

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.

IPR2024-01139

U.S. Patent No. 6,950,469

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 6,950,469**

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	2
IV.	'469 PATENT OVERVIEW	2
	A. Technology Background	2
	B. The Alleged Problem in the Prior Art	8
	C. The Alleged Invention of the '469 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 The GROUND.....	21
	A. Ground 1: Obviousness over TML6 and Fandrianto (Claims 1-7, 9, and 14-51).....	21
	1. Motivation to Combine	21
	2. Claims 1-7, 9, 14-17, 20-22	29
	3. Independent Claim 18	83
	4. Independent Claim 19.....	83
	5. Claims 23-26	86
	6. Claims 27-30	88
	7. Claims 31-33	89
	8. Independent Claim 34	91
	9. Claims 35-41	92
	10. Claims 42-45	94

11.	Claims 46-48	95
12.	Independent Claim 49.....	96
13.	Independent Claims 50-51	96
IX.	NO BASIS FOR DISCRETIONARY INSTITUTION DENIAL.....	96
A.	<i>General Plastic</i>	96
B.	<i>Fintiv</i>	98
C.	§325(d)	99
X.	CONCLUSION.....	101
XI.	MANDATORY NOTICES	101
A.	Real Party in Interest	101
B.	Related Matters.....	101
C.	Notice of Counsel and Service Information.....	102

EXHIBIT LIST

Exhibit	Reference
1001	U.S. Patent No. 6,950,469 (“the ’469 patent”)
1002	File History of the ’469 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)
1014	Iain Richardson, <i>Video Codec Design: Developing Image and Video Compression Systems</i> (2002)
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)
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”)

LISTING OF CHALLENGED CLAIMS

Claim 1	
1[pre][a]	A method of interpolation in video coding in which an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range [is interpolated],
1[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
1[pre][c]	[the image] is interpolated to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the method comprising:
1[a]	a) when values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations are required, interpolating such values directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
1[b]	b) when values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations are required, interpolating such values directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a); and
1[c]	c) when a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location is required, interpolating such a value by taking a weighted average of the value of a first sub-pixel or pixel situated at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located

	diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ vertical location.
Claim 2	
2	A method according to claim 1 wherein a first and a second weight are used in the weighted average referred to in (c), the relative magnitudes of the weights being inversely proportional to the (straight-line diagonal) proximity of the first and the second sub-pixel or pixel to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ vertical location.
Claim 3	
3	A method according to claim 2, wherein in a situation where the first and the second sub-pixel or pixel are symmetrically located with respect to (equidistant from) the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ vertical location, the first and second weights have equal values.
Claim 4	
4	A method according to claim 1 in which the first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations in step b) is used when a sub-pixel at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^N$ unit vertical location is required.
Claim 5	
5	A method according to claim 1 in which the second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations in step b) is used when a sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location is required.
Claim 6	
6	A method according to claim 1 in which, when values for sub-pixels at $\frac{1}{2}^N$ unit horizontal and unit vertical locations, and $\frac{1}{2}^N$ horizontal and $\frac{1}{2}^{N-1}$ vertical locations are required, interpolating such values by taking the average of the values of a first pixel or sub-pixel located at a vertical location corresponding to that of the sub-pixel being calculated and unit horizontal location and a second pixel or sub-pixel located at a vertical location corresponding to that of the sub-pixel being calculated and $\frac{1}{2}^{N-1}$ unit horizontal location.

Claim 7	
7	A method according to claim 1 in which, when values for sub-pixels at unit horizontal and $\frac{1}{2}^N$ unit vertical locations, and $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations are required, interpolating such values by taking the average of the values of a first pixel or sub-pixel located at a horizontal location corresponding to that of the sub-pixel being calculated and unit vertical location and a second pixel or sub-pixel located at a horizontal location corresponding to that of the sub-pixel being calculated and $\frac{1}{2}^{N-1}$ unit vertical location.
Claim 9	
9	A method according to claim 1 in which values for sub-pixels at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations are interpolated by taking the average of values of a sub-pixel located at a $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical location, and a sub-pixel located at a unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location.
Claim 14	
14	A method according to claim 1 in which when values for some sub-pixels at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations are required, such values are interpolated by taking the average of a plurality of nearest neighbouring pixels.
Claim 15	
15	A method according to claim 1 in which N equals an integer selected from a list consisting of the values 2, 3, and 4.
Claim 16	
16	A method according to claim 1 in which in at least one of step a) and step b) interpolating sub-pixel values directly using weighted sums involves the calculation of an intermediate value for the sub-pixel values having a dynamic range greater than the specified dynamic range.
Claim 17	
17	A method according to claim 14 in which the intermediate value for a sub-pixel having $\frac{1}{2}^{N-1}$ sub-pixel resolution is used

	the calculation of a sub-pixel value having $\frac{1}{2}^N$ sub-pixel resolution.
Claim 18	
18 <pre>[a]</pre>	A method of interpolation in video coding in which an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range [is interpolated],
18 <pre>[b]</pre>	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
18 <pre>[c]</pre>	[the image] is interpolated to generate values for sub-pixels at fractional horizontal and vertical locations, the method comprising:
18[a]	a) when values for sub-pixels at half unit horizontal and unit vertical locations, and unit horizontal and half unit vertical locations are required, interpolating such values directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
18[b]	b) when values for sub-pixels at half unit horizontal and half unit vertical locations are required, interpolating such values directly using a weighted sum of values for sub-pixels residing at half unit horizontal and unit vertical locations calculated according to step (a); and
18[c]	c) when values for sub-pixels at quarter unit horizontal and quarter unit vertical locations are required, interpolating such values by taking the average of at least one pair of a first pair of values of a sub-pixel located at a half unit horizontal and unit vertical location, and a sub-pixel located at a unit horizontal and half unit vertical location and a second pair of values of a pixel located at a unit horizontal and unit vertical location, and a sub-pixel located at a half unit horizontal and half unit vertical location.
Claim 19	
19 <pre>[a]</pre>	A method of interpolation in video coding in which an image comprising pixels arranged in rows and columns and

	represented by values having a specified dynamic range [is interpolated],
19[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
19[pre][c]	[the image] is interpolated to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the method comprising:
19[a]	a) when values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations are required, interpolating such values directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
19[b]	b) when a value for a sub-pixel at a sub-pixel horizontal and sub-pixel vertical location is required, interpolating such a value directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for sub-pixels located at a horizontal location corresponding to that of the sub-pixel being calculated.
Claim 20	
20	A method according to claim 1 in which the sub-pixels used in the first weighted sum are sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and the first weighted sum is used to interpolate a value for a sub-pixel at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.
Claim 21	
21	A method according to claim 1 in which the sub-pixels used in the second weighted sum are sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations and the second weighted sum is used to interpolate a value for a sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location.

Claim 22	
22	A method according to claim 1 in which when values for sub-pixels at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations are required, they are interpolated by taking the average of at least one pair of a first pair of values of a sub-pixel located at a $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical location, and a sub-pixel located at a unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location and a second pair of values of a pixel located at a unit horizontal and unit vertical location, and a sub-pixel located at a $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location.
Claim 23	
23[pre][a]	A video coder for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
23[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
23[pre][c] ¹	the video coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^{[x]}$, where x is a positive integer having a maximum value N, the interpolator being adapted to:
23[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
23[b]	b) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a); and

¹ In Claim 23, the “x” is not superscripted. That appears to be an error.

23[c]	c) interpolate a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location by taking a weighted average of the value of a first sub-pixel or pixel situated at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.
Claim 24	
24	A video coder according to claim 23, comprising a video encoder.
Claim 25	
25	A video encoder according to claim 23, comprising a video decoder.
Claim 26	
26	A codec comprising the video encoder of claim 24 or the video decoder of claim 25.
Claim 27	
27[pre][a]	A communications terminal comprising a video coder for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
27[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
27[pre][c]	the video coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the interpolator being adapted to:
27[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit

	vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
27[b]	b) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a); and
27[c]	c) interpolate a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location by taking a weighted average of the value of a first sub-pixel or pixel situated at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ vertical location.
Claim 28	
28	A communications terminal according to claim 27 comprising a video encoder.
Claim 29	
29	A communications terminal according to claim 27 comprising a video decoder.
Claim 30	
30	A communications terminal according to claim 27 having a video codec comprising a video encoder and a video decoder.
Claim 31	
31[pre][a]	A telecommunications system comprising a communications terminal and a network, the telecommunications network and the communications terminal being connected by a communications link over which coded video can be transmitted, the communications terminal comprising a video coder for coding for coding an image comprising pixels

	arranged in rows and columns and represented by values having a specified dynamic range,
31[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
31[pre][c]	the video coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the interpolator being adapted to:
31[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
31[b]	b) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a); and
31[c]	c) interpolate a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location by taking a weighted average of the value of a first sub-pixel or pixel situated at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.
Claim 32	
32	A telecommunications system according to claim 31 which is a mobile telecommunications system comprising a mobile communications terminal and a wireless network, the connection between the mobile communications terminal and the wireless network being formed by a radio link.

Claim 33	
33	A telecommunications system according to claim 31 in which the network enables the communications terminal to communicate with other communications terminals connected to the network over communications links between the other communications terminals and the network.
Claim 34	
34[pre][a]	A telecommunications system comprising a communications terminal and a network, the telecommunications network and the communications terminal being connected by a communications link over which coded video can be transmitted, the network comprising a video coder for coding for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
34[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
34[pre][c]	the video coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the interpolator being adapted to:
34[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
34[b]	b) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a); and
34[c]	c) interpolate a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location by taking a weighted average of the value of a first sub-pixel or pixel situated at a

	$\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.
Claim 35	
35[pre][a]	A video coder for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
35[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
35[pre][c]	the coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the resolution of the sub-pixels being determined by a positive integer N, the interpolator being adapted to:
35[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
35[b]	b) interpolate a value for a sub-pixel at a subpixel horizontal and sub-pixel vertical location is required directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for subpixels located at a horizontal location corresponding to that of the sub-pixel being calculated.
Claim 36	
36	A video coder according to claim 35 in which the interpolator is further adapted to form the first weighted sum using the values of sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and to use the first weighted sum to interpolate a value for a sub-pixel at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.

Claim 37	
37	A video coder according to claim 35 in which the interpolator is further adapted to form the second weighted sum using the values of sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations and to use the second weighted sum to interpolate a value for a sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location.
Claim 38	
38	A video coder according to claim 35 in which the interpolator is further adapted to interpolate values for sub-pixels at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations by taking the average of at least one pair of a first pair of values of a sub-pixel located at a $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical location, and a sub-pixel located at a unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location and a second pair of values of a pixel located at a unit horizontal and unit vertical location, and a subpixel located at a $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical location.
Claim 39	
39	A video coder according to claim 35, comprising a video encoder.
Claim 40	
40	A video encoder according to claim 35, comprising a video decoder.
Claim 41	
41	A codec comprising the video encoder of claim 39 or the video decoder of claim 40.
Claim 42	
42[pre][a]	A communications terminal comprising a video coder for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
42[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,

42[pre][c]	the coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the resolution of the sub-pixels being determined by a positive integer N, the interpolator being adapted to:
42[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
42[b]	b) interpolate a value for a sub-pixel at a subpixel horizontal and sub-pixel vertical location is required directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for subpixels located at a horizontal location corresponding to that of the sub-pixel being calculated.
Claim 43	
43	A communications terminal according to claim 42 comprising a video encoder.
Claim 44	
44	A communications terminal according to claim 42 comprising a video decoder.
Claim 45	
45	A communications terminal according to claim 42 having a video codec comprising a video encoder and a video decoder.
Claim 46	
46[pre][a]	A telecommunications system comprising a communications terminal and a network, the telecommunications network and the communications terminal being connected by a communications link over which coded video can be transmitted, the communications terminal comprising a video coder for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,

46[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
46[pre][c]	the coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the resolution of the sub-pixels being determined by a positive integer N, the interpolator being adapted to:
46[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
46[b]	b) interpolate a value for a sub-pixel at a subpixel horizontal and sub-pixel vertical location is required directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for subpixels located at a horizontal location corresponding to that of the sub-pixel being calculated.
Claim 47	
47	A telecommunications system according to claim 46 which is a mobile telecommunications system comprising a mobile communications terminal and a wireless network, the connection between the mobile communications terminal and the wireless network being formed by a radio link.
Claim 48	
48	A telecommunications system according to claim 46 in which the network enables the communications terminal to communicate with other communications terminals connected to the network over communications links between the other communications terminals and the network.
Claim 49	
49[pre][a]	A telecommunications system comprising a communications terminal and a network, the telecommunications network and the communications terminal being connected by a communications link over which coded video can be transmitted, the network comprising a video coder for coding

	for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
49[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
49[pre][c]	the coder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the resolution of the sub-pixels being determined by a positive integer N, the interpolator being adapted to:
49[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations;
49[b]	b) interpolate a value for a sub-pixel at a subpixel horizontal and sub-pixel vertical location is required directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for subpixels located at a horizontal location corresponding to that of the sub-pixel being calculated.
Claim 50	
50[pre][a]	A codec for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range,
50[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations,
50[pre][c]	the codec comprising a video encoder and a video decoder, each of the video encoder and the video decoder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the fractional horizontal and vertical locations being defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value N, the interpolator of the video encoder and the interpolator of the video decoder each being adapted to:

50[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations
50[b]	b) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations directly using a choice of a first weighted sum of values for sub-pixels residing at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations and a second weighted sum of values for sub-pixels residing at unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations, the first and second weighted sums of values being calculated according to step (a)
50[c]	c) interpolate a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location by taking a weighted average of the value of a first sub-pixel or pixel situated at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel or pixel located at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, variables m, n, p and q taking integer values in the range 1 to N such that the first and second sub-pixels or pixels are located diagonally with respect to the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location.
Claim 51	
51[pre][a]	A codec for coding an image comprising pixels arranged in rows and columns and represented by values having a specified dynamic range
51[pre][b]	the pixels in the rows residing at unit horizontal locations and the pixels in the columns residing at unit vertical locations
51[pre][c]	the codec comprising a video encoder and a video decoder, each of the video encoder and the video decoder comprising an interpolator adapted to generate values for sub-pixels at fractional horizontal and vertical locations, the resolution of the sub-pixels being determined by a positive integer N, the interpolator of the video encoder and the interpolator of the video decoder each being adapted to:
51[a]	a) interpolate values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical

	locations directly using weighted sums of pixels residing at unit horizontal and unit vertical locations
51[b]	b) interpolate a value for a sub-pixel at a sub-pixel horizontal and sub-pixel vertical location directly using a choice of a first weighted sum of values for sub-pixels located at a vertical location corresponding to that of the sub-pixel being calculated and a second weighted sum of values for sub-pixels located at a horizontal location corresponding to that of the sub-pixel being calculated.

I. INTRODUCTION

Amazon.com, Inc. and Amazon.com Services LLC (“Amazon” or “Petitioner”) request *inter partes* review (“IPR”) of claims 1-7, 9, and 14-51 (“Challenged Claims”) of U.S. Patent No. 6,950,469 (“’469 patent”; EX-1001).

II. GROUNDS FOR STANDING

Petitioner certifies that the ’469 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 ’469 patent. EX-1010.

III. IDENTIFICATION OF CHALLENGE

A. Prior Art

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

Prior Art
TML6 (H.26L Test Model Long Term Number 6) , publicly available by March 28, 2001, prior art at least under §102(a) ² (EX-1005)
Fandrianto (U.S. Patent No. 5,594,813) , issued on January 14, 1997, prior art at least under §102(b) (EX-1006)

² All references to 35 U.S.C. §§102, 103 are to the pre-AIA statutory framework.

B. Grounds

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

Ground	Claims	References
1 (§103)	1-7, 9, 14-51	TML6, Fandrianto

IV. '469 PATENT OVERVIEW

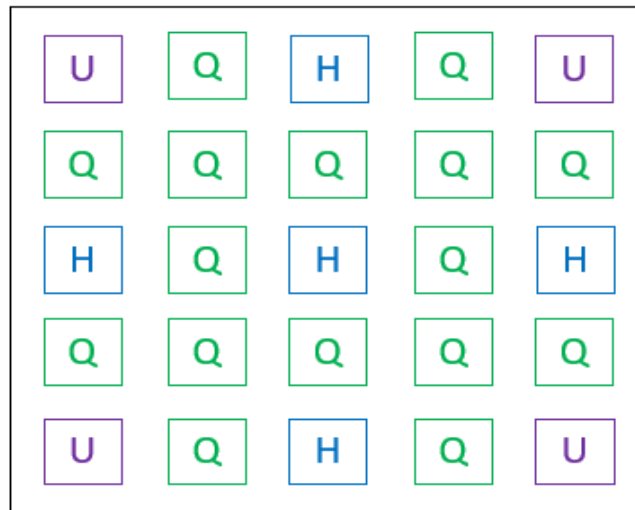
A. Technology Background

Video data consist of sequences of still images, or “frames,” comprising rows and columns of “pixels.” EX-1001, 1:33-35; Ramchandran, ¶¶21-22. “Motion compensated temporal prediction” is a video coding technique that exploits the redundancy typically present between successive frames in a video sequence. EX-1001, 2:57-65; Ramchandran, ¶23. 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, 4:16-24; 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, 4:54-60, 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

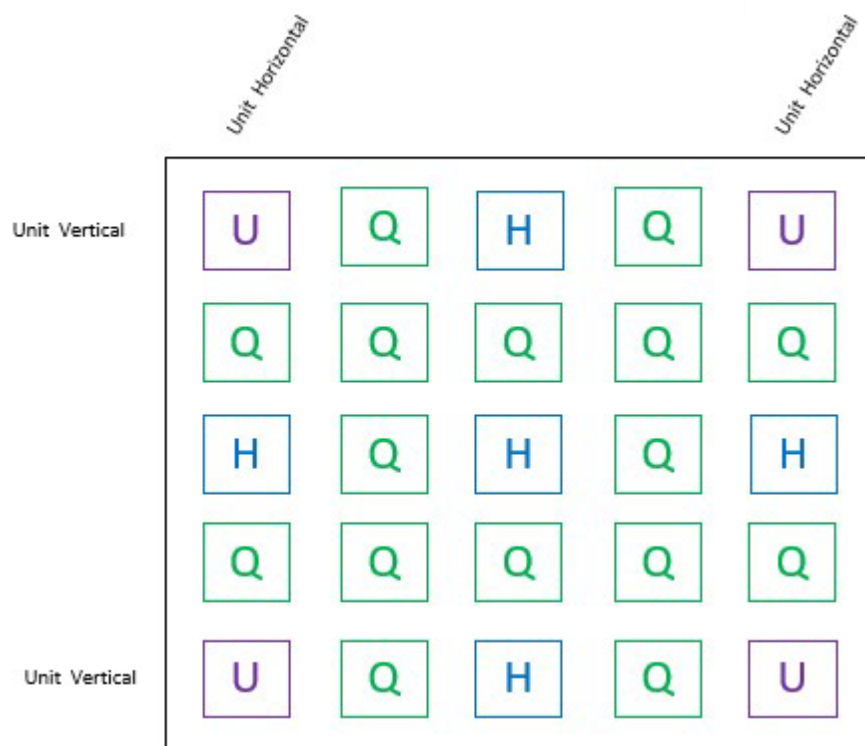
“sub-pixels” must be “interpolated” using actual pixel values. Ramchandran, ¶27. The benefits of sub-pixel interpolation were well-known by the ’469 patent’s earliest possible priority date. *See* VCEG-L45, 15; Fandrianto, 5:26-28; Ramchandran, ¶¶29-33, 43.

For clarity, certain terminology is addressed below. The figure below³ shows four actual pixels, referred to as **unit pixels U** (purple). **Half ($\frac{1}{2}$) sub-pixels H** (blue) are located halfway (horizontally, vertically, or both) between unit pixels. **Quarter ($\frac{1}{4}$) sub-pixels Q** (green) are located quarter-way between unit pixels.

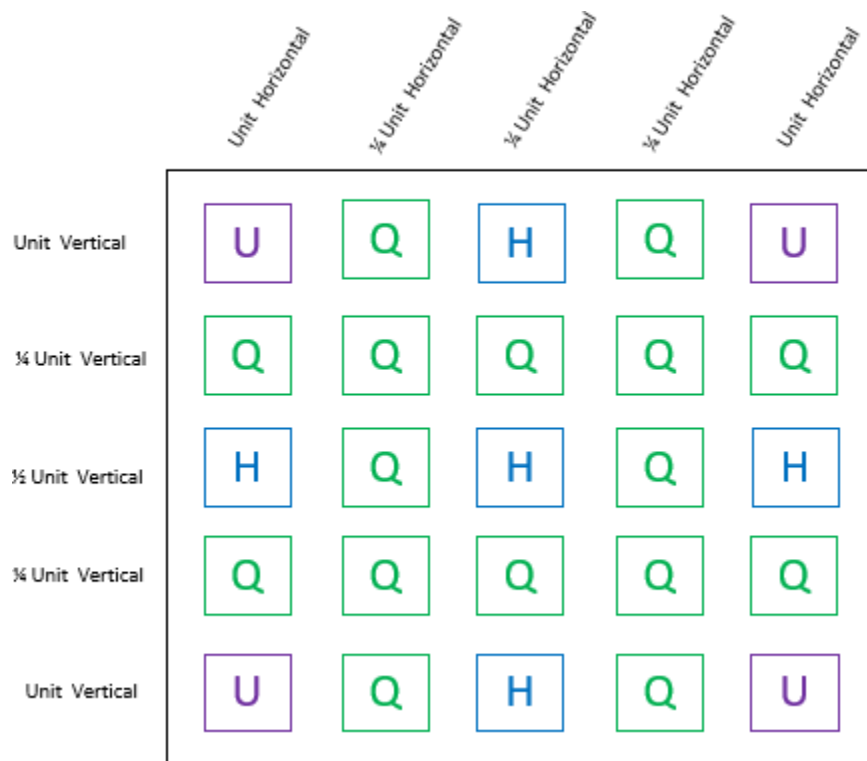


³ Figures like the one below that include a citation to Dr. Ramchandran’s declaration (EX-1003) have been created to graphically illustrate certain concepts in this Petition.

Ramchandran, ¶28. As shown below, **unit horizontal** location refers to a pixel/sub-pixel situated in a **column** that contains unit pixels (purple). Ramchandran, ¶29. Similarly, **unit vertical** location refers to a pixel/sub-pixel situated in a **row** that contains unit pixels. Ramchandran, ¶29. Thus, a unit pixel is situated at both unit horizontal and unit vertical locations.

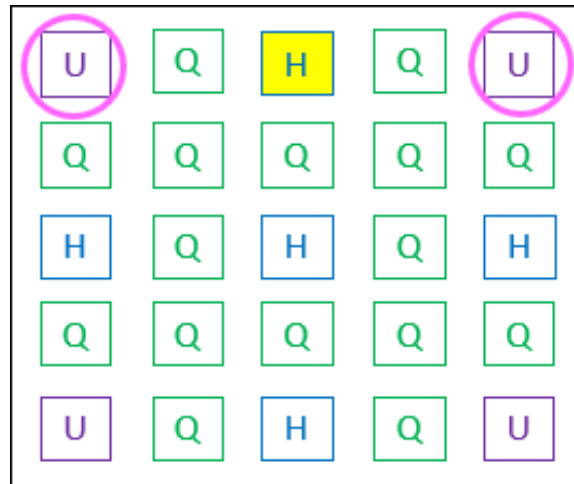


Ramchandran, ¶¶28-29. **Half ($\frac{1}{2}$) unit horizontal location** refers to a position halfway between two unit horizontal locations and **half ($\frac{1}{2}$) unit vertical location** refers to a position halfway between two unit vertical locations. **Quarter ($\frac{1}{4}$) unit horizontal location**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



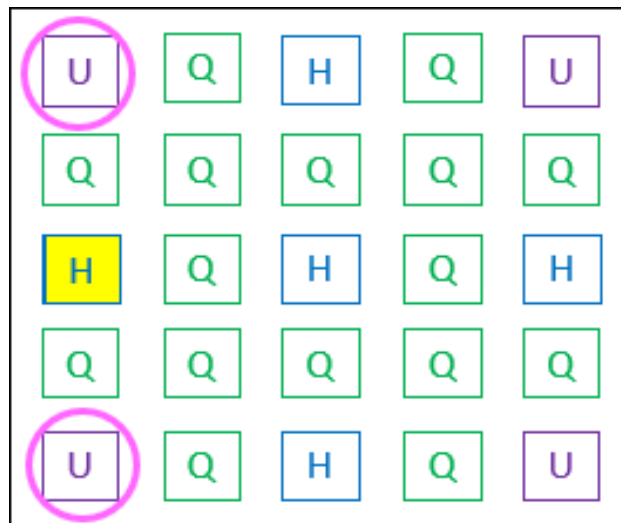
Ramchandran, ¶28.

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



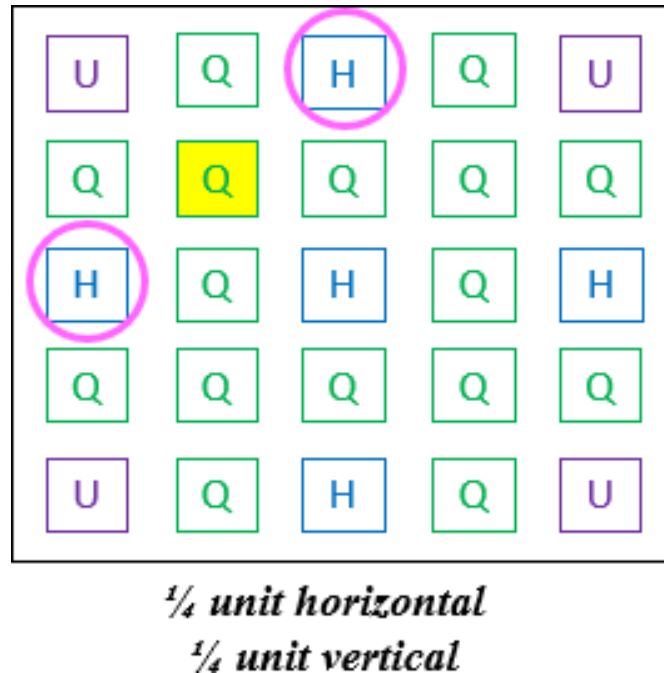
$\frac{1}{2}$ unit horizontal
unit vertical

Ramchandran, ¶¶29-30. Likewise, as shown below, the 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, the highlighted quarter sub-pixel Q (green)—situated at the quarter unit horizontal, quarter unit vertical (or “quarter-quarter”)⁴ location—can be **diagonally interpolated** using previously interpolated half sub-pixels H (blue; circled).



Ramchandran, ¶32.

⁴ This Petition uses the shorthand “quarter-quarter sub-pixel” to refer to a sub-pixel situated at the quarter unit horizontal and quarter unit vertical locations, and similarly uses “half-half sub-pixel” to refer to a sub-pixel situated at the half unit horizontal and half unit vertical locations.

B. The Alleged Problem in the Prior Art

The '469 patent admits that interpolating half and quarter sub-pixels was known. EX-1001, 6:35-41, 7:4-23, 10:4-13:9; Ramchandran, ¶34. For example, the Background describes TML6's sub-pixel interpolation method in 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.

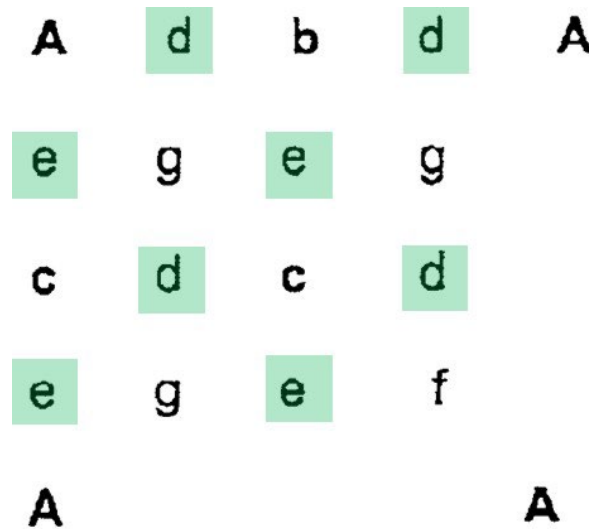


Fig. 13(a)

EX-1001, Figure 13(a), 12:6-44, 21:64-65.

According to the '469 patent, TML6 does not calculate the highlighted quarter sub-pixel "g" (green) (below) using horizontal or vertical interpolation. Instead, sub-pixel "g" is interpolated based on the following four pixels/sub-pixels: pixel "A," sub-pixel "b," and sub-pixels "c" (circled).

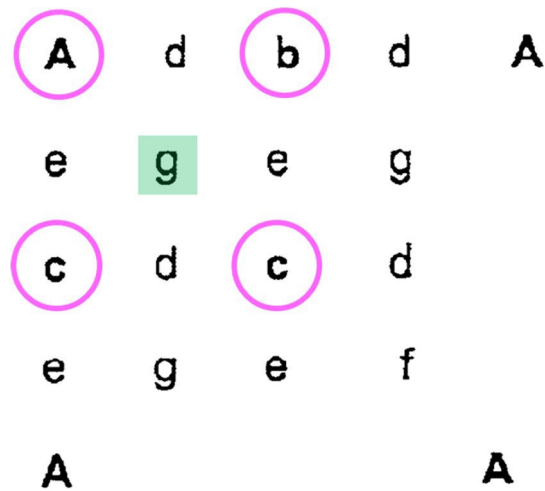


Fig. 13(a)

EX-1001, Figure 13(a), 8:43-53, 12:45-13:9; Ramchandran, ¶¶35-38. The '469 patent asserts, however, interpolating certain sub-pixels based on four other pixels/sub-pixels is too computationally intensive. *See* EX-1001, 37:66-38:24.

C. The Alleged Invention of the '469 Patent

To address this purported problem, as illustrated in Figure 14(a) below, the '469 patent claims as its allegedly novel concept using two diagonally opposite half sub-pixels “b” (blue) to interpolate quarter sub-pixel “h” (green).

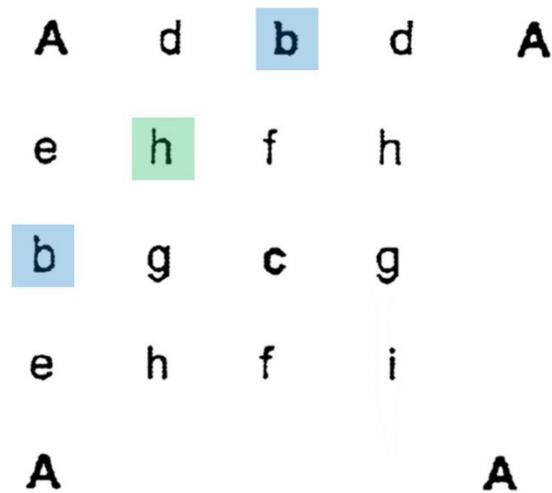


Fig. 14(a)

EX-1001, Figure 14(a), 22:3-4, 36:39-61; Ramchandran, ¶¶39-40. 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, 37:66-38:24; 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 ’469 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.⁵ 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 ’469 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.⁶ *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

⁵ Although the definition of a POSITA proposed in the three prior IPRs challenging the ’469 patent (IPR2020-01685, IPR2021-00035, and IPR2021-00036) differs slightly from the definition proposed here, the Challenged Claims are invalid under both definitions. Ramchandran, ¶42.

⁶ 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, 8:43-53; Wenger, ¶¶15, 21. 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
/tml6pre.zip/lencod/inc/global.h	Pages 35-47

File	Pages in TML6 (EX-1005)
/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, ¶¶44-46. 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 '469 patent.

TML6 implements methods of interpolating half and quarter sub-pixels. TML6, 57-59 (lines 538-632); Ramchandran, ¶¶47-108. As the '469 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, 11:33-39. The '469 patent describes the “interpolation method according to

TML6 as applied in the encoder...with reference to FIGS. 13[a-c].” EX-1001, 11:39-13:39.

First, the '469 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.”

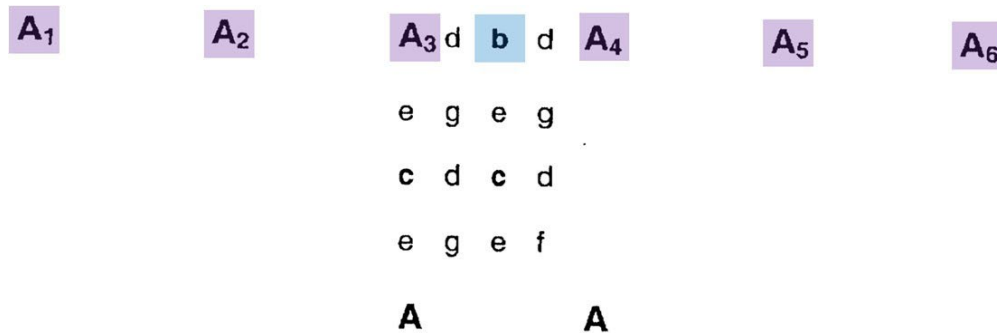
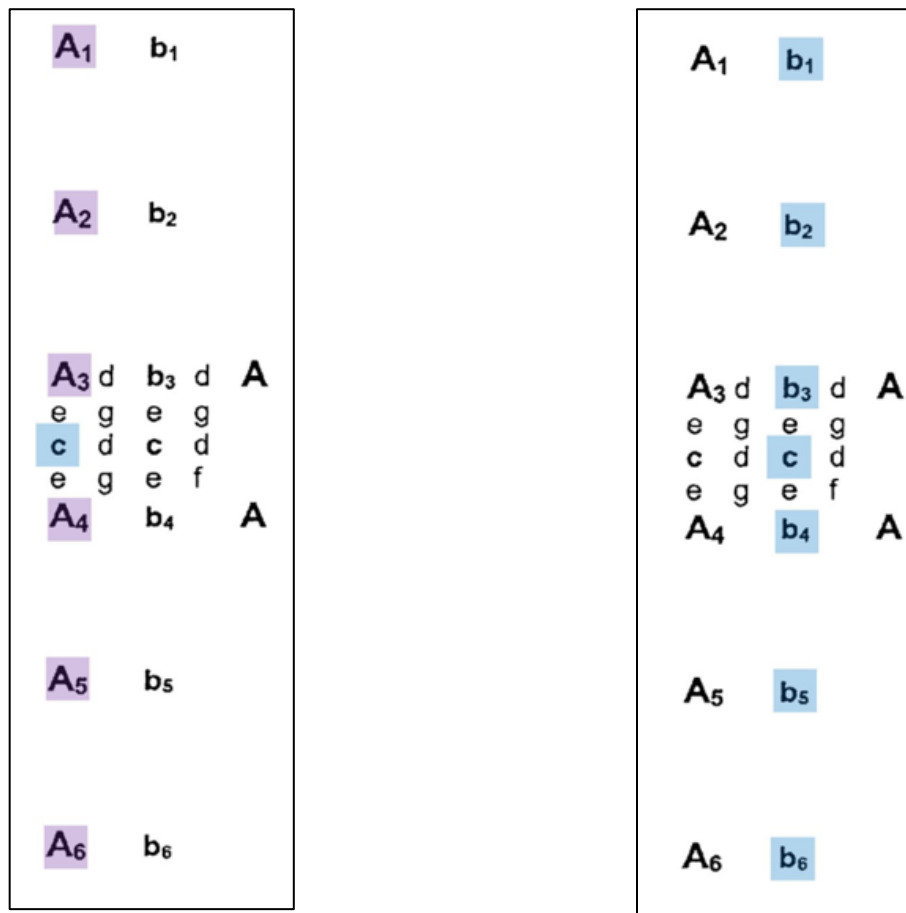


Figure13(b)

EX-1001, Figure 13(b), 11:43-53. 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 '469 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), 11:54-12:5. 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 '469 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].”

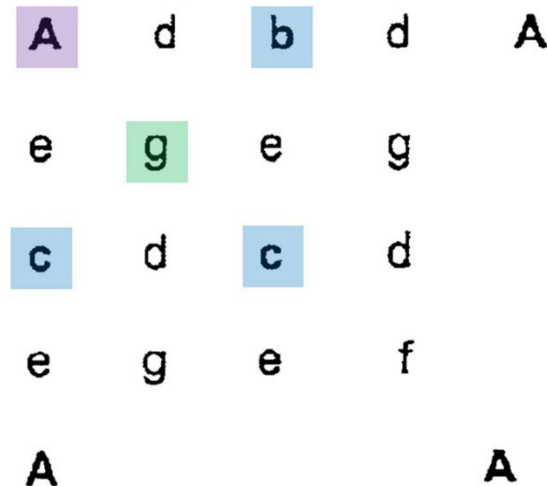


Fig. 13(a)

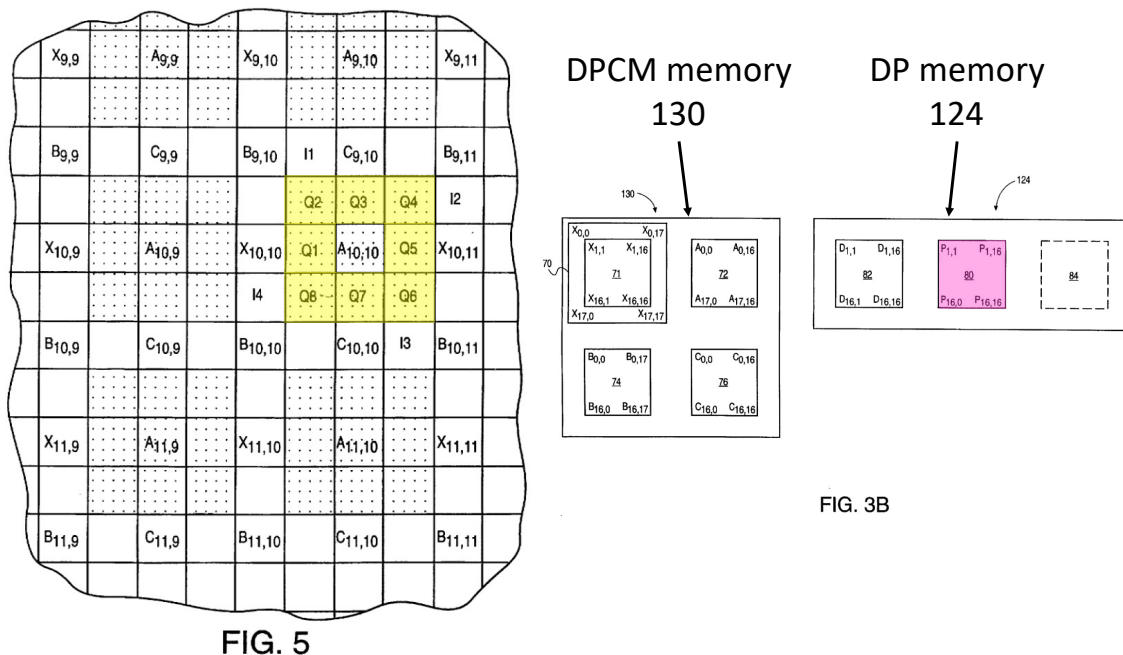
EX-1001, Figure 13(a), 12:45-52. As described for 1[c], the TML6 code confirms this interpolation method. TML6, 58 (lines 593-625); Ramchandran, ¶¶50, 96-101.

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, ¶168.

First, pixel block $X_{0,0}-X_{17,17}$ (purple)⁷ is stored in “X” block 70 (purple) of DPCM memory 130.

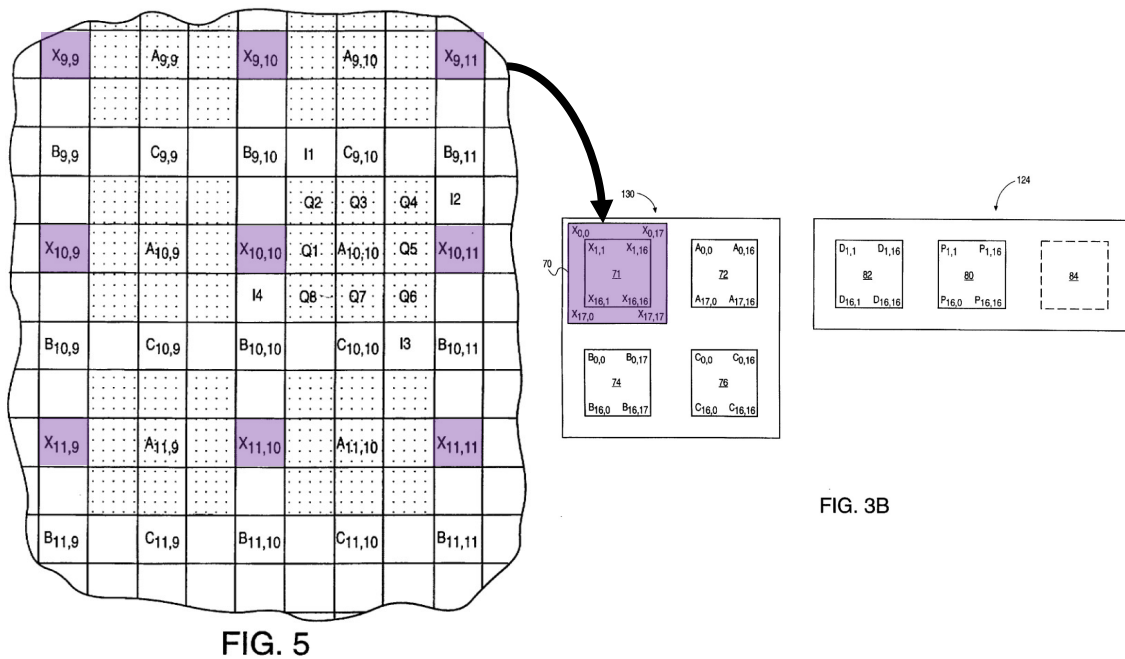


FIG. 3B

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.

⁷ 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.

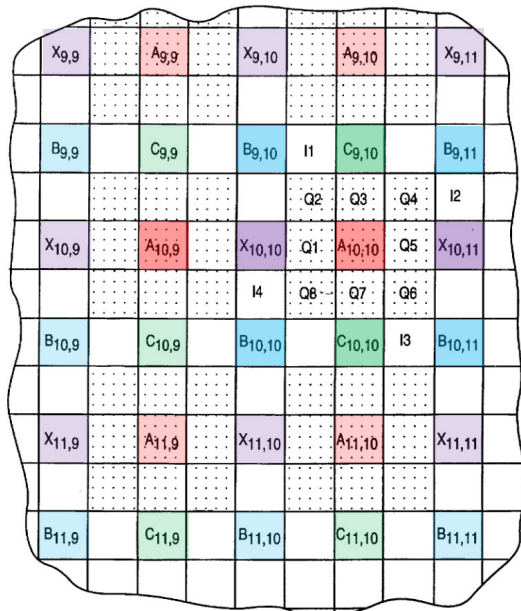


FIG. 5

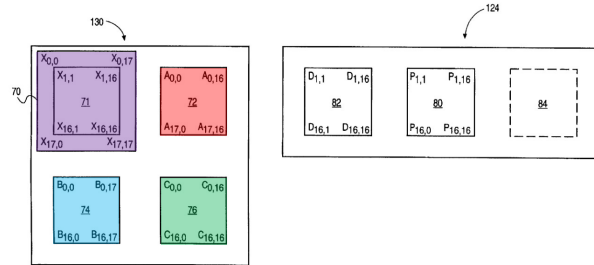


FIG. 3B

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, ¶169.

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, ¶170.

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, ¶171.

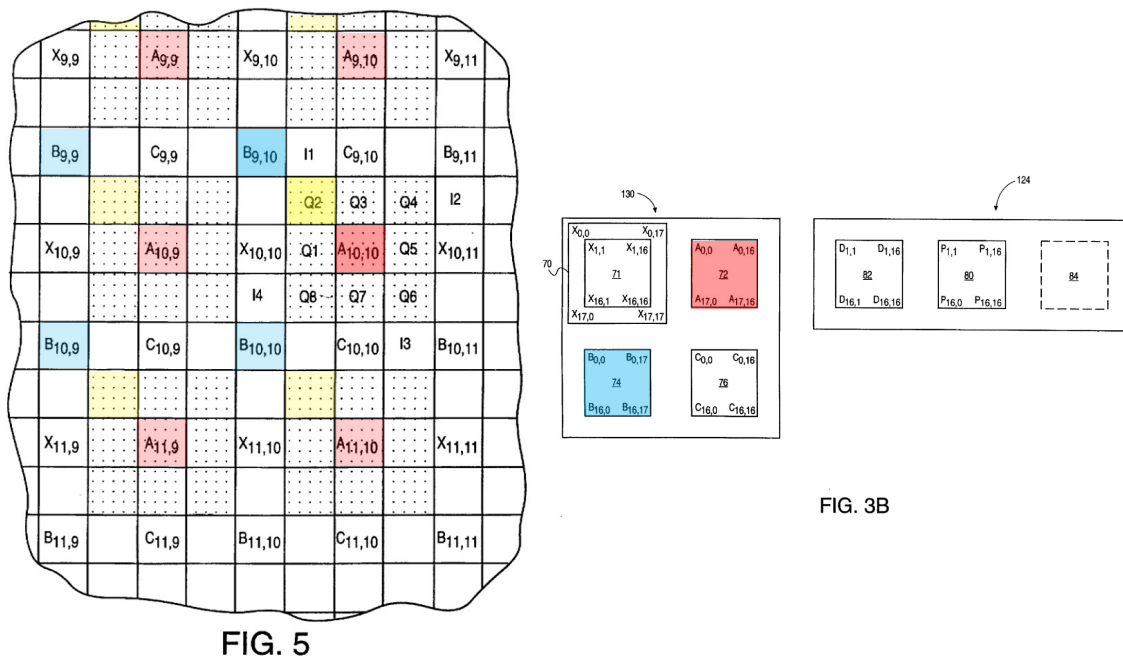


FIG. 3B

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

VIII. DETAILED EXPLANATION OF THE GROUND

A. Ground 1: Obviousness over TML6 and Fandrianto (Claims 1-7, 9, and 14-51)

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, ¶¶119-120. 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”)), 57-58 (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.

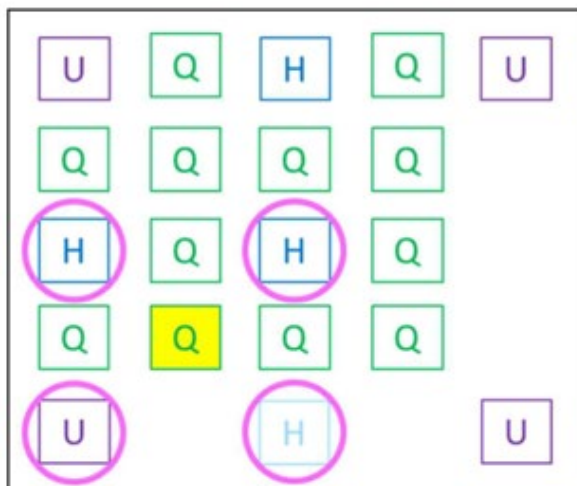
Fandrianto discloses a “limited memory embodiment” capable of interpolating quarter sub-pixels using averages of values for pixels (“X-block”) and/or already interpolated half sub-pixels (“A-block,” “B-block,” and “C-block”). *Id.*, Figure 5, 5:26-32, 7:27-10:58. A POSITA would have been motivated to 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, ¶130.

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, ¶¶125-126.

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 the highlighted quarter-quarter sub-pixel (green) based on the closest unit pixel (purple) and the three closest half sub-pixels (blue). TML6, 58 (lines 593-625). Fandrianto interpolates the highlighted quarter-quarter sub-pixel (green) based on the two closest diagonally opposed half sub-pixels (blue). Fandrianto, Figure 5, 10:51-58.

TML6



Fandrianto

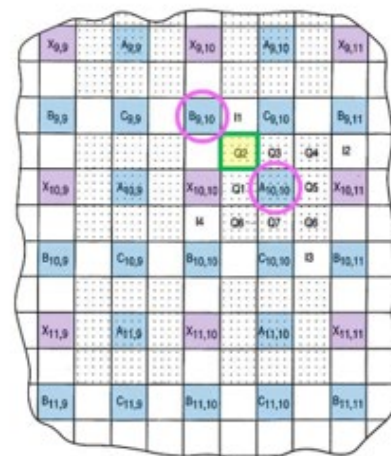
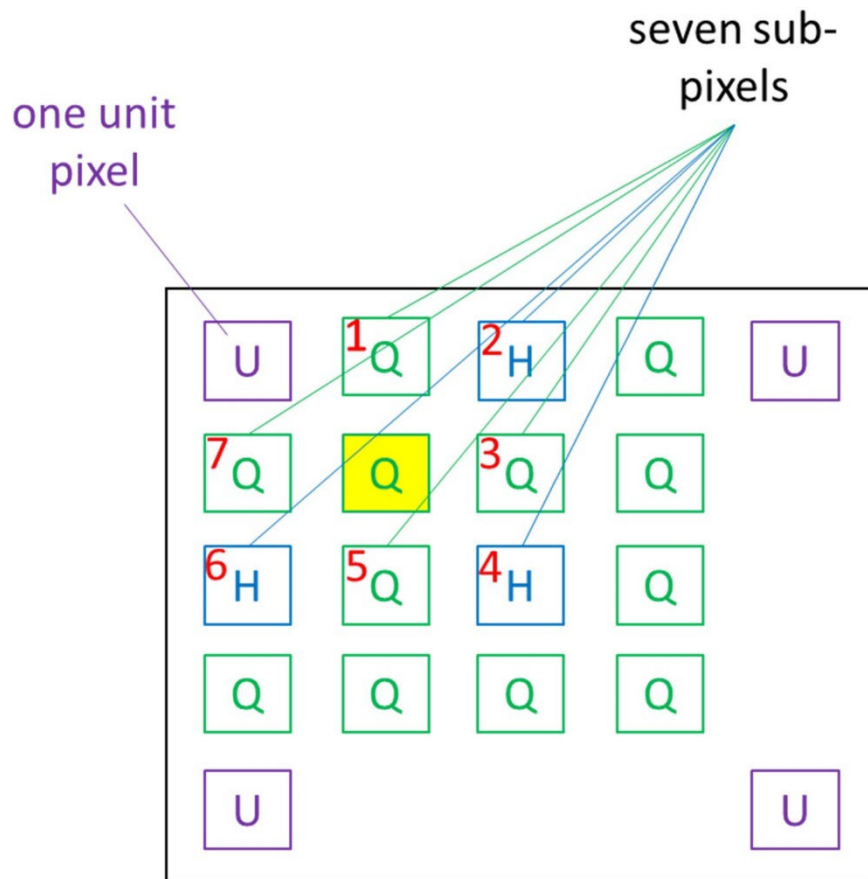


FIG. 5

A POSITA would have substituted Fandrianto's known interpolation method in TML6 to produce the predictable result of interpolating quarter-quarter sub-pixels. Ramchandran, ¶121.

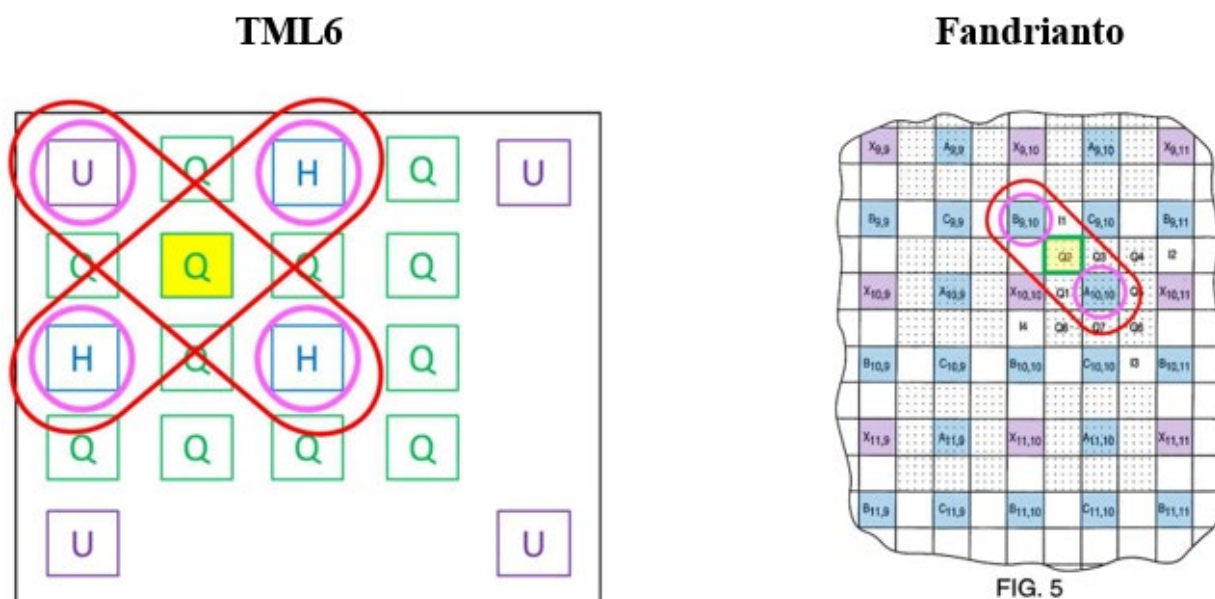
It would have been obvious to try Fandrianto's method to interpolate the quarter-quarter sub-pixels in TML6 because there are a finite number of identifiable and predictable ways to do so. *KSR*, 550 U.S. at 421. Interpolation refers generally to estimating new data points based on assumptions regarding 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. *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, ¶122-123. 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, ¶¶122-124.

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, ¶127.

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:

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

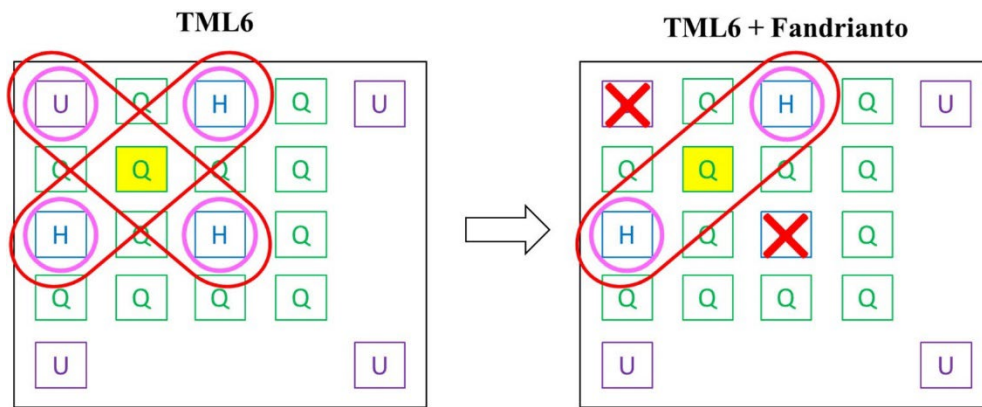
The average of TML6 can be written as:

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

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, ¶128. 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, ¶¶128-129.

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, ¶130.

2. Claims 1-7, 9, 14-17, 20-22

a. Independent Claim 1

(i) 1[pre][a]

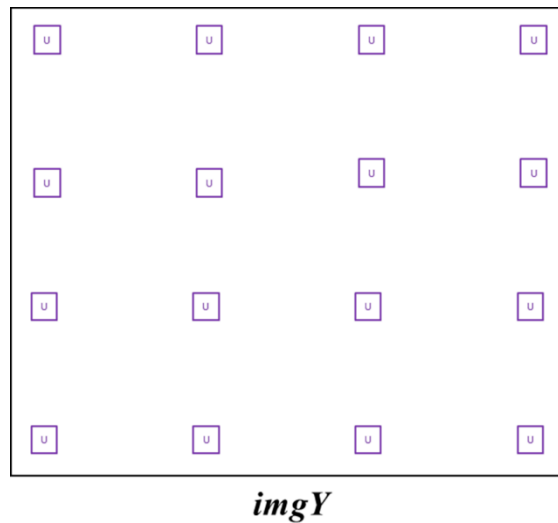
If limiting,⁸ TML6 discloses or renders obvious 1[pre][a]. Ramchandran, ¶¶131-132.

TML6 discloses a method of interpolating an image (frame) of a video signal in video coding. *See* 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 *encode_one_frame()* that “[e]ncodes one frame” of video data. *Id.*, 48-50 (lines 33-162). The *encode_one_frame()* function invokes the function *oneforthpix()* to interpolate sub-pixels for the frame of video data. *Id.*, 49 (lines 102-109), 57-60 (lines 538-719); Ramchandran, ¶¶131-132, 47-108. The ’469 patent confirms TML6

⁸ Petitioner reserves the right to argue in other proceedings the preamble of any challenged claim is limiting.

discloses a method of interpolation in video coding. EX-1001, 8:43-53; *see id.*, 11:33-13:29, Figures 13(a)-13(c); Ramchandran, ¶137.

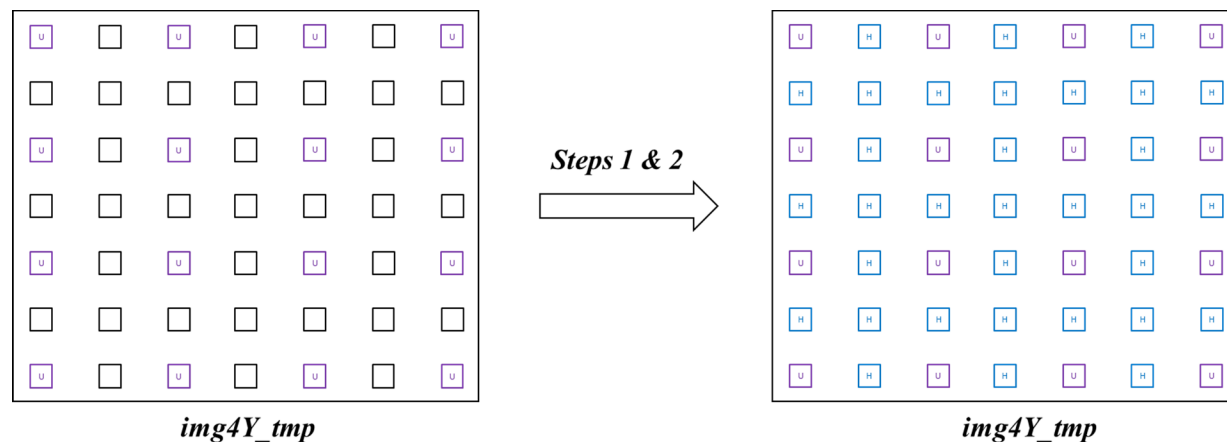
TML6 further discloses that the frame “compris[es] pixels arranged in rows and columns,” as recited. The *oneforthpix()* function uses a data structure *imgY* representing brightness values of unit pixels (purple) for the frame. TML6, 49 (lines 104-105), 57 (line 552).



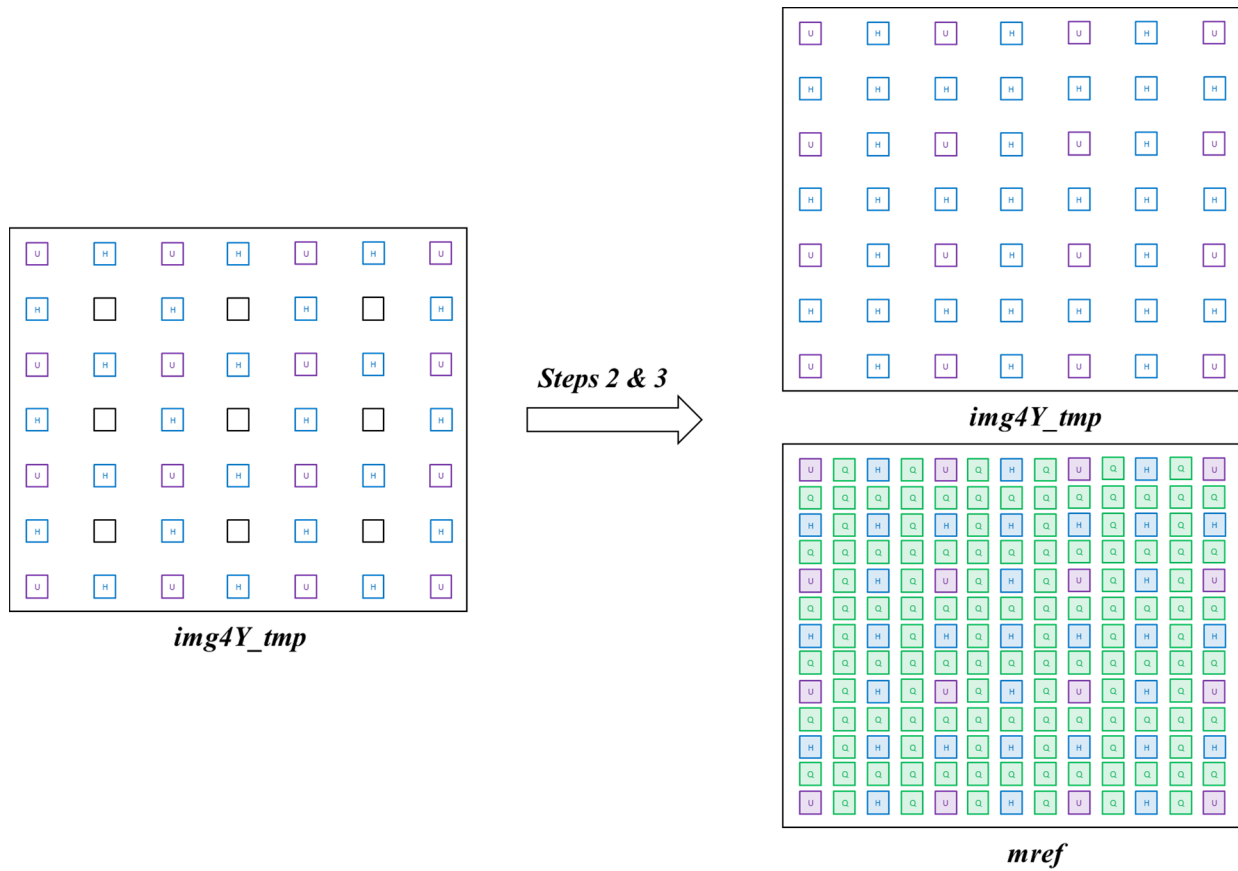
Ramchandran, ¶133. These unit pixels are arranged in an array representing “rows and columns” of pixels. TML6, 93 (lines 1338-1344), 49-50 (lines 97-109), 110-112 (lines 480-600), 57-59 (lines 563-632), 131 (line 149); Ramchandran, ¶¶133-134.

Moreover, the *oneforthpix()* function performs three main steps to interpolate sub-pixel values based on the values of the unit pixels (purple) in the data structure *imgY*. TML6, 57-58 (lines 553-625). In the first two steps, the *oneforthpix()*

function interpolates and stores values for half sub-pixels (blue) in a temporary data structure *img4Y_tmp*.

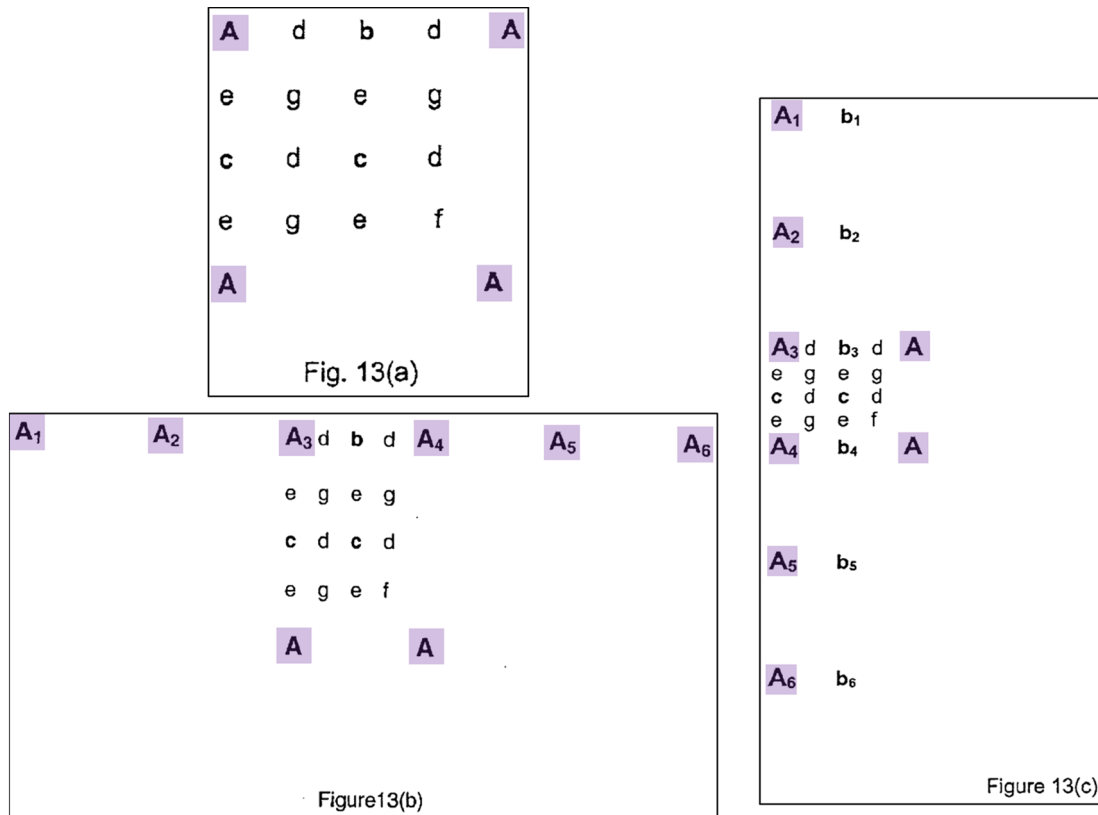


Id.; Ramchandran, ¶135. In steps two and three, the *oneforthpix()* function: stores values for half sub-pixels (blue) in data structure *mref* (based on the values in *img4Y_tmp*); interpolates values for quarter sub-pixels (green); and stores the quarter sub-pixel values in *mref*.



Ramchandran, ¶136 (discussing TML6, 57-58 (lines 576-625)). The unit pixels (purple) in the data structures *img4Y_tmp* and *mref* are likewise arranged in an array representing “rows and columns” of pixels. *Id.*; Ramchandran, ¶¶134-136.

The '469 patent acknowledges TML6 discloses an interpolated image comprising unit pixels (purple) arranged in “rows and columns” as recited in 1[pre][a] and as shown below in Figures 13(a)-13(c).



EX-1001, 8:43-9:23 (“[S]eparate notation, defined in Figure 13a, will be used in the discussion of the sub-pixel value interpolation scheme used in TML6....[T]he letter A represents the location of pixels in the image data representing a frame of a video sequence....”), Ramchandran, ¶138.

The ’469 patent describes “dynamic range” as “the range of values which the sub-pixel values and the weighted sums can take.” EX-1001, 16:53-54. In TML6, the values stored in *imgY* and *mref* are 8-bit values representing integers from 0-255. TML6, 57-58 (lines 563-625), 131 (lines 131-132), 72-73 (lines 215-221 (“currently only 8 bit/pel is supported”)); Ramchandran, ¶¶139-140. The ’469 patent confirms TML6 discloses a dynamic range of “a minimum value of 0 and a

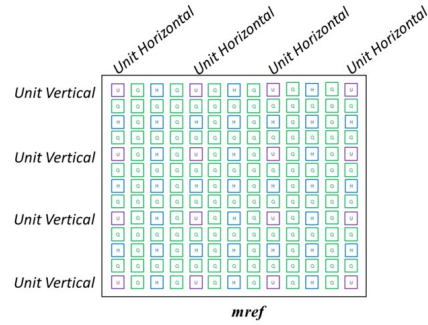
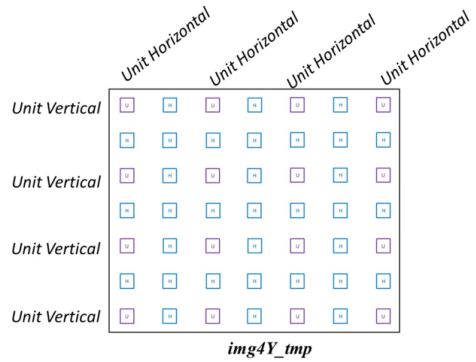
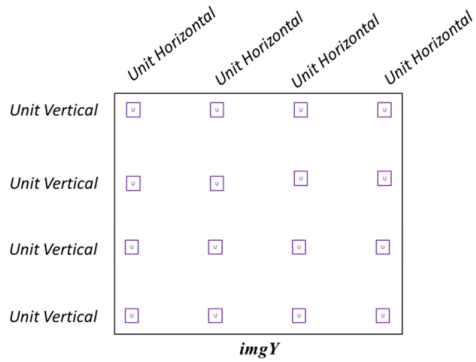
maximum value of 2^n-1 , where n is the number of bits reserved for a pixel value...typically 8.” EX-1001, 9:59-64. The range 0 to 2^8-1 represents a dynamic range 0 to 255. Ramchandran, ¶140.

TML6 thus discloses interpolating, in video coding, an image comprising “rows and columns” of pixels that are “represented by values having a specified dynamic range.” Ramchandran, ¶140.

(ii) 1[pre][b]

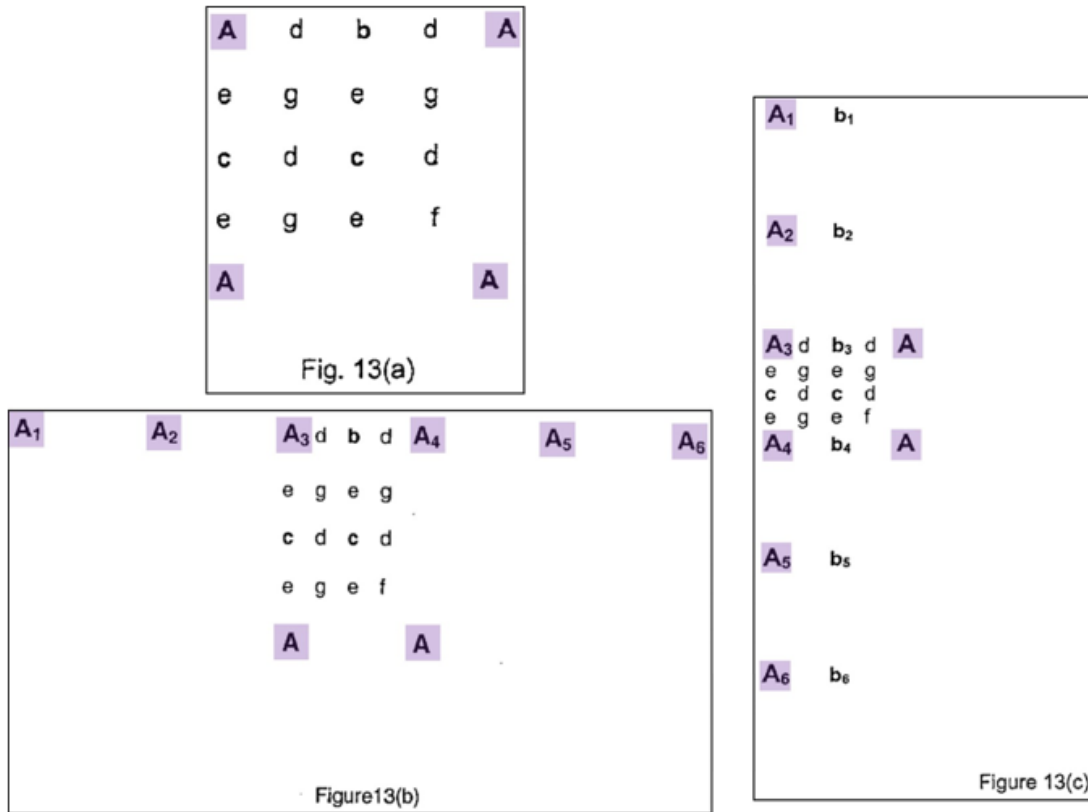
If limiting, TML6 discloses or renders obvious 1[pre][b]. Ramchandran, ¶¶141-142.

As discussed for 1[pre][a], the TML6 data structures *imgY*, *img4Y_tmp*, and *mref*, shown below, include unit pixels (purple) in rows residing at unit horizontal locations and unit pixels in columns residing at unit vertical locations.



TML6, 57-59 (lines 538-632); Ramchandran, ¶¶141-142.

Referring to Figures 13(a)-13(c) (below), the '469 patent confirms in TML6 “pixels A [(purple)] have unit horizontal and unit vertical locations.”



EX-1001, Figures 13a-13c, 8:43-9:23; Ramchandran, ¶143.

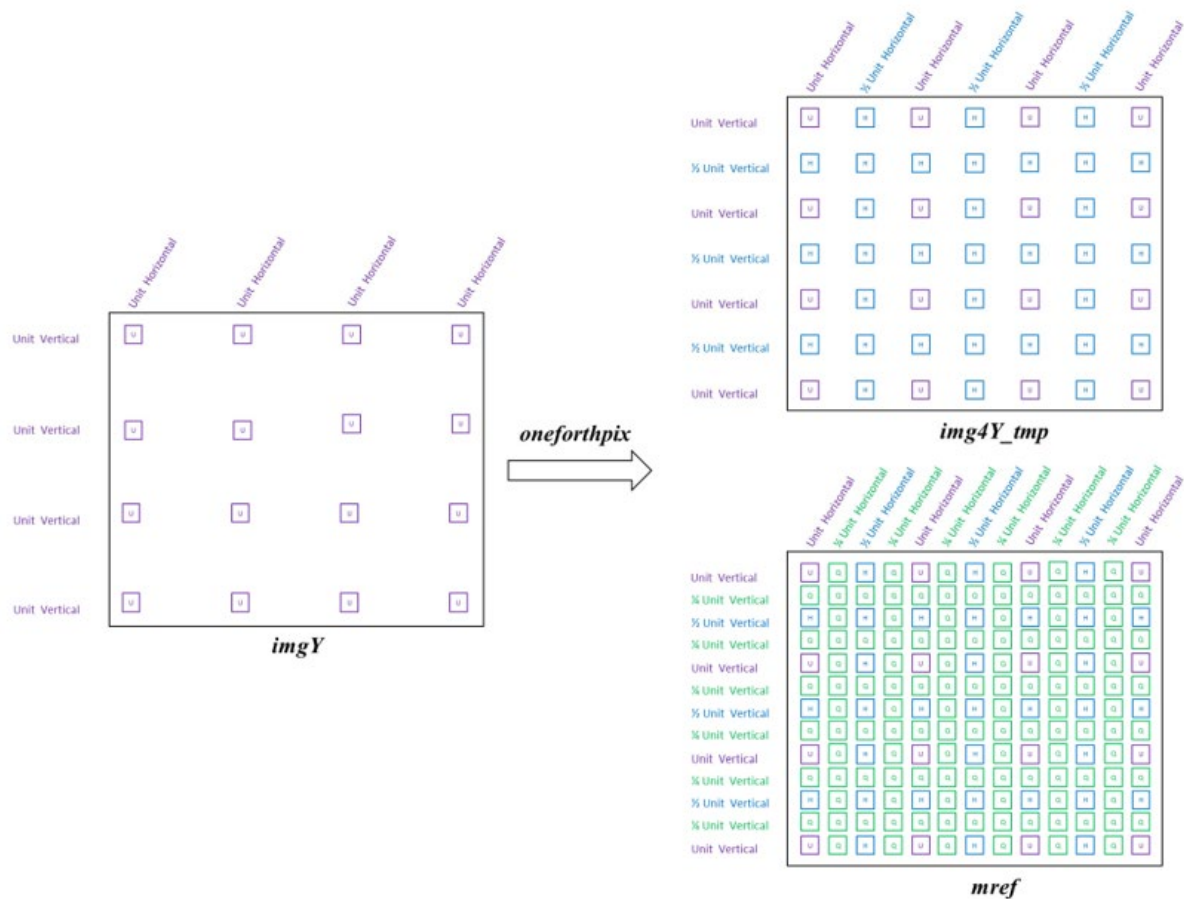
TML6 thus discloses “pixels in the rows residing at unit horizontal locations and...pixels in the columns residing at unit vertical locations,” as recited. Ramchandran, ¶143.

(iii) 1[pre][c]

If limiting, TML6 discloses or renders obvious 1[pre][c]. Ramchandran, ¶¶144-145.

As illustrated below, TML6’s *oneforthpix()* function interpolates the unit pixels (purple) of the image in data structure *imgY* to generate values for sub-pixels

(blue and green) and stores the values in *img4Y_tmp* and *mref*. TML6, 49 (lines 97-109), 57-59 (lines 538-632). As described in further detail for claim elements 1[a]-1[c] *infra*, the *oneforthpix()* function interpolates values for half sub-pixels (blue) and quarter sub-pixels (green) at fractional horizontal and fractional vertical locations based on the unit pixels of the image in *imgY*. *Id.*, 57-59 (lines 538-632). The *oneforthpix()* function then stores the resulting sub-pixel values in *img4Y_tmp* and *mref*. *Id.*



Ramchandran, ¶145. In the data structures *img4Y_tmp* and *mref* shown above, the fractional horizontal and vertical locations include half-unit vertical and half-unit horizontal locations (blue) defined according to $\frac{1}{2}^x$, where $x=1$ ($\frac{1}{2}^1=\frac{1}{2}$). In addition, the quarter-unit vertical and quarter-unit horizontal locations (green) are defined according to $\frac{1}{2}^x$, where $x=2$ ($\frac{1}{2}^2=\frac{1}{4}$). Moreover, TML6 discloses that “x” has a maximum value of $N=2$. Ramchandran, ¶¶144-145, 47-108.

As shown in Figure 13(a) (below), the ’469 patent confirms TML6 interpolates the image to generate values for sub-pixels at half and quarter fractional horizontal and vertical locations (blue and green).

		Unit Horizontal				
		$\frac{1}{4}$ Unit Horizontal				
		$\frac{1}{2}$ Unit Horizontal				
		$\frac{1}{4}$ Unit Horizontal				
		Unit Horizontal				
Unit Vertical	A	d	b	d	A	
$\frac{1}{4}$ Unit Vertical	e	g	e	g		
$\frac{1}{2}$ Unit Vertical	c	d	c	d		
$\frac{1}{4}$ Unit Vertical	e	g	e	f		
Unit Vertical	A				A	

Fig. 13(a)

EX-1001, Figures 13(a)-13(c), 8:43-9:58 (“Sub-pixels b, c, and e shown in [Figure] 13a fall into [the ‘half horizontal location’] category.... ‘[Q]uarter horizontal

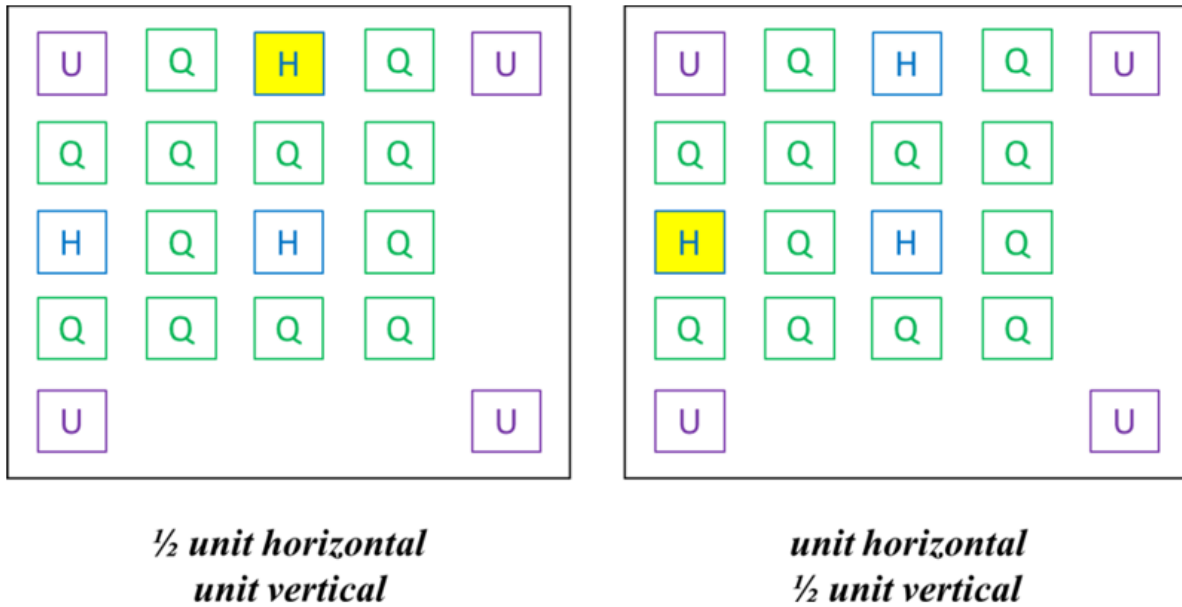
location' refers to any sub-pixel that is constructed in a column which lies at quarter-pixel resolution, such as...sub-pixels d and g in FIG. 13a....”), 11:33-13:29; Ramchandran, ¶146.

TML6 thus “generate[s] values for sub-pixels at fractional horizontal and vertical locations[...defined according to $\frac{1}{2}^x$, where x is a positive integer having a maximum value $N[=2]$,” as recited. Ramchandran, ¶¶145-146.

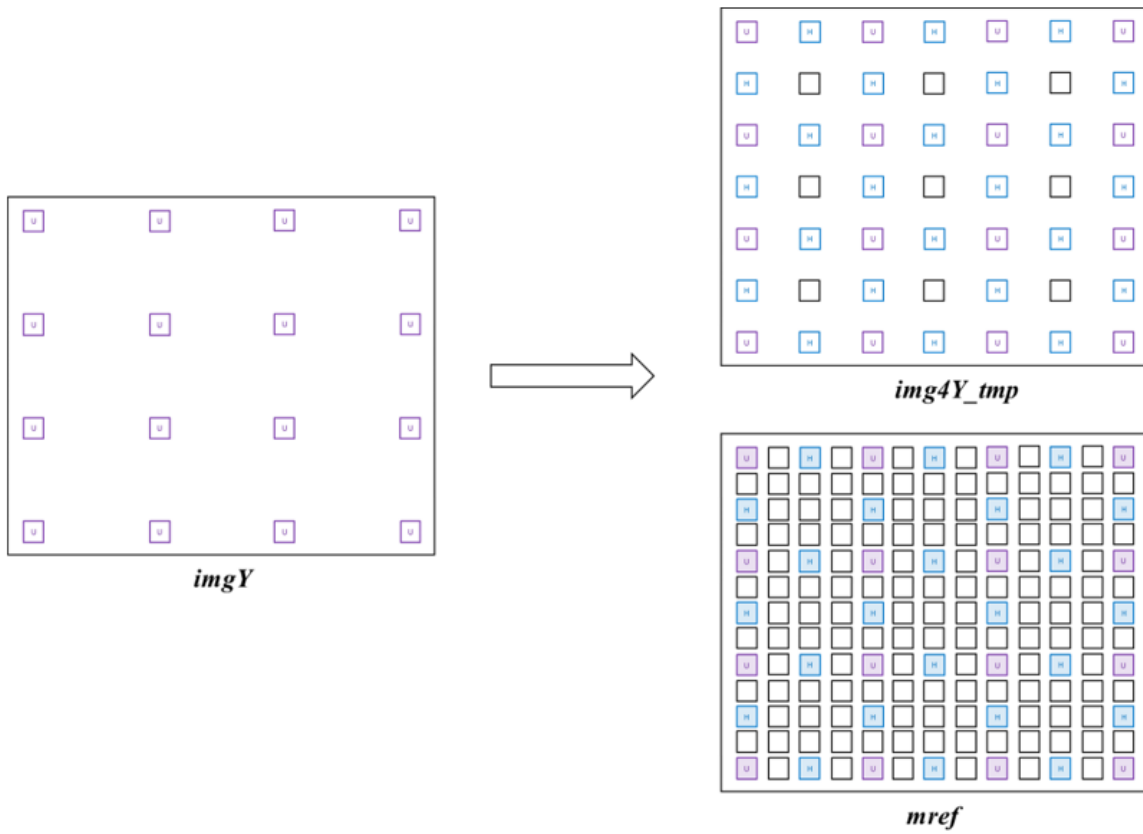
(iv) 1[a]

TML6 discloses or renders obvious 1[a]. Ramchandran, ¶¶147-153.

The TML6 *oneforthpix()* function interpolates values for half sub-pixels (blue) at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations. TML6, 57-58 (lines 538-592). Namely, the half unit ($\frac{1}{2}^{N-1}$; $N=2$) horizontal and unit vertical locations and the unit horizontal and half unit ($\frac{1}{2}^{N-1}$; $N=2$) vertical locations are highlighted in the figures below.

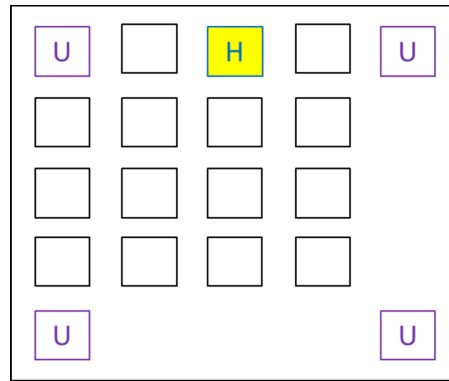


Ramchandran, ¶147. The *oneforthpix()* function interpolates these sub-pixels when they are required for motion estimation. TML6, 57-59 (lines 538-632), 16-24 (lines 734-1180 (“motion_search” function)), 115-116 (lines 769-842 (“Motion vector search”)). As shown below, the *oneforthpix()* function uses the values of the unit pixels (purple) stored in *imgY* to interpolate the half sub-pixels (blue) located at the (1) half unit horizontal and unit vertical locations and (2) unit horizontal and half unit vertical locations and stores the values first in *img4Y_tmp* and then in *mref*.



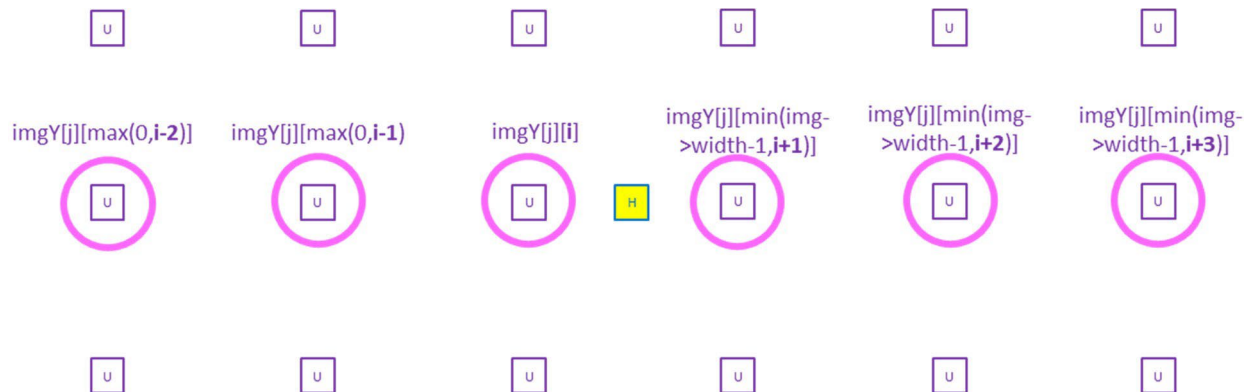
Ramchandran, ¶147 (discussing TML6, 57-58 (lines 538-592)).

More specifically, *oneforthpix()* uses values of the unit pixels (purple) stored in *imgY* to interpolate the half sub-pixels (blue) at the half unit horizontal and unit vertical locations (corresponding to the highlighted sub-pixel below).



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

Ramchandran, ¶148 (discussing TML6, 57-58 (lines 563-575, 588)). These highlighted half sub-pixels (blue) are interpolated in the first step of *onefourthpix()* using a weighted sum of the values of the three adjacent circled unit pixels (purple) to the left and the three adjacent circled unit pixels (purple) to the right of the highlighted sub-pixel.



Ramchandran, ¶148 (discussing TML6, 57-58 (lines 563-575, 588)).

The weights given to the value of each unit pixel (purple) for the weighted sum are stored in the data structure *ONE_FOURTH_TAP*, defined in the file

image.h in the *lencod* folder.⁹ TML6, 57 (lines 568-570), 3 (lines 28-34). A value including the weighted sum is stored in *img4Y_tmp* and then, after being rounded and brought into the dynamic range, in *mref*. *Id.*, 57-58 (lines 572, 588); Ramchandran, ¶149.

Figure 13(b) of the '469 patent (below) confirms TML6's interpolation method for these half sub-pixels. The value for the half sub-pixel “b” (blue) is interpolated “based upon the values of the 6 pixels (A_1 to A_6) [purple] situated in a row at unit horizontal locations and unit vertical locations symmetrically about b, as shown in FIG. 13b, according to the formula $b=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$.”

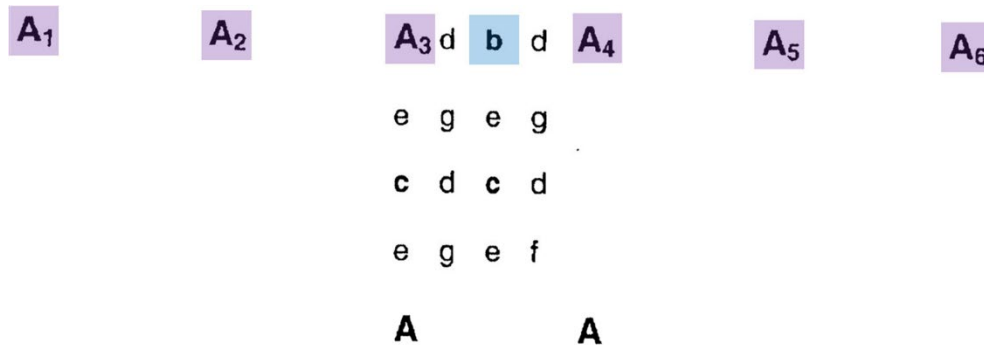
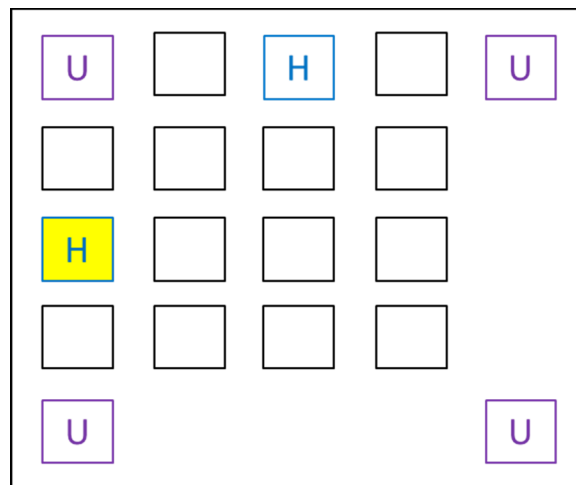


Figure13(b)

⁹ Each weight from *ONE_FOURTH_TAP* in the weighted sum is multiplied by a scaling factor of 32. TML6, 57 (line 572).

EX-1001, Figures 13(a)-13(b), 11:43-53; Ramchandran, ¶150. The '469 patent further confirms TML6's sub-pixel values are interpolated directly. EX-1001, 11:33-39 ("The sub-pixel value interpolation scheme according to TML6[is] referred to herein as direct interpolation[.]...[M]aximum precision is retained throughout. This is achieved by using intermediate values which are neither rounded nor clipped.").

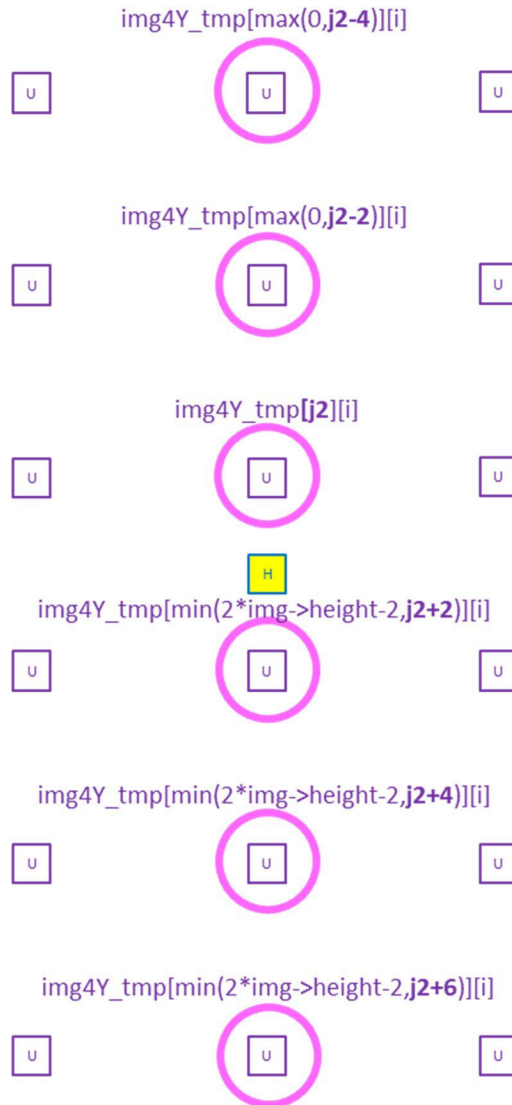
The *oneforthpix()* function also uses values of the unit pixels (purple) stored in *imgY* to interpolate the half sub-pixels (blue; highlighted below) at the unit horizontal and half unit vertical locations.



unit horizontal
½ unit vertical

Ramchandran, ¶151 (discussing TML6, 57-58 (lines 576-592)). As shown below, the *oneforthpix()* function interpolates these highlighted half sub-pixels (blue) using a weighted sum of the values of the three adjacent circled unit pixels (purple) **above**

and the three adjacent circled unit pixels (purple) **below** the highlighted sub-pixel (blue).



Ramchandran, ¶152 (discussing TML6, 57-58 (lines 576-592)).

The weights given to the values of the unit pixels (purple) for the weighted sum are stored in the data structure ***ONE_FOURTH_TAP***, discussed above. TML6, 57-58 (lines 576-592), 3 (lines 28-34); Ramchandran, ¶153. A value including the

weighted sum is stored in *img4Y_tmp* and then in *mref*, after being rounded and brought into the dynamic range. TML6, 57-58 (lines 576-592); Ramchandran, ¶152.

Figure 13(c) of the '469 patent (below) confirms how TML6 interpolates these half sub-pixels. Specifically, the value for the half sub-pixel “c” (blue) is directly 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 $c=(A_1-5A_2+20A_3+20A_4-5A_5+A_6)$.”

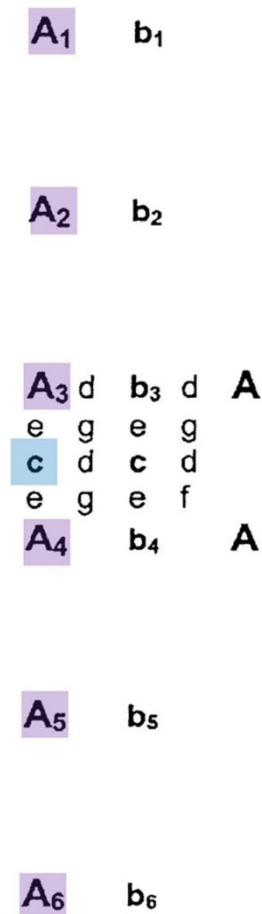


Figure 13(c)

EX-1001, Figures 13(a)-13(c), 11:54-12:5; Ramchandran, ¶153.

TML6 thus discloses “[directly] interpolating [values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and unit vertical locations, and unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations] using weighted sums of pixels residing at unit horizontal and unit vertical locations” when required, as recited. Ramchandran, ¶¶147-153.

(v) 1[b]

The TML6-Fandrianto combination renders obvious 1[b]. Ramchandran, ¶¶154-162.

Regarding TML6, the *onefourthpix()* function in TML6 directly interpolates “values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations” when required for motion estimation, as recited. TML6, 57-59 (lines 538-632), 16-24 (lines 734-1180 (“motion_search” function)), 115-116 (lines 769-842 (“Motion vector search”)). Specifically, the *onefourthpix()* function interpolates sub-pixels at half unit ($\frac{1}{2}^{N-1}$; N=2) horizontal and half unit ($\frac{1}{2}^{N-1}$; N=2) vertical locations using a first weighted sum of the values of the three half sub-pixels immediately above and the three half sub-pixels immediately below.¹⁰ *Id.*, 57-58 (lines 576-592); Ramchandran, ¶¶154-156. A value including the first weighted sum is stored in

¹⁰ As discussed for 1[a], the weights for the first weighted sum are stored in *ONE_FOURTH_TAP*. See TML6, 57 (lines 28-34).

img4Y_tmp and then, after being rounded and brought into the dynamic range, stored in *mref*. TML6, 57-58 (lines 576-592); Ramchandran, ¶¶155-156. The first weighted sum is calculated as described in 1[a] because it uses the same 6-tap filter used in 1[a] and uses the values of sub-pixels from 1[a]. TML6, 57-58 (lines 576-592), 3 (lines 18-24); Ramchandran, ¶157.

Figure 13(c) of the '469 patent (below) confirms TML6 interpolates the value for the half-half sub-pixel “c” (blue) (at half unit horizontal and half unit vertical locations) as a weighted sum of half sub-pixels “b” (green) at half unit horizontal and unit vertical locations.

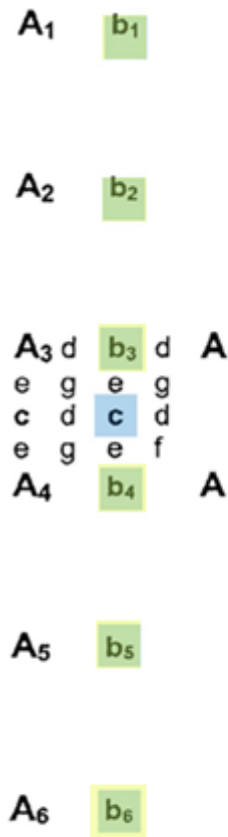


Figure 13(c)

EX-1001, Figure 13(c), 11:54-12:5; Ramchandran, ¶158.

Fandrianto, in turn, discloses interpolating the value for highlighted half-half sub-pixel $C_{9,10}$ when required for motion estimation. Fandrianto, 5:26-32, 6:49-54. This sub-pixel can be interpolated either by (1) averaging the circled half sub-pixels $A_{9,10}$ and $A_{10,10}$ above and below $C_{9,10}$ (vertical interpolation), or (2) averaging the circled half sub-pixels $B_{9,10}$ and $B_{9,11}$ to the right and left of $C_{9,10}$ (horizontal interpolation), illustrated below.

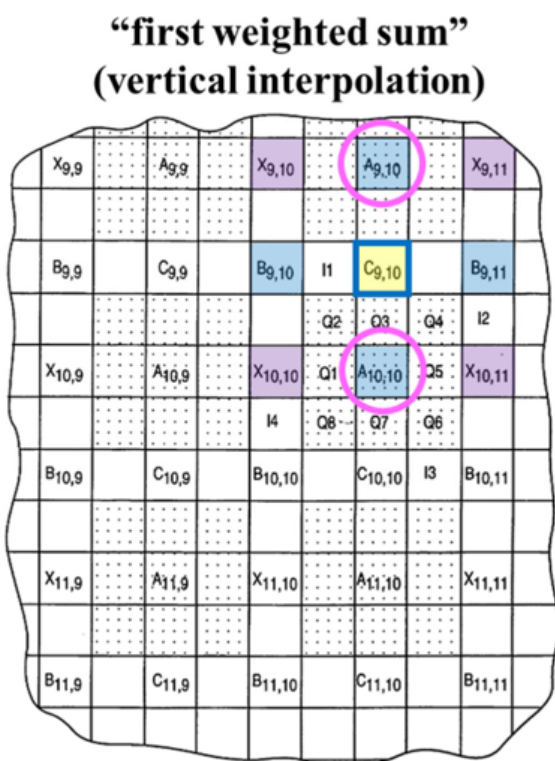


FIG. 5

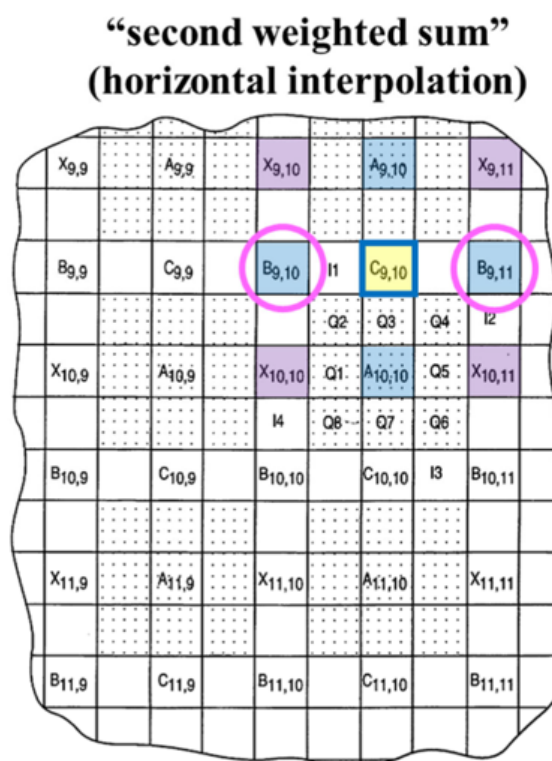


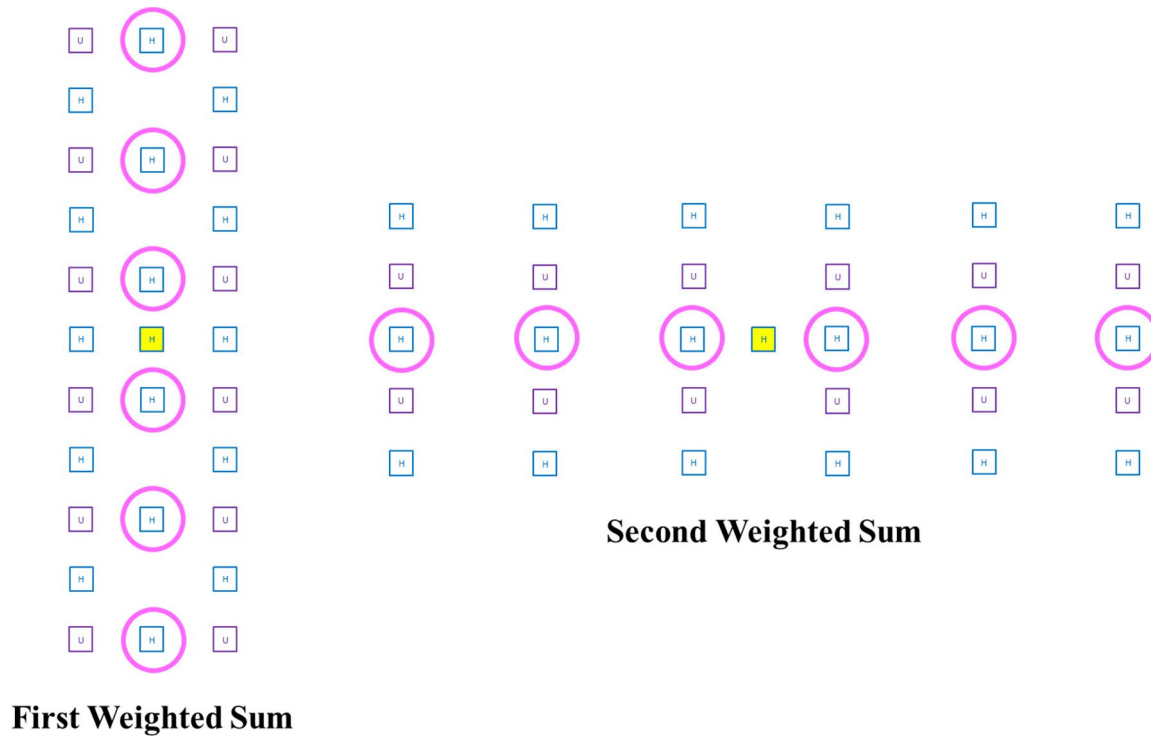
FIG. 5

Id., Figure 5, 6:4-12, 6:49-6:54, 6:18-48, 8:36-10:50. Accordingly, Fandrianto discloses interpolating half-half sub-pixels using either the “first weighted sum” of the “A” sub-pixels at half unit horizontal and unit vertical locations, or the “second weighted sum” of the “B” sub-pixels at unit horizontal and half unit vertical locations, with respective weights in each weighted sum equal to “1.” Ramchandran, ¶159. Furthermore, Fandrianto’s first (vertical) and second (horizontal) weighted sums are calculated “according to step (a)” because, like TML6, they are “interpolat[ed]...directly using weighted sums of pixels residing at unit horizontal and unit vertical locations,” i.e., the “A” and “B” sub-pixels. Fandrianto, 6:49-6:54

(“Horizontal and vertical interpolation are described above in the context of the A block 72 and the B block 74.”); Ramchandran, ¶159.

As discussed *supra* §VIII.A.1, it would have been obvious for a POSITA to modify TML6 based on the teachings of Fandrianto such that the *oneforthpix()* function calculates a first weighted sum of values for sub-pixels residing at half unit horizontal and unit vertical locations (a vertical weighted sum as disclosed in TML6) and a second weighted sum of values for sub-pixels residing at unit horizontal and half unit vertical locations (a horizontal weighted sum as disclosed in Fandrianto), and to interpolate the half-half sub-pixels using a choice of either the first weighted sum or the second weighted sum, as shown below.¹¹ Ramchandran, ¶160.

¹¹ In the TML6-Fandrianto combination, the values for these half-half sub-pixels are stored in the *mref* data structure in the same manner as when the first weighted sum is used. TML6, 57-58 (lines 576-592).



Fandrianto explicitly teaches that a finite number of interpolation techniques can be used to interpolate the “half-half sub-pixels,” including vertical interpolation using a first weighted sum (used in TML6) and horizontal interpolation using a second weighted sum. Fandrianto, 6:4-12, 6:49-7:2. Thus, it would have been obvious to try these finite options, including calculating both first and second weighted sums and using a choice between the two, to interpolate the “half-half sub-pixels” in TML6. Further, revising TML6 to implement this modification would have been straightforward, and a POSITA would have reasonably expected this modification to yield the predictable result of interpolating the “half-half sub-pixels.” In fact, the ’469 patent recognizes that interpolating the “half-half sub-pixel *c*,” shown in Figure

14(a) below, using either a vertical weighted sum of sub-pixels *b* or a horizontal weighted sum of sub-pixels *b* produces the same result.

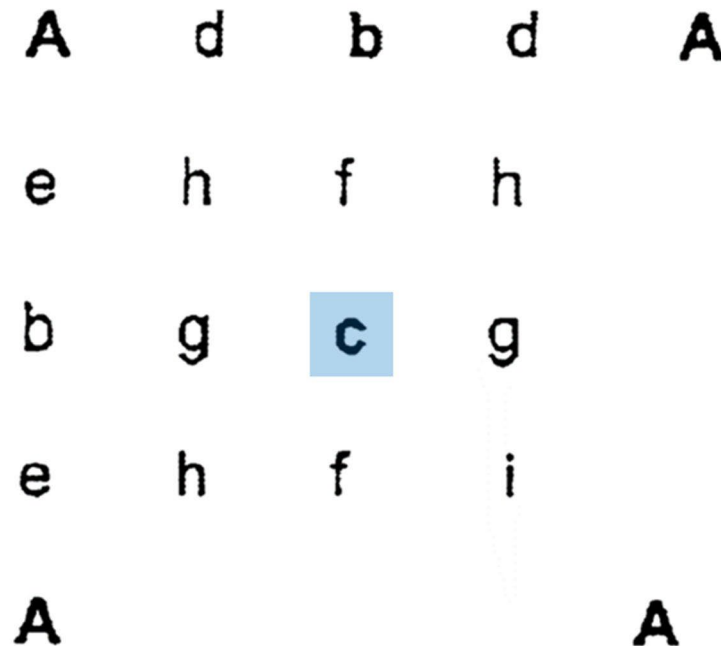


Fig. 14(a)

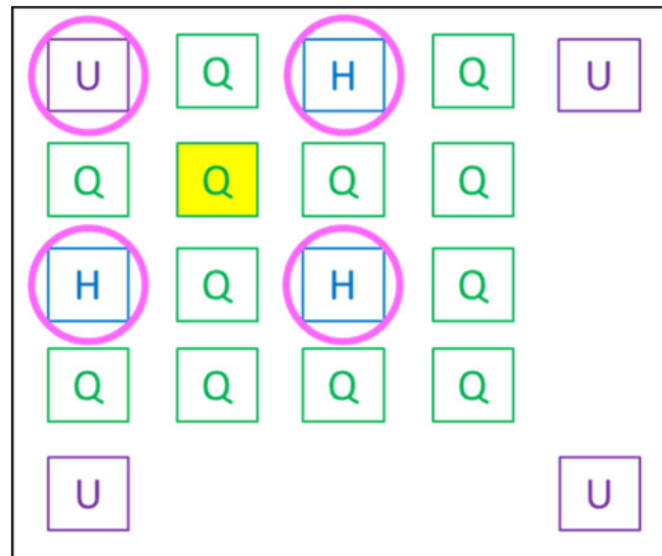
EX-1001, Figure 14(a), 36:5-38 (“It should be noted that the use of intermediate values *b* in the horizontal direction leads to the same result as using intermediate values *b* in the vertical direction.”), 39:16-37 (same); Ramchandran, ¶¶160-161.

TML6 in view of Fandrianto thus renders obvious “using a choice of a first weighted sum...and a second weighted sum,” calculated according to 1[a], when directly interpolating “values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$ unit vertical locations,” as recited. Ramchandran, ¶¶161-162.

(vi) 1[c]

The TML6-Fandrianto combination teaches 1[c]. Ramchandran, ¶¶163-164. First, both TML6 and Fandrianto disclose interpolating a value for a quarter-quarter sub-pixel, i.e., “a value for a sub-pixel situated at a $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location” where $N=2$. Ramchandran, ¶163.

Regarding TML6, when sub-pixels—such as the highlighted quarter-quarter sub-pixel (green) below—are required for motion estimation, *onefourthpix()* interpolates them using a weighted average of the closest unit pixel U (purple and circled) and closest three half sub-pixels H (blue; circled).



TM6, 58-59 (lines 593-632), 16-24 (lines 734-1180 (“*motion_search*” function)), 115-116 (lines 769-842 (“Motion vector search”)); Ramchandran, ¶164.

As shown in Figure 13(a) (below), the ’469 patent confirms TML6 interpolates the value for the quarter-quarter sub-pixel “g” (green) by using a

weighted average of the closest unit pixel A (purple) and closest three half sub-pixels b, c, and c (blue).

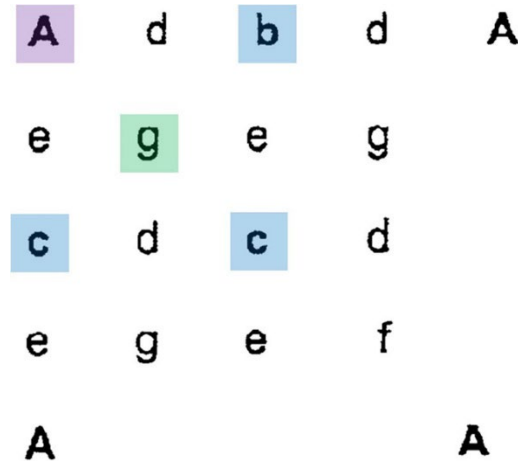


Fig. 13(a)

EX-1001, Figure 13a, 12:45-52; Ramchandran, ¶165.

Turning to Fandrianto, Figure 5 (below) shows four unit pixels (purple within red box) and discloses interpolating values for quarter-quarter sub-pixels Q1-Q8, including the highlighted quarter-quarter sub-pixel Q2 (green), when required for motion estimation. Fandrianto specifically teaches deriving quarter-quarter sub-pixel Q2 “from a **diagonal interpolation** of the data in the A block 72 and the B block 74.” Fandrianto, 10:51-58.¹²

¹² 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

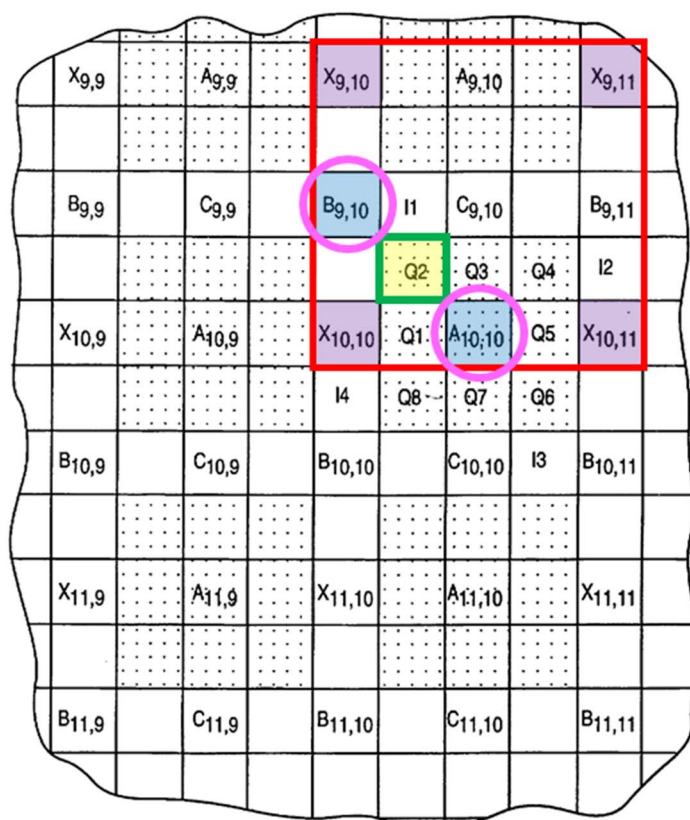


FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:58-9:32, 10:51-58.

In other words, quarter-quarter sub-pixel Q2 (green) is diagonally interpolated using a weighted average of the two diagonally adjacent half sub-pixels A and B (blue and circled). Namely, the sub-pixel Q2 (green) is interpolated by taking a weighted average of the value of the half sub-pixel B (blue) situated at a unit ($\frac{1}{2}^{N-m}$ unit; $N=2, m=2$) horizontal and half unit ($\frac{1}{2}^{N-n}$ unit; $N=2, n=1$) vertical location and

Q1-Q8. Ramchandran, ¶¶168-171 (citing Fandrianto, 7:26-8:20, 10:51-58, Figures 3B, 5).

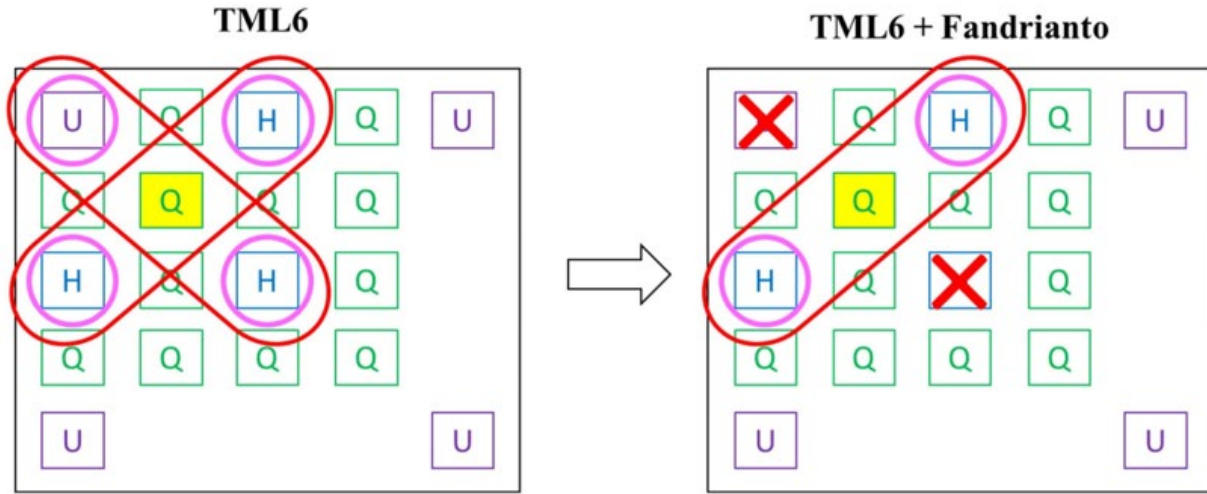
the value of the half sub-pixel A (blue) located at a half unit ($\frac{1}{2}^{N-p}$ unit; $N=2$, $p=1$) horizontal and unit ($\frac{1}{2}^{N-q}$; $N=2$, $q=2$) vertical location, where the weights of the values are “1.” *Id.*, 8:58-9:32; 10:51-58, Ramchandran, ¶¶166-172.

Moreover, as demonstrated above and shown in the table below, the variables m , n , p , and q are integer values in the range 1 to $N=2$:

Variable	Value
m	2
n	1
p	1
q	2

As described *supra* §VIII.A.1, a POSITA would have been motivated to use Fandrianto’s method of interpolating the quarter-quarter sub-pixels Q2 (green) in TML6 by, for example, modifying the code of TML6 to use Fandrianto’s quarter sub-pixel interpolation method or by using Fandrianto’s technique on its hardware. Thus, as shown below, the combined teachings of TML6 and Fandrianto disclose that, when a value for the highlighted quarter-quarter sub-pixel Q (green) is required in TML6, the value would be interpolated by taking a weighted average of the value of a first sub-pixel H (blue and circled (below right)) at a $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and the value of a second sub-pixel H (blue and circled

(below right)) at a $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, as in Fandrianto.



As shown above, the first and second sub-pixels H (blue and circled) are located diagonally adjacent with respect to the highlighted quarter-quarter sub-pixel Q (green), i.e., “the sub-pixel at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical location,” when $N=2$. Ramchandran, ¶¶173-175.

TML6 in view of Fandrianto thus renders obvious interpolating, when needed, sub-pixels at $\frac{1}{2}^N$ unit horizontal and $\frac{1}{2}^N$ unit vertical locations by averaging the values of two diagonally-situated sub-pixels/pixels at (1) $\frac{1}{2}^{N-m}$ unit horizontal and $\frac{1}{2}^{N-n}$ unit vertical location and (2) $\frac{1}{2}^{N-p}$ unit horizontal and $\frac{1}{2}^{N-q}$ unit vertical location, respectively, with m , n , p , and q having the values mapped in the table above (“integer values in the range 1 to $N[=2]$ ”), as recited. Ramchandran, ¶175.

b. Claim 2

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[c] and illustrated below, Fandrianto discloses that the value for the highlighted quarter-quarter sub-pixel Q2 (green) is obtained by a weighted average of the first half sub-pixel A (blue) located at a half unit horizontal and unit vertical location and the second half sub-pixel B (blue) located at a unit horizontal and half unit vertical location, with weights equal to “1.” See Fandrianto, Figures 2, 5, 6:18-7:2, 8:36-10:58; Ramchandran, ¶176.

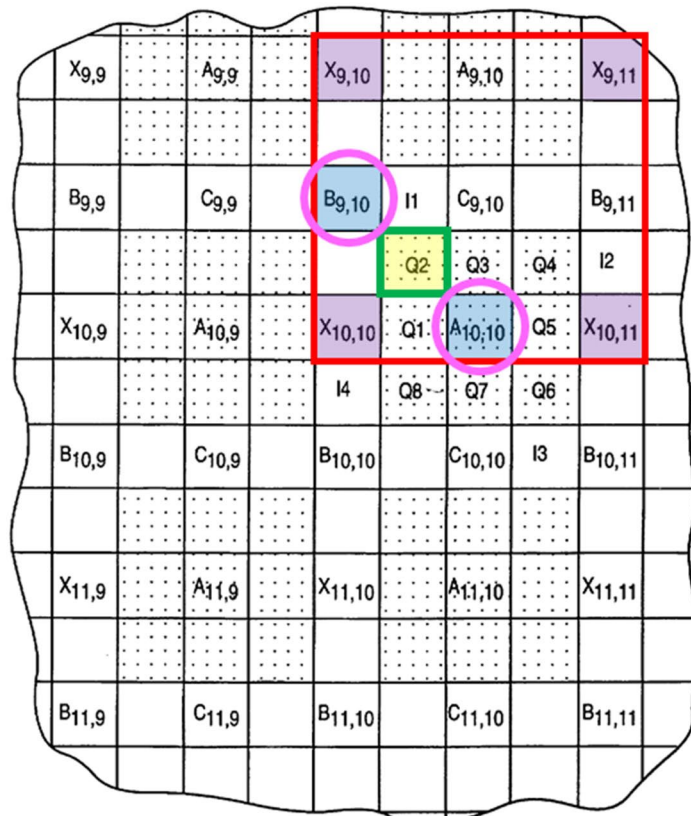


FIG. 5

Fandrianto, Figure 5. Fandrianto's half sub-pixels A and B (blue) are equidistant to the quarter sub-pixel Q2 (green) along the straight-line diagonal (orange).

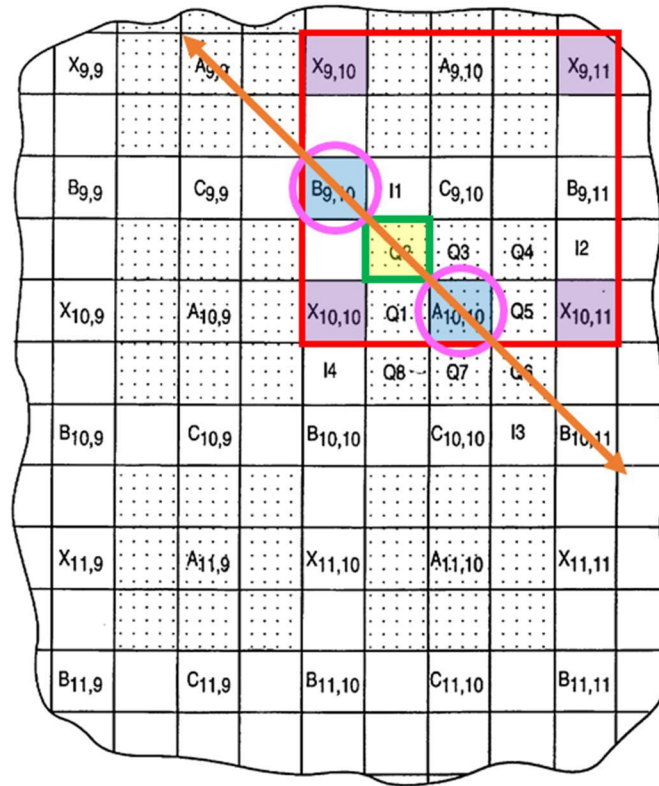


FIG. 5

Thus, the relative magnitudes of the first and second weights (i.e., one) are inversely proportional to the straight-line diagonal (orange) proximity of the first and second sub-pixels A and B (blue) to the sub-pixel Q2 (green) at the quarter ($\frac{1}{2}^N$, $N=2$) unit horizontal and quarter ($\frac{1}{2}^N$, $N=2$) unit vertical location. Ramchandran, ¶177.

c. Claim 3

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for Claim 2 and illustrated below, the first and second half sub-pixels A and B (blue) are symmetrically located with respect to the quarter-

quarter sub-pixel Q2 (green), and the first and second weights (i.e., “1”) have equal values.

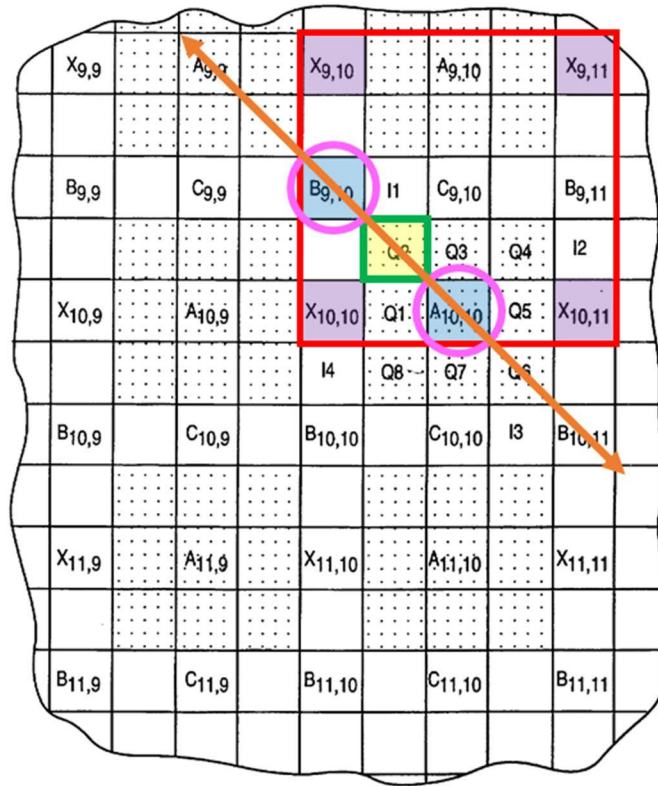
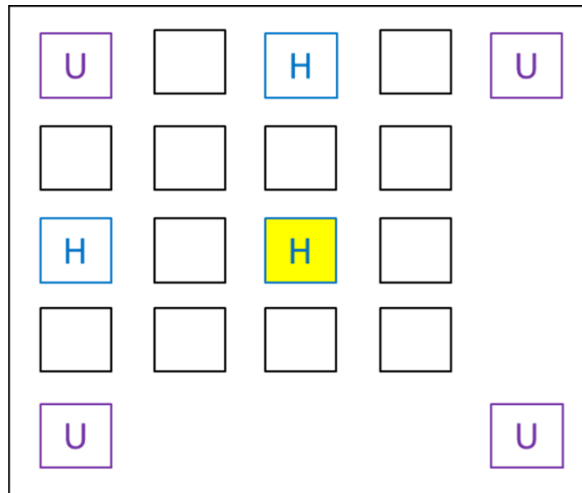


FIG. 5

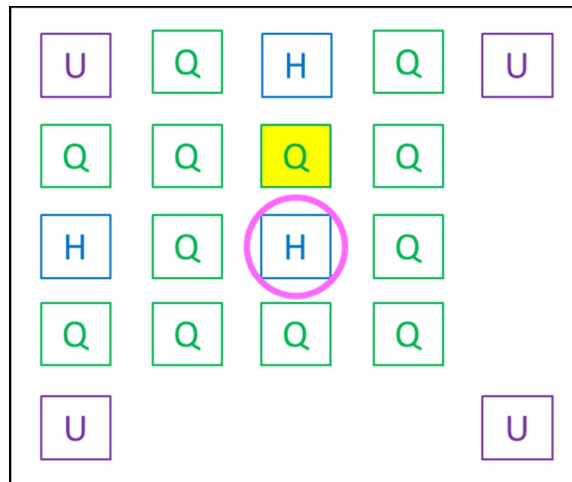
Fandrianto, Figure 5; Ramchandran, ¶178.

d. Claim 4

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[b], the “first weighted sum” calculated in TML6 is used to interpolate the value for the highlighted half-half sub-pixel (blue) below, which is then stored in *mref*.



Ramchandran, ¶179. Claim 4 further requires that this “first weighted sum”—corresponding to the half-half sub-pixel (blue) circled below—is used when a highlighted sub-pixel Q (green) below at $\frac{1}{2}N^{-1}$ ($\frac{1}{2}$; $N=2$) unit horizontal and $\frac{1}{2}N$ ($\frac{1}{4}$; $N=2$) unit vertical location is required (i.e., is interpolated).



Ramchandran, ¶¶179-180. Fandrianto discloses that the value for the half-half sub-pixel “C_{10,10}” (blue and circled below) is used to interpolate the highlighted sub-pixel Q7 (green) below at half unit horizontal and quarter unit vertical location when it is required for motion estimation.

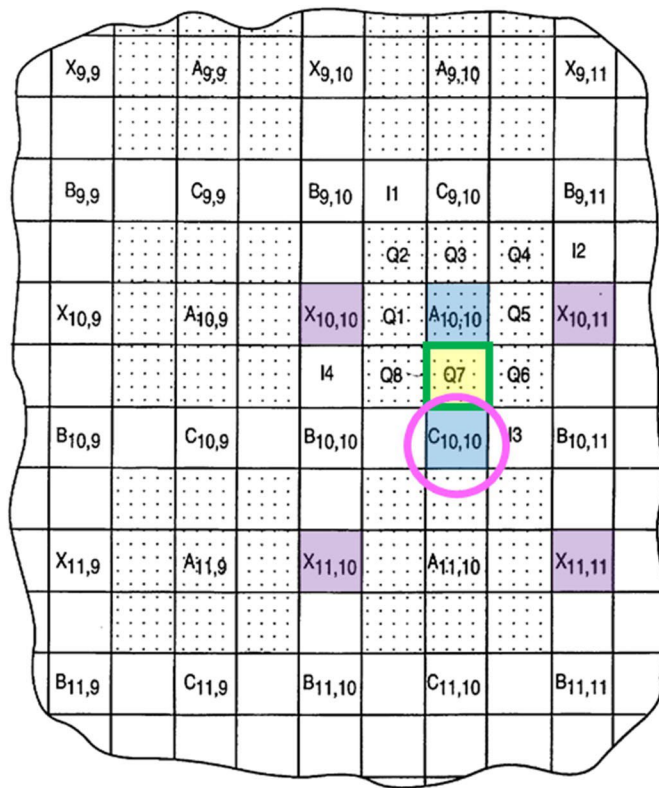


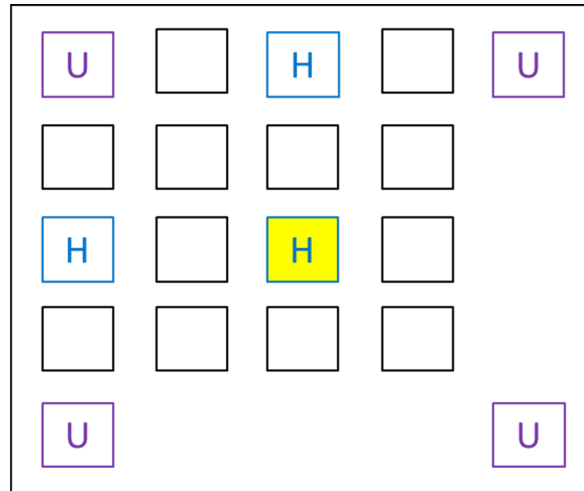
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-35, 10:23-33 (“ $Q7_{r,c} = (C_{r,c} + A_{r,c})/2$ ”); *see also id.*, 8:36-57 (“Q1”), 9:33-43 (“Q3”). Thus, the TML6-Fandrianto combination’s quarter sub-pixel interpolation scheme discussed for 1[c] would use the value for the half-half sub-pixel of TML6 in *mref* (including the first weighted sum) to interpolate the Q7 sub-pixel at $\frac{1}{2}^{N-1}$ ($\frac{1}{2}$; $N=2$) unit horizontal and $\frac{1}{2}^N$ ($\frac{1}{4}$; $N=2$) unit vertical location when it is required. Ramchandran, ¶181.

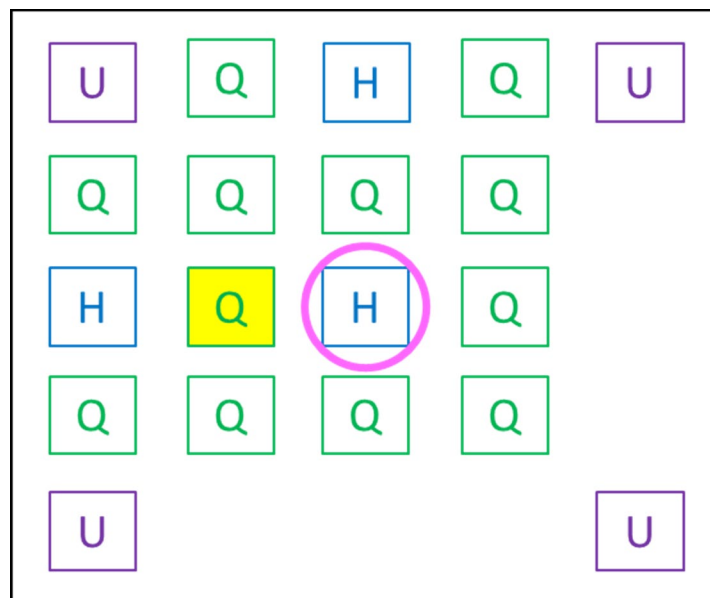
e. Claim 5

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[b], the “second weighted sum” in the TML6-

Fandrianto combination is used to interpolate the value for the highlighted half-half sub-pixel (blue) below, the value of which is stored in *mref*.



Claim 5 further states that this “second weighted sum” (corresponding to the value of the half-half sub-pixel (blue and circled below) is used when a highlighted sub-pixel Q (green) below at $\frac{1}{2}^N$ ($\frac{1}{4}$; $N=2$) unit horizontal and $\frac{1}{2}^{N-1}$ ($\frac{1}{2}$; $N=2$) unit vertical location is required (i.e., is interpolated).



Ramchandran, ¶¶182-183.

As shown below, Fandrianto discloses using the value for the half-half sub-pixel $C_{10,10}$ (blue; circled) to interpolate the highlighted sub-pixel I1 (green) below at $\frac{1}{2}N^{-1}$ ($\frac{1}{2}$; $N=2$) unit horizontal and $\frac{1}{2}N$ ($\frac{1}{4}$; $N=2$) unit vertical location when it is required for motion estimation.

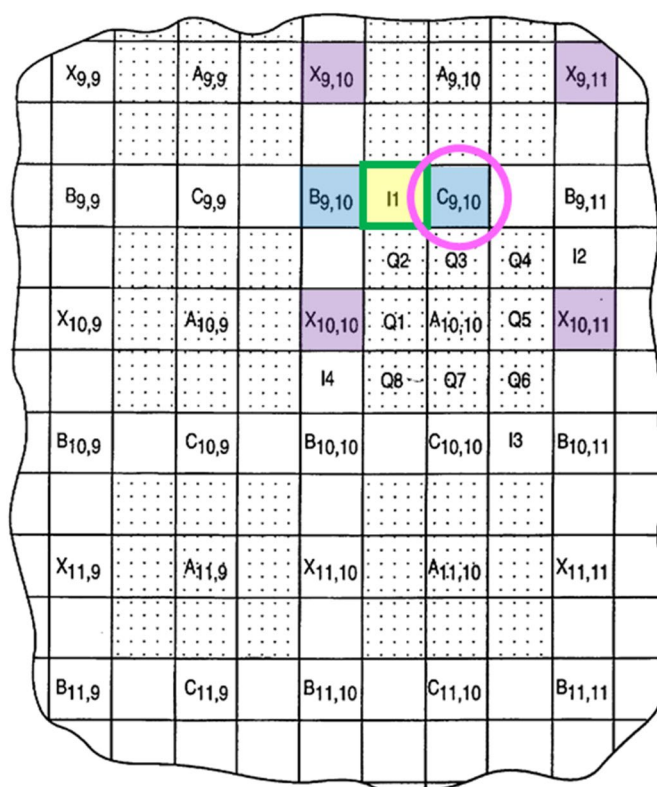


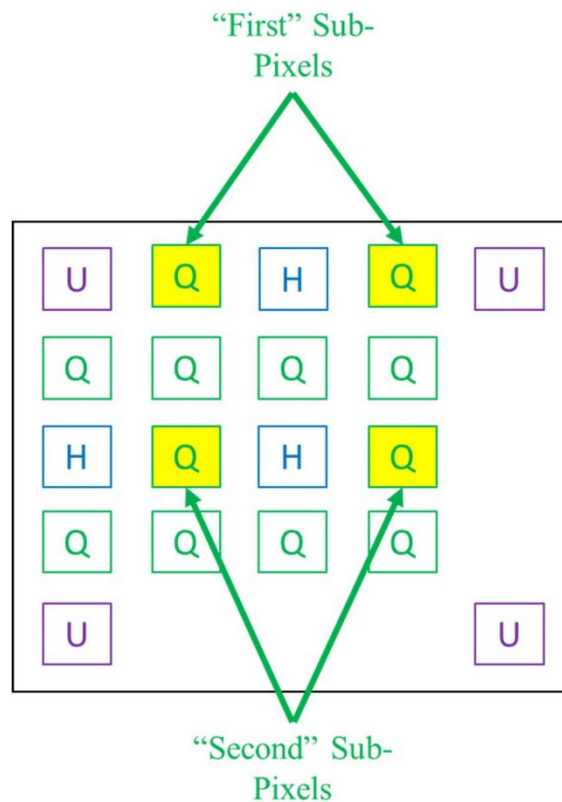
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-35, 8:58-9:32 (“ $I_{r,c}=(B_{r-1,c}+C_{r-1,c})/2$ ”); *see also id.*, 8:36-57 (“Q1”), 10:6-22 (“I3”). Thus, the TML6-Fandrianto combination’s quarter sub-pixel interpolation scheme discussed above regarding Element 1[c] would use the value for the half-half sub-pixel of TML6 in *mref* (including the

second weighted sum) to interpolate the I1 sub-pixel at $\frac{1}{2}^N$ ($\frac{1}{4}$; N=2) unit horizontal and $\frac{1}{2}^{N-1}$ ($\frac{1}{2}$; N=2) unit vertical location when it is required. *Id.*; Ramchandran, ¶184.

f. Claim 6

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As illustrated below, Claim 6 requires interpolating values of two sub-pixels: (1) “[first] sub-pixels at $\frac{1}{2}^N$ [$\frac{1}{4}$; N=2] unit horizontal and unit vertical locations” and (2) “[second] sub-pixels at... $\frac{1}{2}^N$ [$\frac{1}{4}$; N=2] [unit] horizontal and $\frac{1}{2}^{N-1}$ [$\frac{1}{2}$; N=2] [unit] vertical locations.”



Ramchandran, ¶185.

Fandrianto discloses interpolating sub-pixels when they are required for motion estimation. *E.g.*, Fandrianto, 5:26-32. As illustrated in Figure 5 below (left), Fandrianto interpolates the highlighted first sub-pixel “Q1” (green) at quarter unit horizontal and unit vertical location by averaging the values of pixel $X_{10,10}$ (purple and circled) and sub-pixel $A_{10,10}$ (blue and circled). Similarly, as shown below (right), Fandrianto interpolates the highlighted first sub-pixel “Q5” (green) at quarter unit horizontal and unit vertical location by averaging the values of pixel $X_{10,11}$ (purple and circled) and sub-pixel $A_{10,10}$ (blue and circled).

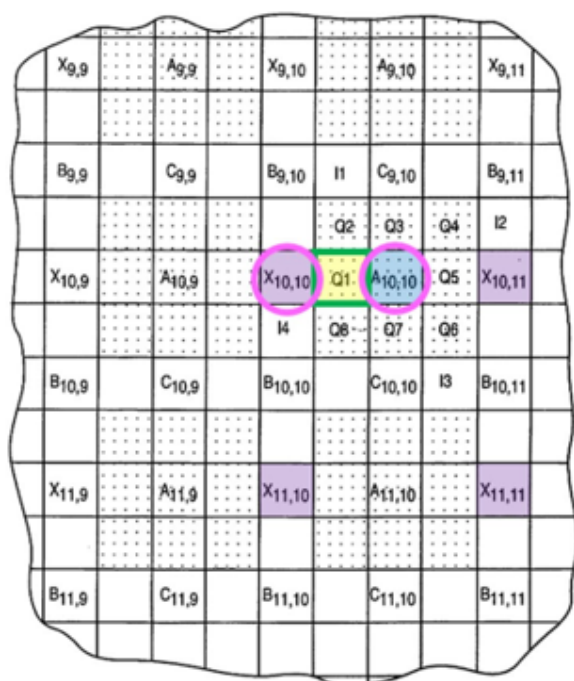


FIG. 5

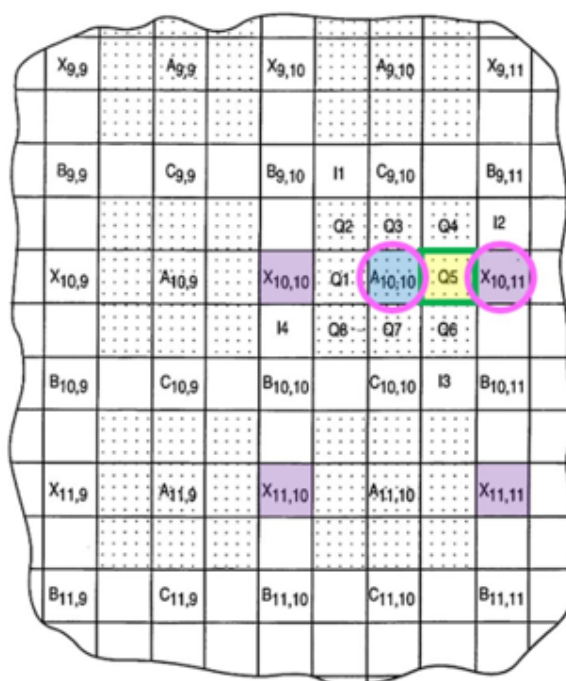


FIG. 5

Id., Figure 5, 5:26-32, 8:29-57 (“ $Q1_{r,c}=(X_{r,c}+A_{r,c})/2$ ”), 9:63-10:5 (“ $Q5_{r,c}=(X_{r,c}+1+A_{r,c})/2$ ”). The circled pixels (purple) are located at a vertical

location corresponding to that of the highlighted sub-pixels (green) being calculated and at a unit horizontal location, and the circled half sub-pixels (blue) are located at a vertical location corresponding to that of the highlighted sub-pixels (green) being calculated and at a half unit ($\frac{1}{2}N^{-1}$; $N=2$) horizontal location. Ramchandran, ¶185.

Furthermore, as shown below (left), Fandrianto interpolates the highlighted second sub-pixel “I1” (green) at quarter unit horizontal and half unit vertical location by averaging the values of sub-pixel $B_{9,10}$ (blue and circled) and sub-pixel $C_{9,10}$ (blue and circled). Similarly, as shown below (right), Fandrianto interpolates the highlighted second sub-pixel “I3” (green) at quarter unit horizontal and half unit vertical location by averaging the values of a sub-pixel $B_{10,11}$ (blue and circled) and a sub-pixel $C_{10,10}$ (blue and circled).

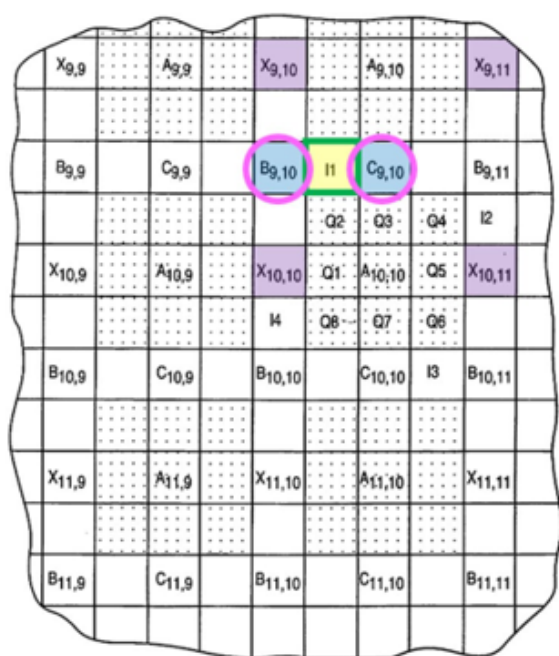


FIG. 5

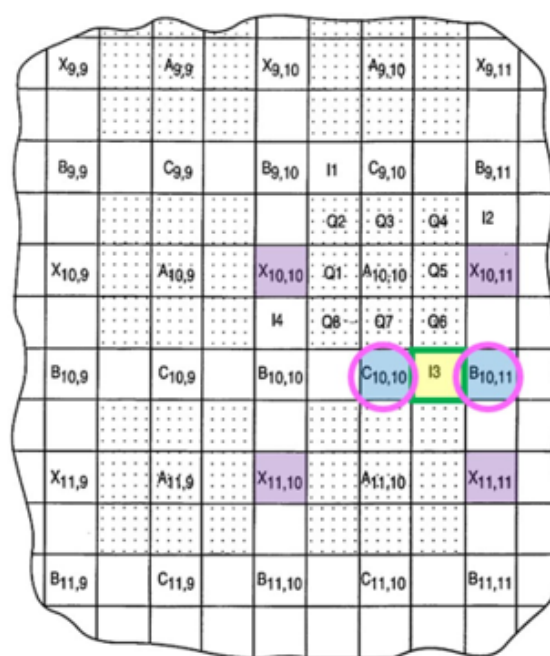


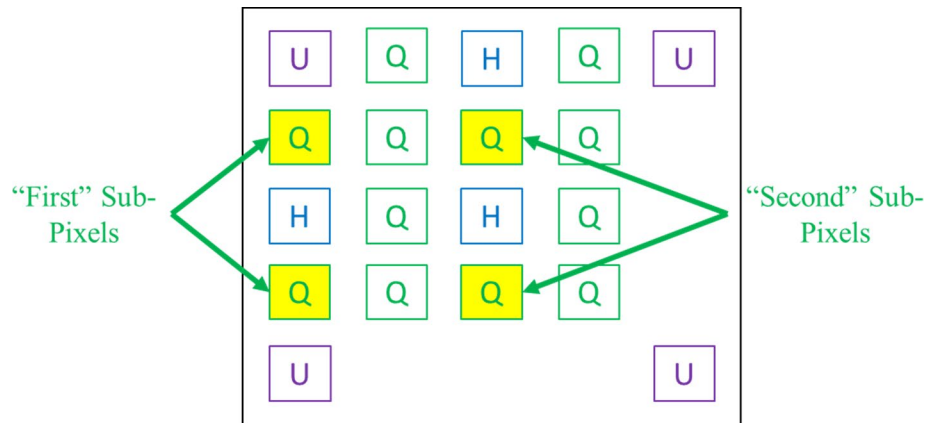
FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 8:58-9:32 (“ $I1_{r,c}=(B_{r-1,c}+C_{r-1,c})/2$ ”), 10:6-22 (“ $I3_{r,c}=(B_{r,c+1}+C_{r,c})/2$ ”). The circled sub-pixels (blue) are located at a vertical location corresponding to that of the highlighted sub-pixels (green) being calculated and at a unit horizontal location or at a half unit ($\frac{1}{2}^{N-1}$; $N=2$) horizontal location. *Id.*; Ramchandran, ¶186.

As discussed *supra* §VIII.A.1 and for 1[c], it would have been obvious to substitute Fandrianto’s quarter sub-pixel interpolation scheme for that of TML6’s. Thus, per the combination, TML6 would interpolate the first sub-pixels and the second sub-pixels required by Claim 6 using values for the pixels and sub-pixels from the *mref* data structure of TML6. *See also* Claim 4; Ramchandran, ¶187.

g. Claim 7

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. Claim 7 requires interpolating values of two sub-pixels: (1) “[first] sub-pixels at unit horizontal and $\frac{1}{2}^N$ [$\frac{1}{4}$; $N=2$] unit vertical locations”; and (2) “[second] sub-pixels at... $\frac{1}{2}^{N-1}$ [$\frac{1}{2}$; $N=2$] unit horizontal and $\frac{1}{2}^N$ [$\frac{1}{4}$; $N=2$] unit vertical locations.”



Ramchandran, ¶188. As required by Claim 7 and shown below on the left, Fandrianto interpolates the highlighted first sub-pixel “I4” (green) at unit horizontal and quarter unit vertical location by averaging the values of a pixel $X_{10,10}$ (purple and circled) and a sub-pixel $B_{10,10}$ (blue and circled) when it is required for motion estimation. Similarly, as shown below on the right, Fandrianto interpolates the highlighted first sub-pixel “I2” (green) at unit horizontal and quarter unit vertical location by averaging the values of a pixel $X_{10,11}$ (purple and circled) and a sub-pixel $B_{9,11}$ (blue and circled) when it is required for motion estimation.

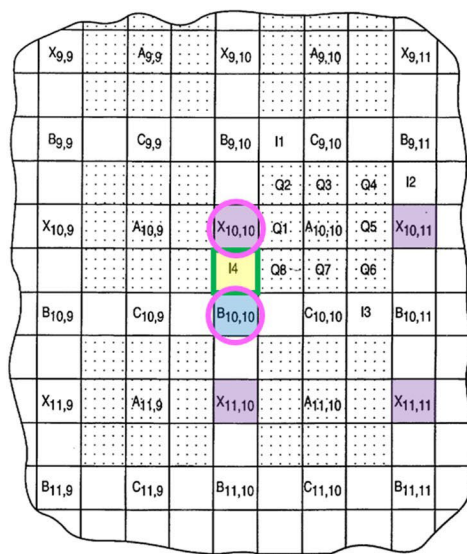


FIG. 5

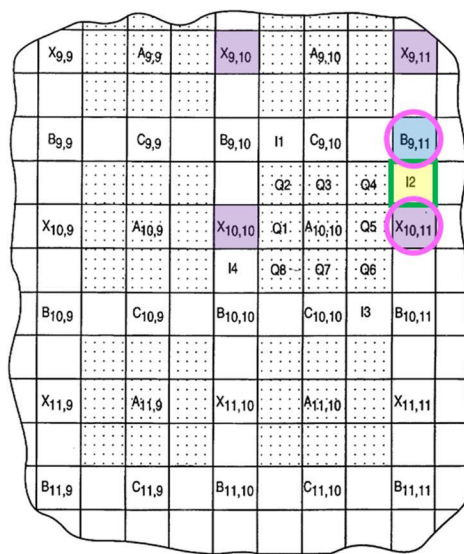


FIG. 5

Fandrianto, Figure 5, 5:26-32, 8:29-57, 10:34-10:50 (“ $I_{4,r,c}=(B_{r,c}+X_{r,c})/2$ ”), 9:45-62 (“ $I_{2,r,c}=(B_{r-1,c}+1+X_{r,c+1})/2$ ”). The circled pixels (purple) are located at a horizontal location corresponding to that of the highlighted sub-pixels (green) being calculated and at a unit vertical location. The circled half sub-pixels (blue) are located at a horizontal location corresponding to that of the highlighted sub-pixels (green) being calculated and at a $\frac{1}{2}N^{-1}$ ($\frac{1}{2}$; $N=2$) unit vertical location. *Id.*; Ramchandran, ¶188.

As required by Claim 7 and shown below on the left, Fandrianto interpolates the highlighted second sub-pixels “Q7” (green) at half unit horizontal and quarter unit vertical locations by averaging the values of a pixel A10,10 (blue and circled) and a sub-pixel C10,10 (blue and circled) when it is required for motion estimation. Similarly, as shown below on the right, Fandrianto interpolates the highlighted second sub-pixel “Q3” (green) at half unit horizontal and quarter unit vertical

location by averaging the values of a pixel $A_{10,10}$ (blue and circled) and a sub-pixel $C_{9,10}$ (blue and circled) when it is required for motion estimation.

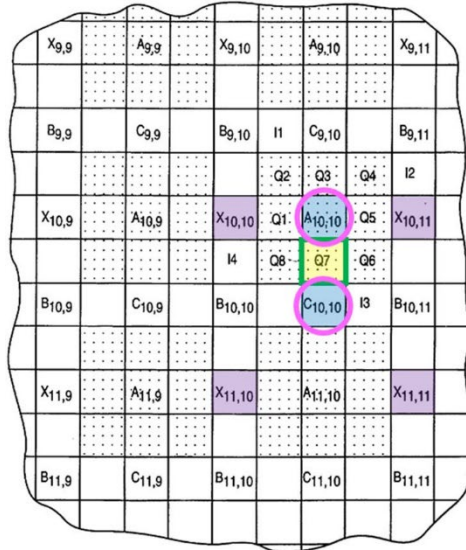


FIG. 5

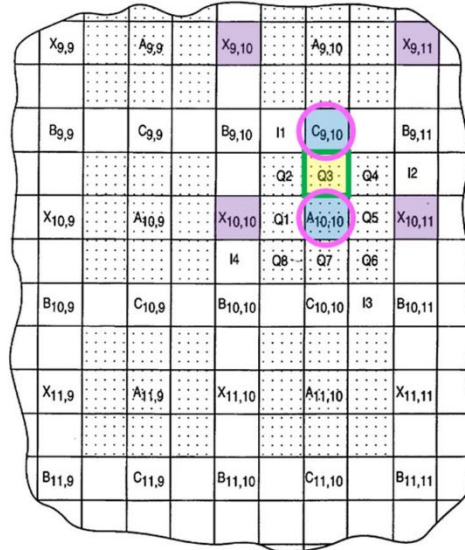


FIG. 5

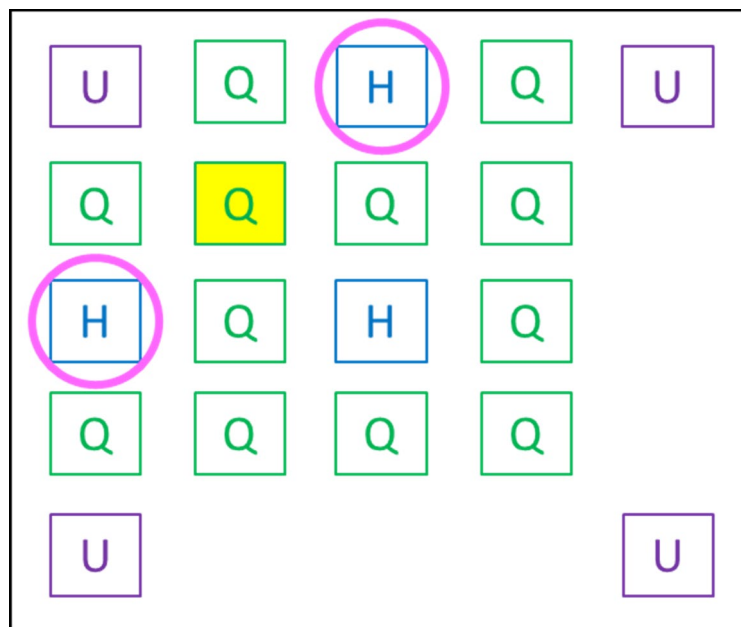
Fandrianto, Figure 5, 5:26-32, 8:29-57, 10:23-33 (“ $Q7_{r,c}=(C_{r,c}+A_{r,c})/2$ ”), 9:32-44 (“ $Q3_{r,c}=(C_{r-1,c+1}+A_{r,c+1})/2$ ”). The circled sub-pixels (blue) are located at a horizontal location corresponding to that of the highlighted sub-pixels (green) being calculated and at a unit vertical location or at a $\frac{1}{2}N^{-1}$ ($\frac{1}{2}$; $N=2$) unit vertical location. *Id.*; Ramchandran, ¶189.

As discussed for 1[c], it would have been obvious to substitute Fandrianto’s quarter sub-pixel interpolation scheme for that of TML6. Thus, in the combination, TML6 would interpolate the first sub-pixels and the second sub-pixels required by Claim 7 using values for the pixels and sub-pixels from the *mref* data structure of TML6. *See also* Claim 4; Ramchandran, ¶190.

h. Claim 9

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[c] and shown below, the combination discloses that the highlighted sub-pixel (green) at quarter ($\frac{1}{2}^N$; $N=2$) unit horizontal and quarter ($\frac{1}{2}^N$; $N=2$) unit vertical locations are interpolated by averaging values of a circled sub-pixel (blue) at a half unit ($\frac{1}{2}^{N-1}$; $N=2$) horizontal and unit vertical location and a circled sub-pixel (blue) located at a unit horizontal and half unit ($\frac{1}{2}^{N-1}$; $N=2$) vertical location. Ramchandran, ¶191.

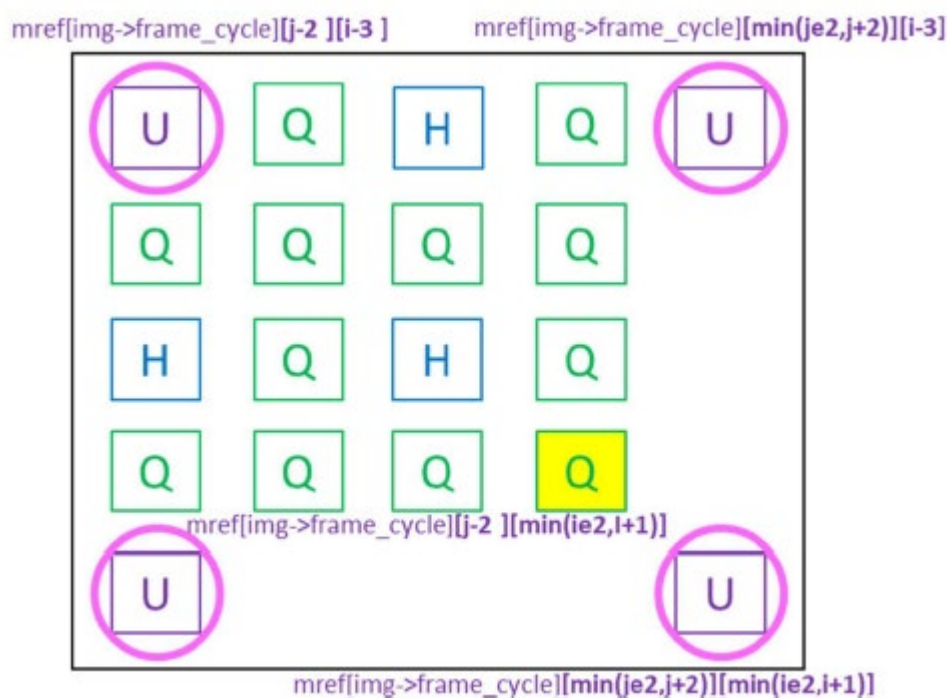
TML6 + Fandrianto



i. Claim 14

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. TML6 discloses interpolating values for some of the sub-pixels at quarter unit ($\frac{1}{2}^N$; $N=2$) horizontal and quarter unit ($\frac{1}{2}^N$; $N=2$) vertical locations,

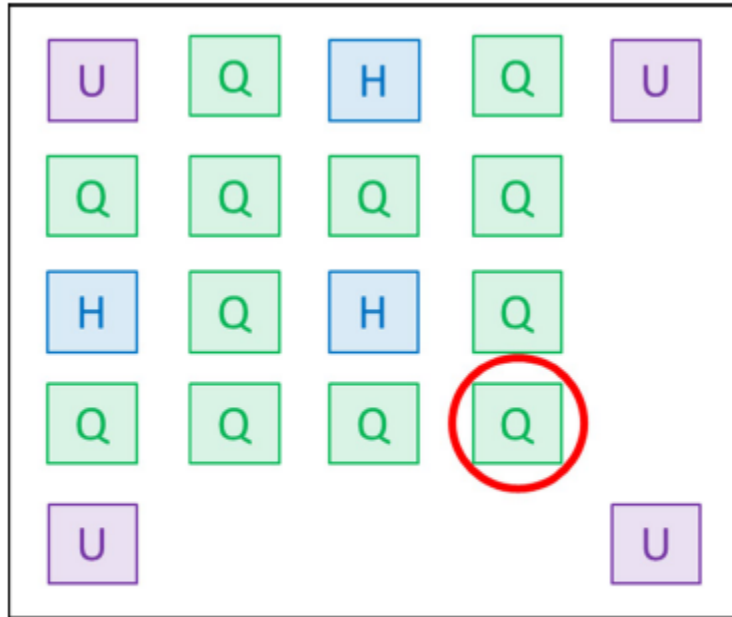
when they are required for motion compensation, by taking the average of the nearest neighbouring unit pixels. Namely, the third step of the *oneforthpix()* function interpolates the highlighted quarter sub-pixels (green) (located at the quarter unit horizontal and quarter unit vertical locations in the bottom right corner of the block of pixels being processed) by taking the average of the four closest unit pixels (purple and circled).



TML6, 57-58 (lines 593-625); *see also id.*, 16-24 (lines 734-1180 (“motion_search” function)), 115-116 (lines 769-842 (“Motion vector search”)); Ramchandran, ¶192.

The values of these four unit pixels (purple) used to calculate the averages for the highlighted quarter sub-pixels (green) are retrieved from the data structure *mref*.

TML6, 58 (lines 615-618). The averages are normalized, rounded to a range of 0-255, and stored back in *mref*, as shown and circled below.



mref

Id., 58 (lines 618-621); Ramchandran, ¶193.

The '469 patent, confirms TML6 interpolates the sub-pixels “f” (green) (at quarter unit horizontal and quarter unit vertical locations) by averaging the nearest neighboring unit-unit pixels “A” (purple).

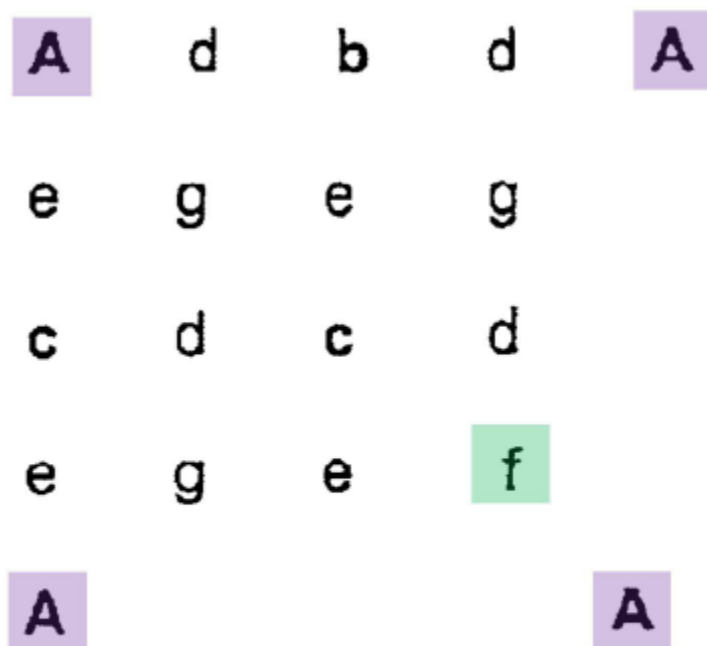


Fig. 13(a)

EX-1001, Figure 13(a), 12:53-58 “[t]he value for $\frac{1}{4}$ resolution sub-pixel f is interpolated by averaging the values of the 4 closest pixels at unit horizontal and vertical locations, according to $f=(A_1+A_2+A_3+A_4+2)/4$, where the locations of pixels A_1 , A_2 , A_3 , and A_4 are the four nearest original pixels”); Ramchandran, ¶194.

j. Claim 15

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[pre][c], $N=2$ in the combined TML6-Fandrianto system. Ramchandran, ¶195.

k. Claim 16

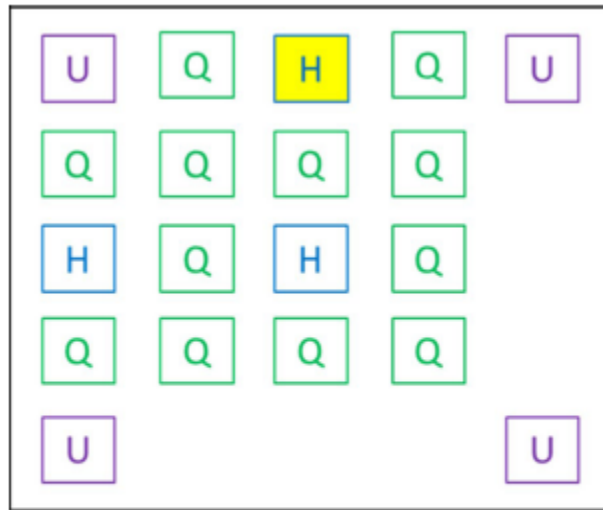
The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed above for 1[a], TML6's *oneforthpix()* function interpolates sub-pixels at (1) half unit horizontal and unit vertical locations and (2) unit horizontal and half unit vertical locations to create intermediate values "*is*." As discussed for 1[b], this function interpolates sub-pixels at half unit horizontal and half unit vertical locations to create intermediate values. Values representing these intermediate values are stored in the *img4Y_tmp* data structure. TML6, 57-58 (lines 538-592); Ramchandran, ¶196.

These intermediate values "*is*" may fall outside the dynamic range of 0-255, and thus, they are normalized, clipped, and rounded so that they are integer values within this range before they are stored in the *mref* data structure. TML6, 57-58 (538-592); Ramchandran, ¶197.

l. Claim 17

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[a] and shown below, TML6's *oneforthpix()* function interpolates the highlighted half sub-pixels (blue) at half unit ($1/2^{N-1}$; $N=2$) horizontal and unit vertical locations to create intermediate values "*is*." Values

representing these intermediate values “*is*” are stored in the *img4Y_tmp* data structure.¹³



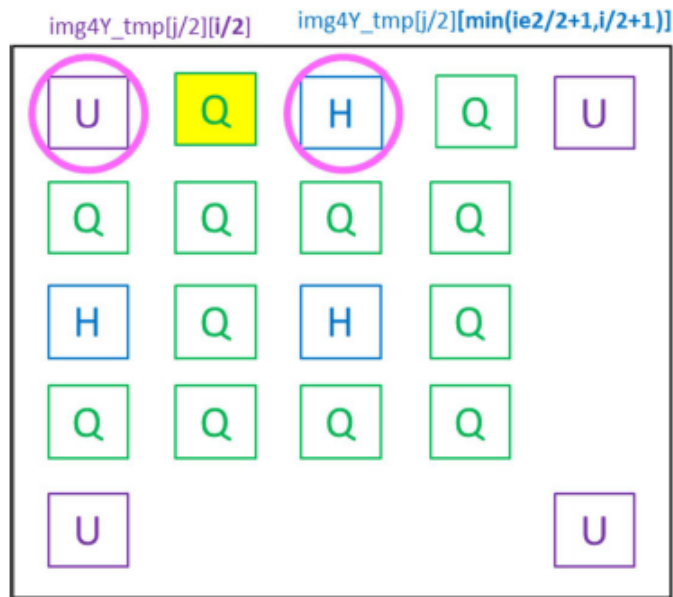
*½ unit horizontal
unit vertical*

TML6, 57 (lines 563-575); Ramchandran, ¶198.

TML6 further discloses interpolating values for quarter ($1/2^N$; $N=2$) resolution sub-pixels using these intermediate values. For example, values for the highlighted quarter resolution sub-pixels (green) at quarter unit horizontal and unit vertical locations are interpolated using values for the nearest unit pixel (purple and circled)

¹³ The “*is*” value is multiplied by 32 when stored in *img4Y_tmp*, creating another intermediate value. TML6, 57 (lines 563-575).

and the nearest half ($1/2^{N-1}$; $N=2$) resolution sub-pixel (blue and circled) at a half unit horizontal and unit vertical location, as shown below.



TML6, 58 (lines 593-625). As shown above, the values for the circled half resolution sub-pixels (blue) at half unit horizontal and unit vertical locations are the intermediate values stored in the *img4Y_tmp*. *Id.*, 57 (lines 563-625). Thus, the intermediate value for a sub-pixel having half ($1/2^{N-1}$, $N=2$) sub-pixel resolution is “used [in] the calculation of” a sub-pixel value having quarter ($1/2^N$, $N=2$) sub-pixel resolution. Ramchandran, ¶199.

The '469 patent confirms TML6 uses the intermediate values of the half sub-pixels situated at half unit horizontal and unit vertical locations to interpolate values for the quarter sub-pixels at quarter unit horizontal and unit vertical locations. For example, as shown below, the '469 patent explains intermediate values for half sub-

pixel “b” (located at half unit horizontal and unit vertical locations) and unit pixel “A” (located at unit horizontal and unit vertical locations) are used in the calculation of the value for the quarter resolution sub-pixel value “d” (located at quarter unit horizontal and unit vertical locations).

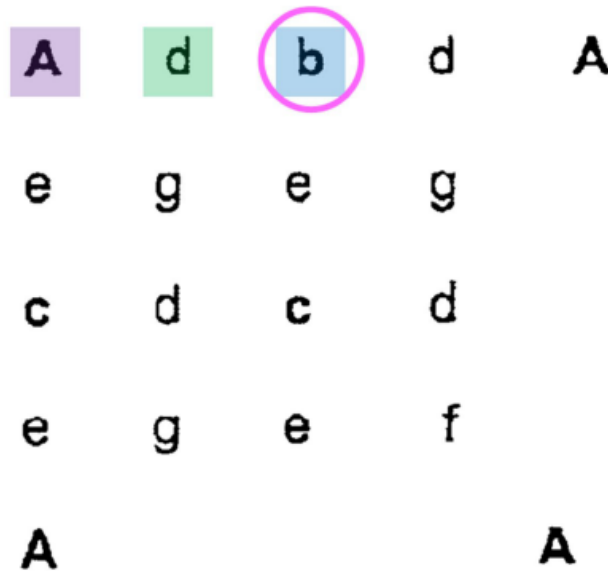


Fig. 13(a)

EX-1001, Figure 13(a), 12:6-24 (“Values for $\frac{1}{4}$ resolution sub-pixels d [green] located at quarter horizontal and unit vertical locations, are calculated from the value of the immediately neighbouring pixel [purple] at unit horizontal and unit vertical location (pixel A) and the intermediate value b [blue] calculated in step (1) for the immediately neighbouring $\frac{1}{2}$ resolution sub-pixel at half horizontal and unit vertical

location ($\frac{1}{2}$ resolution sub-pixel b), according to $d=(32A+b+32)/64$.”), 11:46-50; Ramchandran, ¶200.

m. Claim 20

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for Claim 4, the TML6-Fandrianto combination discloses that the half sub-pixels used in the first weighted sum—determined in 1[b]—are half sub-pixels residing at half unit ($\frac{1}{2}^{N-1}$; $N=2$) horizontal and unit vertical locations, and that this weighted sum is used to interpolate a value for the half-half sub-pixel. As further explained for Claim 4, the TML6-Fandrianto combination discloses that these half-half sub-pixel values (i.e., the first weighted sum) are used to interpolate values for the quarter sub-pixels located at $\frac{1}{2}^{N-1}$ ($\frac{1}{2}$; $N=2$) unit horizontal and $\frac{1}{2}^N$ ($\frac{1}{4}$; $N=2$) unit vertical location. Ramchandran, ¶201.

n. Claim 21

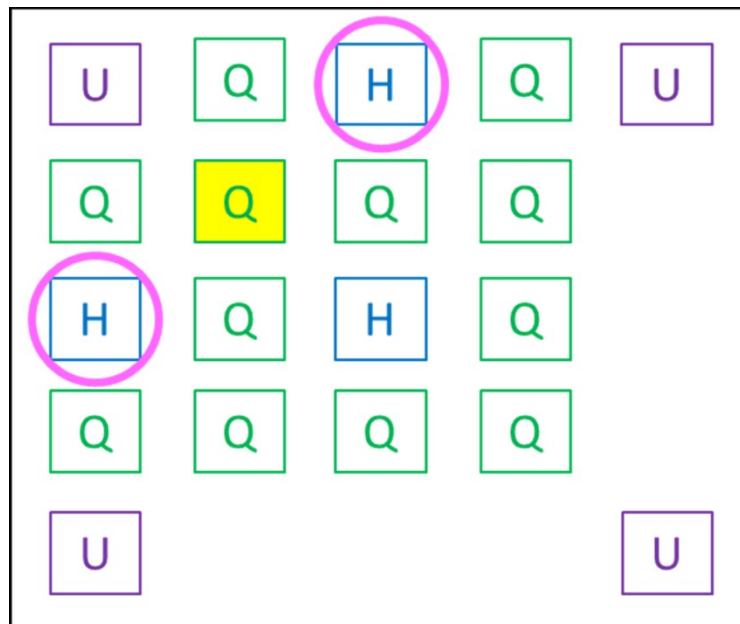
The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for Claim 5, the TML6-Fandrianto combination discloses that the half sub-pixels used in the second weighted sum—determined in 1[b]—are half sub-pixels residing at unit horizontal and half unit ($\frac{1}{2}^{N-1}$; $N=2$) vertical locations, and that this weighted sum is used to interpolate a value for the half-half sub-pixel. As further explained for Claim 5, the TML6-Fandrianto combination discloses that these half-half sub-pixel values (i.e., the second weighted sum) are

used to interpolate values for the quarter sub-pixels located at $\frac{1}{2}^N$ ($\frac{1}{4}$; $N=2$) unit horizontal and $\frac{1}{2}^{N-1}$ ($\frac{1}{2}$; $N=2$) unit vertical location. Ramchandran, ¶202.

o. Claim 22

The TML6-Fandrianto combination renders obvious the additional limitations of this claim. As discussed for 1[c], Claim 9, and shown below, when values for sub-pixels (green) at quarter unit ($\frac{1}{2}^N$; $N=2$) horizontal and quarter unit ($\frac{1}{2}^N$; $N=2$) vertical locations are required, they are interpolated by taking the average of a first pair of values of (1) a circled sub-pixel (blue) located at a half unit ($\frac{1}{2}^{N-1}$; $N=2$) horizontal and unit vertical location and (2) a circled sub-pixel (blue) located at a unit horizontal and half unit ($\frac{1}{2}^{N-1}$; $N=2$) vertical location.

TML6 + Fandrianto



Ramchandran, ¶203. If this claim requires both a first pair of values and a second pair of values to perform the interpolation of the highlighted sub-pixel, TML6 discloses using both pairs—i.e., four values—as discussed *supra* §VIII.A.1 and for 1[c]. Ramchandran, ¶203.

3. Independent Claim 18

Claim 18 is substantially the same as Claim 1 but is directed to the specific case where $N=2$. Ramchandran, ¶204. As explained for 1[pre][c], in the combined TML6-Fandrianto system, $N=2$. Accordingly, the combined system teaches this claim. If 18[c] requires both a first and second pair of values, TML6 discloses using both pairs (i.e., four values) as discussed *supra* §VIII.A.1 and for 1[c]. Ramchandran, ¶¶204-205.

4. Independent Claim 19

a. Elements 19[pre][a]-19[a]

19[pre][a]-19[a] are identical to 1[pre][a]-1[a], respectively, and thus the TML6-Fandrianto combination renders them obvious for the same reasons. Ramchandran, ¶¶206-224.

b. Element 19[b]

The TML6-Fandrianto combination renders obvious 19[b]. Ramchandran, ¶¶225-232. As discussed for 1[b], the TML6-Fandrianto combination renders obvious directly interpolating values for sub-pixels at $\frac{1}{2}^{N-1}$ unit horizontal and $\frac{1}{2}^{N-1}$

unit vertical locations using a choice of a first weighted sum and a second weighted sum.

The first weighted sum is a sum of sub-pixel values “located at a **vertical** location corresponding to that of the sub-pixel being calculated,” as recited, because it is **vertically interpolated** using sub-pixels situated vertically (i.e., above and below) with respect to the sub-pixel being calculated, as illustrated below with reference to the ’469 patent’s Figure 13(c) discussion of TML6 (left) and Fandrianto (right).

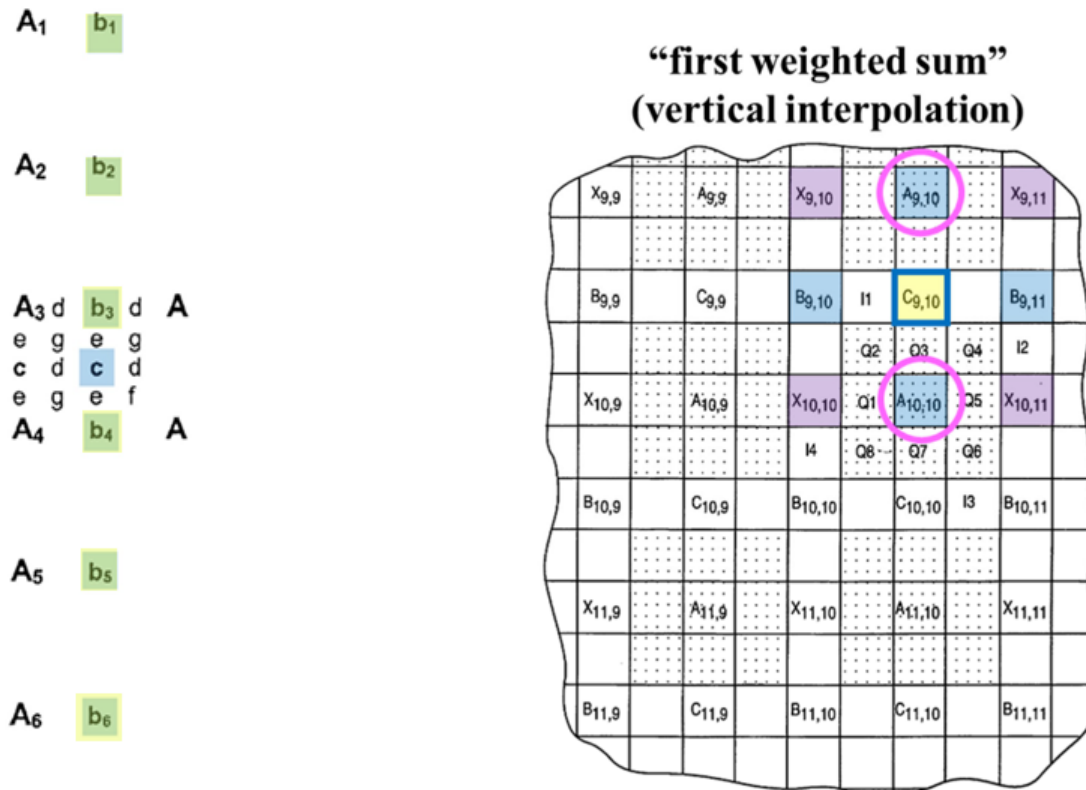
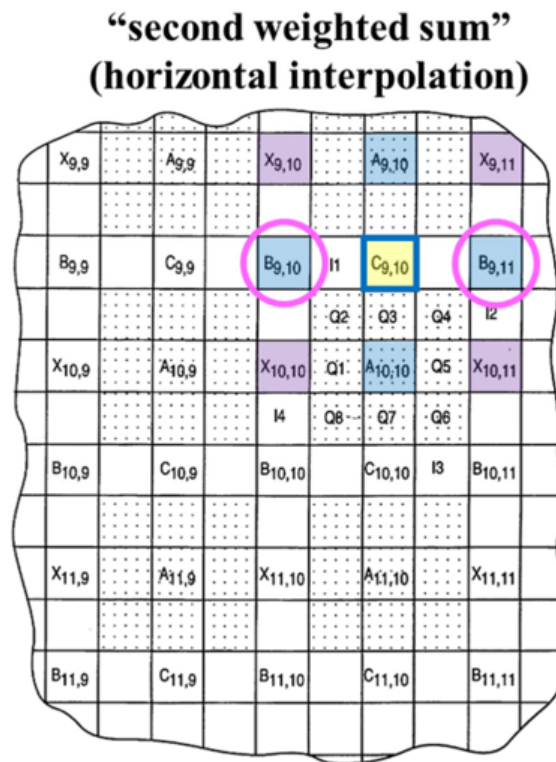


Figure 13(c)

FIG. 5

EX-1001, Figure 13(c), 11:54-12:5; Fandrianto, Figure 5, 6:4-12, 6:18-6:54, 8:36-10:50; Ramchandran, ¶229.

Likewise, the second weighted sum is a sum of sub-pixel values “located at a **horizontal** location corresponding to that of the sub-pixel being calculated,” as recited, because it is **horizontally interpolated** using sub-pixels situated horizontally (i.e., left and right) with respect to the sub-pixel being calculated, as illustrated below with reference to Fandrianto.



Fandrianto, Figure 5, 6:4-12, 6:18-6:54, 8:36-10:50; Ramchandran, ¶229-231.

The TML6-Fandrianto combination thus discloses 19[b] for the same reasons it discloses 1[b]. Ramchandran, ¶232.

5. Claims 23-26

a. Independent Claim 23

Claim 23 is substantively the same as Claim 1, except it is written in terms of “[a] **video coder...comprising an interpolator...adapted to**” perform the method steps of Claim 1. Ramchandran, ¶233.

If “video coder” is limiting, TML6 and Fandrianto disclose it. TML6 encodes and decodes video data. *See* TML6, 155 (line 5), 69-70 (lines 3, 86), 1 (lines 3, 18). TML6 was prepared by VCEG of ITU-T, which develops video coding standards for implementation in hardware or software coders and in services such as videoconferencing and video broadcast. Ramchandran, ¶234; Wenger, ¶¶15, 19-21; VCEG-L45, 1. Fandrianto also discloses a “video coder.” Fandrianto, 1:16-67, Figure 2. A POSITA would have thus understood that TML6 and Fandrianto teach a “video coder.” Ramchandran, ¶¶233-237, 105-110.

If “interpolator” is limiting, TML6 and Fandrianto disclose it. TML6 discloses a sub-pixel interpolator. TML6, 48 (line 3) (“Code one image/slice”). As discussed for 1[pre][a], TML6 includes a function *oneforthpix()* that interpolates sub-pixels for a frame of video data. Fandrianto, in turn, describes arithmetic logic unit (ALU) 154 within “illustrative programmable architecture 100” that receives

pixel data and performs pixel averaging arithmetic to generate intermediate pixel values during half-pixel and quarter-pixel interpolation processes. Fandrianto, 3:55-4:6, 4:38-42, 4:56-64, 6:26-32, 6:61-65, 9:4-32, Figure 2. These interpolation operations involve averaging pixel groups retrieved from the DPCM memory 130 or a combination of DP memory 124 and DPCM memory 130 and storing the resulting interpolated values in a designated memory block, demonstrating the functionality of an interpolator. Ramchandran, ¶234. A POSITA would have thus understood that TML6 and Fandrianto both disclose “[a] video coder...comprising an interpolator.” Ramchandran, ¶234.

b. Claim 24

The TML6-Fandrianto combination renders obvious a “video encoder” for the same reasons discussed for Claim 23. Ramchandran, ¶¶233-235.

c. Claim 25

The TML6-Fandrianto combination renders obvious a “video decoder” for the same reasons discussed for Claim 23. Ramchandran, ¶¶233-236, 105-110.

d. Claim 26

TML6 and Fandrianto teach a “codec” comprising a video encoder or decoder. Ramchandran, ¶¶233-237. A codec is a “hardware- or software-based video compression/decompression adapter[.]” Riley, 9, 12; *see also* CRC-Dictionary, 28. As discussed for Claim 23, TML6 and Fandrianto both encode/decode (i.e.,

compress/decompress) video data, and thus disclose a video codec. Ramchandran, ¶237.

6. Claims 27-30

a. Independent Claim 27

Claim 27 is substantively the same as Claim 23, except it is written in terms of “[a] **communications terminal** comprising a video coder...comprising an interpolator...adapted to” perform the steps of Claim 1. Ramchandran, ¶238.

The ’469 patent describes a “communications terminal” as “comprising a user interface,...processor[, and]...transmitting[/receiving] block[s]” in addition to “a video coder.” EX-1001, 18:30-34. If “communications terminal” is limiting, TML6 and Fandrianto teach it. A POSITA would have understood that TML6 is designed for a video coder in a communications terminal. Ramchandran, ¶239. VCEG developed TML6 for use in applications that receive packetized data over IP networks, which a POSITA would understand requires communications terminals. VCEG-L45, 34-36. Furthermore, Fandrianto’s technique is designed for implementation by a processor in digital television broadcast and video telephony applications, which require communications terminals. Fandrianto, 1:12-41, 3:55-4:17, 11:32-14:40, 20:38-51, Figure 2; Ramchandran, ¶239.

b. Claim 28

As discussed for Claim 27, the TML6-Fandrianto combination renders obvious a communications terminal comprising a video encoder. *See* VCEG-L45, 34-36; Fandrianto, 1:12-41; Ramchandran, ¶240.

c. Claim 29

As discussed for Claim 27, the TML6-Fandrianto combination renders obvious a communications terminal comprising a video decoder. *See* VCEG-L45, 34-36; Fandrianto, 1:12-41; Ramchandran, ¶240.

d. Claim 30

As discussed for Claims 26 and 27, the TML6-Fandrianto combination renders obvious a communications terminal having a video codec comprising a video encoder/decoder, as recited in Claim 30. *See* VCEG-L45, 34-36; Fandrianto, 1:12-41; Ramchandran, ¶240.

7. Claims 31-33

a. Independent Claim 31

Claim 31 is substantively the same as Claim 27, except it is written in terms of “[a] **telecommunications system** comprising a communications terminal and a network, the **telecommunications network** and the communications terminal being **connected by a communications link** over which coded video can be transmitted,

the communications terminal comprising a video coder...comprising an interpolator...adapted to” perform the steps of Claim 1. Ramchandran, ¶241.

TML6 and Fandrianto teach a “communications terminal” for the reasons discussed for Claim 27. If a “telecommunications system,” “telecommunications network,” and a “communications link” connecting the telecommunications network and the communications terminal are limiting, it was well-known to implement video coders in hardware or software in terminals (e.g., wired/wireless devices) within a telecommunications system comprising a network of terminals that transmit coded video over communications links. *See* VCEG-L45, 34-36; Nakaya, 11:64-12:47, 13:47-50, Figure 14; Riley, 7-14. A POSITA would have thus understood that the TML6-Fandrianto combination discloses these limitations for the reasons discussed for Claims 23 and 27. Ramchandran, ¶242.

b. Claim 32

The TML6-Fandrianto combination renders Claim 32 obvious. As discussed for Claim 31, this combination uses a video coder in a telecommunications system comprising a communications terminal and network. TML6 and Fandrianto further teach that the connection between the mobile communications terminal and the wireless network is formed by a radio link. As discussed for Claim 31, both TML6 and Fandrianto teach video telephony systems. A POSITA would have understood that, in contrast to cable television, a television broadcast is transmitted in a wireless

network over the airwaves (i.e., via a radio link) from a transmitter to receiver (i.e., a communications terminal), and that video telephony would require a communications link over the telephony network. Ramchandran, ¶243.

c. Claim 33

In the combined TML6-Fandrianto system, the network enables the communications terminal to communicate with other communications terminals over communications links between the other communications terminals and the network for reasons similar to the reasons discussed above for Claims 31-32. Moreover, a POSITA would have understood that, by definition, a network in a telecommunications system enables the communications terminal to communicate with other communications terminals connected to the network via links. Ramchandran, ¶244.

8. Independent Claim 34

Claim 34 is identical to Claim 31, except it is written in terms of “the **network** comprising a video coder” instead of “the communication terminal comprising a video coder.” Ramchandran, ¶245. The TML6-Fandrianto combination renders Claim 34 obvious for the same reasons it renders Claim 31 obvious because a POSITA would have understood that a network includes one or more communications terminals, and if one communications terminal comprises a video

coder (as set forth in Claim 31), the network would necessarily comprise a video coder. Ramchandran, ¶245.

9. Claims 35-41

a. Independent Claim 35

Claim 35 is similar to Claim 19 with minor differences. First, Claim 35 is written in terms of “[a] **video coder...comprising an interpolator...adapted to**” perform the method steps of Claim 19. Ramchandran, ¶246. As discussed for Claim 23, if limiting, TML6 and Fandrianto teach these additional limitations.

Second, 35[pre][c] differs slightly from 19[pre][c]. In 35[pre][c], the “interpolator...generate[s] values for sub-pixels at fractional horizontal and vertical locations, **the resolution of the sub-pixels being determined by a positive integer N.**” As discussed for Claim 1, in the TML6-Fandrianto system, $N=2$, which is a “positive integer.” Furthermore, “the resolution of the sub-pixels [are] determined by” N , i.e., $1/2^N=1/2^2=1/4$ (quarter) resolution. Thus, the TML6-Fandrianto combination discloses Claim 35 for the reasons discussed for Claims 1 and 19. Ramchandran, ¶¶246-248.

b. Claim 36

Claim 36 is substantively the same as Claim 20, except it is written in terms of “[a] **video coder...in which the interpolator is...adapted to**” perform the method

of Claim 20. As discussed for Claims 23 and 35, if limiting, TML6 and Fandrianto teach these additional limitations. Ramchandran, ¶¶249-250.

c. Claim 37

Claim 37 is substantively the same as Claim 21, except it is written in terms of “[a] **video coder**...in which the interpolator is...adapted to” perform the method of Claim 21. As discussed for Claims 23 and 35, if limiting, TML6 and Fandrianto teach these additional limitations. Ramchandran, ¶¶251-252.

d. Claim 38

Claim 38 is substantively the same as Claim 22, except it is written in terms of “[a] **video coder**...in which the interpolator is...adapted to” perform the method of Claim 22. As discussed for Claims 23 and 35, if limiting, TML6 and Fandrianto teach these additional limitations. Ramchandran, ¶¶253-258.

e. Claim 39

As explained for Claim 35, TML6 and Fandrianto teach a video coder comprising a video encoder. Ramchandran, ¶259.

f. Claim 40

As explained for Claim 35, TML6 and Fandrianto teach a video coder comprising a video decoder. Ramchandran, ¶260. A POSITA would have further understood that a motion compensated video encoder includes a video decoder. *Id.*

g. Claim 41

As explained for Claims 26 and 35, TML6 and Fandrianto teach a codec comprising a video encoder/decoder. Ramchandran, ¶¶261-262.

10. Claims 42-45

a. Independent Claim 42

Claim 42 is substantively the same as Claim 35, except it is written in terms of “[a] **communications terminal** comprising a video coder...comprising an interpolator...adapted to” perform the steps of Claim 1. Ramchandran, ¶263. The TML6-Fandrianto combination teaches Claim 42 for the reasons discussed for Claims 27 and 35. Ramchandran, ¶264.

b. Claim 43

As discussed for Claims 35 and 39-42, the TML6-Fandrianto combination discloses a communications terminal comprising a video encoder, as recited in this claim. Ramchandran, ¶265.

c. Claim 44

As discussed for Claims 35 and 39-42, the TML6-Fandrianto combination discloses a communications terminal comprising a video decoder, as recited in this claim. Ramchandran, ¶265.

d. Claim 45

As discussed for Claims 35 and 39-42, the TML6-Fandrianto combination discloses a video codec comprising a video encoder/decoder, as recited in Claim 45. Ramchandran, ¶265.

11. Claims 46-48

a. Independent Claim 46

Claim 46 is substantively the same as Claim 35, except it is written in terms of “[a] **telecommunications system** comprising a communications terminal and a network, the **telecommunications network** and the communications terminal being **connected by a communications link** over which coded video can be transmitted, the communications terminal comprising a video coder...comprising an interpolator...adapted to” perform the steps of Claim 1. Ramchandran, ¶266. The TML6-Fandrianto combination thus renders Claim 46 obvious for the reasons discussed for Claims 31 and 35. Ramchandran, ¶267.

b. Claim 47

The TML6-Fandrianto combination renders obvious the additional limitations of this claim for the same reasons discussed for Claims 31-32 and 35. Ramchandran, ¶268.

c. Claim 48

The TML6-Fandrianto combination renders obvious the additional limitations of this claim for the same reasons discussed for Claims 31-32 and 35. Ramchandran, ¶269.

12. Independent Claim 49

Claim 49 is identical to Claim 46 except Claim 46 recites a “**communications terminal** comprising a video coder,” whereas Claim 49 recites a “**network** comprising a video coder.” The TML6-Fandrianto combination renders Claim 49 obvious for the same reasons discussed for Claim 34. Ramchandran, ¶270.

13. Independent Claims 50-51

Claims 50 and 51 are substantially the same as Claims 23 and 35, respectively, except they are written in terms of “[a] **codec**...comprising a video encoder and a video decoder.” Ramchandran, ¶271. The TML6-Fandrianto combination renders Claims 50-51 obvious for the reasons discussed for Claims 23, 26, and 35. Ramchandran, ¶¶271-275.

IX. NO BASIS FOR DISCRETIONARY INSTITUTION DENIAL

A. General Plastic

This is Amazon’s first petition for IPR of the ’469 patent. The ’469 patent was challenged in three prior IPRs filed in 2020 by Lenovo (United States) Inc. (“Lenovo”): IPR2020-01685, IPR2021-00035, and IPR2021-00036. 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 ’469 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 [’469] 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 ’469 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 ’469 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 ’469 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 '469 patent. Although the '469 patent accurately summarizes certain portions of the TML6 sub-pixel interpolation scheme in its Background section (EX-1001, 11:33-13:39), 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. 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 '469 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 ’469 patent’s Background summarizes certain aspects of TML6’s sub-pixel interpolation scheme with reference to Figures 13(a)-(c). EX-1001, 11:33-13:39. But the ’469 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) (instituting IPR because “[a]lthough [reference] was considered by the Examiner, the *combinations* of references set forth in the Petition were not considered by the Examiner”). Fandrianto was cited in an IDS but never applied against the claims nor discussed by the Examiner. EX-1002, 311. 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 also Shenzhen Zhiyi Tech. Co. v. iRobot Corp.*, IPR2017-02137, Paper 9, 9-10 (Apr. 2, 2018) (Examiner “initial[ing] a disclosure statement confirming that he had considered” the prior art reference insufficient to warrant denying institution under §325(d) because “[t]he [prior art] reference was not applied against the claims and there is no evidence that the Examiner considered the particular disclosures cited by Petitioner in the Petition”).

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 ’469 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.)

- *Nokia Techs. Oy v. Apple Inc.*, No. 2-16-cv-01440 (E.D. Tex.)
- *In the Matter of Certain Electronic Devices, Including Computers, Tablet Computers, and Components and Modules Thereof*, Inv. No. 337-TA-1208 (USITC)
- IPR2020-01685
- IPR2021-00035
- IPR2021-00036

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 6,950,469, 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 12,520. 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. 6,950,469 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, LLP
99 Hawley Lane
Stratford, CT 06614

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