-pixels being arranged in rows and columns,**

If limiting,⁷ TML6 discloses 1[pre][a]. Ramchandran, ¶133.

TML6 discloses a method for sub-pixel value interpolation. TML6, 48 (line 3) (“Code one image/slice”). TML6 includes a file *image.c* (in the *src* folder of the *encoder* folder) containing a function titled *encode_one_frame()* that “[e]ncodes one frame” of video data. TML6, 48-50 (lines 33-162). The *encode_one_frame()* function calls the functions *oneforthpix()* and *oneforthpix_2()* to interpolate sub-pixels for the frame of video data. TML6, 49 (lines 102-109), 57-60 (lines 538-719); Ramchandran, ¶¶133-134, 47-108.

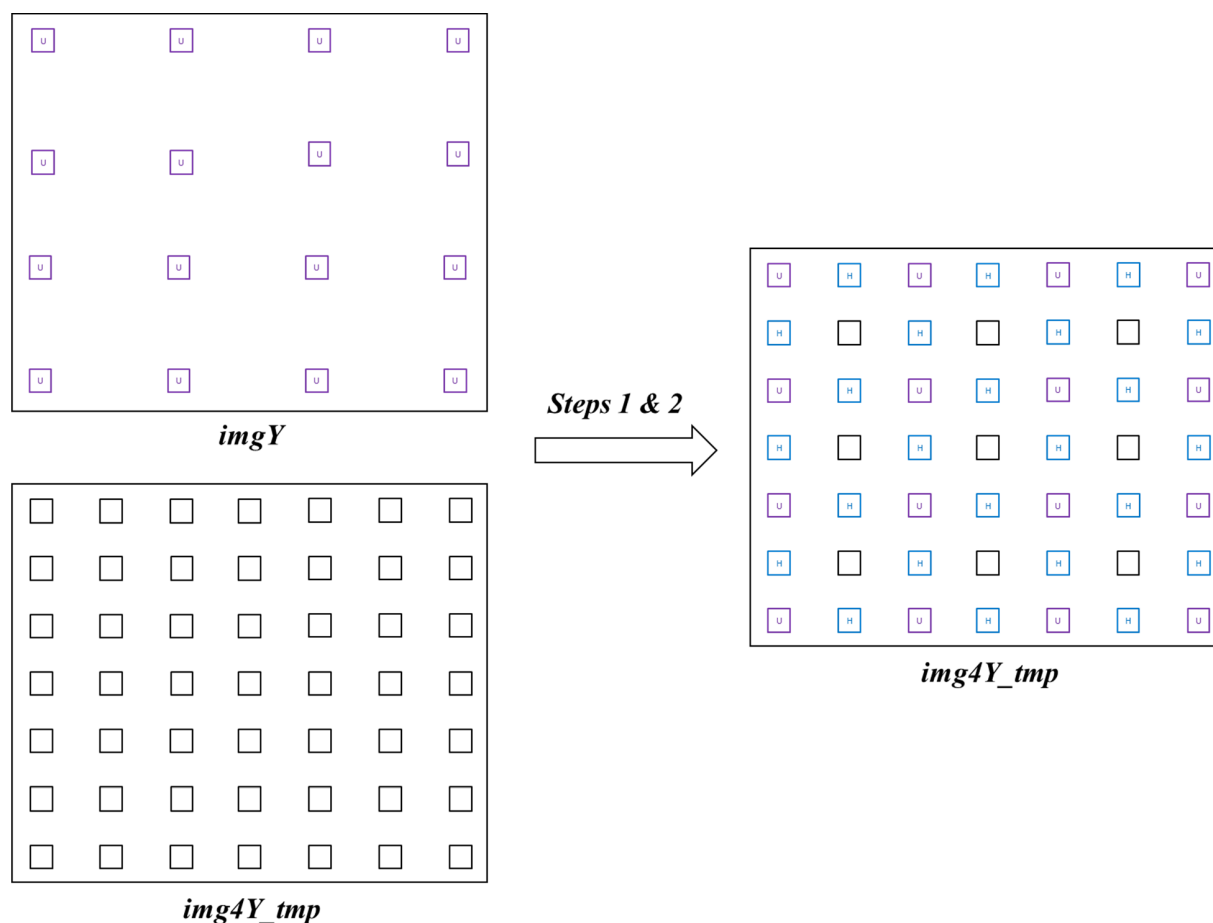
⁷ Petitioner reserves the right to argue in other proceedings the preamble of any challenged claim is limiting.

Namely, as shown below, the *imgY* data structure stores data values representing the brightness—or “luminance”—of unit pixels “U” (purple) for a frame of video data.



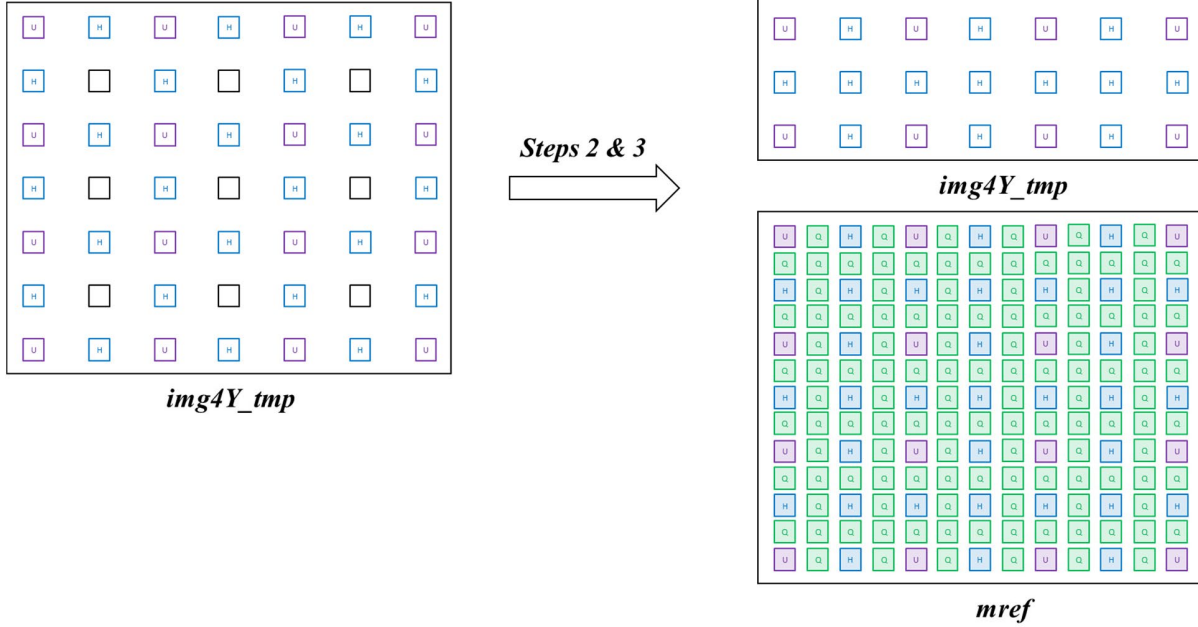
TML6, 93 (lines 1338-1344), 49-50 (lines 97-109), 110-112 (lines 480-600), 57-59 (lines 563-632), 131 (line 149). Then, as described in further detail for 1[a]-1[c], the *oneforthpix()* function performs three main steps to interpolate sub-pixel values based on these stored unit pixel values. TML6, 57-58 (lines 553-625); Ramchandran, ¶135.

In the first two steps (shown below), the *oneforthpix()* function creates a temporary data structure *img4Y_tmp*, interpolates values for some of the half sub-pixels (blue) based on the unit pixels (purple), and stores these values in *img4Y_tmp*.



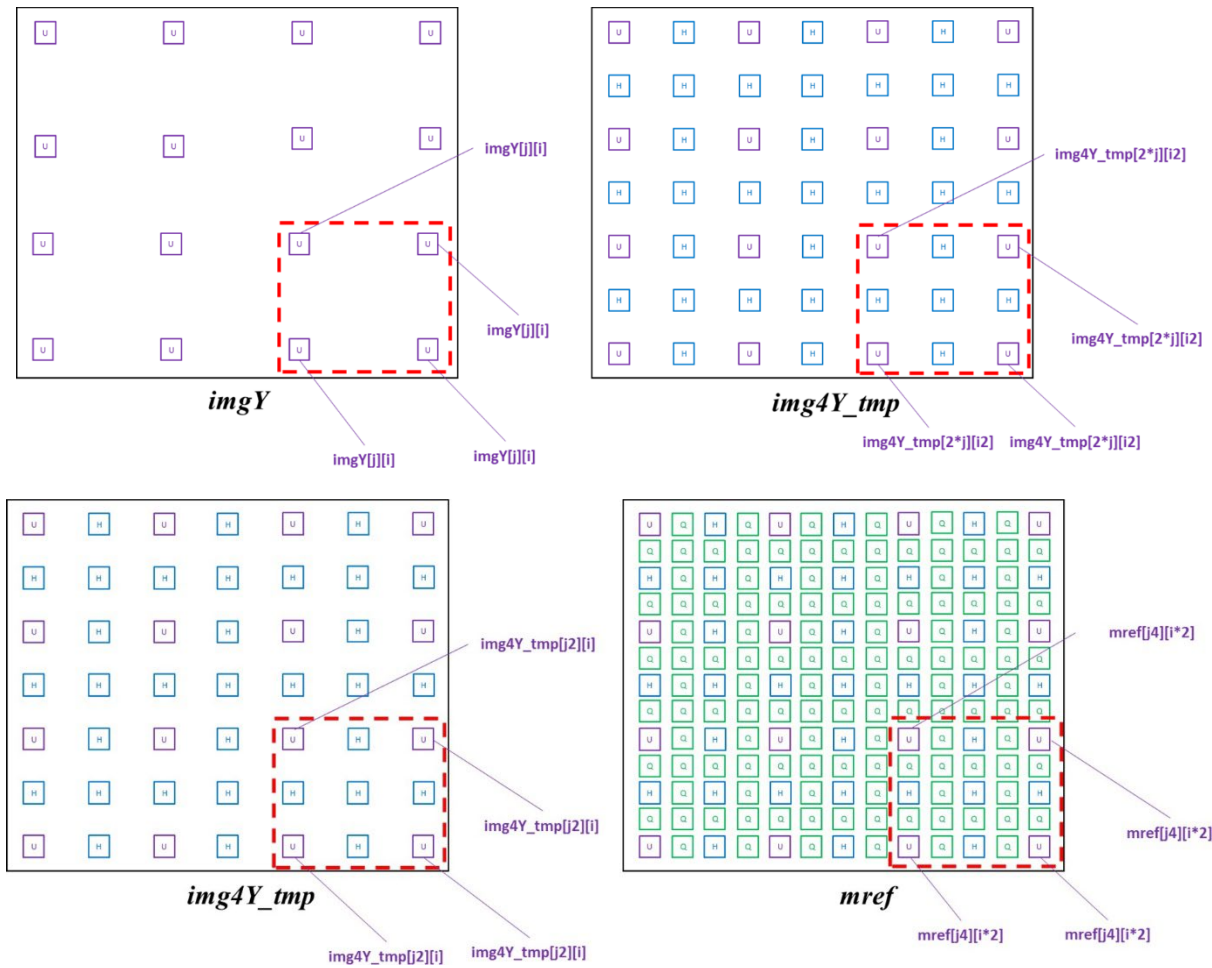
TML6, 57-58 (lines 563-592); Ramchandran, ¶136.

In steps two and three (shown below), the *oneforthpix()* function interpolates values for additional half sub-pixels (blue)—some based on unit pixel values, and others based on previously-interpolated half sub-pixel values—and stores these values in *img4Y_tmp* (below upper right). The *oneforthpix()* function uses these half sub-pixel values and unit pixels (purple) in *img4Y_tmp* to interpolate values for quarter sub-pixels (green) and stores the quarter and half sub-pixel values in *mref* (below lower right).



TML6, 57-58 (lines 576-625); Ramchandran, ¶137.

Therefore, as shown below, *img4Y_tmp* and *mref* include half sub-pixels (blue) and quarter sub-pixels (green) that are situated within a rectangular bounded region (red) defined by four corner unit pixels (purple) with no intermediate pixels between the corners. Furthermore, the pixels (purple) and sub-pixels (blue and green) are arranged in rows and columns.



TML6, 57-58 (lines 554-592); Ramchandran, ¶¶137-138.

The '599 patent acknowledges TML6 discloses a method for sub-pixel interpolation to determine values for sub-pixels. EX-1001, Figures 13(a)-13(c), 8:50-9:48, 11:44-13:44; Ramchandran, ¶139. Furthermore, when describing TML6 with reference to Figure 13(a) (below), the '599 patent acknowledges that TML6 interpolates half sub-pixels “b” and “c” (blue) and quarter sub-pixels “d” to “g” (green), which are situated within a rectangular bounded region (red) defined by four

corner pixels “A” (purple) with no intermediate pixels between the corners and with the pixels and sub-pixels in rows and columns.

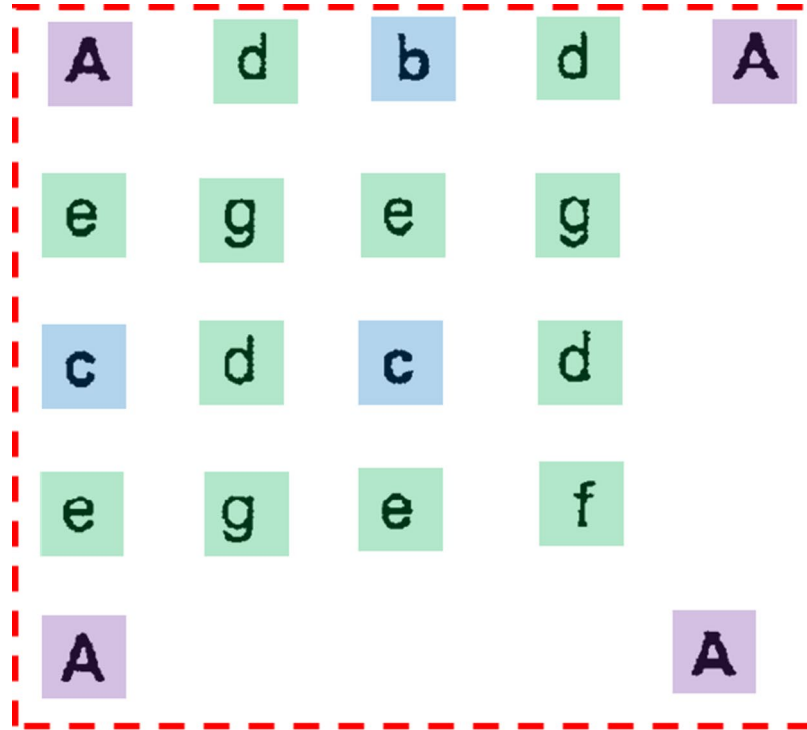


Fig. 13(a)

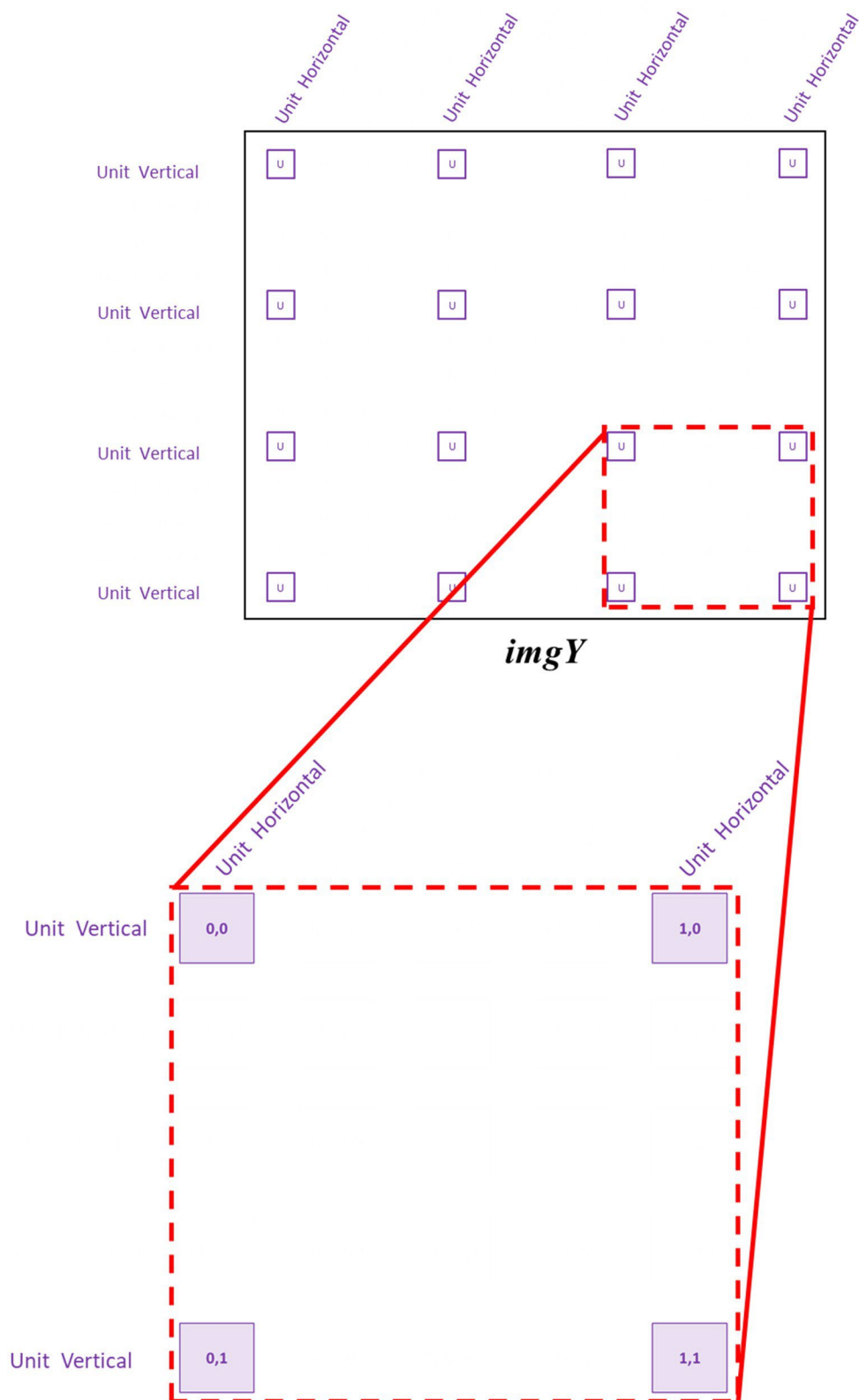
EX-1001, Figure 13(a), 8:50-9:48 (“[T]he letter A represents the location of pixels in the image data representing a frame of a video sequence.”), 11:44-13:44 (describing, with reference to Figure 13(a), TML6’s interpolation of half sub-pixels “b” and “c” and quarter sub-pixels “d”-“g” using “pixels A”). As shown above, the pixels “A” (purple) and sub-pixels “b” to “g” (blue and green) are arranged in rows and columns. Ramchandran, ¶140.

TML6 thus discloses “interpolati[ng]...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 arranged in rows and columns,” as recited. Ramchandran, ¶¶133-140.

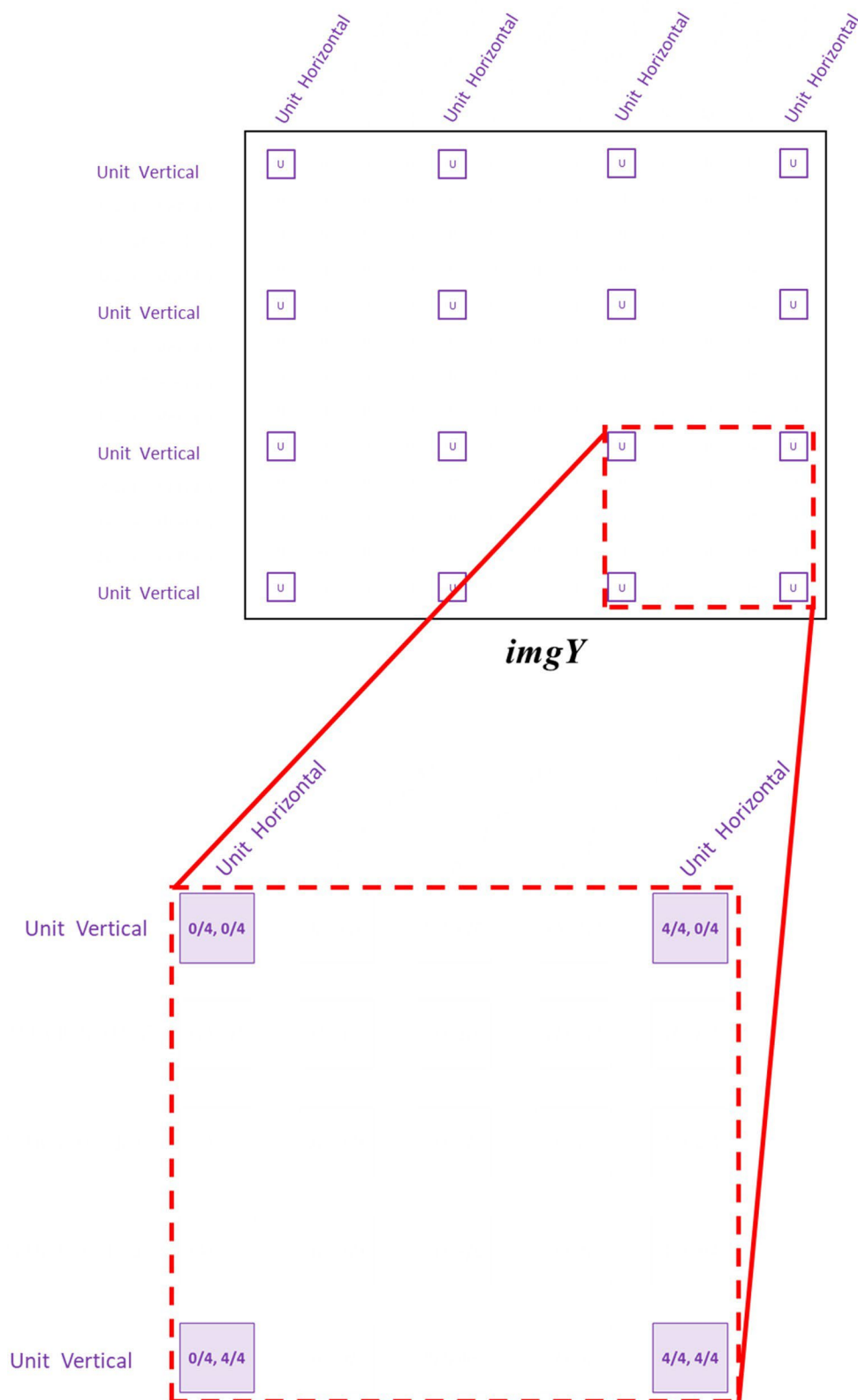
(ii) 1[pre][b]

If limiting, TML6 discloses pixel and sub-pixel locations being representable mathematically within the rectangular bounded region, as claimed. Ramchandran, ¶¶141-142.

As discussed for 1[pre][a] and shown below, *imgY* stores values for unit pixels U (purple), four of which are located in the rectangular bounded region (red).

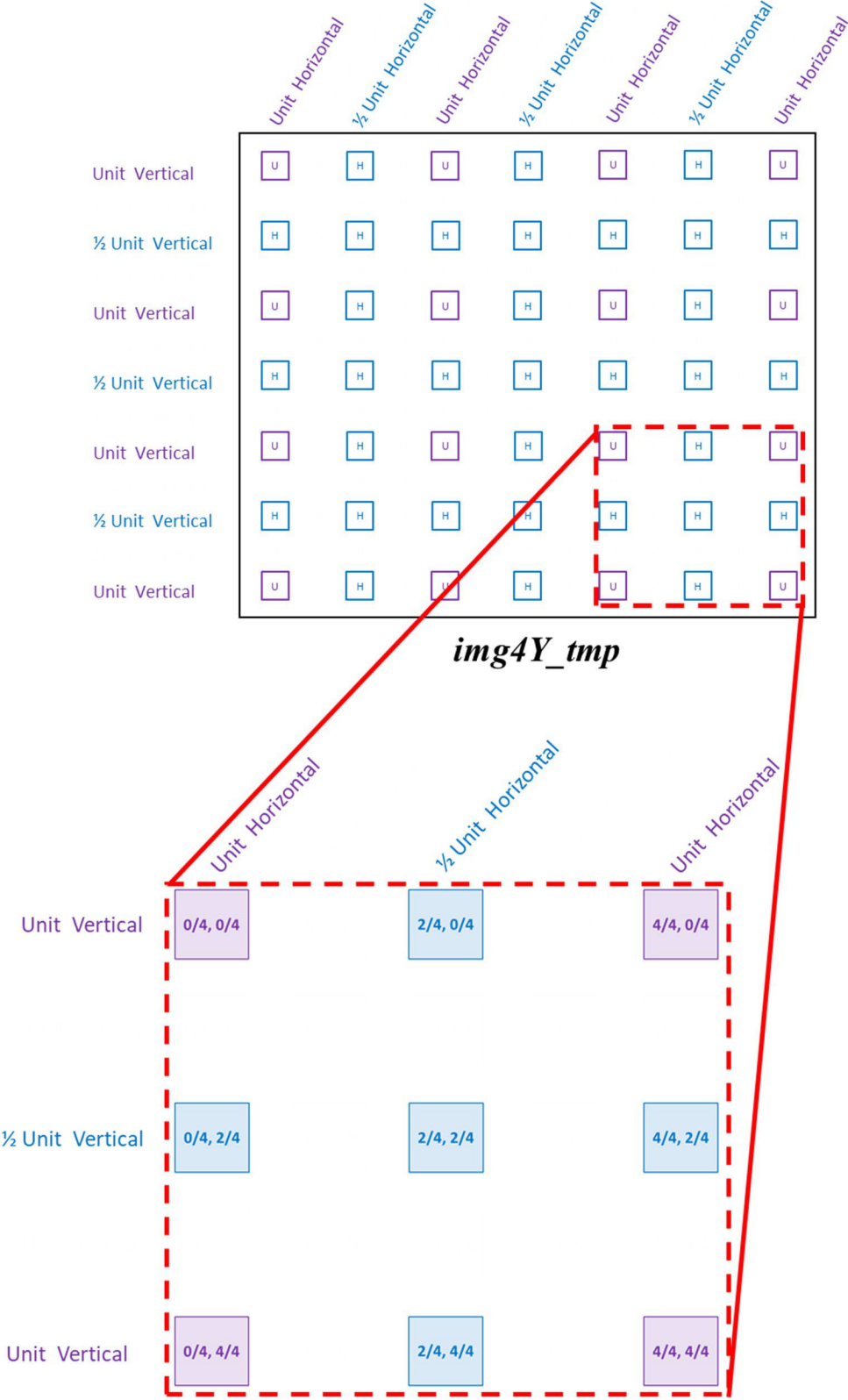


Within this bounded region, the four pixels U are representable mathematically using the Cartesian coordinate system. Namely, the upper left pixel U has the coordinates (0,0), the upper right pixel U has the coordinates (1,0), the lower left pixel U has the coordinates (0,1), and the lower right pixel U has the coordinates (1,1). These coordinates are equivalently represented with the notation “0/4” (which equals 0) and “4/4” (which equals 1), as shown below.



Ramchandran, ¶¶142-143.

As further described above for 1[pre][a] and shown below, the *oneforthpix()* function interpolates half sub-pixels (blue) based on the unit pixels U (purple) in *imgY* and stores these pixel and sub-pixel values in *img4Y_tmp*. Since the half sub-pixels (blue) are located halfway between the unit pixels U (purple) in the vertical and/or horizontal direction, they can be expressed with the coordinates shown below.



Ramchandran, ¶144.

Similarly, as described above for `l[pre][a]` and shown below, the *oneforthpix()* function interpolates quarter sub-pixels (green) based on the unit pixels U (purple) and the previously-interpolated half sub-pixels (blue) and stores the pixel and sub-pixel values in *mref*. Since the quarter sub-pixels (green) are located a quarter-way between the unit pixels U (purple) in the vertical and/or horizontal direction, they can be expressed with the coordinates shown below.



Ramchandran, ¶145.

Accordingly, TML6 discloses locations of unit pixels (purple) and sub-pixels (blue and green) are representable mathematically within the rectangular bounded region (red) using the coordinate notation $K/2^N$ for the horizontal coordinates and notation $L/2^N$ for the vertical coordinates. Namely, as shown above, the horizontal coordinates include:

$0/4$ ($K=0$; $N=2$),

$1/4$ ($K=1$; $N=2$),

$2/4$ ($K=2$; $N=2$),

$3/4$ ($K=3$; $N=2$), and

$4/4$ ($K=4$; $N=2$).

Similarly, the vertical coordinates include:

$0/4$ ($L=0$; $N=2$),

$1/4$ ($L=1$; $N=2$),

$2/4$ ($L=2$; $N=2$),

$3/4$ ($L=3$; $N=2$), and

$4/4$ ($L=4$; $N=2$).

Furthermore, K and L are positive integers having respective values between zero and 2^N (i.e., $2^2=4$).⁸ Ramchandran, ¶¶146-147.

Finally, $N=2$, which is a positive integer greater than one. Moreover, as noted above, TML6 interpolates up to quarter sub-pixels, and therefore, N represents a particular degree of sub-pixel value interpolation—i.e., interpolating quarter ($1/2^N$; $N=2$) sub-pixels. Ramchandran, ¶148.

Also, as shown in Figure 13(a) (below), the '599 patent confirms TML6 discloses locations of pixels (purple), half sub-pixels (blue), and quarter sub-pixels (green) are representable mathematically within the rectangular bounded region (red) using the coordinate notation $K/2^N, L/2^N$. Specifically, as shown below, K and L are integers having values between zero and 4 (i.e., 2^N ; $N=2$), and $N=2$, which is an integer greater than one and represents a particular degree of sub-pixel value interpolation (i.e., interpolating quarter ($1/2^N$; $N=2$) sub-pixels).

⁸ Claim 1 recites K and L are “**positive integers** having respective values between zero and 2^N ” yet later recites that K and L can equal **zero** ($1[a]-1[c]$, *infra*). Petitioner assumes K and L can be zero for this proceeding but reserves the right to argue Claim 1 is invalid as indefinite in other proceedings.

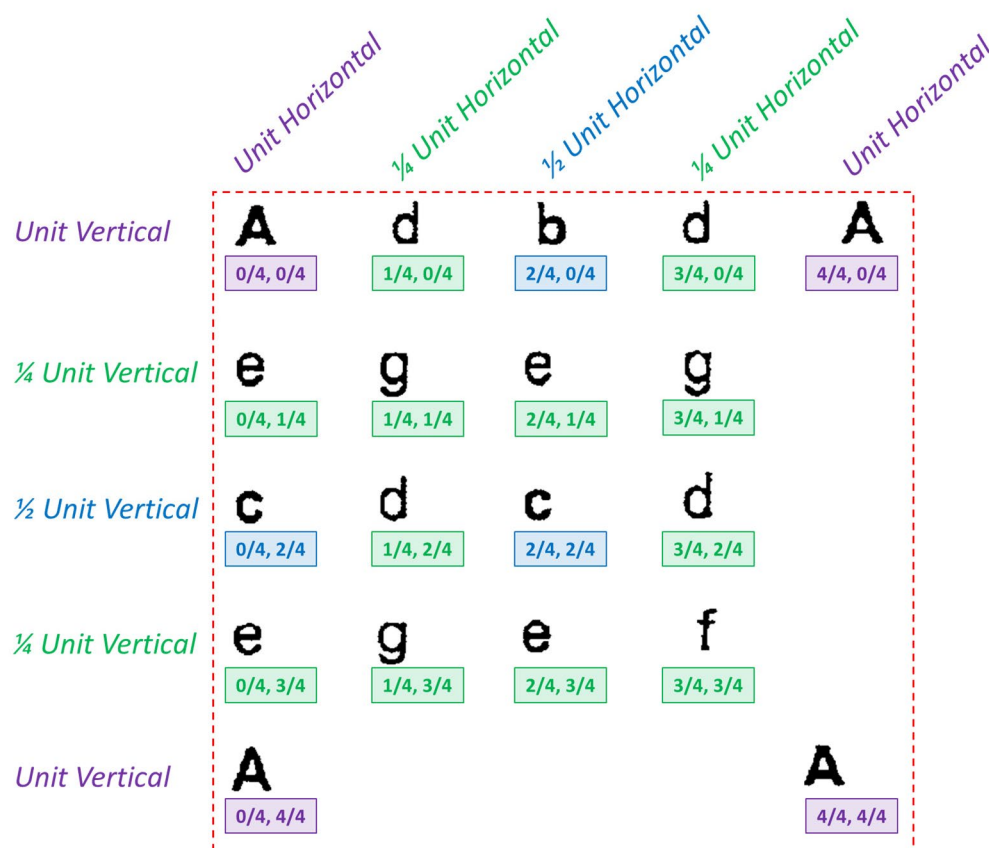


Fig. 13(a)

EX-1001, Figures 13(a)-13(c), 8:50-9:65 (“[T]he letter A represents the location of pixels in the image data representing a frame of a video sequence.... The term ‘unit horizontal location’ is used to describe the location of any sub-pixel that is constructed in a column of the original image data.... ‘Half horizontal location’ is used to describe the location of any sub-pixel that is constructed in a column that lies at half pixel resolution. Sub-pixels b, c, and e...fall into this category.... ‘Quarter horizontal location’ refers to any sub-pixel that is constructed

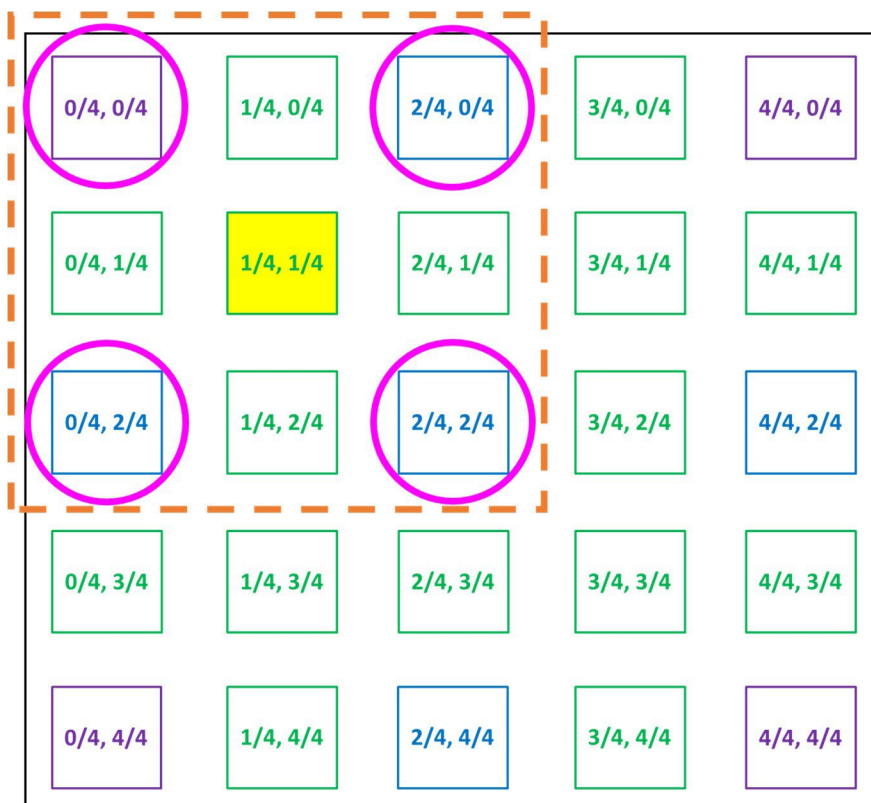
in a column which lies at quarter-pixel resolution, such as...sub-pixels d and g in FIG. 13a....”), 11:44-13:44; Ramchandran, ¶149.

(iii) 1[a]

The TML6-Fandrianto combination teaches 1[a]. Ramchandran, ¶150.

First, TML6 and Fandrianto both disclose “interpolating...value[s] for...sub-pixel[s] having co-ordinates with odd values of both K and L,” e.g., **K=1** and **L=1 or 3**, as recited. Ramchandran, ¶150. For example, as shown below, TML6’s *oneforthpix()* function interpolates the highlighted quarter sub-pixels (green) at the coordinates $(1/4, 1/4)$,⁹ i.e., $K/2^N, L/2^N$ ($K=1, L=1, N=2$).

⁹ Pixels and sub-pixels located at coordinates (x,y) are referred to as “(x,y) pixels” and “(x,y) sub-pixels.”



TML6, 58 (lines 593-625); This highlighted (1/4,1/4) sub-pixel (green) is interpolated according to a weighted average of the one pixel (purple) and three half sub-pixels (blue) referenced in 1[a] and circled above, with weights equal to “1.”

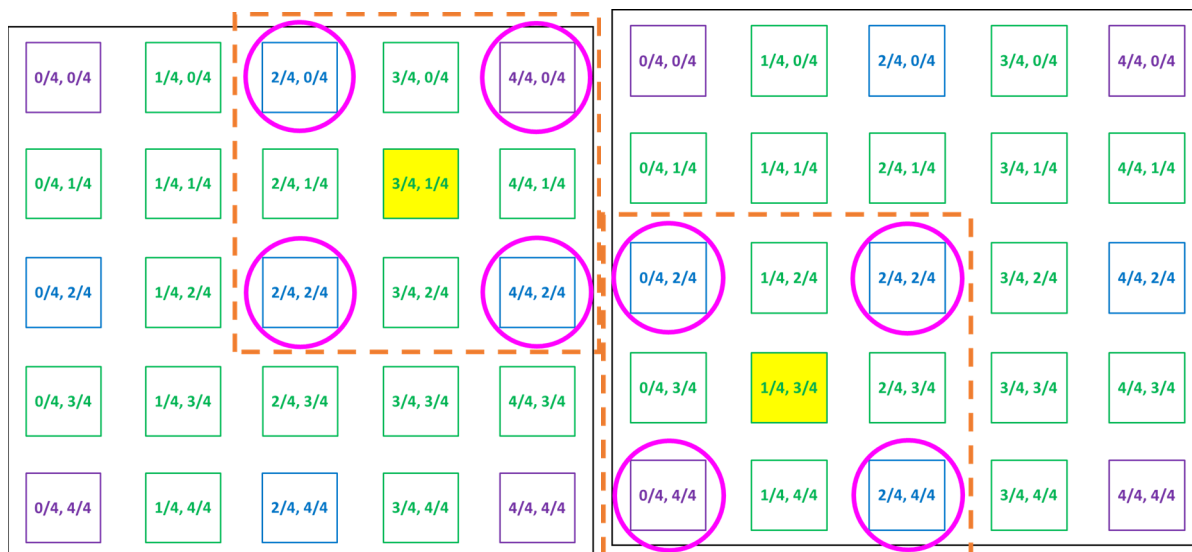
Specifically, the highlighted (1/4,1/4) sub-pixel is interpolated using a weighted average of:

- The “nearest-neighbouring pixel” (i.e., 0/4,0/4 pixel);
- The (2/4,2/4) sub-pixel “situated at the co-ordinates $\frac{1}{2}, \frac{1}{2}$ ” (i.e., $\frac{1}{2}, \frac{1}{2} = 2/4, 2/4$);

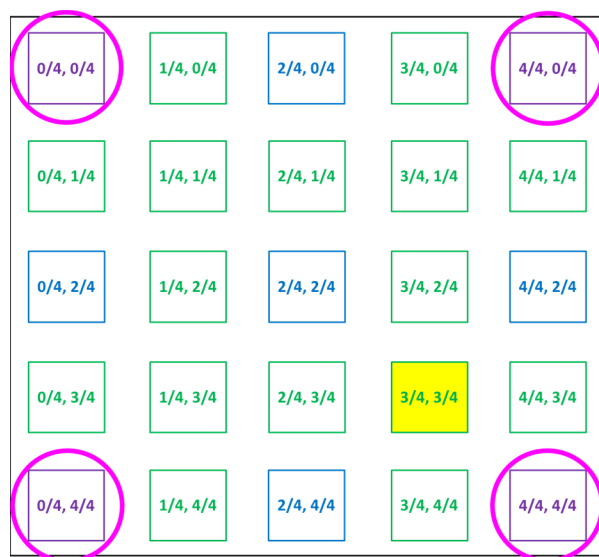
- The 0/4,2/4 sub-pixel “having co-ordinates with even values of K and L, including zero” (i.e., co-ordinates $K/2^N, L/2^N$; $K=0$; $L=2$; $N=2$) and “diagonally-opposed” to the 2/4,0/4 sub-pixel; and
- The 2/4,0/4 sub-pixel “having co-ordinates with even values of K and L, including zero” (i.e., co-ordinates $K/2^N, L/2^N$; $K=2$; $L=0$; $N=2$) and “diagonally-opposed” to the 0/4,2/4 sub-pixel.

TML6, 58 (lines 593-625). As shown above, these sub-pixels are within a quadrant (orange) of the rectangular bounded region defined by corner pixels having co-ordinates (1/2,1/2)—i.e., (2/4,2/4) sub-pixel (blue, circled)—and the nearest-neighbouring pixel—i.e., (0/4,0/4) pixel (purple and circled). Ramchandran, ¶¶150-151, 47-108.

Similarly, the highlighted (1/4,3/4) sub-pixel (below right) at the coordinates (1/4,3/4)—i.e., $K/2^N, L/2^N$ ($K=1, L=3, N=2$)—and the highlighted (3/4,1/4) sub-pixel (below left) at the coordinates (3/4,1/4)—i.e., $K/2^N, L/2^N$ ($K=3, L=1, N=2$)—along with corresponding quarter sub-pixels in other rectangular bounded regions are similarly interpolated as the (1/4,1/4) sub-pixel.



TML6, 58-59 (lines 593-632). Finally, the highlighted (3/4,3/4) sub-pixel is interpolated using the nearest four unit pixels (purple).



Id.; Ramchandran, ¶152.

As shown in Figure 13(a) (below), the '599 patent confirms TML6 interpolates (1/4,1/4) sub-pixel “g” (green) by using a weighted average of the same unit pixel A (purple) and three half sub-pixels b, c, and c (blue).

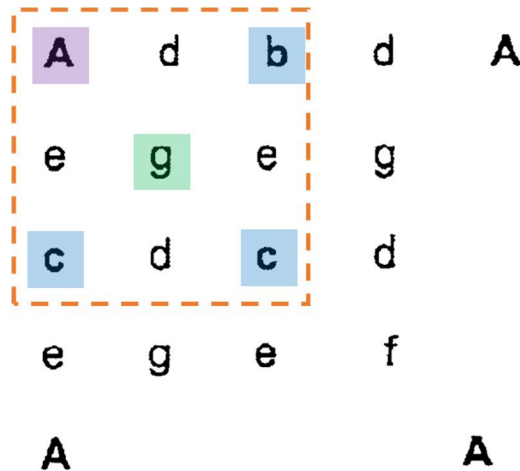


Fig. 13(a)

EX-1001, Figure 13(a), 12:58-65 (“Values for...g are computed using the value of the nearest original pixel A and the intermediate values of the three nearest-neighbouring half resolution sub-pixels [b, c, and c], according to $g = (1024A + 32b + 32c_1 + c_2 + 2048) / 4096$ ”), 12:66-13:3 (“ $\frac{1}{4}$ resolution sub-pixel f is interpolated by averaging the...4 closest pixels....”); Ramchandran, ¶153.

Thus, TML6 interpolates the highlighted $(\frac{1}{4}, \frac{1}{4})$, $(\frac{1}{4}, \frac{3}{4})$, and $(\frac{3}{4}, \frac{1}{4})$ sub-pixels in the figures above by using a weighted average of (1) “the value of a nearest-neighbouring pixel,” (2) “the value of the sub-pixel situated at co-ordinates $\frac{1}{2}, \frac{1}{2}$,” and (3) the “the values of a pair of diagonally-opposed sub-pixels having co-ordinates with even values of both K and L, including zero.” However, TML6 arguably does not teach interpolating the highlighted $(\frac{1}{4}, \frac{1}{4})$, $(\frac{1}{4}, \frac{3}{4})$, $(\frac{3}{4}, \frac{1}{4})$, or $(\frac{3}{4}, \frac{3}{4})$ sub-pixel according to a predetermined choice of (1) a weighted average

of a pair of the nearest-neighbouring pixel and the sub-pixel situated at co-ordinates $(1/2, 1/2)$ and (2) a weighted average of the pair of diagonally-opposed sub-pixels having co-ordinates with even values of both K and L, including zero. If TML6 does not suggest this feature, TML6 and Fandrianto together do. Ramchandran, ¶154.

Namely, Fandrianto, in Figure 5 (below), shows four unit pixels X (purple within a rectangular bounded region (red box)) and interpolates a value for the highlighted $(1/4, 3/4)$ sub-pixel Q2 (green).

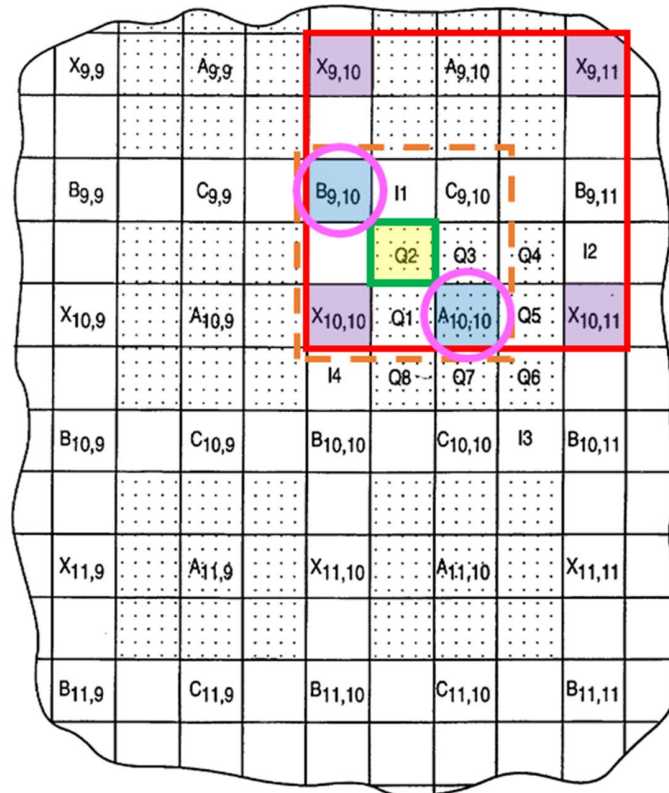


FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:58-9:32, 10:51-58. This $(1/4, 3/4)$ sub-pixel Q2 (green) has co-ordinates $K/2^N, L/2^N$ ($K=1, L=3, N=2$) with odd values of K and L

and is interpolated by using a weighted average of diagonally-opposed sub-pixels—i.e., $(0/4, 2/4)$ sub-pixel B and $(2/4, 4/4)$ sub-pixel A (both blue, circled)). The $(0/4, 2/4)$ sub-pixel B and the $(2/4, 4/4)$ sub-pixel A have co-ordinates $K/2^N, L/2^N$ ($K=0$ or 2 , $L=2$ or 4 , $N=2$) with even values of K and L , including zero. Fandrianto, 8:58-9:32, 10:51-58, 8:36-10:58; Ramchandran, ¶¶160.¹⁰ As shown above, the diagonally-opposed $(0/4, 2/4)$ sub-pixel B and $(2/4, 4/4)$ sub-pixel A are within a quadrant (orange box) of the rectangular bounded region (red box) defined by corner pixels having co-ordinates $(1/2, 1/2)$ —i.e., $(2/4, 2/4)$ sub-pixel $C_{9,10}$ —and the nearest-neighbouring pixel—i.e., $(0/4, 4/4)$ pixel $X_{10,10}$. Ramchandran, ¶¶155-159.

Fandrianto alternatively teaches interpolating highlighted $(1/4, 3/4)$ sub-pixel Q2 using a weighted average of values of the nearest-neighbouring pixel—i.e., $(0/4, 4/4)$ pixel X (purple, circled)—and sub-pixel C, at co-ordinates $(1/2, 1/2)$ (blue, circled).

¹⁰ As explained *supra* §VII.B, A block 72 and B block 74 contain “A” and “B” data, respectively, because Fandrianto does not require a specific order for deriving pixels Q1-Q8. Ramchandran, ¶¶156-158 (citing Fandrianto, 7:26-8:20, 10:51-58, Figures 3B, 5).

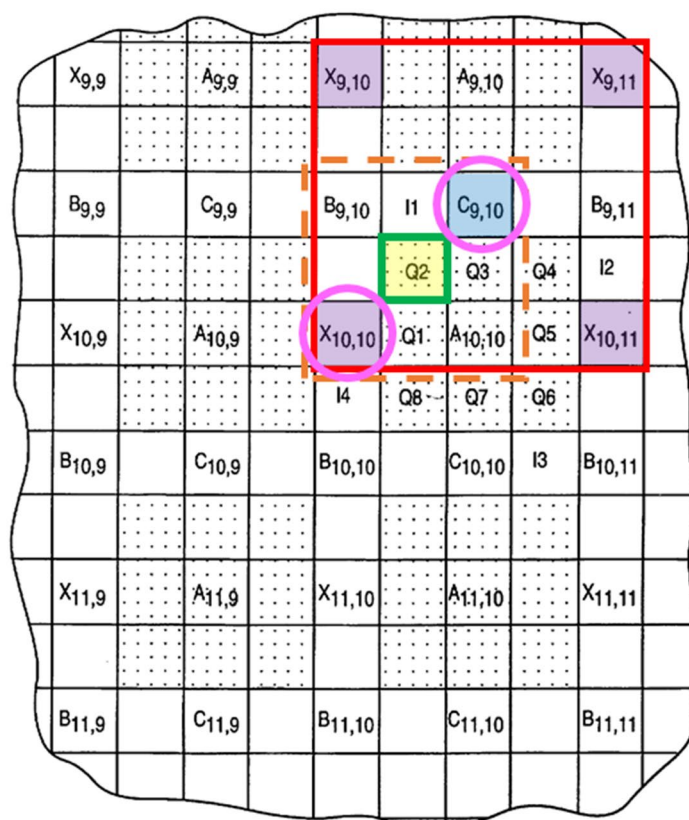
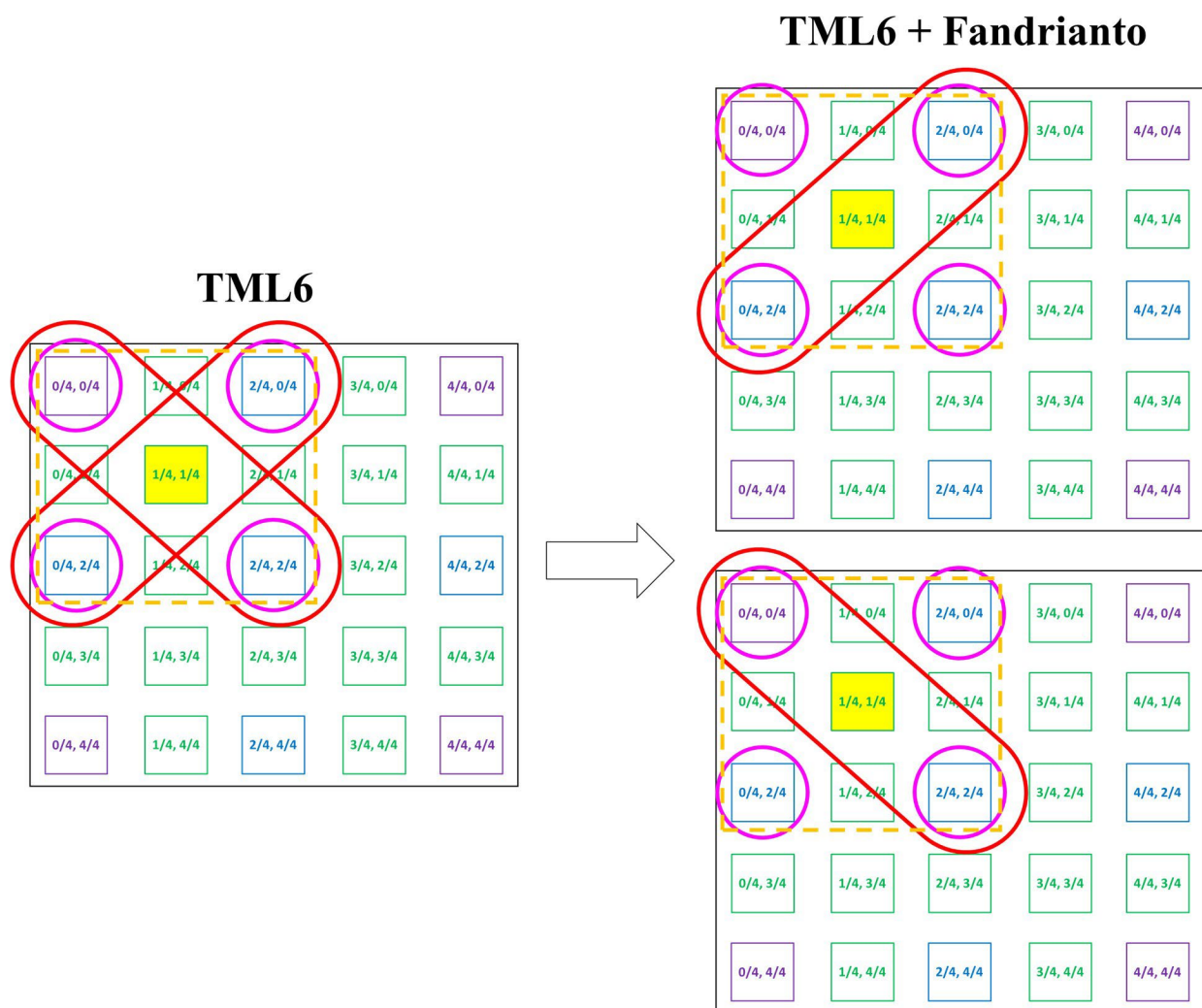


FIG. 5

Fandrianto, Figure 5, 5:26-32, 7:52-55 (“[W]hen **diagonal interpolation** is also used, all quarter pixel estimation search blocks [including sub-pixel Q2 (green)] are interpolated from the full [unit pixel X (purple)] and half pixel [C (blue)] estimation search blocks.”), 8:58-9:32, 10:51-58. Therefore, Fandrianto discloses diagonally interpolating (1/4,3/4) sub-pixel Q2 using the nearest-neighbouring pixel $X_{10,10}$ and the sub-pixel $C_{9,10}$ with coordinates (1/2,1/2), as recited. *Id.* Like the other disclosed interpolations, this interpolation is a weighted average of two values of the pixel $X_{10,10}$ and sub-pixel $C_{9,10}$, with weights equal to “1.” Fandrianto, Figure 5, 6:18-7:2, 8:36-10:50; Ramchandran, ¶161.

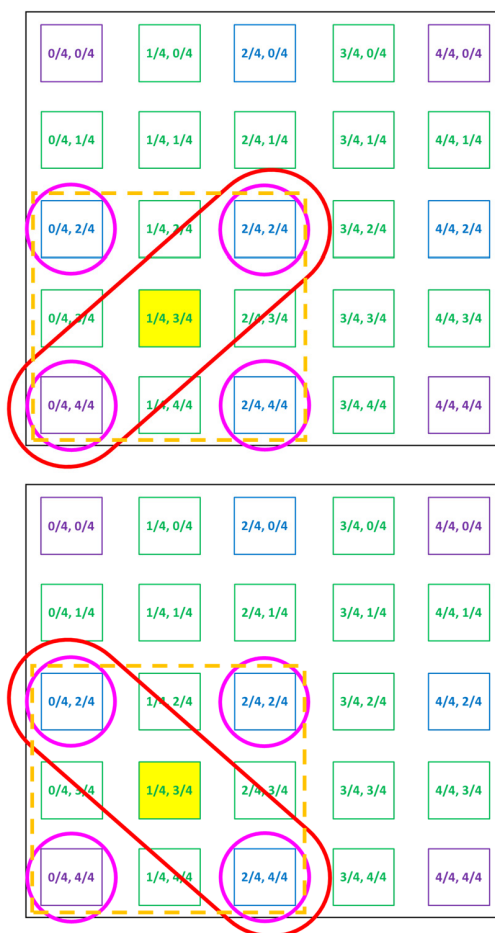
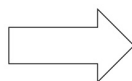
As discussed *supra* §VIII.A.1, a POSITA would have been motivated to use Fandrianto's method of interpolating the $(1/4, 3/4)$ sub-pixel Q2 (green) to interpolate the $(1/4, 1/4)$ sub-pixel, $(1/4, 3/4)$ sub-pixel, $(3/4, 1/4)$ sub-pixel, and/or $(3/4, 3/4)$ sub-pixel in TML6 by, for example, modifying the TML6 code to use Fandrianto's method, or by using Fandrianto's technique on its hardware. Thus, as shown below, the TML6-Fandrianto combination interpolates TML6's $(1/4, 1/4)$ sub-pixel (green) using (below right) a weighted average of a nearest-neighbouring $(0/4, 0/4)$ pixel (purple) and the $(2/4, 2/4)$ sub-pixel (blue) at co-ordinates $(1/2, 1/2)$, or (upper right) a weighted average of a pair of diagonally-opposed sub-pixels—i.e., $(0/4, 2/4)$ and $(2/4, 0/4)$ sub-pixels)—having co-ordinates with even values of both K and L, including zero—i.e., $K=0$ or 2 , $L=0$ or 2 —within a quadrant (orange box) of the rectangular bounded region defined by corner pixels having co-ordinates $(1/2, 1/2)$ and the nearest-neighbouring pixel, as in Fandrianto.

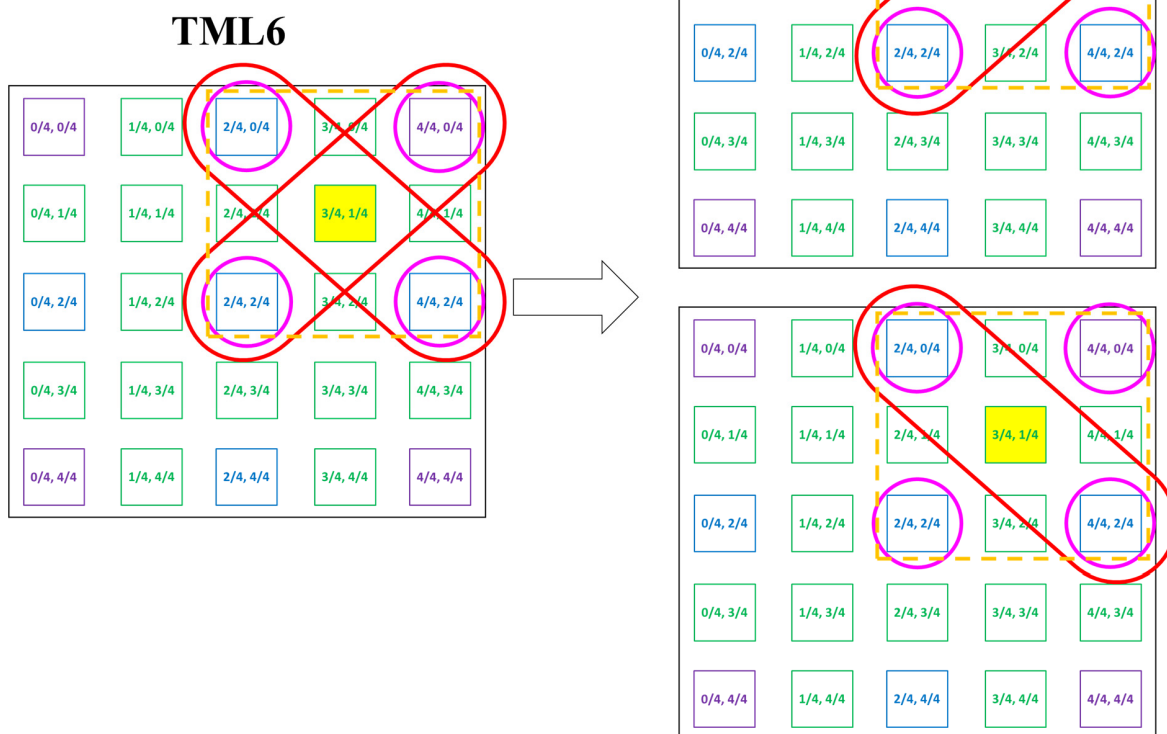


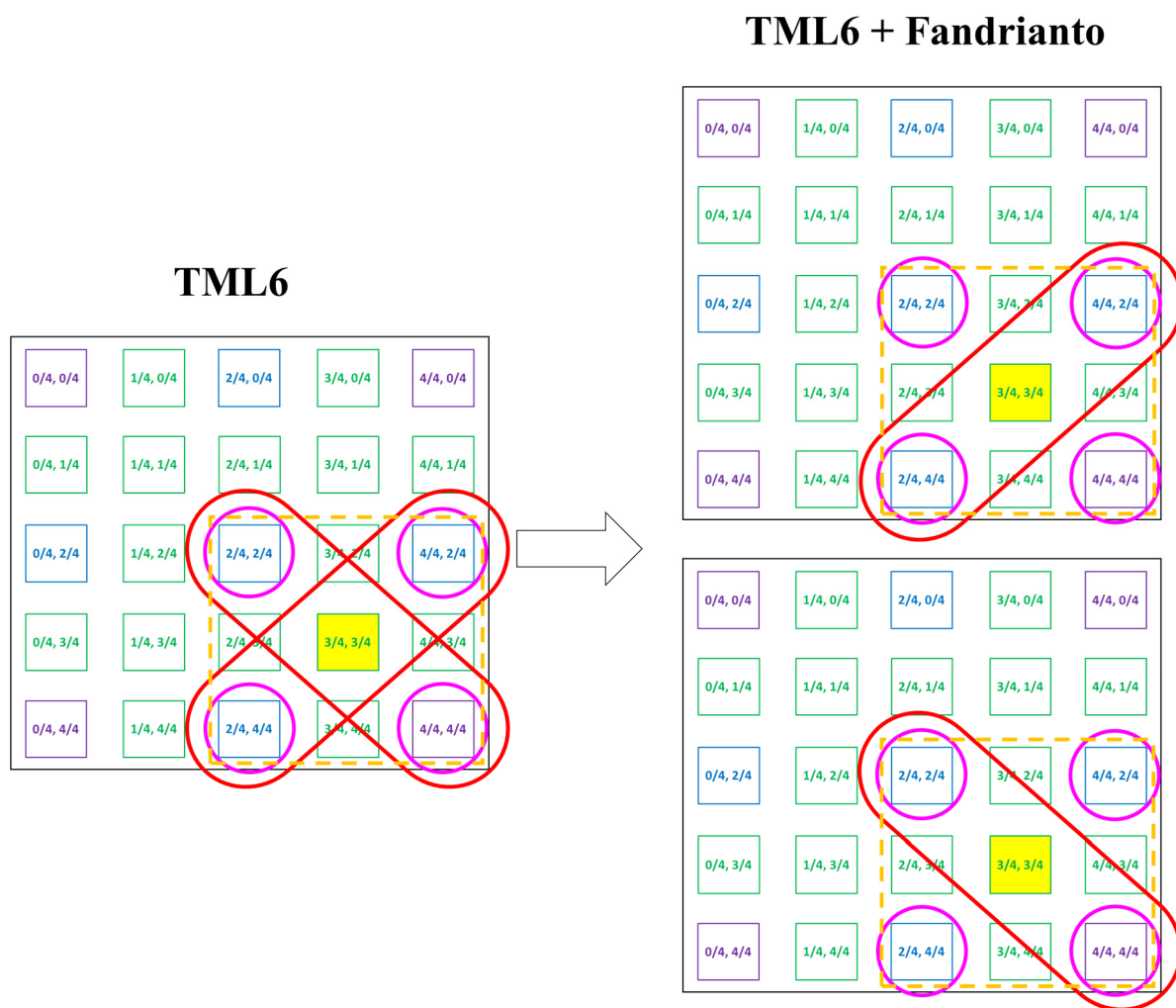
Similarly, as shown below, the TML6-Fandrianto combination interpolates TML6's (1/4,3/4), (3/4,1/4), and (3/4,3/4) sub-pixels.

TML6 + Fandrianto

TML6







Ramchandran, ¶162.

Furthermore, as discussed *supra* §VIII.A.1, it would have been obvious for a POSITA to interpolate TML6's (1/4,1/4), (1/4,3/4), (3/4,1/4), and/or (3/4,3/4) sub-pixels using a predetermined choice of the two diagonal methods of Fandrianto, with each method used to interpolate at least one sub-pixel. Fandrianto explicitly teaches a finite number of interpolation techniques for interpolating these sub-pixels, including vertical and two diagonal interpolations. Thus, it would have been obvious

to try these finite options, including making a predetermined choice between these two diagonal interpolation options to interpolate any of the $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and/or $(3/4, 3/4)$ sub-pixels. Revising TML6 or using Fandrianto's hardware to implement this modification would have been straightforward, and a POSITA would have reasonably expected the modification to yield the predictable result of interpolating $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and/or $(3/4, 3/4)$ sub-pixels based on values of nearby pixels and/or sub-pixels. Ramchandran, ¶163.

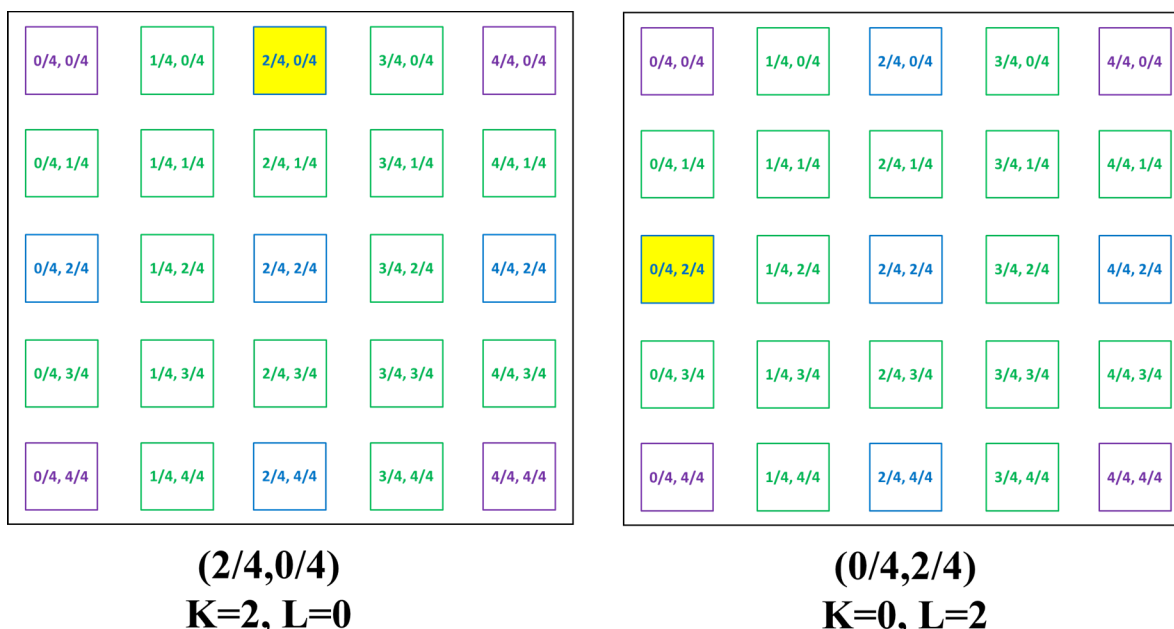
The TML6-Fandrianto combination thus interpolates $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and/or $(3/4, 3/4)$ sub-pixels —i.e., “sub-pixel[s] having co-ordinates with odd values of both K and L”—“according to a predetermined choice of a weighted average of” the pixels/sub-pixels recited in 1[a]. Ramchandran, ¶163.

(iv) 1[b]

TML6 discloses or renders obvious 1[b]. Ramchandran, ¶164.

As shown below, TML6's *oneforthpix()* function interpolates values for $(2/4, 0/4)$ sub-pixels having co-ordinates $(K/2^N, L/2^N)$ with K equal to an even value and L equal zero ($K=2, L=0, N=2$) and for $(0/4, 2/4)$ sub-pixels having co-ordinates $(K/2^N, L/2^N)$ with $K=0$ and L equal to an even value ($K=0, L=2, N=2$).¹¹

¹¹ TML6 and Fandrianto both disclose interpolating “sub-pixel values for sub-pixels” because they both interpolate corresponding sub-pixels in other rectangular

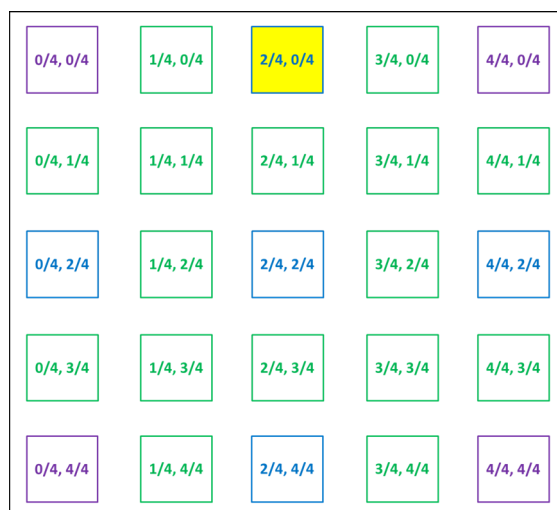


TML6, 57-58 (lines 538-592); Ramchandran, ¶164. As discussed for 1[a], these $(2/4, 0/4)$ and $(0/4, 2/4)$ sub-pixels are used to interpolate values of sub-pixels having

bounded regions in the same manner as described herein. See TML6, 93 (lines 1338-1344), 57-59 (lines 538-632); Fandrianto, Figure 5, 5:26-6:48; Ramchandran, ¶164. However, if 1[b] requires interpolating multiple sub-pixels having co-ordinates with K equal to an even value and L equal zero and multiple sub-pixels having co-ordinates with K equal to zero and L equal to an even value “within [the] rectangular bounded region” of 1[pre][a], Petitioner reserves the right to argue that this claim is invalid as indefinite in other proceedings because there is only one such sub-pixel when $N=2$. Ramchandran, ¶164.

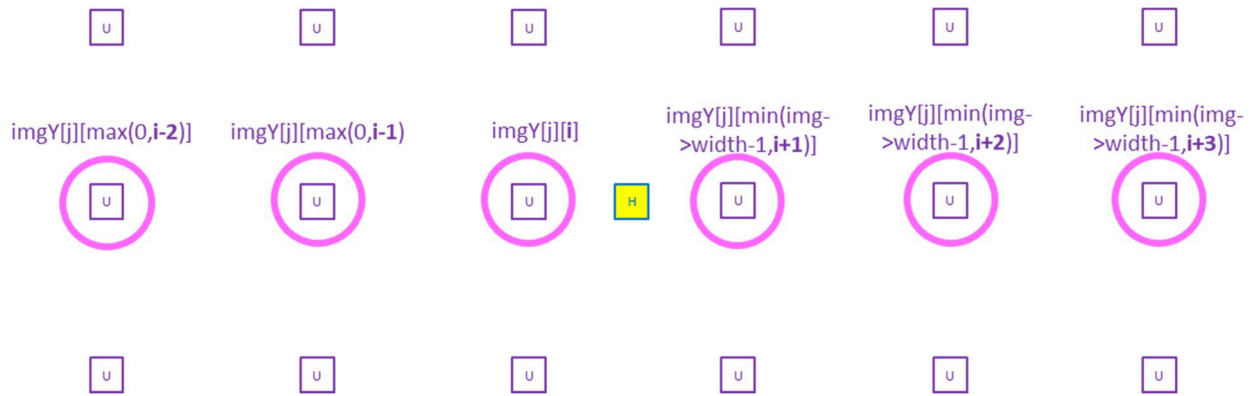
co-ordinates $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and/or $(3/4, 3/4)$ with odd values of both K and L (**K=1 or 3, L=1 or 3**). Ramchandran, ¶¶164, 47-108.

As illustrated below, TML6's *oneforthpix()* function interpolates values for these $(2/4, 0/4)$ and $(0/4, 2/4)$ sub-pixels using the values of unit pixels (purple) located in rows and columns, respectively. Specifically, *oneforthpix()* interpolates values for the highlighted $(2/4, 0/4)$ sub-pixels (blue) by using values of unit pixels that are stored in *imgY* and that are in the same row as the highlighted sub-pixel.



$(2/4, 0/4)$
K=2, L=0

TML6, 57-58 (lines 538-592); Ramchandran, ¶165. As shown below, these highlighted $(2/4, 0/4)$ sub-pixels (blue) are interpolated in the first step of *oneforthpix()* using a weighted sum of values of the three adjacent circled unit pixels (purple) to the left and the three adjacent circled unit pixels to the right of the highlighted sub-pixel.



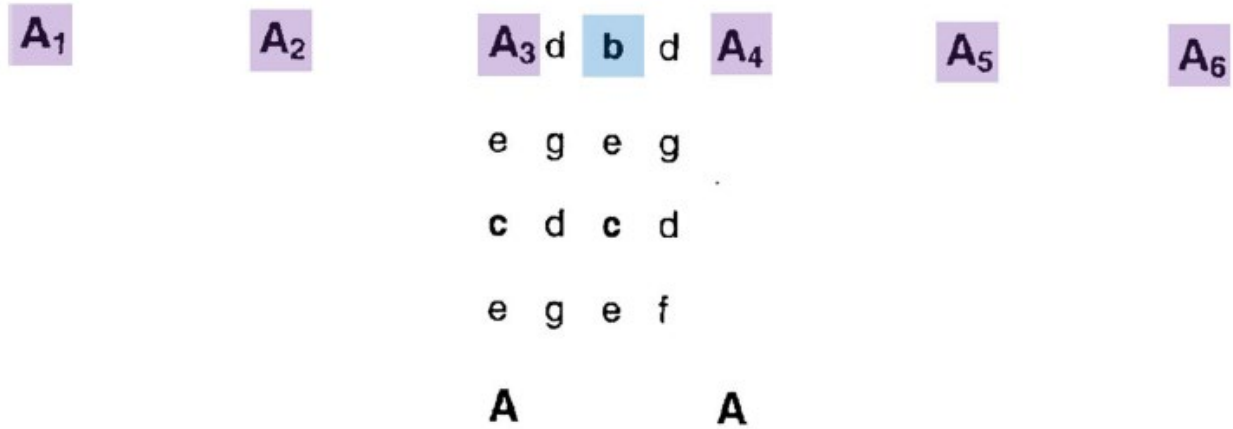
Ramchandran, ¶165 (discussing TML6, 57-58 (lines 563-575, 588)).

The weights given to the value of each unit pixel (purple) for this weighted sum are stored in **ONE_FOURTH_TAP** (defined in the file *image.h* in the *lencod* folder).¹² TML6, 57 (lines 568-570), 3 (lines 28-34). A value including this weighted sum is stored in *img4Y_tmp* and, after being rounded and brought into the dynamic range, is stored in *mref*. TML6, 57-58 (lines 572, 588); Ramchandran, ¶165.

Figure 13(b) of the '599 patent (below) confirms how TML6 interpolates these (1/2,0) sub-pixels. Namely, (2/4,0/4) sub-pixel “b” (blue) is interpolated “based upon the values of the 6 pixels (A₁ to A₆) [purple] situated in a row at unit

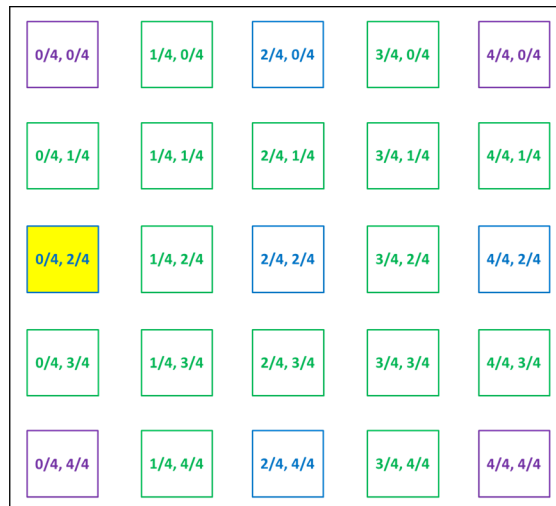
¹² As shown at line 572, the weighted sum is multiplied by 32, i.e., each of the weights from **ONE_FOURTH_TAP** in the weighted sum is multiplied by a scaling factor, 32. TML6, 57 (line 572).

horizontal locations and unit vertical locations symmetrically about b , as shown in FIG. 13*b*, according to the formula $b=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$.”



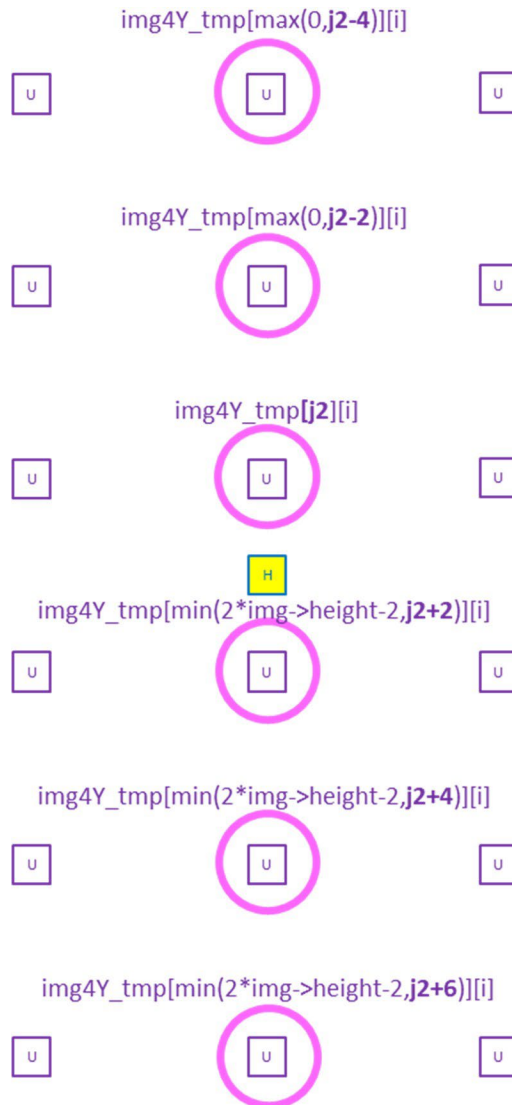
EX-1001, Figures 13(a)-13(b), 11:54-65; Ramchandran, ¶166.

Similarly, as illustrated below, *oneforthpix()* interpolates values for the highlighted (0/4,2/4) sub-pixels (blue) by using values of unit pixels (purple) stored in *imgY* and located in the same column as the highlighted sub-pixels.



(0/4,2/4)
K=0, L=2

Ramchandran, ¶167. These highlighted (0/4,2/4) sub-pixels (blue) are interpolated in *onefourthpix()*'s second step using a weighted sum of values of the three adjacent circled unit pixels (purple) above and the three adjacent circled unit pixels below the highlighted sub-pixel, as shown below.



Ramchandran, ¶167 (discussing TML6, 57-58 (lines 576-592)).

The weights of the values of the unit pixels (purple) for this weighted sum are stored in *ONE_FOURTH_TAP*, discussed above. TML6, 57-58 (lines 576-592), 3

(lines 28-34). A value including this weighted sum is stored in *img4Y_tmp* and, after being rounded and brought into the dynamic range, is stored in *mref*. TML6, 57-58 (lines 576-592); Ramchandran, ¶168.

Figure 13(c) of the '599 patent (below) confirms how TML6 interpolates these half sub-pixels. Specifically, (0/4,2/4) sub-pixel “c” (blue) is interpolated “based upon the values of the 6 pixels (A_1 to A_6) [purple] situated in a column at unit horizontal locations and unit vertical locations symmetrically about c [blue], according to the formula $c=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$.”

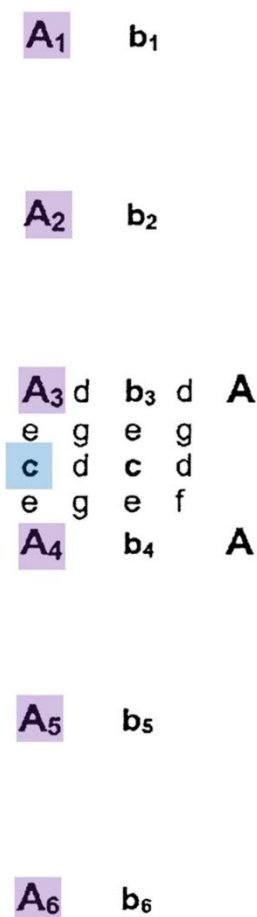


Figure 13(c)

EX-1001, Figures 13(a)-13(c), 11:66-12:17; Ramchandran, ¶169.

TML6 thus discloses using $(2/4,0/4)$ sub-pixels ($K=2$, $L=0$, $N=2$; “sub-pixels having co-ordinates with K equal to an even value and L equal zero”) and $(0/4,2/4)$ sub-pixels ($K=0$, $L=2$, $N=2$; “sub-pixels having co-ordinates with K equal to zero and L equal to an even value”) “in the interpolation of” $(1/4,1/4)$, $(1/4,3/4)$, $(3/4,1/4)$, and/or $(3/4,3/4)$ sub-pixels ($K=1$ or 3 , $L=1$ or 3 , $N=2$; “sub-pixels having co-ordinates with odd values of both K and L ”) “using weighted sums of the [pixel]

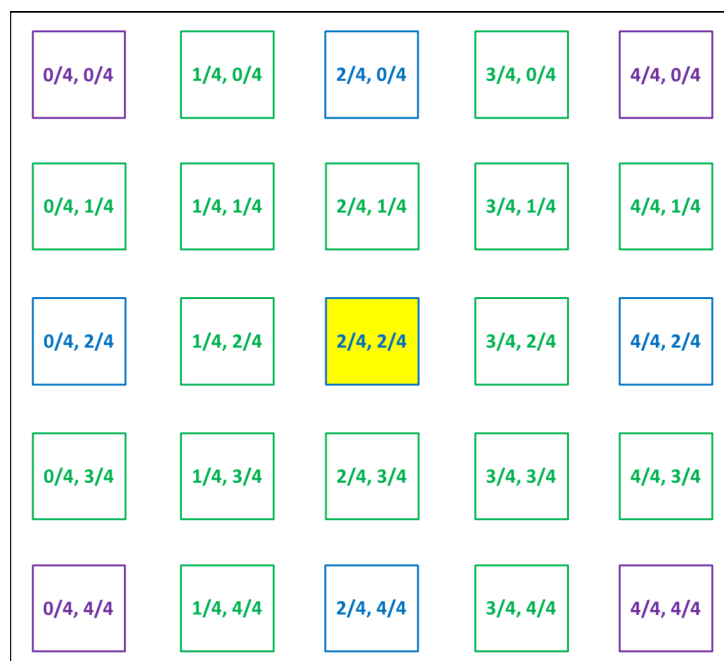
values” derived from weights stored in *ONE_FOURTH_TAP*, as recited. Ramchandran, ¶¶164-168.

(v) 1[c]

The TML6-Fandrianto combination teaches 1[c]. Ramchandran, ¶170.

As shown below, TML6’s *oneforthpix()* function interpolates values for (2/4,2/4) sub-pixels (blue) (highlighted below) having co-ordinates ($K/2^N, L/2^N$; $K=L=N=2$) with even values of both K and L.¹³

¹³ As discussed for 1[b], TML6 and Fandrianto disclose “interpolating sub-pixel values for sub-pixels” in other rectangular bounded regions. If 1[c] requires interpolating multiple sub-pixels in one rectangular bounded region with even values of K and L, Petitioner reserves the right to argue that the claim is invalid as indefinite in other proceedings since there is only one such sub-pixel when $N=2$.



(2/4,2/4)
K=2, L=2

TML6, 57-58 (lines 576-592). As discussed for 1[a], these (2/4,2/4) sub-pixels are “situated at co-ordinates $\frac{1}{2}, \frac{1}{2}$ ” and are used in the interpolation of the values for the (1/4,1/4), (1/4,3/4), (3/4,1/4), and/or (3/4,3/4) sub-pixels having coordinates (K/2^N,L/2^N; K=1 or 3, L=1 or 3, N=2) with odd values of both K and L. Ramchandran, ¶¶170, 47-108.

The highlighted (2/4,2/4) sub-pixels (blue) (below) are interpolated in the second step of the *oneforthpix()* function using a weighted sum (hereinafter “first weighted sum”) of:

- The values of the circled $(2/4,0/4)$ sub-pixel and $(2/4,4/4)$ sub-pixel (blue) (in the orange box) having co-ordinates $(K/2^N, L/2^N; K=2, L=0 \text{ or } 4, N=2)$ with K equal to an even value and L equal to zero,¹⁴ and

¹⁴ Element 1[c] arguably requires interpolating sub-pixels (with even values of K and L) using a weighted sum of:

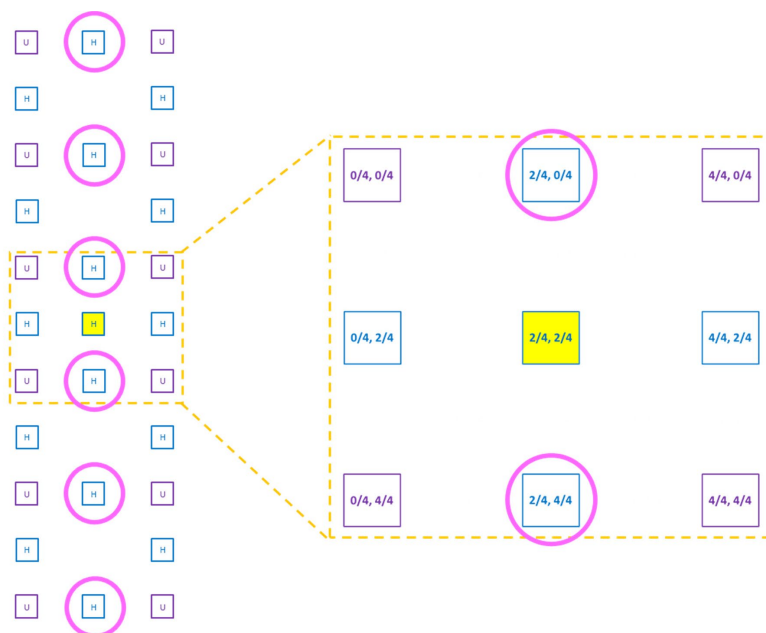
- (1) at least two sub-pixels having coordinates with K equal to an even value and L equal to zero (both within the same bounded region), and
- (2) corresponding sub-pixels in adjacent rectangular bounded regions.

However, the '599 patent interpolates the $(2/4,2/4)$ sub-pixel c using a weighted sum of:

- (1) only one sub-pixel having coordinates with K equal to an even value (i.e., 2) and L equal to zero (i.e., the $2/4,0/4$ sub-pixel b_3) and a second $2/4,4/4$ sub-pixel b_4 having coordinates with K equal to an even value (i.e., 2) and L equal to four (and not zero),
- (2) and corresponding sub-pixels b_1 and b_2 and sub-pixels b_5 and b_6 in adjacent rectangular bounded regions.

EX-1001, Figures 14(a)-14(c), 39:32-54 (“ $c=(b_1-5b_2+20b_3+20b_4-5b_5+b_6+512)$ ”). If the specification supports this claim, TML6 interpolates these sub-pixels in the same

- The values of the circled $(2/4,0/4)$ sub-pixel and $(2/4,4/4)$ sub-pixel having corresponding co-ordinates in immediately adjacent rectangular bounded regions.



First Weighted Sum

TML6, 57-58 (lines 576-592). A value with this first weighted sum is stored in *img4Y_tmp* and, after being rounded and brought into the dynamic range, stored in *mref*. TML6, 57-58 (lines 576-592); Ramchandran, ¶171.

Figure 13(c) of the '599 patent (below) confirms how TML6 interpolates $(2/4,2/4)$ sub-pixels c (blue).

way, as described below and confirmed by the '599 patent. EX-1001, Figure 13(c), 11:66-12:17 (“ $c=(b_1-5b_2+20b_3+20b_4-5b_5+b_6)$ ”).

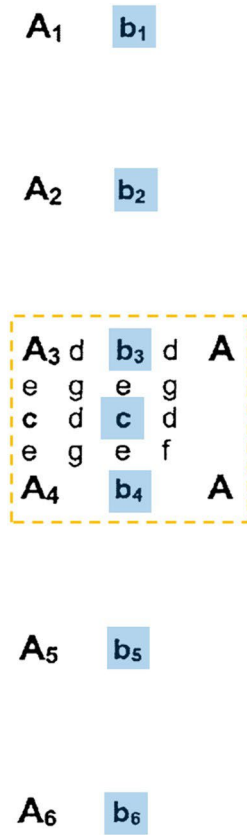
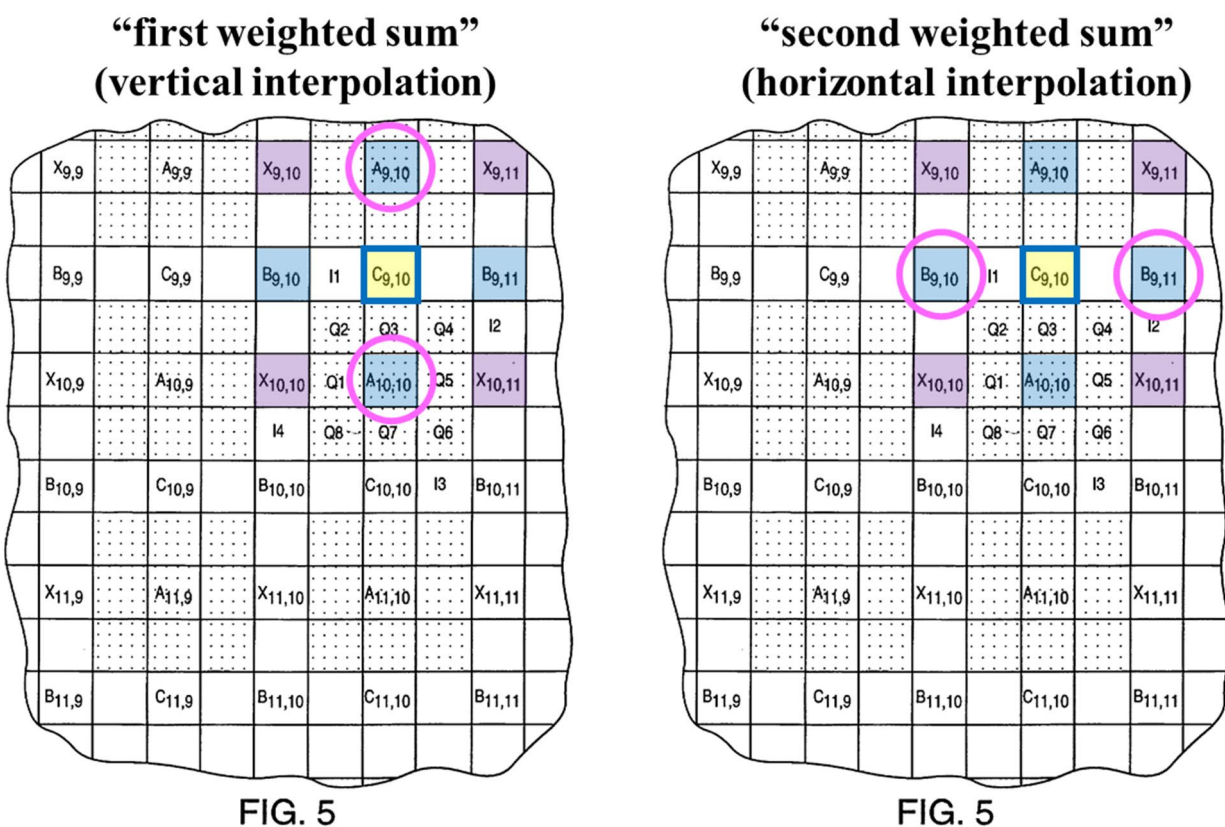


Figure 13(c)

EX-1001, Figure 13(c), 11:66-12:17 (“ $c=(b_1-5b_2+20b_3+20b_4-5b_5+b_6)$ ”);
Ramchandran, ¶172.

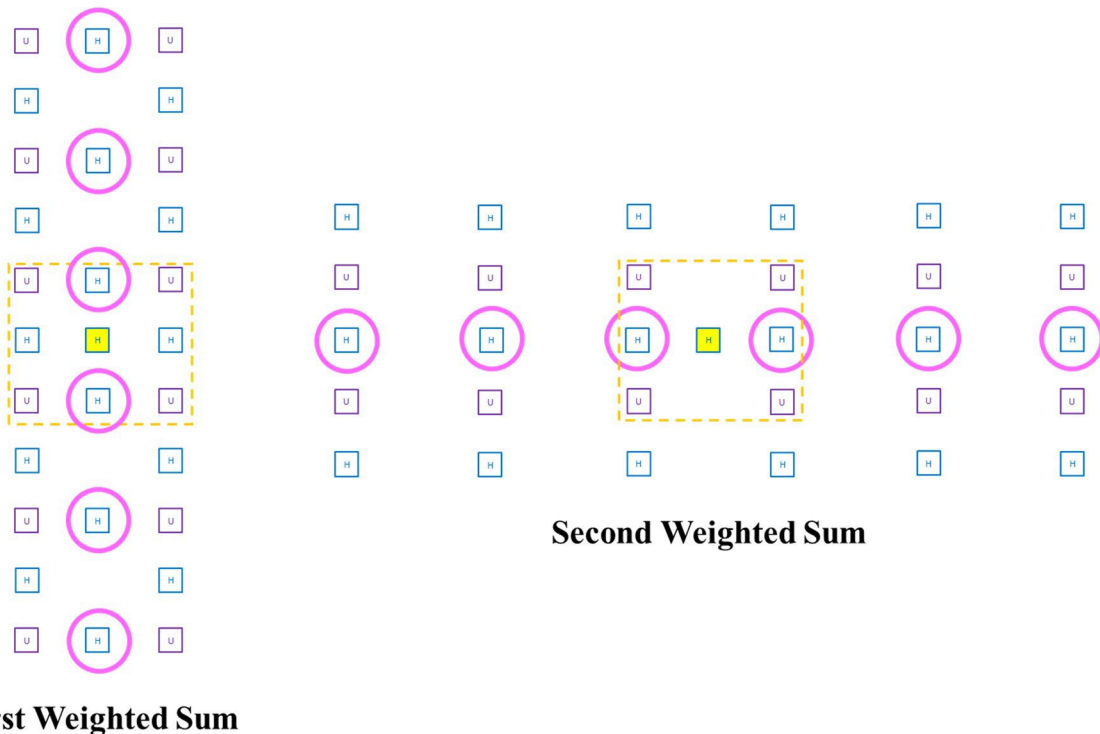
Fandrianto further interpolates the highlighted (2/4,2/4) sub-pixel C (blue, below) either by (1) averaging the circled (2/4,0/4) sub-pixel A above and the (2/4,4/4) sub-pixel A below the highlighted sub-pixel (“vertical interpolation”) or (2) averaging the circled 0/4,2/4 sub-pixel B to the left and the (4/4,2/4) sub-pixel B to the right of the highlighted (2/4,2/4) sub-pixel C (“horizontal interpolation”).



Fandrianto, 5:26-32, 6:4-12, 6:49-54 (“The C block 76 is formed by progressive interpolation of preferably either the A block 72 vertically, or the B block 74 horizontally.”), 6:18-48, 8:36-10:50. Accordingly, Fandrianto interpolates the highlighted (2/4,2/4) sub-pixels using either the “first weighted sum” of sub-pixel values having co-ordinates with K equal to an even value 2 and L=0 (“A block”) or another weighted sum (“second weighted sum”) of sub-pixel values having co-ordinates with K equal to zero and L equal to an even value (“B block”), with weights in each weighted sum equal to “1.” *Id.*; Ramchandran, ¶173.

As discussed *supra* §VIII.A.1, it would have been obvious to modify the teachings of TML6 based on the teachings of Fandrianto such that *oneforthpix()*

interpolates the 2/4,2/4 sub-pixels using a predetermined choice of either the first weighted sum of TML6 (summing six values of sub-pixels having co-ordinates with K equal to an even value (two) and L equal to zero) or the second weighted sum of TML6 as modified by Fandrianto (also summing six values of sub-pixels, but horizontally like Fandrianto with sub-pixels having co-ordinates with K equal to zero and L equal to an even value (two)), as shown below.



Ramchandran, ¶174.

Namely, Fandrianto explicitly teaches that a finite number of interpolation techniques could have been used to interpolate the 2/4,2/4 sub-pixels, including vertical interpolation using a first weighted sum (as also used in TML6), and horizontal interpolation using a second weighted sum. Fandrianto, 6:4-12, 6:49-7:2

(“The C block 76 is formed by progressive interpolation of preferably either the A block 72 vertically, or the B block 74 horizontally.”). Thus, it would have been obvious to try these finite options, including calculating first and second weighted sums and using a predetermined choice between the two to interpolate the 2/4,2/4 sub-pixels in TML6. Further, revising TML6’s code to implement this modification would have been straightforward, and a POSITA would reasonably expect this modification to yield the predictable result of interpolating the 2/4,2/4 sub-pixels. In fact, the ’599 patent recognizes (with reference to Figure 14(a)) that interpolating the 2/4,2/4 sub-pixels “c” using either a vertical weighted sum of sub-pixels “b” or a horizontal weighted sum of sub-pixels “b” produces the same result. EX-1001, Figure 14(a), 36:20-54 (“[T]he use of intermediate values b in the horizontal direction leads to the same result as using intermediate values b in the vertical direction.”), 39:32-54 (same); Ramchandran, ¶175.

Accordingly, the TML6-Fandrianto combination teaches a predetermined choice between the weighted sums of values, as claimed. Ramchandran, ¶¶173-175.

b. Claim 2

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for 1[a] and shown in the two annotated versions of Figure 5 below, Fandrianto interpolates highlighted (1/4,3/4) sub-pixel Q2—having co-

ordinates with odd values of both K ($K/2^N$, $K=1$, $N=2$) and L ($L/2^N$, $L=3$, $N=2$)—according to a predetermined choice of two weighted averages.

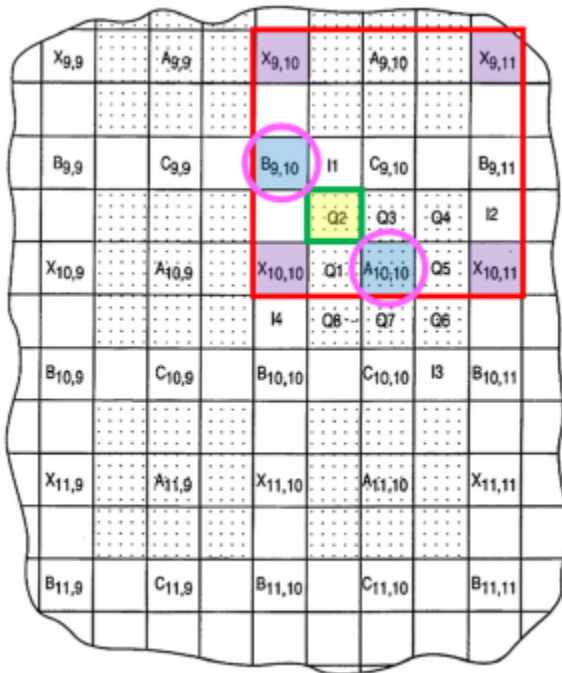


FIG. 5

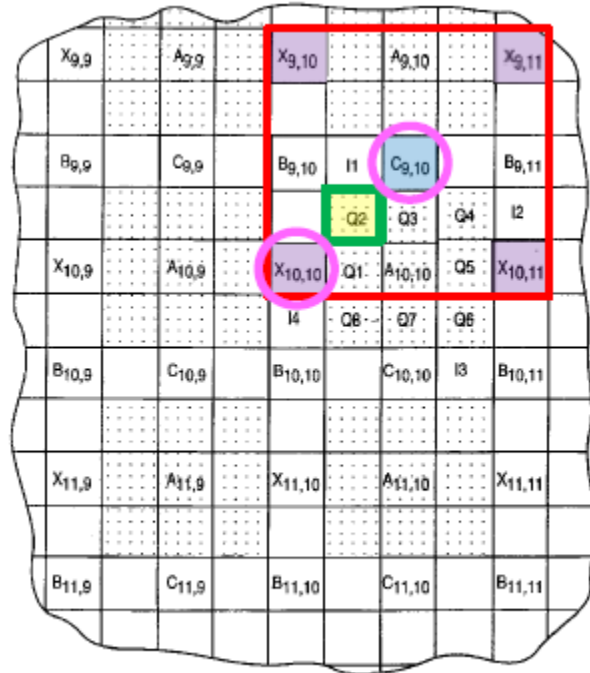


FIG. 5

Fandrianto, Figures 2, 5, 5:26-32, 6:18-7:55, 8:36-10:58. The first weighted average is calculated using the pair of diagonally-opposed (0/4,2/4) subpixel “B” and (2/4,4/4) sub-pixel “A” (blue; circled left) having coordinates with even values of K ($K/2^N$, $K=0$ or 2, $N=2$) and L ($L/2^N$, $L=2$ or 4, $N=2$). The second weighted average involves the nearest-neighbouring pixel X (purple; circled right) and (2/4,2/4) sub-pixel C (blue; circled right) at coordinates (1/2,1/2). Ramchandran, ¶176.

As discussed for 1[a] and shown below, when sub-pixel Q2 (green) is interpolated using the first weighted average ($Q2=(A+B)/2$) of the diagonally-

opposed (0/4,2/4) sub-pixel B and (2/4,4/4) sub-pixel A (blue; circled), the weights both equal “1.”¹⁵

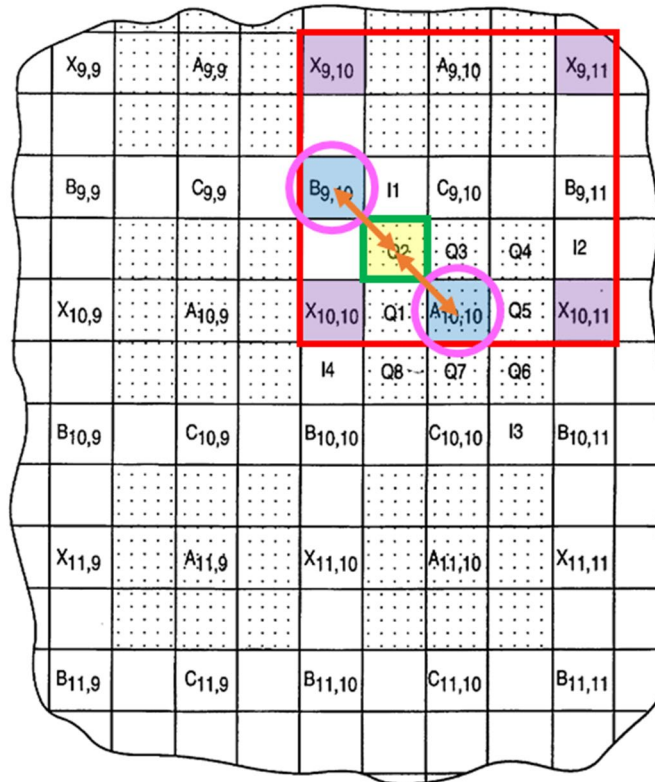


FIG. 5

Fandrianto, Figure 5, 8:58-9:32; 10:51-58; 8:36-10:58; Ramchandran, ¶177.

¹⁵ As explained *supra* §VII.B, A block 72 and B block 74 contain “A” and “B” data, respectively, because Fandrianto does not require a specific order for deriving pixels Q1-Q8. Ramchandran, ¶¶156-158 (citing Fandrianto, 7:26-8:20, 10:51-58, Figures 3B, 5).

Furthermore, as discussed for 1[a] and shown below, when sub-pixel Q2 (green) is interpolated using the second weighted average ($Q2=(X+C)/2$) of nearest-neighbouring pixel X (purple; circled) and sub-pixel C (blue; circled) at coordinates $(1/2,1/2)$, the weights of pixel X and sub-pixel C likewise both equal “1.”

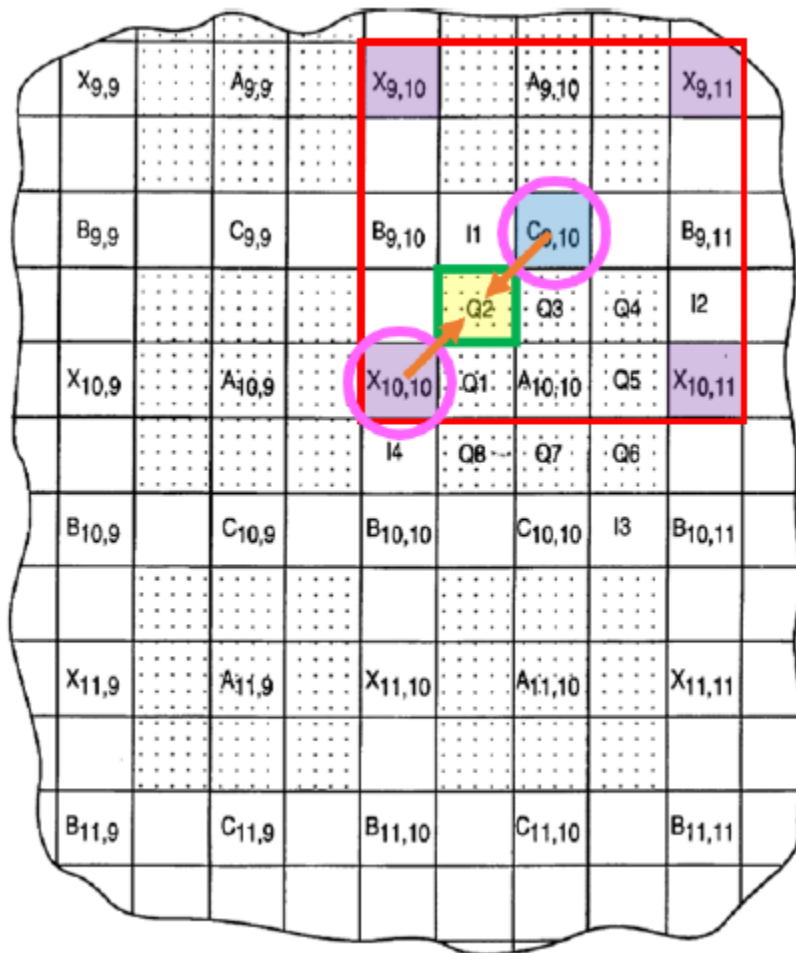


FIG. 5

Fandrianto, Figure 5, 5:26-32, 7:52-55 (“[W]hen **diagonal interpolation** is also used, all quarter pixel estimation search blocks [including sub-pixel Q2 (green)] are

interpolated from the full [unit pixel X (purple)] and half pixel [C (blue)] estimation search blocks.”), 8:58-9:32, 10:51-58; Ramchandran, ¶178.

Thus, when the chosen weighted average is the second weighted average involving the nearest-neighbouring pixel X (purple) and the (2/4,2/4) sub-pixel C (blue), the magnitudes of the first and second weights (“1”) in the weighted average are inversely proportional to the respective straight-line diagonal distances (orange arrows above) of the nearest-neighbouring pixel X (purple) and sub-pixel C (blue) to the interpolated highlighted (1/4,3/4) sub-pixel Q2 (K=1, L=3). Namely, pixel X (purple) and sub-pixel C (blue) are equidistant from sub-pixel Q2 (green), and thus, the magnitudes of their weights (“1”) are inversely proportional to their distances from sub-pixel Q2. Ramchandran, ¶179.

If the claim requires affirmatively “selecting” the relative magnitudes of the first and second weights of the weighted average to be inversely proportional to the respective distances of the pixel X and sub-pixel C to the sub-pixel Q2, such features would have been obvious. Namely, the closer pixel X and sub-pixel C are to sub-pixel Q2, the more likely their values are similar to subpixel Q2. Thus, the “closer” pixel X and sub-pixel C should have larger weights. Conversely, the farther pixel X and sub-pixel C are from Q2, the less likely the values are like Q2. Accordingly, the “farther” pixel X and sub-pixel C should have smaller weights. Thus, a POSITA would have known to select the relative magnitudes of the weights of the pixel X

and sub-pixel C in the weighted average to be inversely proportional to their respective distances from Q2.

Further, a POSITA would have known to use linear interpolation when interpolating between two points, as acknowledged in the '599 patent. EX-1001, 10:45-62, Tables 1-2; *see* VCEG-M28 (2-tap filters). In linear interpolation, the weights are inversely proportional to the distances from the known points to the interpolated point. Britannica, 356; Ramchandran, ¶180.

Accordingly, the TML6-Fandrianto combination interpolates TML6's $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and $(3/4, 3/4)$ sub-pixels having coordinates with odd values of both K and L ($K/2^N$, **K=1 or 3**, $N=2$) and L ($L/2^N$, **L=1 or 3**, $N=2$), as claimed. Ramchandran, ¶181.

c. Claim 3

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for Claim 2 and shown below, the first and second weights have equal values ("1") when the nearest-neighbouring pixel X (purple) and the sub-pixel C (blue)—at $(1/2, 1/2)$ —are equidistant from the $(1/4, 3/4)$ sub-pixel Q2 (green) ($K=1$, $L=3$) being interpolated.

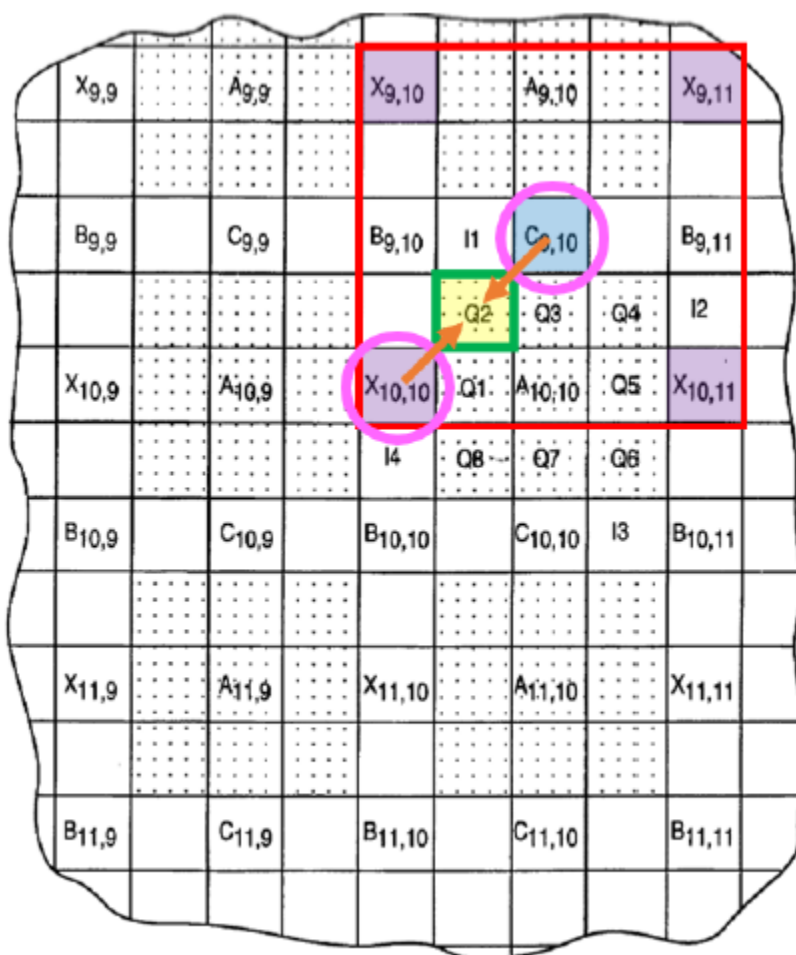


FIG. 5

Fandrianto, Figure 5; Claim 2, *supra*; Ramchandran, ¶182.

d. Claim 4

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for 1[a] and shown below, Fandrianto interpolates highlighted (1/4,3/4) sub-pixel Q2 having co-ordinates with odd values of both K ($K/2^N$; $K=1$, $N=2$) and L ($L/2^N$; $L=3$, $N=2$) according to a predetermined choice of two weighted averages.

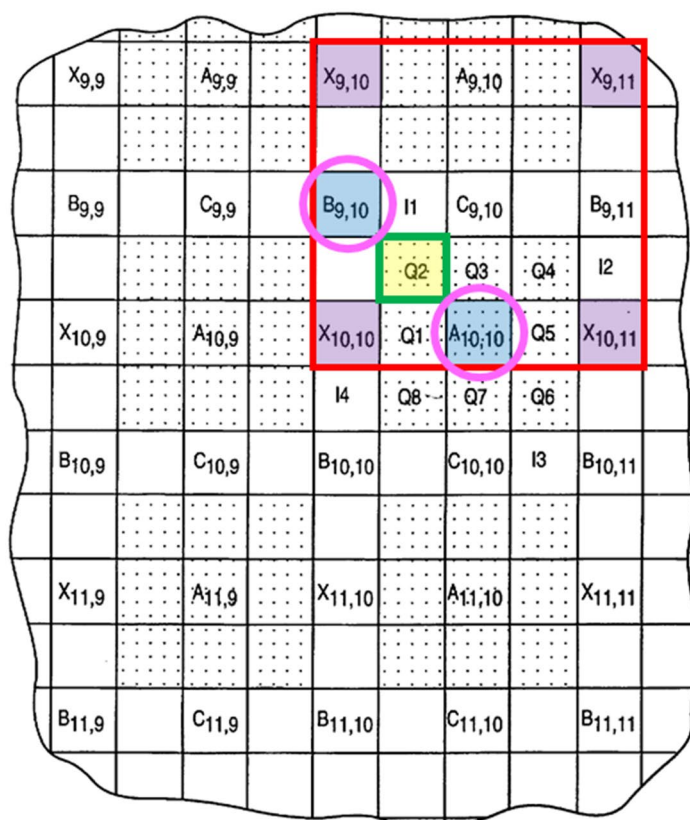


FIG. 5

Fandrianto, Figures 2, 5, 6:18-7:2, 8:36-10:58; Ramchandran, ¶183.

When the chosen weighted average involves the diagonally-opposed (0/4,2/4) sub-pixel B and (2/4,4/4) sub-pixel A (blue; circled above) having co-ordinates with even values of both K ($K/2^N$, $K=0$ or 2 , $N=2$) and L ($L/2^N$, $L=2$ or 4 , $N=2$), the magnitudes of the first and second weights (“1”) in the weighted average of sub-pixels A and B are inversely proportional to the straight-line diagonal distances (orange arrows) of A and B (having co-ordinates with even values of both K and L) to the highlighted sub-pixel Q2 (green) (having co-ordinates with odd values of K and L).

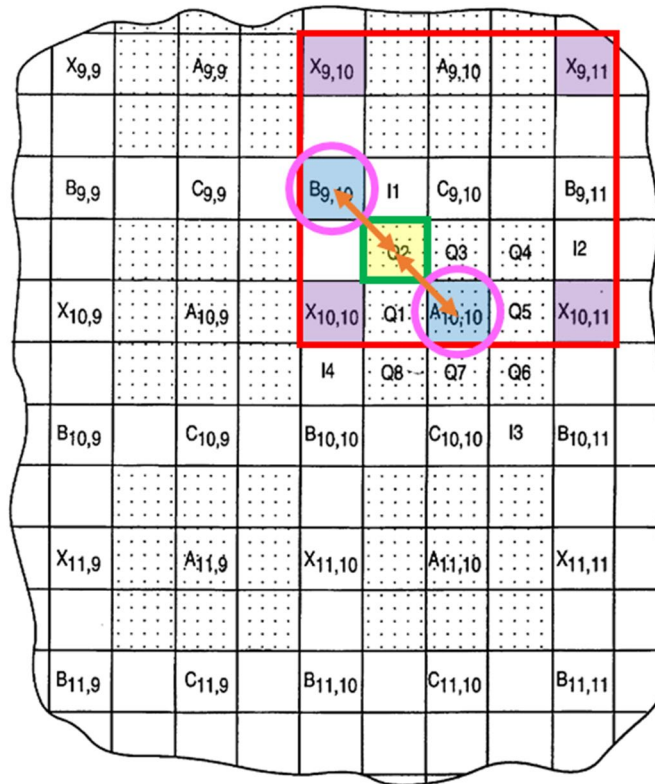


FIG. 5

Fandrianto, Figure 5. Namely, sub-pixels A and B (blue) are equidistant from sub-pixel Q2, and the magnitudes of their weights (“1”) are inversely proportional to their distances (orange arrows) from sub-pixel Q2. Ramchandran, ¶184.

If the claim requires affirmatively “selecting” the magnitudes of the first and second weights of the weighted average to be inversely proportional to the respective distances of sub-pixels A and B to the sub-pixel Q2, such features would have been obvious. Namely, the closer A and B are to Q2, the more likely their values are similar to Q2. Thus, the “closer” sub-pixels A and B should have larger weights. Conversely, the farther A and B are from Q2, the less likely their values are similar

to Q2. Accordingly, the “farther” sub-pixels A and B should have smaller weights. Thus, a POSITA would have known to select the relative magnitudes of the weights of A and B in the weighted average to be inversely proportional to their distances from Q2. Ramchandran, ¶¶185-186.

Accordingly, the TML6-Fandrianto combination interpolates TML6’s $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and $(3/4, 3/4)$ sub-pixels having co-ordinates with odd values of both K ($K/2^N$, **K=1 or 3**, $N=2$) and L ($L/2^N$, **L=1 or 3**, $N=2$), as claimed. Ramchandran, ¶187.

e. Claim 5

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for Claim 4 and shown below, the first and second weights have equal values (“1”) when the sub-pixels A and B (blue) (having co-ordinates with even values of both K and L) are equidistant from the sub-pixel Q2 (green) (having co-ordinates with odd values of both K and L) being interpolated.

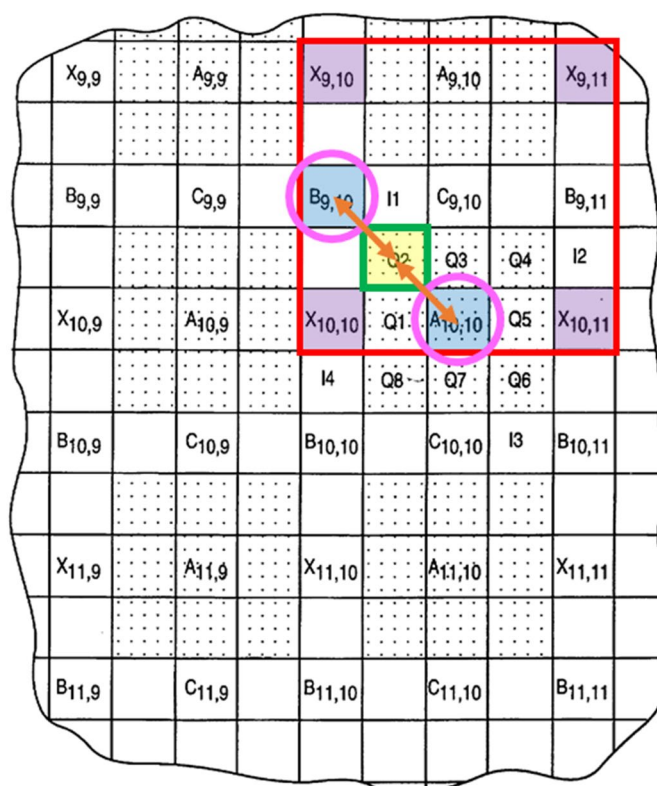


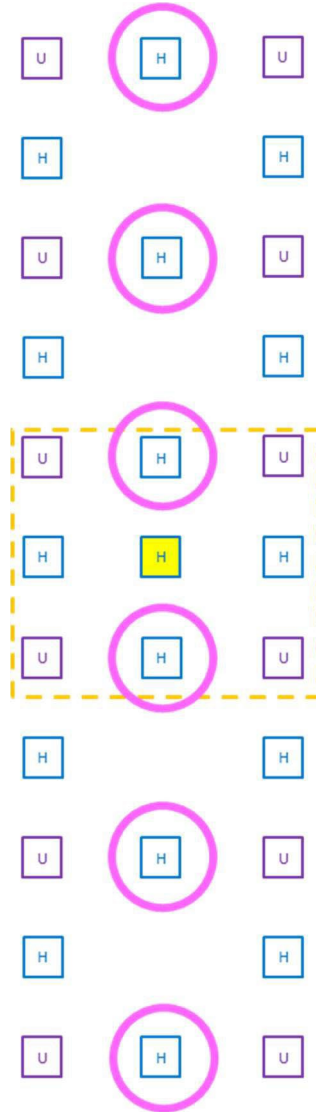
FIG. 5

Fandrianto, Figure 5; Ramchandran, ¶188.

f. Claim 6

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for 1[c], values for (2/4,2/4) sub-pixels having co-ordinates with even values of both K and L are interpolated using a predetermined choice of a first weighted sum of certain sub-pixel values and a second weighted sum of other sub-pixel values. Claim 6 repeats verbatim much of 1[c] and further states that the (2/4,2/4) sub-pixels are interpolated using the first weighted sum “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.” Ramchandran, ¶189.

As explained for 1[c], TML6 interpolates the (2/4,2/4) sub-pixels (highlighted below) using the first weighted sum of the sub-pixel values (circled).



Ramchandran, ¶190.

Furthermore, Fandrianto interpolates highlighted (2/4,1/4) sub-pixel “Q7” having co-ordinates with K equal to an even value and L equal to an odd value

($K/2^N, L/2^N$; $K=2, L=1, N=2$), and corresponding Q7 sub-pixels in other rectangular bounded regions, when required for motion estimation.

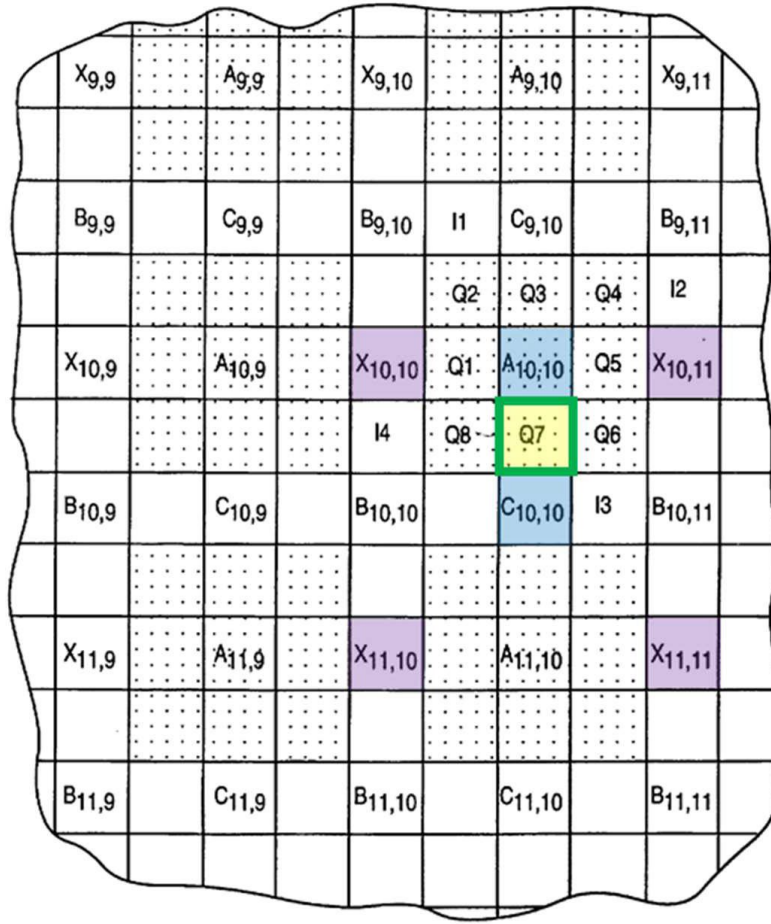
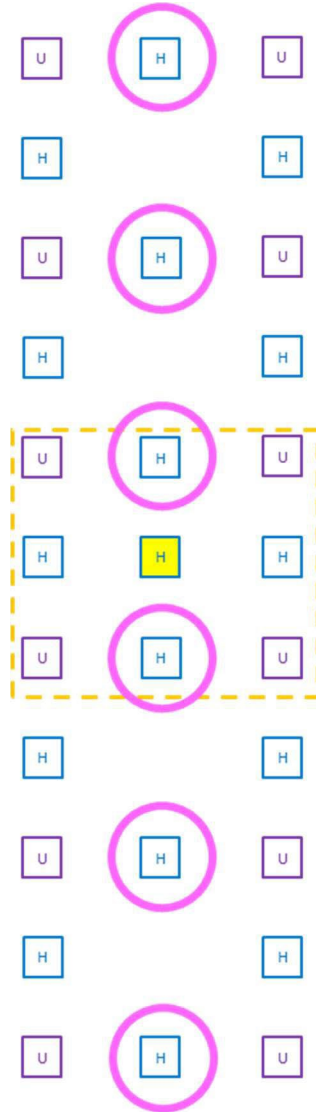


FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-35, 10:23-33; *see also id.*, 8:36-57, 9:33-43. As shown above, interpolating the highlighted Q7 sub-pixel (green) uses the value of the (2/4,2/4) sub-pixel $C_{10,10}$ (blue), and thus, this sub-pixel $C_{10,10}$ would be interpolated when interpolating the sub-pixel Q7 is required. Fandrianto, Figure 5, 5:26-7:19, 8:29-35, 10:23-33 (“ $Q7_{r,c}=(C_{r,c}+A_{r,c})/2$ ”); *see id.*, 8:36-57, 9:33-43. Therefore, the TML6-Fandrianto combination’s sub-pixel interpolation technique

interpolates $(2/4, 2/4)$ sub-pixels using the first weighted sum (shown below) when the $(2/4, 1/4)$ sub-pixel is also required.



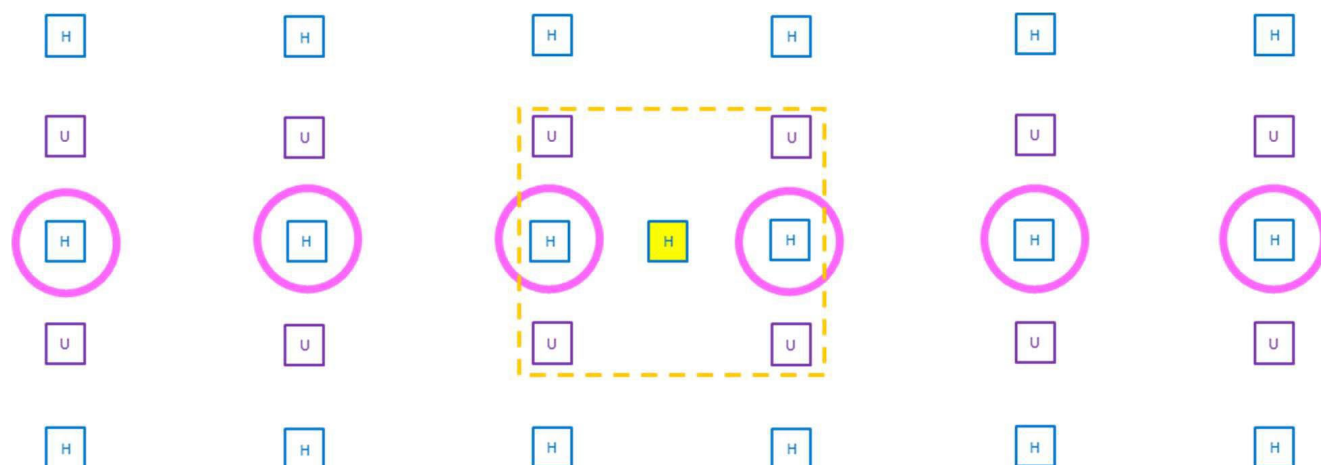
Ramchandran, ¶191.

g. Claim 7

The TML6-Fandrianto combination discloses the additional features of this claim. As discussed for 1[c], values for $(2/4, 2/4)$ sub-pixels having co-ordinates with even values of both K and L are interpolated using a predetermined choice of a

first weighted sum of certain sub-pixel values and a second weighted sum of other sub-pixel values. Claim 7 repeats verbatim much of 1[c] and further states that the sub-pixel values—e.g., (2/4,2/4)—are interpolated using the second weighted sum “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.” Ramchandran, ¶192.

As explained for 1[c] and shown below, the TML6-Fandrianto combination interpolates the highlighted (2/4,2/4) sub-pixels using the second weighted sum of the circled sub-pixels.



Ramchandran, ¶193.

Furthermore, Fandrianto interpolates highlighted (1/4,2/4) sub-pixel “I1” having co-ordinates with K equal to an odd value and L equal to an even value ($K/2^N, L/2^N$; $K=1, L=2, N=2$), and corresponding “I1” sub-pixels in other rectangular bounded regions, when required for motion estimation.

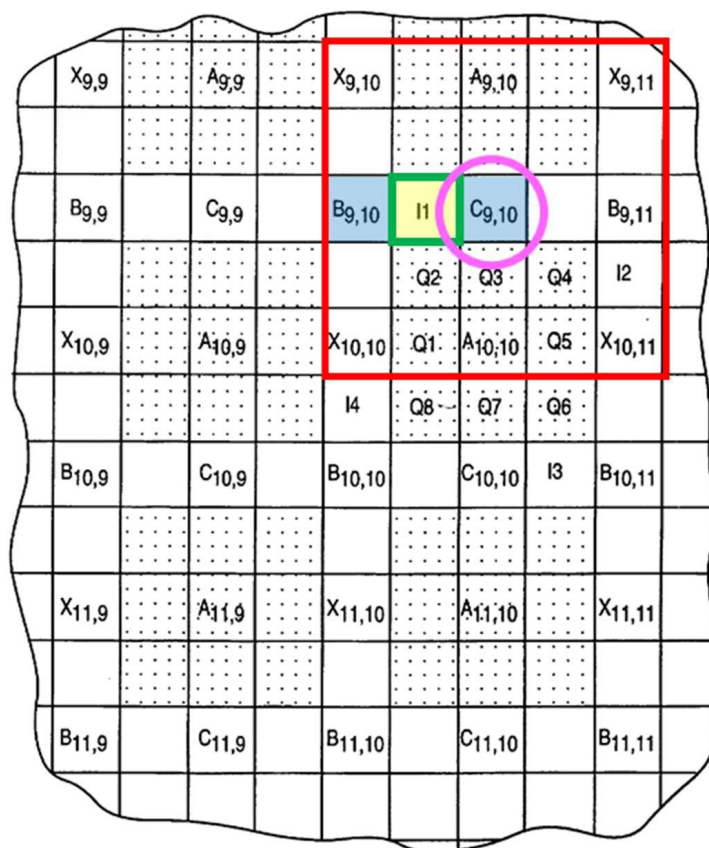
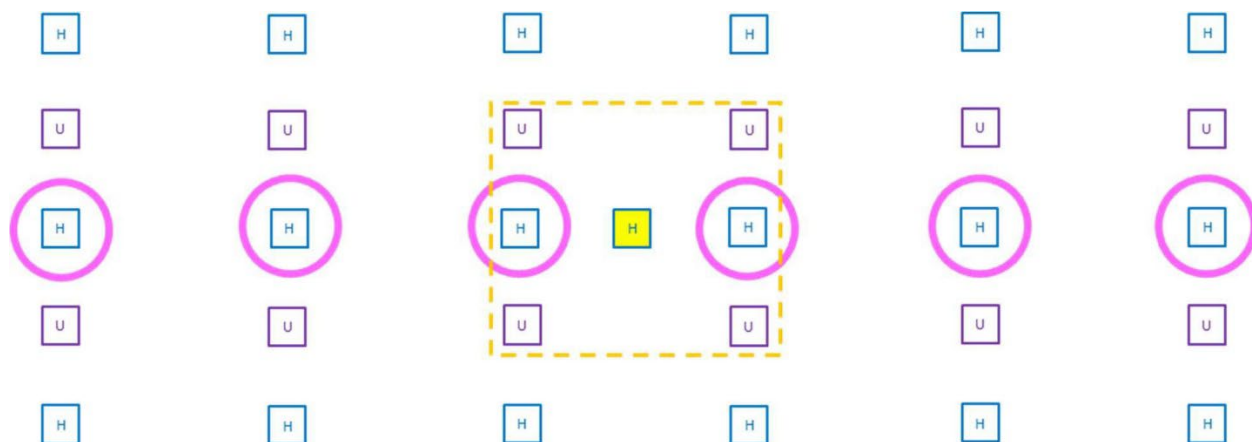


FIG. 5

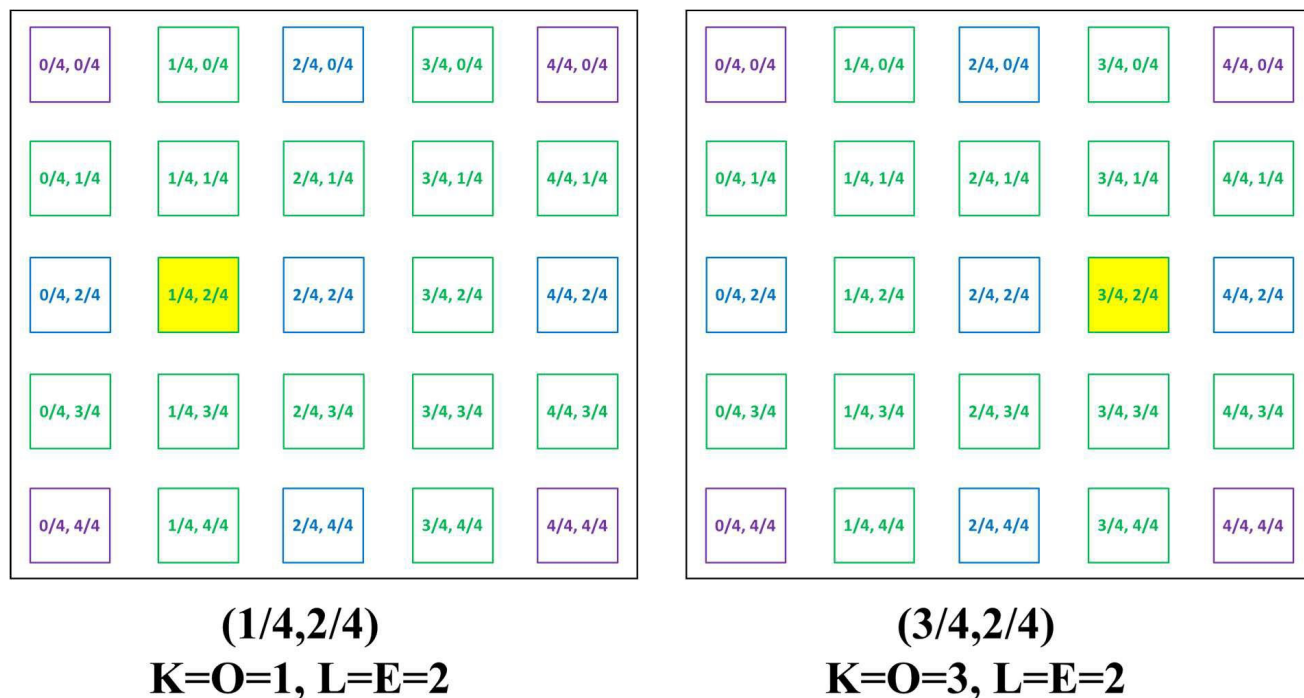
Fandrianto, Figure 5, 5:26-32, 8:29-35, 8:58-9:32 (“ $I_{1,r,c} = (B_{r-1,c} + C_{r-1,c})/2$ ”), 10:6-22 (“I3”). As shown above, interpolating the highlighted I1 sub-pixel uses the (2/4,2/4) sub-pixel C_{9,10} (blue), and thus, this sub-pixel C_{9,10} would be interpolated when interpolating the sub-pixel I1 is required. Fandrianto, Figure 5, 5:26-7:19, 8:29-35, 8:58-9:32; *see id.*, 10:6-22. Therefore, the TML6-Fandrianto combination’s sub-pixel interpolation technique interpolates (2/4,2/4) sub-pixels using the second weighted sum (shown below) when the (1/4,2/4) sub-pixel is also required.



Ramchandran, ¶194.

h. Claim 8

The TML6-Fandrianto combination discloses the additional features of this claim. For example, as shown below, the combination interpolates (1/4,2/4) and (3/4,2/4) sub-pixels having co-ordinates with K equal to an odd value O (e.g., 1 or 3), and L equal to an even value E (e.g., 2), which is not zero or 2^N (i.e., 4) ($K/2^N, L/2^N$; **K=1 or 3, L=2, N=2**).



Ramchandran, ¶195.

Furthermore, as shown below (left), Fandrianto interpolates the highlighted (1/4, 2/4) sub-pixel “I1” by averaging a first (0/4, 2/4) sub-pixel B_{9,10} (blue and circled) having coordinates (K/2^N, L/2^N; K=0, L=2, N=2) with K equal to the even value O-1 (e.g., 0 (1-1)) and L equal to E (e.g., 2) and a second (2/4, 2/4) sub-pixel C_{9,10} (blue and circled) having coordinates (K/2^N, L/2^N; K=2, L=2, N=2) with K equal to the even value O+1 (e.g., 2 (1+1)) and L equal to E (e.g., 2). Similarly, as shown below (right), Fandrianto interpolates the highlighted (3/4, 2/4) sub-pixel “I3” by averaging first (2/4, 2/4) sub-pixel C_{10,10} (blue and circled) having coordinates (K/2^N, L/2^N; K=2, L=2, N=2) with K equal to the even value O-1 (e.g., 2 (3-1)) and L equal to E (e.g., 2) and a second 4/4, 2/4 sub-pixel B_{10,11} (blue and circled) having

coordinates $(K/2^N, L/2^N; K=4, L=2, N=2)$ with K equal to the even value $O+1$ (e.g., $4(3+1)$) and L equal to E (e.g., 2).

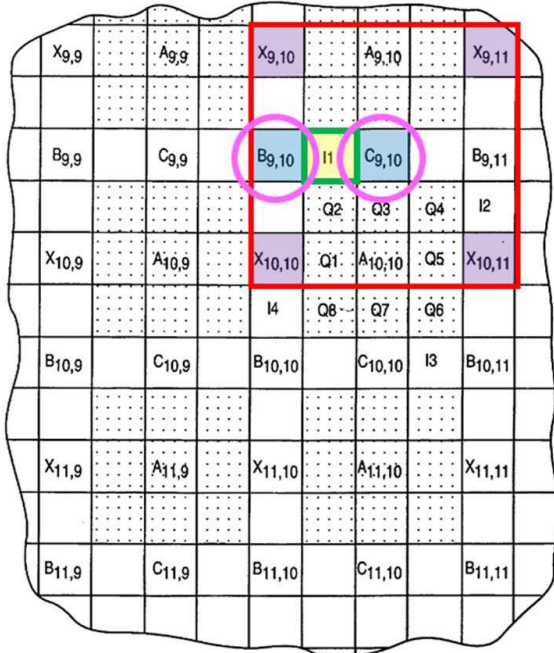


FIG. 5

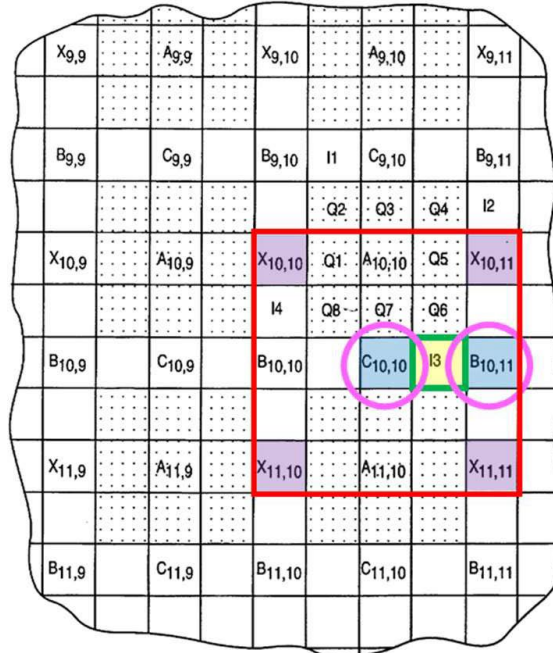


FIG. 5

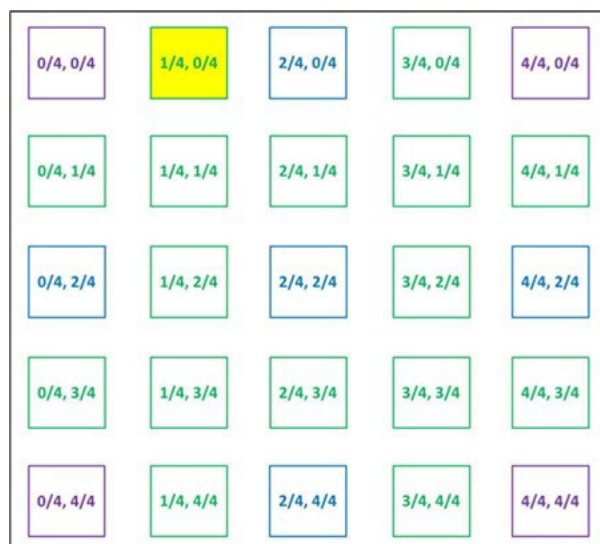
Fandrianto, Figure 5, 5:26-32, 8:29-57, 8:58-9:32 ($I_{1,r,c} = (B_{r-1,c} + C_{r-1,c})/2$), 10:6-22 ($I_{3,r,c} = (B_{r,c} + 1 + C_{r,c})/2$); Ramchandran, ¶196.

As discussed *supra* §I.A.1, implementing Fandrianto's technique to interpolate the claimed sub-pixels using values in TML6's *mref* was obvious. Ramchandran, ¶197.

i. Claim 9

The TML6-Fandrianto combination discloses the additional features of this claim. The combination interpolates highlighted $(1/4, 0/4)$ sub-pixel $(K/2^N, L/2^N;$

K=1, L=0, N=2) “having coordinates with K equal to 1 and L equal to zero,” as recited.



(1/4,0/4)
K=1, L=0

Ramchandran, ¶198.

Furthermore, as shown below, Fandrianto interpolates the highlighted (1/4,0/4) sub-pixel “Q1” (green) by averaging (0/4,0/4) pixel $X_{10,10}$ (purple and circled) having coordinates $(K/2^N, L/2^N; K=0, L=0, N=2)$ with K equal to zero and L equal to zero and a (2/4,0/4) sub-pixel $A_{10,10}$ (blue and circled) $(K/2^N, L/2^N; K=2, L=0, N=2)$ “having coordinates with K equal to 2 and L equal to zero,” as recited.

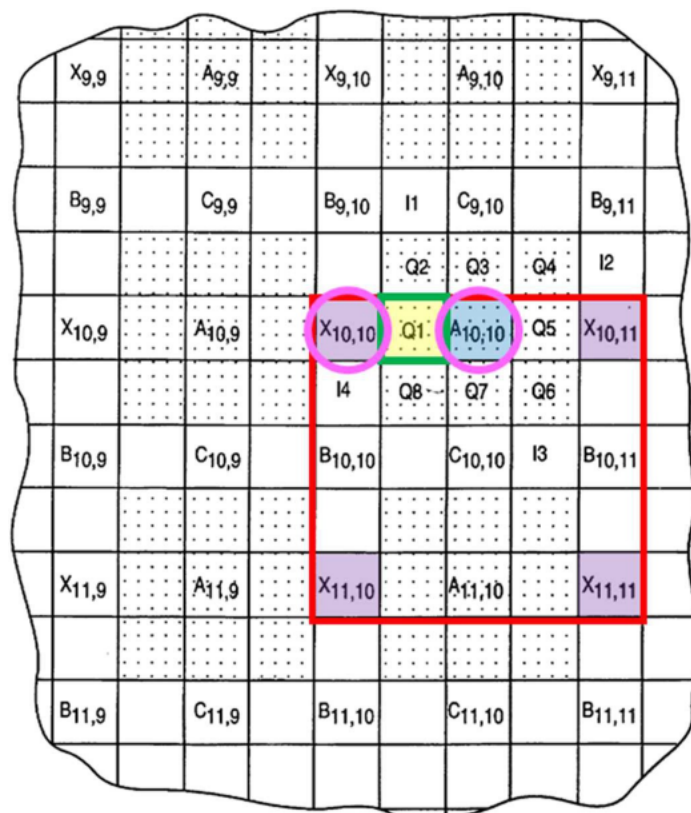


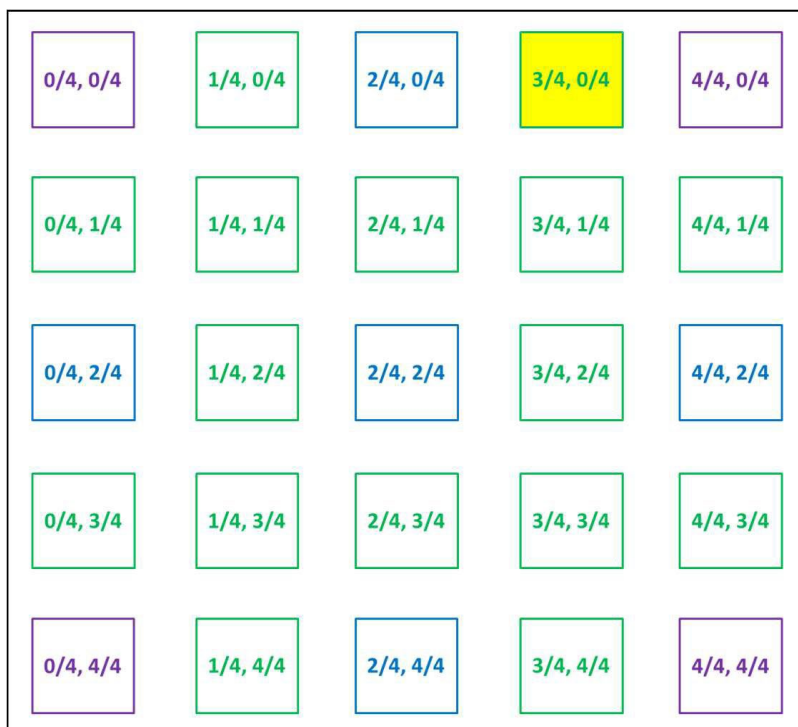
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57 (“ $Q_{1,r,c} = (X_{r,c} + A_{r,c})/2$ ”); Ramchandran, ¶199.

As discussed *supra* §VIII.A.1, implementing Fandrianto’s technique to interpolate the claimed sub-pixel using values in TML6’s mref was obvious. Ramchandran, ¶200.

j. Claim 10

The TML6-Fandrianto combination discloses the additional features of this claim. For example, the combination interpolates the highlighted (3/4,0/4) sub-pixel (below) having coordinates ($K/2^N, L/2^N$; $K=3, L=0, N=2$) with $K=2^N-1$ (e.g., 3) and $L=0$.



(3/4,0/4)
K=3, L=0

Ramchandran, ¶201.

Furthermore, Fandrianto interpolates the highlighted (3/4,0/4) sub-pixel “Q5” (below) by averaging (4/4,0/4) pixel $X_{10,11}$ (purple and circled) having coordinates $(K/2^N, L/2^N; K=4, L=0, N=2)$ with $K=2^N$ (e.g., 4) and $L=0$ and a (2/4,0/4) sub-pixel $A_{10,10}$ (blue and circled) having coordinates $(K/2^N, L/2^N; K=2, L=0, N=2)$ with $K=2^{N-2}$ (e.g., 2) and $L=0$.

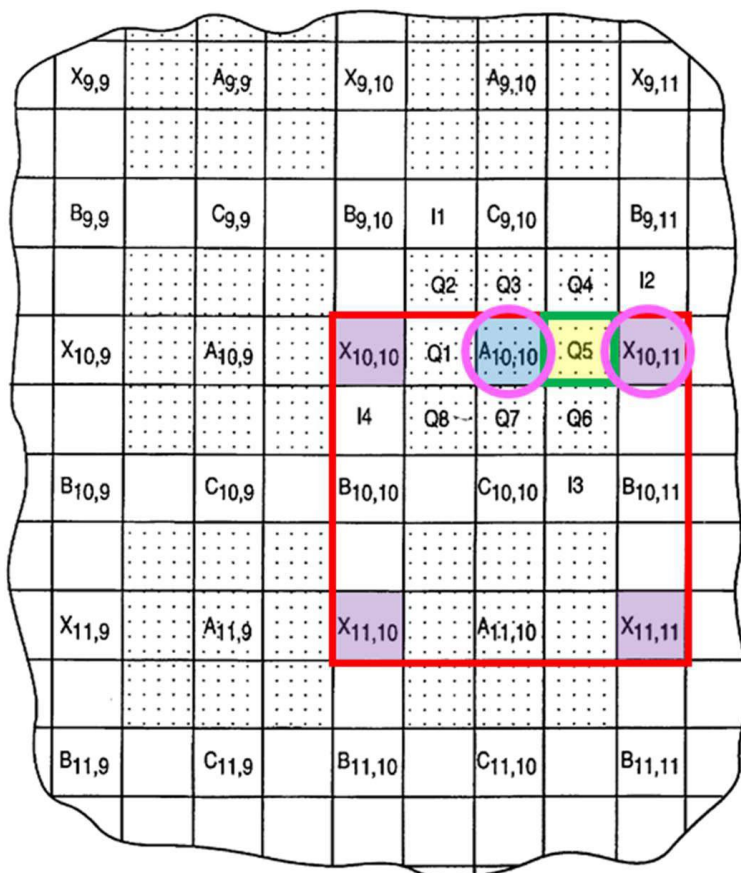


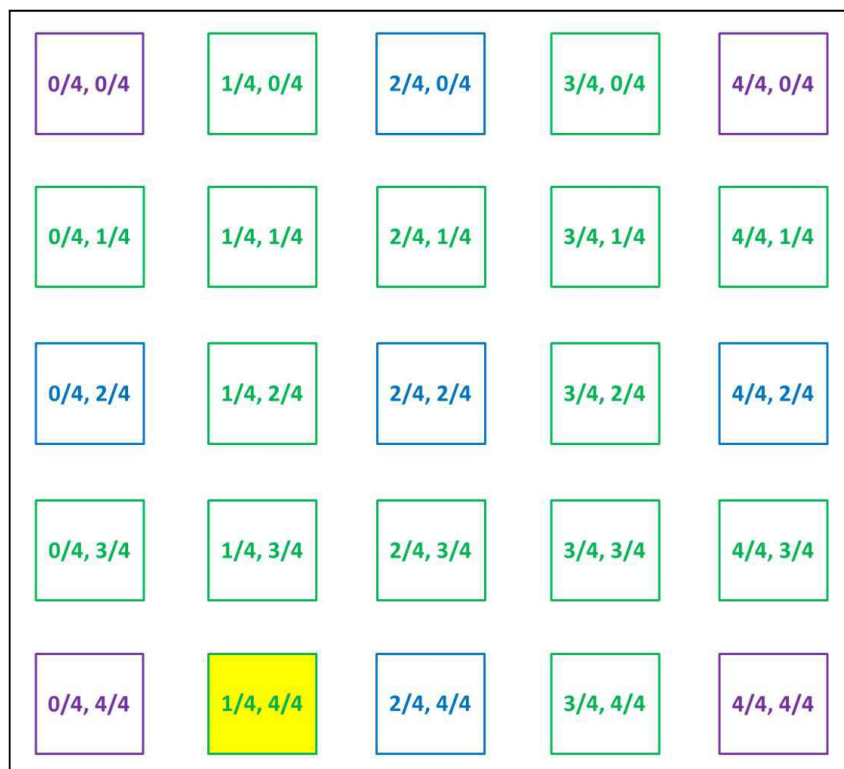
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 9:63-10:5 (“ $Q5_{r,c}=(X_{r,c+1}+A_{r,c})/2$ ”);
Ramchandran, ¶202.

As discussed *supra* §VIII.A.1, implementing Fandrianto’s technique to
interpolate the claimed sub-pixel using values in TML6’s *mref* was obvious.
Ramchandran, ¶203.

k. Claim 11

The TML6-Fandrianto combination discloses the additional features of this
claim. For example, the combination interpolates highlighted (1/4,4/4) sub-pixel
(below) having coordinates ($K/2^N, L/2^N$; $K=1, L=4, N=2$) with $K=1$ and $L=2^N$.



(1/4,4/4)
K=1, L=4

Ramchandran, ¶204.

Furthermore, Fandrianto interpolates the highlighted (1/4,4/4) sub-pixel “Q1” (green) (below) by averaging (0/4,4/4) pixel $X_{10,10}$ (purple and circled) having coordinates $(K/2^N, L/2^N; K=0, L=4, N=2)$ with $K=0$ and $L=2^N$ (e.g., 4) and a (2/4,4/4) sub-pixel $A_{10,10}$ (blue and circled) having coordinates $(K/2^N, L/2^N; K=2, L=4, N=2)$ with $K=2$ and $L=2^N$.

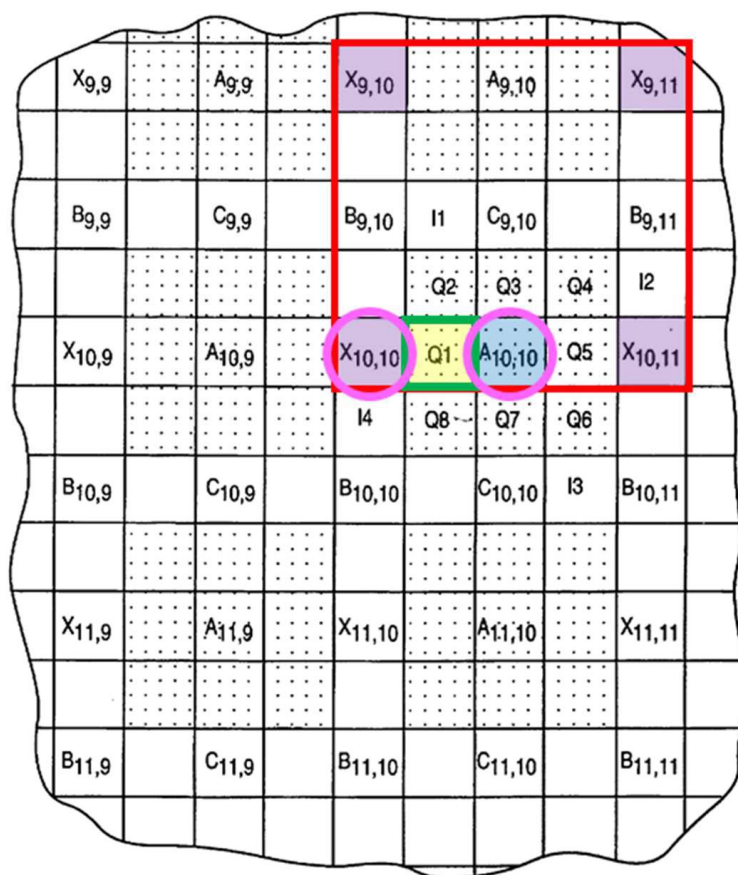


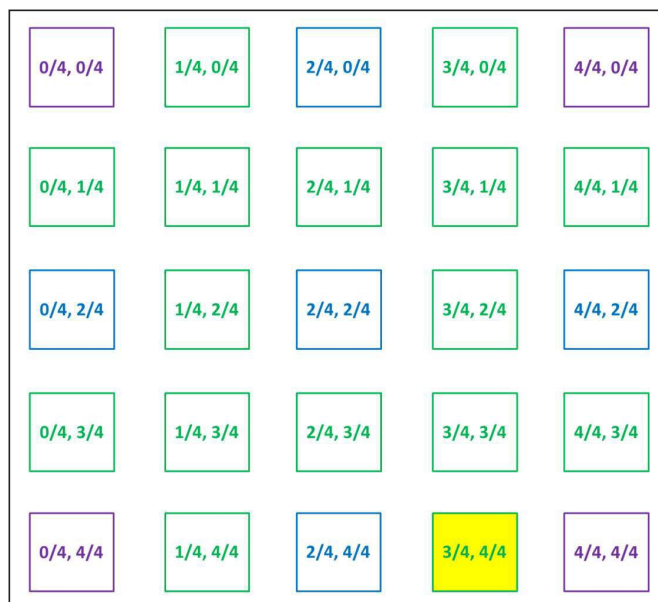
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57 (“ $Q1_{r,c} = (X_{r,c} + A_{r,c})/2$ ”); Ramchandran, ¶205.

As discussed *supra* §VIII.A.1, implementing Fandrianto’s technique to interpolate the claimed sub-pixel using values in TML6’s *mref* was obvious. Ramchandran, ¶206.

I. Claim 12

The TML6-Fandrianto combination discloses the additional features of this claim. As shown below, the combination interpolates highlighted (3/4,4/4) sub-pixel having coordinates with $K=2^N-1$ and $L=2^N$ ($K/2^N, L/2^N$; $K=3, L=4, N=2$).



$$\frac{3}{4}, \frac{4}{4})$$

$$\mathbf{K=1, L=4}$$

Ramchandran, ¶207.

Furthermore, Fandrianto interpolates the highlighted (3/4,4/4) sub-pixel Q5 (green) (below) by averaging (4/4,4/4) pixel $X_{10,11}$ (purple and circled) having coordinates with $K=2^N$ and $L=2^N$ ($K/2^N, L/2^N$; $\mathbf{K=4, L=4, N=2}$) and (2/4,4/4) sub-pixel $A_{10,10}$ (blue and circled) having coordinates with $K=2^N-2$ and $L=2^N$ ($K/2^N, L/2^N$; $\mathbf{K=2, L=4, N=2}$).

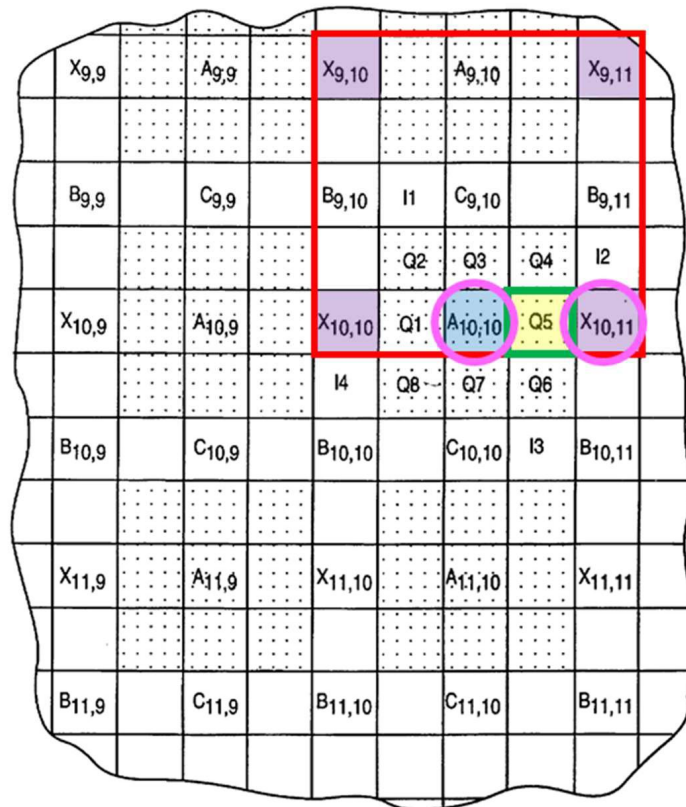


FIG. 5

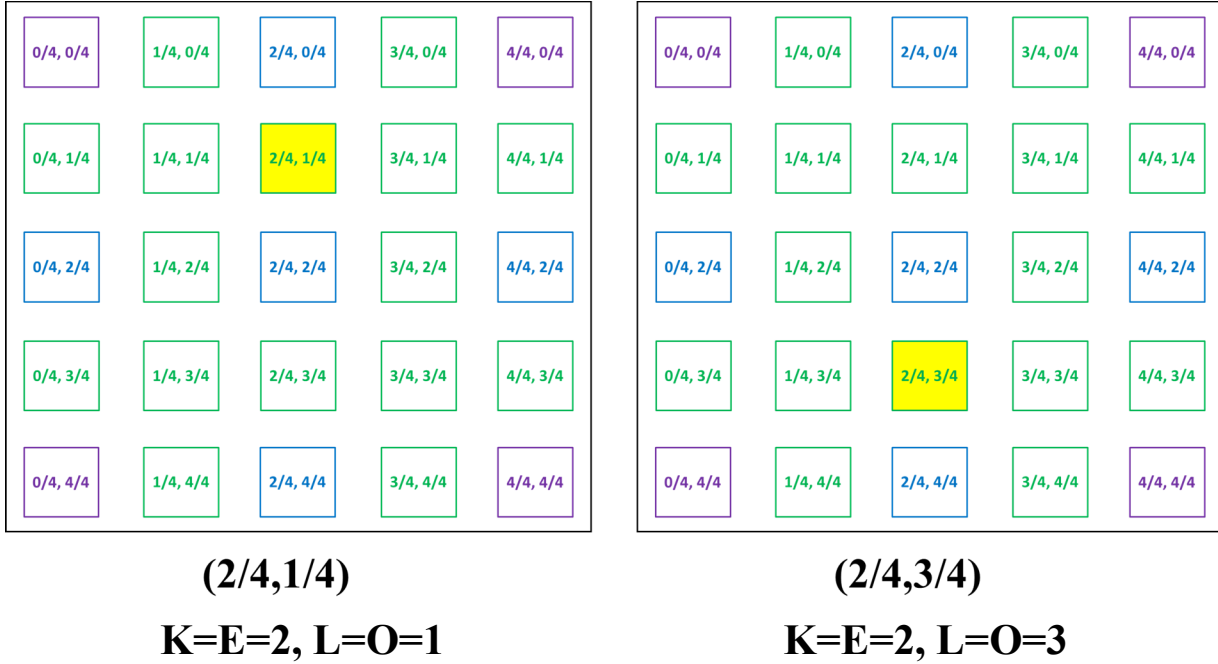
Fandrianto, Figure 5, 5:26-32, 8:29-57, 9:63-10:5; Ramchandran, ¶208.

As discussed *supra* §VIII.A.1, implementing Fandrianto's technique to interpolate the claimed sub-pixel using values in TML6's *mref* was obvious. Ramchandran, ¶209.

m. Claim 13

The TML6-Fandrianto combination discloses the additional features of this claim. Specifically, the combination interpolates highlighted (2/4,1/4) and (2/4,3/4) sub-pixels (below) having co-ordinates with K equal to an even value E not

including zero or 2^N ($E=4$) and L equal to an odd value O ($O=1$ or 3) ($K/2^N, L/2^N$; $K=2, L=1$ or $3, N=2$).



Ramchandran, ¶210.

As shown below, Fandrianto interpolates highlighted $(2/4, 1/4)$ sub-pixel Q7 (below left) by averaging first $(2/4, 0/4)$ sub-pixel $A_{10,10}$ (blue, circled) having coordinates $(K/2^N, L/2^N; K=2, L=0, N=2)$ with K equal to E (e.g., 2) and L equal to the even value $O-1$ (e.g., $0=1-1$) and a second $(2/4, 2/4)$ sub-pixel $C_{10,10}$ (blue, circled) having coordinates $(K/2^N, L/2^N; K=2, L=2, N=2)$ with K equal to E (e.g., 2) and L equal to the even value $O+1$ (e.g., $2=1+1$). Furthermore, as shown below (right), Fandrianto interpolates highlighted $(2/4, 3/4)$ sub-pixel Q3 by averaging first $(2/4, 2/4)$ sub-pixel $C_{9,10}$ (blue and circled) having coordinates $(K/2^N, L/2^N; K=2, L=2,$

N=2) with K equal to E (e.g., 2) and L equal to the even value O-1 (e.g., 2=3-1) and a second (2/4,4/4) sub-pixel $A_{10,10}$ (blue and circled) having coordinates $(K/2^N, L/2^N; K=2, L=4, N=2)$ with K equal to E (e.g., 2) and L equal to the even value O+1 (e.g., 4=3+1).

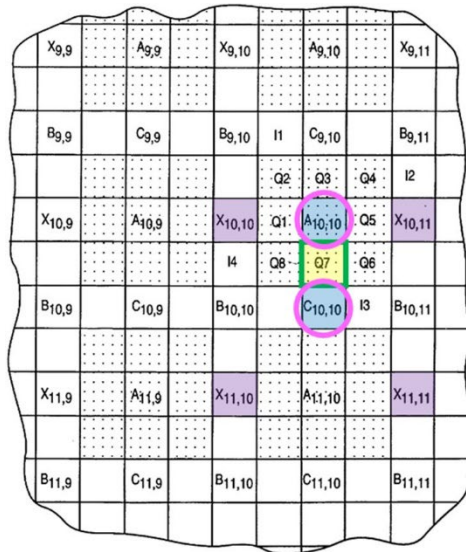


FIG. 5

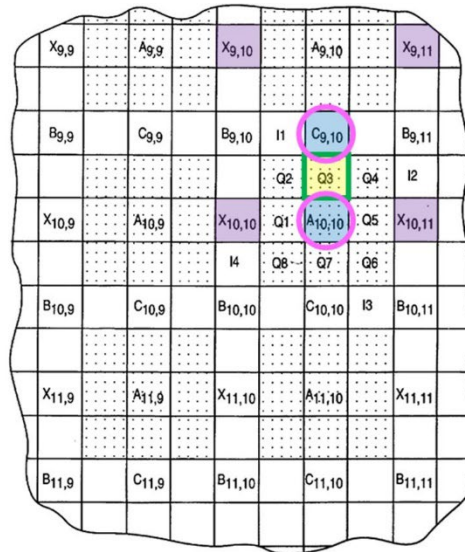


FIG. 5

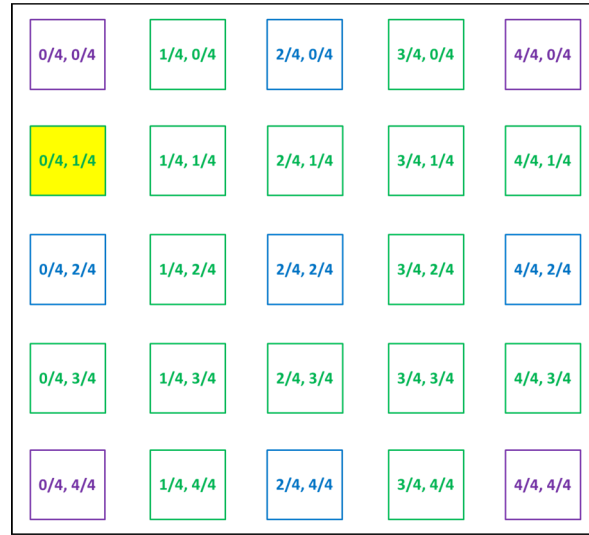
Fandrianto, Figure 5, 5:26-32, 8:29-57, 10:23-33, 9:32-44; Ramchandran, ¶211.

As discussed *supra* §VIII.A.1, implementing Fandrianto's technique to interpolate the claimed sub-pixel using values in TML6's *mref* was obvious.

Ramchandran, ¶212.

n. Claim 14

The TML6-Fandrianto combination discloses the additional features of this claim. The combination interpolates highlighted (0/4,1/4) sub-pixels (below) having co-ordinates with K=0 and L=1 ($K/2^N, L/2^N; K=0, L=1, N=2$).



(0/4,1/4)
K=0, L=1

Ramchandran, ¶213.

Furthermore, Fandrianto interpolates highlighted (0/4,1/4) sub-pixel I4 (below) by averaging (0/4,0/4) pixel $X_{10,10}$ (purple, circled) having co-ordinates with $K=0$ and $L=0$ ($K/2^N, L/2^N$; $\mathbf{K=0}, \mathbf{L=0}, N=2$) and (0/4,2/4) sub-pixel $B_{10,10}$ (blue, circled) having co-ordinates with $K=0$ and $L=2$ ($K/2^N, L/2^N$; $\mathbf{K=0}, \mathbf{L=2}, N=2$).

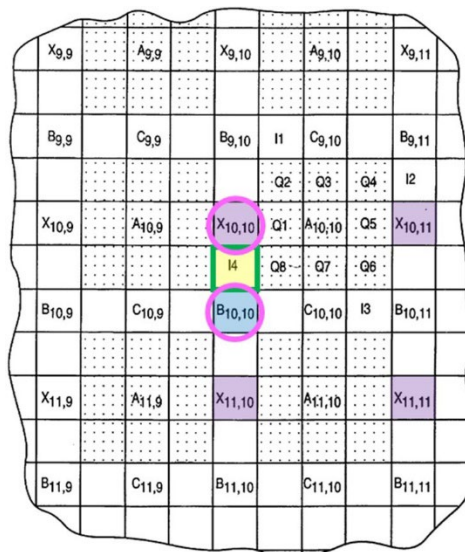


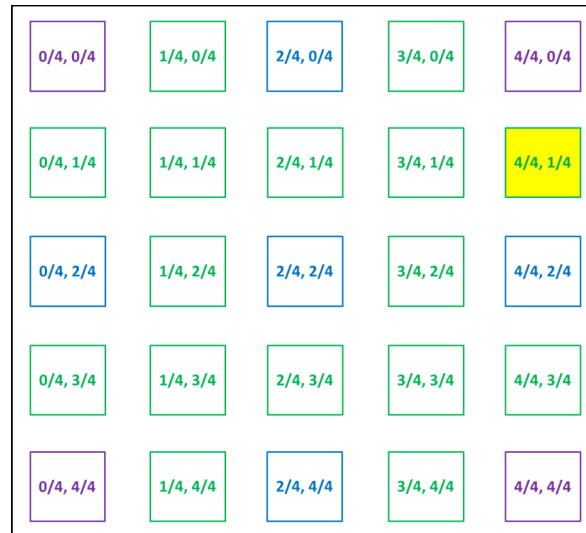
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 10:34-10:50 (“ $I_{r,c}=(B_{r,c}+X_{r,c})/2$ ”);
Ramchandran, ¶214.

As discussed *supra* §VIII.A.1, implementing Fandrianto’s technique to interpolate the claimed sub-pixel using values in TML6’s *mref* was obvious.
Ramchandran, ¶215.

o. Claim 15

Claim 15 appears to have an error, but assuming the “sub-pixel having co-ordinates with K equal to zero and L equal to 2” refers to the sub-pixel having co-ordinates with $K=2^N$ and $L=2$, the TML6-Fandrianto combination teaches the additional features of Claim 15. As shown below, the combination interpolates highlighted (4/4,1/4) sub-pixels having co-ordinates with $K=2^N$ (i.e., 4) and $L=1$ ($K/2^N, L/2^N$; $K=4, L=1, N=2$).



(4/4,1/4)
K=4, L=1

Ramchandran, ¶216.

Fandrianto interpolates highlighted (4/4,1/4) sub-pixel “I4” (below) by averaging the (4/4,0/4) pixel $X_{10,10}$ (purple, circled) having co-ordinates $(K/2^N, L/2^N; K=4, L=0, N=2)$ with $K=2^N$ (e.g., 4) and $L=0$ and (4/4,2/4) sub-pixel $B_{10,10}$ (blue, circled) having co-ordinates $(K/2^N, L/2^N; K=4, L=2, N=2)$ with $K=2^N$ and $L=2$.

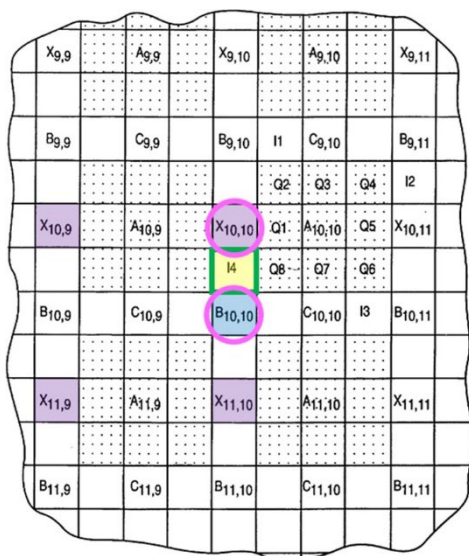


FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 10:34-10:50 (“ $I_{4,r,c} = (B_{r,c} + X_{r,c})/2$ ”);

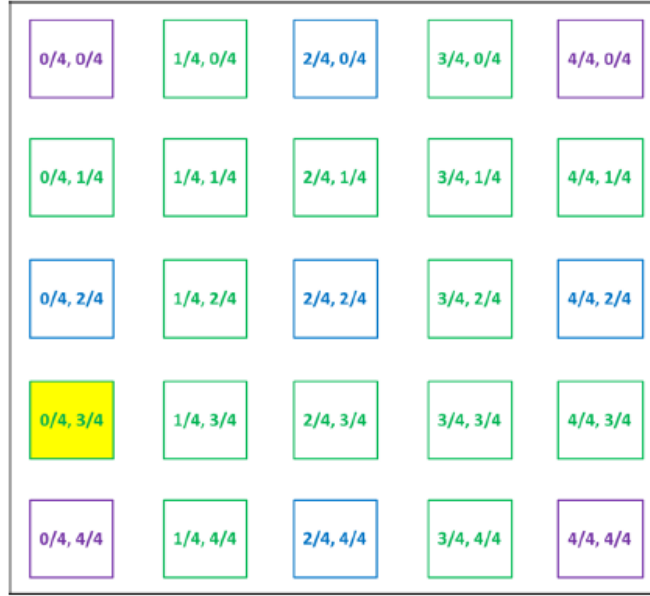
Ramchandran, ¶217.

As discussed *supra* §VIII.A.1, substituting Fandrianto’s interpolation technique for TML6’s would have been obvious. Ramchandran, ¶218.

p. Claim 16

The TML6-Fandrianto combination discloses the additional features of this claim. The combination interpolates values for highlighted (0/4,3/4) sub-pixel

(below) having co-ordinates with $K=0$ and $L=2^N-1$ (e.g., $3=2^2-1$) ($K/2^N, L/2^N$; $\mathbf{K=0}$, $\mathbf{L=3}$, $N=2$).



(0/4,3/4)
K=0, L=3

Ramchandran, ¶219.

As shown below,¹⁶ Fandrianto interpolates highlighted (0/4,3/4) sub-pixel “I2” by averaging (0/4,4/4) pixel $X_{10,11}$ (purple, circled) having co-ordinates ($K/2^N, L/2^N$; $K=0$, $L=4$, $N=2$) with $K=0$ and $L=2^N$ (e.g., 4) and (0/4,2/4) sub-pixel $B_{9,11}$ (blue, circled) having co-ordinates ($K/2^N, L/2^N$; $K=0$, $L=2$, $N=2$) with $K=0$ and $L=2^N-2$ ($2=2^2-2$).

¹⁶ The figure has annotations illustrating unit pixels $X_{9,12}$ and $X_{10,12}$.

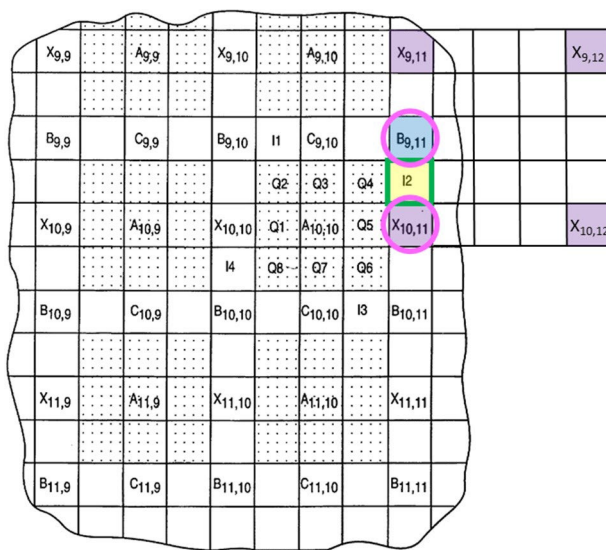


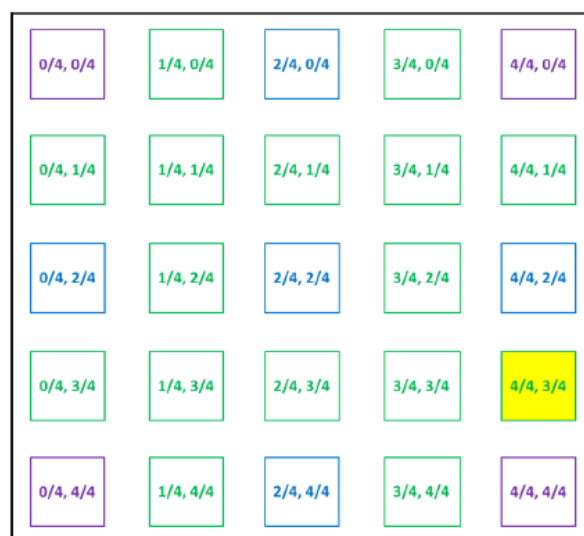
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 9:45-62 ($I_{2,r,c} = (B_{r-1,c+1} + X_{r,c+1})/2$);
Ramchandran, ¶220.

As discussed *supra* §VIII.A.1, implementing Fandrianto's technique to interpolate the claimed sub-pixel by averaging values in TML6's *mref* was obvious.
Ramchandran, ¶221.

q. Claim 17

The TML6-Fandrianto combination discloses the additional features of this claim. The combination interpolates highlighted (4/4,3/4) sub-pixel (below) having co-ordinates ($K/2^N, L/2^N$; $K=4, L=3, N=2$) with $K=2^N$ (4) and $L=2^N-1$ ($3=2^2-1$).



(4/4, 3/4)
K=4, L=3

1[pre][a], *supra*; Ramchandran, ¶222.

Fandrianto interpolates highlighted (4/4, 3/4)¹⁷ sub-pixel “I2” (below) by averaging (4/4, 4/4) pixel $X_{10,11}$ (purple, circled) having co-ordinates ($K/2^N, L/2^N$; $K=4, L=4, N=2$) with $K=2^N$ (i.e., 4) and $L=2^N$ and (4/4, 2/4) sub-pixel $B_{9,11}$ (blue, circled) having co-ordinates ($K/2^N, L/2^N$; $K=4, L=2, N=2$) with $K=2^N$ (i.e., 4) and $L=2^N-2$ ($2=2^2-2$).

¹⁷ Under the claimed co-ordinate notation (1[pre][b], *supra*), sub-pixel 4/4, 3/4 is equivalent to sub-pixel 0/4, 3/4 (discussed for Claim 16, *supra*) because 4/4, 3/4 is defined by the rectangular bounded region about $C_{9,11}$ and 0/4, 3/4 is defined by the rectangular bounded region about $C_{9,10}$.

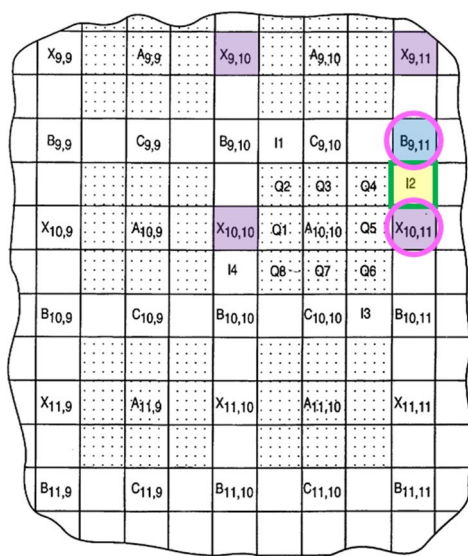


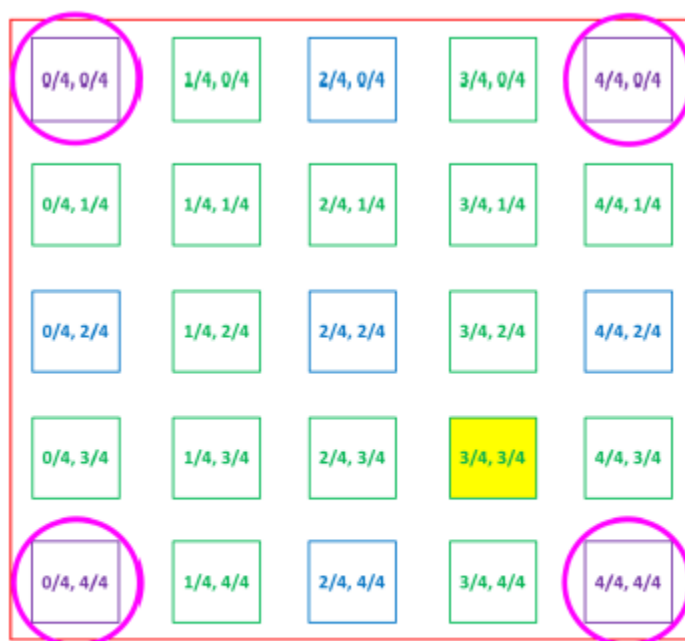
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 9:45-62 ($I_{2,r,c} = (B_{r-1,c+1} + X_{r,c+1})/2$);
Ramchandran, ¶223.

As discussed *supra* §VIII.A.1, substituting Fandrianto's interpolation technique for TML6's would have been obvious. Ramchandran, ¶224.

r. Claim 18

The TML6-Fandrianto combination discloses the additional features of this claim. As shown below, TML6 interpolates highlighted (3/4,3/4) sub-pixel having coordinates $(K/2^N, L/2^N; K=3, L=3, N=2)$ with $K=2^N-1$ (i.e., $3=2^2-1$) and $L=2^N-1$ (i.e., $3=2^2-1$). Namely, the third step of the *oneforthpix()* function interpolates (3/4,3/4) sub-pixel by taking the weighted average of the corner pixels (purple; circled) defining the rectangular bounded region (red), as shown below.



$(3/4, 3/4)$
 $K=3, L=3$

TML6, 57-58 (lines 593-625); Ramchandran, ¶225.

The four unit pixels (purple) used to calculate the weighted average for highlighted $(3/4, 3/4)$ sub-pixel are retrieved from *mref* and have weights equaling “1.” TML6, 58 (lines 615-618). Moreover, this weighted average, after being normalized and rounded to be within the dynamic range of 0 to 255, is stored back in the data structure *mref* as the value of the $(3/4, 3/4)$ sub-pixel. TML6, 58 (lines 618-621); Ramchandran, ¶226.

The '599 patent confirms TML6 interpolates $(3/4, 3/4)$ sub-pixels “f” (green) (having coordinates with $K=L=2^{N-1}=3$) by taking a weighted average of the four corner pixels “A” (purple) that define the rectangular bounded region.

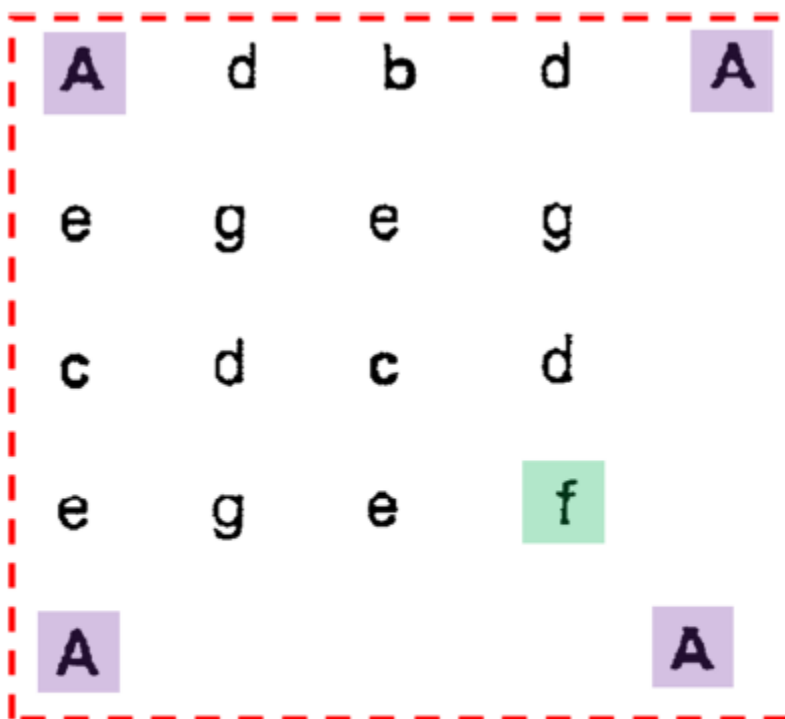


Fig. 13(a)

EX-1001, Figure 13(a), 12:66-13:3; Ramchandran, ¶227.

Accordingly, the TML6-Fandrianto combination interpolates TML6's (3/4,3/4) sub-pixel having coordinates with $K=2^N-1$ ($K=3$) and $L=2^N-1$ ($L=3$), as claimed. Ramchandran, ¶228.

s. Claim 19

As discussed for Claim 1, in the TML6-Fandrianto combination, $N=2$. Ramchandran, ¶229.

3. Independent Claim 20

Claim 20 is substantially the same as Claim 1 but is directed to the specific case where $N=2$. As discussed for Claim 1, in the TML6-Fandrianto combination, $N=2$. Ramchandran, ¶230.

4. Claims 22-42¹⁸ and 44-51

a. Independent Claim 22

Claim 22 is substantively the same as Claim 1, except it is written in terms of “[a]n **apparatus...comprising[] circuitry** operable to” perform the steps of Claim 1. Ramchandran, ¶231.

If “apparatus” is limiting, TML6 and Fandrianto disclose it or render it obvious. Both TML6 and Fandrianto disclose a video coder. *See* TML6, 155 (line 5), 69-70 (lines 3, 86), 1 (lines 3, 18); Fandrianto, 1:16-67, Figure 2. TML6 was developed for implementation in hardware or software and for use in applications that receive packetized data over IP networks, which a POSITA would have understood requires an apparatus. Ramchandran, ¶232; Wenger, ¶¶15, 19-21; VCEG-L45, 1, 34-36. Fandrianto, in turn, teaches a programmable motion estimator including an arithmetic logic unit (ALU) that performs pixel averaging arithmetic

¹⁸ Claims 27 and 28 depend from Claims 44 and 46, respectively, and are addressed after those claims.

for half- and quarter-pixel interpolation, which a POSITA would have also understood requires an apparatus. Fandrianto, Figure 2, 1:12-41, 3:55-4:16, 7:27-37, 8:15-20; Ramchandran, ¶233.

TML6 discloses reference code created to execute on a processor that would require circuitry to execute the code. *See* TML6. The '599 patent confirms TML6 requires such circuitry. EX-1001, 13:23-44 (“TML6... requires more silicon area in ASICs and requires more computations in some CPUs”). Therefore, TML6 discloses or renders obvious circuitry to execute the interpolations in the *oneforthpix()* function, and thus discloses “circuitry operable to interpolate.” Ramchandran, ¶232.

Furthermore, Fandrianto discloses an apparatus including circuitry for interpolating sub-pixels. Fandrianto 1:61-67 (“effective video coding requires the intimate integration of...integrated circuit technology, and digital storage media....”), Figures 2-5, 2:48-58, 3:55-4:53, 5:26-7:2, 7:27-10:58; Ramchandran, ¶233.

The TML6-Fandrianto combination thus discloses “circuitry operable to interpolate [the] sub-pixel value[s].” Ramchandran, ¶234.

b. Claims 23-24, 26, and 29-41

Claims 23-24, 26, and 29-41 are substantively the same as Claims 2-4 and 6-18, respectively, except they are written in terms of “[a]n apparatus...operable to

interpolate” the respective method steps. Ramchandran, ¶¶235-238. The TML6-Fandrianto combination renders them obvious as discussed for Claim 22. Ramchandran, ¶235.

c. Claim 25

The TML6-Fandrianto combination renders the additional features of this claim obvious. As discussed for 1[a] and Claim 2, Fandrianto interpolates the $(1/4, 3/4)$ sub-pixel Q2 (green)—having coordinates with odd values of both K ($K/2^N$, $K=1$, $N=2$) and L ($L/2^N$, $L=3$, $N=2$)—according to a predetermined choice of two weighted averages. Ramchandran, ¶239. As further explained for 1[a] and Claim 2 and as shown below, Fandrianto discloses that one of the weighted averages used to interpolate highlighted $(1/4, 3/4)$ sub-pixel Q2 is the weighted average ($Q2 = (A + B)/2$) of the pair of diagonally-opposed $(0/4, 2/4)$ sub-pixel B and $(2/4, 4/4)$ sub-pixel A (blue; circled).¹⁹ Moreover, the first and second weights for A and B used in the weighted average both equal “1.”

¹⁹ As explained *supra* §VII.B, A block 72 and B block 74 contain “A” and “B” data, respectively, because Fandrianto does not require a specific order for deriving pixels Q1-Q8. Ramchandran, ¶¶156-158 (citing Fandrianto, 7:26-8:20, 10:51-58, Figures 3B, 5).

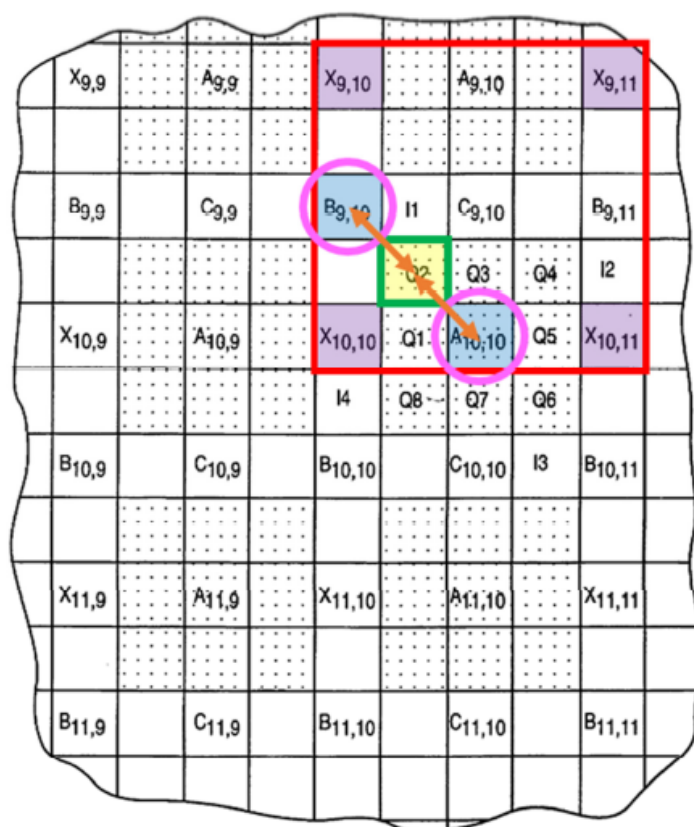


FIG. 5

Fandrianto, Figure. 5, 8:58-9:32; 10:51-58; 8:36-10:58; Ramchandran, ¶240.

As shown above, the sub-pixels A and B (both blue and circled) are equidistant (orange arrows) from the (1/4,3/4) sub-pixel whose value Q2 is being interpolated. Thus, Fandrianto discloses using first and second weights with equal values (both equaling “1”) in the weighted average ($Q = (A + B)/2$) when the (0/4,2/4) sub-pixel B and (2/4,4/4) sub-pixel A (blue)—having coordinates with even values of both K (K=0 or 2) and L (L=2 or 4)—are equidistant from the (1/4,3/4) sub-pixel Q2 having (coordinates with odd values of both K (K=1) and L (L=3)) whose value (Q2) is being interpolated. Ramchandran, ¶240.

Accordingly, the TML6-Fandrianto combination interpolates TML6's $(1/4, 1/4)$, $(1/4, 3/4)$, $(3/4, 1/4)$, and $(3/4, 3/4)$ sub-pixels having coordinates with odd values of both K and L ($K/2^N$, **K=1 or 3**, $N=2$) and L ($L/2^N$, **L=1 or 3**, $N=2$), as recited by Claim 25. Ramchandran, ¶241.

d. Claim 42

As discussed for Claim 1, in the TML6-Fandrianto combination, $N=2$. Ramchandran, ¶242.

e. Claim 44

As discussed for Claim 22, a POSITA would have understood the apparatus of Claim 22 to require a video encoder. Ramchandran, ¶243. Accordingly, TML6 and Fandrianto teach “a video encoder comprising an apparatus for sub-pixel value interpolation” as recited in Claim 22. Ramchandran, ¶244.

f. Claim 27

A POSITA would have understood that TML6 and Fandrianto are designed for implementation in “[a] communications terminal comprising a video encoder.” Ramchandran, ¶246. VCEG developed TML6 for use in applications that receive packetized data over IP networks, which requires communications terminals. VCEG-L45, 34-36. Furthermore, Fandrianto's technique is designed for digital television broadcast and video telephony systems. Fandrianto, 1:12-41; Ramchandran, ¶246. As discussed for Claim 44, TML6 and Fandrianto both teach a

“video encoder.” Accordingly, TML6 and Fandrianto teach a communications terminal comprising a video encoder. Ramchandran, ¶¶244-246.

g. Claim 45

Claim 45 is substantively the same as Claim 44, except that it is directed to a “**still image** encoder” rather than a “**video** encoder.” Ramchandran, ¶247. As Dr. Ramchandran explains, video comprises a series of still images played in rapid succession to create the illusion of motion. Ramchandran, ¶247. Each image contains numerous pixels, like a typical still image. When compressing or encoding video, techniques such as transmitting pixel data for a single frame and reusing it for subsequent frames, or using motion vectors to indicate movement, rely on encoding individual frames, or still images. Ramchandran, ¶247.

TML6 expressly teaches video formats, e.g., QCIF and CIF. TML6, 1. A POSITA familiar with such formats would have understood the underlying principles of image resolution and frame composition such that they could easily apply these principles to extract and save individual frames as still images in formats like JPEG or PNG, since video frames are essentially a sequence of still images. The conversion between video frames and still images was a common practice, facilitated by available software tools and hardware capabilities at that time. Therefore, a “still image encoder” would have been within the expertise of a POSITA. Ramchandran, ¶247.

h. Claim 46

If “video decoder” is limiting, a POSITA would have understood the apparatus of Claim 22 to be implemented in a video decoder. Ramchandran, ¶243. As discussed for Claim 22, TML6 is designed for a video coder in an apparatus. Ramchandran, ¶244. TML6 encodes and decodes video data that includes either a video encoder or decoder. Ramchandran, ¶244. Accordingly, TML6 and Fandrianto teach a “video decoder comprising an apparatus for sub-pixel interpolation” as recited in Claim 22. Ramchandran, ¶245.

i. Claim 28

For similar reasons discussed in Claim 27, a POSITA would have understood that TML6 is designed for communications terminal comprising a video decoder. Ramchandran, ¶246. As discussed for Claim 46, TML6 and Fandrianto teach a “video decoder.” Accordingly, TML6 and Fandrianto teach a communications terminal comprising a video decoder. Ramchandran, ¶¶244-246.

j. Claim 47

Claim 47 is substantively the same Claim 27, except that it is directed to a “still image encoder” rather than a “video encoder.” As discussed for Claim 45, a POSITA would have understood that a video encoder must be capable of processing each frame individually as a still image, and by extension, acts as a still image encoder. Ramchandran, ¶247. Accordingly, TML6 and Fandrianto teach “[a]

communications terminal comprising a still image encoder.” Ramchandran, ¶¶246-247.

k. Claim 48

Claim 48 is substantively the same as Claim 46 except that it is directed to a “**still image** decoder” rather than a “**video** decoder.” Ramchandran, ¶247. For similar reasons as discussed in Claim 45, a POSITA would have understood that a video decoder must be capable of processing each frame individually as a still image, and by extension, acts as a still image decoder. Ramchandran, ¶247.

l. Claim 49

TML6 discloses or renders obvious a “codec” that includes a video encoder and decoder, as recited by Claim 49. Ramchandran, ¶¶243-244. A codec is a device or software “that performs encoding and decoding of communications protocols.” CRC-Dictionary, 28. As discussed for Claim 22, TML6 and Fandrianto both encode/decode video data and thus disclose a video codec. Ramchandran, ¶¶243-246.

m. Claim 50

Claim 50 is substantively the same as Claim 28 except it is directed to a “**still image** decoder” rather than a “**video** decoder.” Ramchandran, ¶248. As discussed for Claims 28 and 48, TML6 and Fandrianto teach the “still image decoder” and “communications terminal” of Claim 50. Ramchandran, ¶248. Accordingly, TML6

and Fandrianto teach a “communications terminal comprising a still image decoder.”
Ramchandran, ¶248.

n. Claim 51

Claim 51 is substantively the same as Claim 49 except it is directed to a “**still image** encoder...and...decoder” rather than a “**video** encoder...and...decoder.”
Ramchandran, ¶249. As discussed for Claims 45 and 48, a POSITA would have understood that a video encoder/decoder must be capable of processing each frame individually as a still image, and by extension, acts as a still image encoder/decoder.
Ramchandran, ¶249. Accordingly, TML6 and Fandrianto teach the “codec” and “still image encoder...and...decoder” of Claim 51. Ramchandran, ¶249.

B. Ground 2: Claims 21 and 43 Are Obvious Over TML6 in View of Fandrianto and the Knowledge of a POSITA.

1. Independent Claim 21

Claim 21 is substantially the same as Claim 20 but is directed to “**eighth** resolution” rather than “quarter resolution...interpolation.” The difference in resolution is reflected in both the coordinate notations and the sub-pixels recited in Claim 21. Ramchandran, ¶250.

A POSITA would have been motivated to modify the TML6-Fandrianto system to achieve eighth resolution interpolation. Ramchandran, ¶257 As discussed for 1[pre][b], TML6—and thus the combined TML6-Fandrianto system—discloses the case where $N=2$. Setting $N=3$ would have been an obvious, efficient, and

straightforward method of achieving eighth resolution interpolation that would have provided improved video quality. Ramchandran, ¶250.

A POSITA would have been motivated to modify the TML6-Fandrianto combination for eighth pixel resolution because the prior art expressly motivates the modification. *See KSR*, 550 U.S. at 418-419. By 2001, sub-pixel interpolation was known to enhance the resolution of digital images. Ramchandran, ¶¶251-253. Eighth resolution sub-pixel accuracy offered a desirable balance between image quality and computational complexity. For instance, the draft successor to TML6—TML8, published in July 2001—implemented 1/8 pixel accuracy “[f]or a higher complexity or higher coding efficiency profile.” VCEG-N10, 1, 14; Ramchandran, ¶255.

A POSITA would have recognized that setting $N = 3$ —instead of 2, as disclosed in TML6, *see* 1[pre][b]—would achieve eighth pixel resolution in the TML6-Fandrianto system. As Dr. Ramchandran explains, $N=3$ was specifically known to be a desirable value to enhance the resolution and smoothness of an approximated function. Ramchandran, ¶254 (citing Press, 106-107).

A POSITA would have recognized setting $N=3$ to be the use of a known technique (higher-order interpolation) to improve similar devices (digital video systems) in the same way (increasing resolution and smoothness using sub-pixel interpolation). Ramchandran, ¶253. A POSITA would have further recognized that

modifying the TML6-Fandrianto combination to achieve eighth pixel resolution would have been the application of a known technique (1/8 sub-pixel interpolation) to a known device (video encoder/decoder) ready for improvement to yield predictable results (video coding system with improved video fidelity). *KSR*, 550 U.S. at 417; Ramchandran, ¶254.

A POSITA would have had a reasonable expectation of success with predictable results because setting $N=3$ instead of 2 in the TML6-Fandrianto combination would have been a trivial modification within the skill of a POSITA, and the system would have worked in the same way with that modification. Ramchandran, ¶256. Furthermore, as explained above, eighth resolution sub-pixel value interpolation was achievable on hardware available in 2001, making it a practical choice for high-resolution displays and imaging applications of the time. Ramchandran, ¶256.

2. Claim 43

The TML6-Fandrianto combination in view of the knowledge of a POSITA renders obvious the additional limitations of this claim for the reasons discussed for Claims 21-22. Ramchandran, ¶258.

IX. NO BASIS FOR DISCRETIONARY INSTITUTION DENIAL

A. *General Plastic*

This is Amazon’s first petition for IPR of the ’599 patent. The ’599 patent was challenged in three prior IPRs filed in 2020 by Lenovo (United States) Inc. (“Lenovo”): IPR2021-00037, IPR2021-00038, and IPR2021-00039. Although *General Plastic* addressed IPR petitions challenging the same patent filed by “the same petitioner,” the Board in *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062, Paper 11 (Apr. 2, 2019) (precedential) extended *General Plastic* factor 1 to include IPR petitions challenging the same patent filed by petitioners sharing a “significant relationship...with respect to [the challenged patent].” *Valve* at 9-10.

Amazon has never had a “significant relationship” with Lenovo regarding the ’599 patent. See *Ford Motor Co. v. Neo Wireless LLC*, IPR2023-00763, Paper 28, 7 (Mar. 22, 2024). In *Ford*, the Director vacated and remanded the Board’s discretionary denial because the first and second petitioners’ “relationship...[was] premised on the allegation that they each infringe the same patent, but with different allegedly infringing products and in different district court proceedings.” *Ford* at 10. The same is true of Amazon and Lenovo. Patent Owner filed suit against Lenovo in the Eastern District of North Carolina on September 25, 2019. EX-1019. Patent Owner and Lenovo stipulated to dismissal of that case on April 12, 2021. EX-1020. Patent Owner filed suit against Amazon in the District of Delaware on

October 31, 2023—over two-and-a-half years after dismissal of the Lenovo case—alleging infringement of different products. EX-1010. Amazon and Lenovo thus lack the “closely aligned interests with respect to the [’599] patent” necessary for a “significant relationship” to exist. *Ford* at 9-10.

“[W]here, as here, the first and second petitioners are neither the same party, nor possess a significant relationship under *Valve, General Plastic* factor one necessarily outweighs the other *General Plastic* factors.” *Videndum Production Solutions, Inc. v. Rotolight Ltd.*, IPR2023-01218, Paper 12, 6 (Apr. 19, 2024). Moreover, there have been no communications, written or oral, between Amazon and Lenovo relating to the ’599 patent or the filing of this Petition, including its financing, preparation, editing, review, approval, or filing. No individuals acting for or on behalf of Lenovo participated or assisted in any way with any of these activities. There have been no payments or agreements between Lenovo and Amazon, directly or indirectly, in connection with this Petition or the ’599 patent.

The Board “frequently decline[s] to exercise discretionary denial under *General Plastic* where there is no ‘significant relationship’ between parties challenging the same patent.” *Samsung Bioepis Co. Ltd. v. Regeneron Pharms., Inc.*, IPR2023-00442, Paper 10 at 46 (July 19, 2023). The Board should do so here.

B. *Fintiv*

There are two co-pending district court proceedings involving the '599 patent. Both were initiated in late 2023, and no schedules have been entered. Any trial would therefore be expected well after the FWD deadline, likely December 2025. Further, it is unclear whether any overlap exists between this Petition and issues that might be raised in the district court, as the parties have not yet exchanged preliminary infringement or invalidity contentions. Discretionary denial under *Fintiv* is thus unjustified.

C. §325(d)

The Board should likewise not exercise its discretion under §325(d) to deny institution. *See Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential); *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, 9-11 (Feb. 13, 2020) (precedential).

The Examiner did not substantively consider either TML6 or Fandrianto alone or in combination during prosecution of the application that led to the '599 patent. Although the '599 patent accurately summarizes certain portions of the TML6 sub-pixel interpolation scheme in its Background section (EX-1001, 11:44-13:53), the TML6 source code relied on herein (EX-1005) includes far greater coverage of the components and functionalities of the TML6 codec and was thus not considered by

the Examiner. *See Westinghouse Air Brake Techs. Corp. v. Siemens Indus., Inc.*, IPR2017-02044, Paper 12, 34 (Feb. 7, 2018) (declining under §325(d) to deny petition that relied on a prior art reference that was merely summarized in the Background section of the challenged patent). The '599 patent itself indicates TML6 was not considered substantively because TML6 is not listed among the numerous references cited. EX-1001, (56). There is also no evidence that TML6 was provided to the Office by the applicant, such as on an IDS. *See* EX-1002.

TML6 is also not “substantially the same [as] prior art or arguments previously...presented to the Office” under §325(d). The '599 patent's Background summarizes certain aspects of TML6's sub-pixel interpolation scheme with reference to Figures 13(a)-(c). EX-1001, 11:44-13:53. But the '599 patent's two-column description lacks the detailed information found in TML6's nearly 10,000 lines of code, for example, how the *oneforthpix()* function interacts with the *imgY*, *img4Y_tmp*, and *mref* data structures. *See supra* Claim 1. Accordingly, TML6 was not considered by the Examiner.

In any event, denial under §325(d) is unwarranted because the Examiner did not consider the specific combination of TML6 and Fandrianto. *Volkswagen Grp. of Am., Inc. v. Mich. Motor Techs. LLC*, IPR2020-00169, Paper 8, 26 (June 2, 2020). Fandrianto was cited in an IDS but never applied against the claims nor discussed by the Examiner. EX-1002, 211. A reference that “was neither applied against the

claims nor discussed by the Examiner” does not weigh in favor of exercising the Board’s discretion under §325(d) to deny a petition. *See, e.g., Zip-Top LLC v. Stasher, Inc.*, IPR2018-01216, Paper 14, 35-36 (Jan. 17, 2019) (“[M]ere citation to a reference by the Examiner does not establish that the Examiner substantively considered the merits of [the reference].”); *see Shenzhen Zhiyi Tech. Co. v. iRobot Corp.*, IPR2017-02137, Paper 9, 9-10 (Apr. 2, 2018).

X. CONCLUSION

Inter partes review of the Challenged Claims is respectfully requested.

XI. MANDATORY NOTICES

A. Real Party in Interest

Petitioner identifies themselves as real parties in interest.

B. Related Matters

To the best of Petitioner’s knowledge, the ’599 patent has been involved in the following related matters:

- *Nokia Techs. Oy v. Lenovo (Shanghai) Elecs. Tech. Co.*, No. 5:19-cv-00427 (E.D.N.C.)
- *Nokia Techs. Oy v. Amazon.com, Inc.*, No. 1:23-cv-01236 (D. Del.)
- *Nokia Techs. Oy v. HP Inc.*, No. 1:23-cv-01237 (D. Del.)
- IPR2021-00037
- IPR2021-00038

- IPR2021-00039

C. Notice of Counsel and Service Information

Lead Counsel	Backup Counsel
Jessica Kaiser, Reg. No. 58,937 PERKINS COIE LLP 1900 Sixteenth Street, Suite 1400 Denver, CO 80202 Tel: (303) 291-2300 Kaiser-ptab@perkinscoie.com	Robin Brewer, Reg. No. 60,951 PERKINS COIE LLP 505 Howard Street, Suite 1000 San Francisco, CA 94105 Tel: (415) 344-7000 Brewer-ptab@perkinscoie.com Jonathan R. Carter, Reg. No. 75,145 PERKINS COIE LLP 1155 Avenue of the Americas 22nd Floor New York, NY 10036 Tel: (212) 261-6826 Fax: (212) 977-1649 Carter-ptab@perkinscoie.com

Petitioner consents to electronic service at the following email address:

Amazon-NokiaIPR@perkinscoie.com.

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

Date: July 5, 2024

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(d)

Pursuant to 37 C.F.R. §42.24(a), Petitioner hereby certifies that portions of the above-captioned Petition for *inter partes* review of U.S. Patent No. 7,280,599, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,977. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ *Jessica Kaiser* /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: July 5, 2024
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *inter partes* review of U.S. Patent No. 7,280,599 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

Perman & Green
425 Post Road
Fairfield, CT 06430-6232

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Brian E. Farnan
bfarnan@farnanlaw.com
Michael J. Farnan
mfarnan@farnanlaw.com
Warren H. Lipschitz
wlipschitz@mckoolsmith.com
Alexandra F. Easley
aeasley@mckoolsmith.com
Joshua Budwin
jbudwin@mckoolsmith.com
R. Mitch Verboncoeur
mverboncoeur@mckoolsmith.com
Josh Newcomer
jnewcomer@mckoolsmith.com
Kevin Burgess
kburgess@mckoolsmith.com
Theodore Stevenson, III
ted.stevenson@alston.com
John D. Haynes
john.haynes@alston.com
Nicholas T. Tsui

nick.tsui@alston.com
Shawn Gannon
shawn.gannon@alston.com

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: July 5, 2024