

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,
Petitioner

v.

TELEFONAKTIEBOLAGET LM ERICSSON,
Patent Owner.

IPR2024-01008

U.S. Patent No. 9,641,841

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,641,841**

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.....	2
III.	STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED.....	2
A.	Prior Art	2
B.	Grounds for Challenge	3
IV.	TECHNOLOGY BACKGROUND.....	4
V.	'841 PATENT OVERVIEW	7
A.	Specification.....	8
B.	Prosecution History	12
VI.	LEVEL OF ORDINARY SKILL	13
VII.	CLAIM CONSTRUCTION	13
VIII.	DETAILED EXPLANATION OF GROUNDS.....	15
A.	Ground 1A: Obviousness over Bjontegaard and Bjontegaard's Code	15
1.	Bjontegaard and Bjontegaard's Code	15
2.	Claim 1	18
a.	1[pre].....	18
b.	1[a]	19
c.	1[b].....	21
d.	1[c]	21
e.	1[d].....	23
3.	Claim 3	24
a.	3[a]	24
b.	3[b].....	25

TABLE OF CONTENTS
(continued)

	Page
c. 3[c]	25
d. 3[d]	25
e. 3[e]	25
f. 3[f]	26
g. 3[g]	26
h. 3[h]	26
4. Claim 5	26
a. 5[pre]	26
b. 5[a]	27
c. 5[b]	27
d. 5[c]	27
e. 5[d]	28
f. 5[e]	28
g. 5[f]	28
5. Claim 7	28
a. 7[a]	28
b. 7[b]	28
c. 7[c]	28
d. 7[d]	28
6. Claim 9	29
a. 9[a]	29

TABLE OF CONTENTS
(continued)

	Page
b. 9[b].....	29
7. Claim 10.....	30
a. 10[a].....	30
b. 10[b].....	35
c. 10[c].....	35
d. 10[d].....	36
e. 10[e].....	38
f. 10[f].....	39
g. 10[g].....	39
8. Claim 12.....	40
a. 12[a].....	40
9. Claim 13.....	40
a. 13[pre].....	40
b. 13[a].....	41
c. 13[b].....	41
10. Claim 14.....	41
a. 14[a].....	41
11. Claim 15.....	42
a. 15[pre].....	42
b. 15[a].....	42
c. 15[b].....	42
d. 15[c].....	43

TABLE OF CONTENTS
(continued)

	Page
12. Claim 16.....	43
a. 16 [pre].....	43
b. 16[a]	44
c. 16[b]	44
d. 16[c]	44
e. 16[d]	44
f. 16[e]	44
g. 16[f].....	44
B. Ground 1B: Obviousness over Bjontegaard, Bjontegaard's Code, H.241, and B205/AAPA.....	45
1. H.241	45
2. B205/Applicant-Admitted Prior Art	45
3. Claim 2	46
a. 2[a]	46
b. 2[b]	49
4. Claim 6	49
a. 6[a]	49
b. 6[b]	50
5. Claim 11	50
a. 11[a]	50
b. 11[b]	51
c. 11[c]	51
C. Ground 2A: Obviousness over Aksu and Gonzalez	51

TABLE OF CONTENTS
(continued)

	Page
1. Aksu	52
2. Gonzalez.....	53
3. Claim 1	54
a. 1[pre].....	54
b. 1[a]	55
c. 1[b].....	59
d. 1[c]	59
e. 1[d].....	61
4. Claim 3.....	62
a. 3[a]	62
b. 3[b].....	62
c. 3[c]	62
d. 3[d].....	63
e. 3[e]	64
f. 3[f].....	65
g. 3[g].....	65
h. 3[h].....	65
5. Claim 5.....	65
a. 5[pre].....	65
b. 5[a]	66
c. 5[b].....	66
d. 5[c]	66

TABLE OF CONTENTS
(continued)

	Page
e. 5[d].....	66
f. 5[e].....	67
g. 5[f].....	67
6. Claim 7.....	67
a. 7[a].....	67
b. 7[b].....	67
c. 7[c].....	67
d. 7[d].....	67
7. Claim 9.....	67
a. 9[a].....	67
b. 9[b].....	68
8. Claim 12.....	68
a. 12[a].....	68
9. Claim 13.....	69
a. 13[pre].....	69
b. 13[a].....	69
c. 13[b].....	69
10. Claim 14.....	70
a. 14[a].....	70
11. Claim 15.....	70
a. 15[pre].....	70
b. 15[a].....	70

TABLE OF CONTENTS
(continued)

	Page
c. 15[b].....	70
d. 15[c].....	71
12. Claim 16.....	71
a. 16 [pre].....	71
b. 16[a].....	71
c. 16[b].....	72
d. 16[c].....	72
e. 16[d].....	72
f. 16[e].....	72
g. 16[f].....	72
D. Ground 2B: Obviousness over Aksu, Gonzalez, H.241, and B205 (AAPA).....	72
1. Claim 2.....	72
a. 2[a].....	72
b. 2[b].....	73
2. Claim 6.....	73
a. 6[a].....	73
b. 6[b].....	73
3. Claim 11.....	74
a. 11[a].....	74
b. 11[b].....	75
c. 11[c].....	75

TABLE OF CONTENTS
(continued)

	Page
E. Ground 2C: Obviousness over Aksu in view of Gonzalez, Fu, and B118	75
1. Fu	75
2. B118.....	76
3. Claim 10.....	77
a. 10[a]	77
b. 10[b].....	79
c. 10[c]	80
d. 10[d].....	80
e. 10[e]	82
f. 10[f].....	83
g. 10[g].....	83
IX. DISCRETIONARY FACTORS FAVOR INSTITUTION	84
A. Fintiv	84
B. 325(d)	85
X. MANDATORY NOTICES	86
A. Real Party in Interest.....	86
B. Related Matters	86
C. Notice of Counsel and Service Information.....	87
D. Service Information.....	88
XI. CONCLUSION.....	88

PETITIONER'S EXHIBIT LIST

Exhibit	Reference
1001	U.S. Patent No. 9,641,841 (“’841 Patent”)
1002	File History of the ’841 Patent
1003	Declaration of Dr. Dan Schonfeld in Support of Petition
1004	Dr. Dan Schonfeld CV
1005	Declaration of Dr. Clifford Reader
1006	Dr. Clifford Reader CV
1007	Declaration of June Munford
1008	U.S. Provisional Pat. App. No. 61/447,862
1009	Gisle Bjontegaard et al., <i>Definition of the new coding elements from Telenor</i> , VCEG Document Q15-J-28 (May 5, 2000) (“Bjontegaard”)
1010	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, File System Printout (q15j29.zip) (May 9, 2000) (“Bjontegaard’s Code”)
1011	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, ldecod.c (q15j29.zip\ldecod\src\ldecod.c) (May 9, 2000) (“Bjontegaard’s Code”)
1012	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, image.c (q15j29.zip\ldecod\src\image.c) (May 9, 2000) (“Bjontegaard’s Code”)
1013	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, lencod.c (q15j29.zip\lencod\src\lencod.c) (May 9, 2000) (“Bjontegaard’s Code”)
1014	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, image.c (q15j29.zip\lencod\src\image.c) (May 9, 2000) (“Bjontegaard’s Code”)

Exhibit	Reference
1015	Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , VCEG Document Q15-J-29, Readme (q15j29.zip\readme.txt) (May 9, 2000) (“Bjontegaard’s Code”)
1016	Gisle Bjontegaard et al., <i>Deblocking filter for 4x4 based coding</i> , VCEG Document Q15-J-27 (May 9, 2000) (“Bjontegaard’s Cover”)
1017	Telecommunication Standardization Sector of ITU, <i>ITU Recommendation H.241: Extended video procedures and control signals for H.300-series terminals</i> (May 29, 2006) (“H.241”)
1018	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>Test Model under Consideration</i> , Document JCTVC-B205, v.7 (Oct. 9, 2010) (“B205”)
1019	International Application Publication No. WO 01/54415 (“Aksu”)
1020	Rafael C. Gonzalez & Richard E. Woods, <i>Digital Image Processing</i> (3d ed.), Pearson-Prentice Hall (2008) (“Gonzalez”)
1021	International Application Publication No. WO 2009/033390
1022	English Translation of International Application Publication No. WO 2009/033390 (“Fu”)
1023	Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, <i>Working Draft Number 2, Revision 2 (WD-2)</i> , Document JVT-B118r2 (Mar. 15, 2002) (“B118”)
1024	Peter List et al., <i>Adaptive Deblocking Filter</i> (July 2003)
1025	U.S. Application Publication No. 2006/0078048 (“Bjontegaard ’048”)
1026	<i>In the Matter of Certain Electronic Computing Devices, and Components and Modules Thereof</i> , Investigation No. 337-TA-1387, Joint Claim Construction Chart (ITC May 8, 2024)
1027	<i>In the Matter of Certain Electronic Computing Devices, and Components and Modules Thereof</i> , Investigation No. 337-TA-1387, Complainant’s Opening Claim Construction Brief (ITC May 15, 2024)

Exhibit	Reference
1028	David Marr, <i>Vision</i> , MIT Publishing (2010)
1029	Ross Gaunt, <i>Color Spaces in Digital Video</i> (May 1997)
1030	BDTi, How video compression works, EETimes (Aug. 6, 2007), https://www.eetimes.com/how-video-compression-works/
1031	Synopi, <i>Best Video Codec for Live Streaming</i> , https://www.synopi.com/best-video-codec-for-live-streaming
1032	Mahdi Nazm Bojnordi, et al., <i>A Fast Two Dimensional Deblocking Filter for H.264/AVC Video Coding</i> , IEEE, 2006 Canadian Conference on Electrical and Computer Engineering, 2017 (Jan. 15, 2007)
1033	Chung-Ming Chen & Chung-Ho Chen, <i>Parallel Processing for Deblocking Filter in H.264/AVC</i> , Proceeding of the fourth IASTED International conference on CIIT, 2005:188 (2005)
1034	Purdue University, <i>CE 503 – Photogrammetry I</i> (2002), https://engineering.purdue.edu/~bethel/digim1.pdf
1035	Telecommunication Standardization Sector of ITU, <i>ITU Recommendation H.263: Video coding for low bit rate communication</i> (Feb. 1998)
1036	Telecommunication Standardization Sector of ITU, <i>ITU Recommendation H.264: Advanced video coding for generic audiovisual services</i> (Mar. 2005)
1037	Telecommunication Standardization Sector of ITU, <i>ITU Recommendation H.264: Advanced video coding for generic audiovisual services</i> (Mar. 2010)
1038	UB Video Inc., Emerging H.26L Standard: Overview and TMS320C64x Digital Media Platform Implementation, White Paper (2002), https://www.ti.com/pdfs/vf/vidimg/h26l_wp.pdf
1039	Chun-Lung Hsu & Yu-Sheng Huang, <i>A Fast-Deblocking Boundary-strength Based Architecture Design of Deblocking Filter in H.264/AVC Applications</i> , 52 J. of Signal Processing Systems 211 (2008)

Exhibit	Reference
1040	Ian E. Richardson, <i>The H.264 Advanced Video Compression Standard</i> (2d. ed, Wiley 2010)
1041	Ramesh Jain et al., <i>Machine Vision</i> , McGraw-Hill (1995)
1042	<i>JCT-VC – Joint Collaborative Team on Video Coding</i> , https://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/jctvc.aspx
1043	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>CE12: Ericsson's and MediaTek's deblocking filter</i> , Document JCTVC-F118 (July 2, 2011)
1044	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>Simplification of chroma deblocking filter in JCTVC-F118</i> , Document JCTVC-F120 (July 1, 2011)
1045	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>Improving Deblocking filter efficiency</i> , Document JCTVC-F256 (July 1, 2011)
1046	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>CE12: SK Telecom/SKKU Deblocking Filter</i> , Document JCTVC-F258 (July 1, 2011)
1047	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>Report on subjective viewing test for deblocking filter proposals</i> , Document JCTVC-F767 (July 21, 2011)
1048	Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, <i>Meeting report of the sixth meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Torino, IT, 14–22 July 2011</i> , Document JCTVC-F800 (Nov. 18, 2011)
1049	Mamta Juneja & Parvinder Singh Sandhu, <i>Performance Evaluation of Edge Detection Techniques for Images in Spatial</i>

Exhibit	Reference
	<i>Domain</i> , 1:5 Int'l J. of Comput. Theory and Engineering 1793 (Dec. 2009)
1050	Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, <i>Joint Model Number 1, Revision 1 (JM-1r1)</i> , Document JVT-A003r1 (Jan. 18, 2002)
1051	Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, <i>AHG Report: Loop Filter</i> , Document JVT-B011 (Jan. 25, 2002)
1052	Kin-Hung Lam, <i>Reduced Complexity Deblocking Filter for H.264 Video Coding</i> , Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Sys. and Comput., 2005 (Mar. 13, 2006)
1053	Yuan Li et al., <i>A Novel Deblocking Filter Algorithm in H.264 for Real Time Implementation</i> , Third Int'l Conf. on Multimedia and Ubiquitous Engineering (Nov. 10, 2009)
1054	O.L. MacSorley, <i>High-Speed Arithmetic in Binary Computers</i> , 49:1 Proc. of the IRE 67 (Jan. 1961)
1055	Dave Shapton, <i>Microsoft Media Player</i> , Sound on Sound (Jan. 2001), https://www.soundonsound.com/people/microsoft-media-player
1056	J. Ren & N. Kehtarnavaz, <i>Algorithmic Optimization for H.264 Deblocking Filter on Portable Devices</i> , 2007 IEEE Int'l Symposium on Consumer Elecs. (Nov. 12, 2007)
1057	Amir Said, <i>An Image Multiresolution Representation for Lossless and Lossy Compression</i> , 5:9 IEEE Transactions on Image Processing 1303 (Sept. 1996)
1058	International Patent Application Publication No. WO2011/12796
1059	Kemal Ugur et al., <i>High Performance, Low Complexity Video Coding and the Emerging HEVC Standard</i> , 20:12 IEEE Transactions on Circuits and Sys. for Video Tech., 1688 (Nov. 18, 2010)
1060	U.S. Patent No. 7,283,588
1061	U.S. Patent No. 7,430,336

Exhibit	Reference
1062	U.S. Patent No. 7,692,682
1063	U.S. Patent No. 10,142,659
1064	U.S. Application Publication No. 2005/0243911 (“Kwon”)
1065	U.S. Application Publication No. 2006/0078048
1066	U.S. Application Publication No. 2009/0141808
1067	Video Coding Experts Group (VCEG), <i>Reduced Complexity Loop Filter</i> , ITU-T Study Group 16, Document VCEG-L25 (Dec. 18, 2000)
1068	Video Coding Experts Group (VCEG), <i>H.26L Test Model Long Term Number 6 (TML-6) draft0</i> , ITU-T Study Group 16, Document VCEG-L45d0 (Mar. 3, 2001)
1069	Video Coding Experts Group (VCEG), <i>Core Experiment results on Low Complexity Loop Filtering</i> , ITU-T Study Group 16, Document VCEG-M21 (Mar. 27, 2001)
1070	Huifeng Shen et al., <i>R-D Optimal Motion Estimation for Fast H.264/AVC Bit-Rate Reduction</i> , Microsoft (Jan. 2006), https://www.microsoft.com/en-us/research/publication/r-d-optimal-motion-estimation-fast-h-264avc-bit-rate-reduction/
1071	Web Archive, <i>VLC Feature List</i> , VideoLAN (Jun 5, 2003), https://web.archive.org/web/20030605113020/ https://www.videolan.org/vlc/features.html
1072	Dan Lavry, <i>Understanding FIR (Finite Impulse Response) Filters – An Intuitive Approach</i> , Lavry Engineering (1997), https://lavryengineering.com/pdfs/lavry-understanding-fir-filters.pdf
1073	Mei-Yin Shen & C.-C. Jay Kuo, <i>Review of Postprocessing Techniques for Compression Artifact Removal</i> , 9:1 J. of Visual Comm’n and Image Rep. (Mar. 1998)
1074	Gary Sullivan, <i>Q.15/16 Meeting Report</i> , ITU-T, Study Group 16, Doc. E(WP 3/16) (Feb. 17, 2000)
1075	<i>About ITU</i> , https://www.itu.int/en/about/Pages/default.aspx

Exhibit	Reference
1076	<i>About MPEG</i> , https://www.mpeg.org/about-mpeg/
1077	<i>H.241 Download</i> , https://www.itu.int/rec/T-REC-H.241-200605-S
1078	Thomas Wiegand et al., <i>Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization</i> , 20 IEEE Trans. on Circuits and Sys. For Video Tech. 1661 (Dec. 2010)
1079	Gary J. Sullivan et al., <i>Using the Draft H.26L Video Coding Standard for Mobile Applications</i> , IEEE (2001)
1080	<i>ITU Homepage</i> , https://www.itu.int/en/Pages/default.aspx
1081	All Meetings, VCEG, www.itu.int - /wftp3/av-arch/video-site/
1082	Osaka Meeting (J) – Document Register, VCEG, https://www.itu.int/wftp3/av-arch/video-site/0005_Osa/
1083	“Deblocking,” ITU Search Results, https://www.itu.int/search
1084	<i>ITU-T Homepage</i> , https://www.itu.int/en/ITU-T/Pages/default.aspx
1085	<i>ITU-T In Brief</i> , https://www.itu.int/en/ITU-T/about/Pages/default.aspx
1086	<i>ITU-T H.241 Recommendation</i> , https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11656&lang=en
1087	<i>ITU-T Recommendation by Series, H Series</i> , https://www.itu.int/itu-t/recommendations/index.aspx?ser=H
1088	<i>ITU-T Recommendation by Series, H Series</i> , https://www.itu.int/itu-t/recommendations/index.aspx?ser=H , WayBack Machine (Feb. 20, 2010)
1089	<i>ITU-T Recommendations and Other Publications</i> , https://www.itu.int/en/ITU-T/publications/Pages/default.aspx , Wayback Machine (July 14, 2010)
1090	<i>ITU-T Recommendations and Other Publications</i> , https://www.itu.int/en/ITU-T/publications/Pages/default.aspx

Exhibit	Reference
1091	<i>ITU-T Recommendations</i> , https://www.itu.int/en/ITU-T/publications/Pages/recs.aspx
1092	<i>ITU-T Study Group 16 List of Questions under Study (Study Period 1997-2000)</i> , https://www.itu.int/ITU-T/1997-2000/com16/questions_old.html
1093	All Meetings, JCTVC, phenix.int-evry.fr/jct/doc_end_user/all_meeting.php
1094	Geneva Meeting (B) – Document Register Archive, JCTVC, https://www.itu.int/wftp3/av-arch/jctvc-site/2010_07_B_Geneva/
1095	Geneva Meeting (B) – Document Register, JCTVC, phenix.int-evry.fr/jct/doc_end_user/current_meeting.php?id_meeting=20&search_id_group=1&search_sub_group=1
1096	Guangzhou Meeting (C) – Document Register, JCTVC, phenix.int-evry.fr/jct/doc_end_user/current_meeting.php?id_meeting=142&search_id_group=1&search_sub_group=1
1097	JCTVC, <i>Terms of Reference of the Joint Collaborative Team on Video Coding Standard Development</i> (Jan. 2010)
1098	Gary J. Sullivan & Jens-Rainer Ohm, <i>Meeting report of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Dresden, DE, 15-23 April, 2010</i> , JCT-VC-A200 (June 2010)
1099	Gary J. Sullivan & Jens-Rainer Ohm, <i>Meeting report of the second meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Geneva, CH, 21-28 July, 2010</i> , JCT-VC-B200 (October 7, 2010)
1100	Gary J. Sullivan & Jens-Rainer Ohm, <i>Meeting report of the third meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Guangzhou, CN, 7–15 October, 2010</i> , JCT-VC-C400 (Jan. 16, 2011)
1101	<i>Joint Video Team (JVT)</i> , https://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jvt.aspx

Exhibit	Reference
1102	All Meetings Archive, JVT, https://www.itu.int/wftp3/av-arch/jvt-site/
1103	Geneva Meeting (B) – Document Register Archive, JVT, https://www.itu.int/wftp3/av-arch/jvt-site/2002_01_Geneva/
1104	Gary Sullivan, <i>List of Documents of 2nd JVT Meeting (Geneva, 29 Jan – Feb 1, 2001</i> , JVT-B00r0 (Jan. 28, 2002)
1105	Gary Sullivan et al., <i>JVT (of ISO/IEC MPEG and ITU-T Q.6/16 VCEG) 2nd Meeting Report, Jan 29-Feb 1 2002</i> , JVT-B001d1 (Feb. 5, 2002)
1106	<i>List of Documents for the Ninth Meeting ITU-T Q.15/SG16 Video Coding Experts Group</i> , VCEG Q15-I-00r0 (Sept. 25, 1999)
1107	<i>List of Documents for the Tenth Meeting ITU-T Q.15/SG16 Video Coding Experts Group</i> , VCEG Q15-J-00r0 (May 12, 2000)
1108	<i>List of Documents for the Tenth Meeting ITU-T Q.15/SG16 Video Coding Experts Group</i> , VCEG Q15-J-00r2 (Mar. 14, 2001)
1009	Gary Sullivan, <i>Q.15/16 Meeting report</i> , VCEG Q15-J-01 (Temp. Doc. No. -E(WP 3/16)) (Feb. 17, 2000)
1010	Gary Sullivan, <i>Announcement of the Tenth Meeting of the Experts Group for ITU-T Q.15/SG16 – Advanced Video Coding Osaka, Japan, May 16-19, 2000</i> , VCEG Q15-J-T0d2 (Apr. 11, 2000)
1111	<i>Terms of Reference for Joint Video Team (JVT) Activities</i> , http://www.itu.int/ITU-T/studygroups/com16/jvt/JVTTToR.pdf , Wayback Machine (Apr. 21, 2003)
1112	Gary Sullivan et al., <i>Invitation to the Second JVT Meeting Geneva, Switzerland, Jan. 29 – Feb. 1, 2002</i> , JVT-B-TD-00r1 (Jan. 24, 2002)
1113	Cliff Reader, <i>Origins of the performance of H.264/AVC: an account of the development of H.263 and H.264 coding tools</i> , 5909 Proc. of SPIE (2005)

Exhibit	Reference
1114	Ron Yan, et al., <i>Error Robust Coding for the FGS Enhancement Bitstream</i> (2001), with bibliographical printout
1115	European Patent No. 2,249,572
1116	U.S. Application No. 2010/0135389
1117	Thomas Stockhammer & Stephan Wenger, <i>Standard-Compliant Enhancements of JVT Coded Video for Transmission over Fixed and Wireless IP</i> (2002), with bibliographical printout
1118	International Patent Application No. WO2003/063501
1119	European Patent No. 1,649,697
1120	U.S. Patent No. 7,227,901
1121	U.S. Application No. 2007/0014358
1122	U.S. Application No. 2008/0260041
1123	U.S. Application No. 2007/0200923
1124	U.S. Application No. 2008/0240252
1125	U.S. Application No. 2008/0037656
1126	French Patent No. 2,927,758 (machine translation and original)
1127	Hsueh-Ming Hang et al., <i>Towards the Next Video Standard: High Efficiency Video Coding</i> , Proc. 2d APSIPA Annual Summit and Conf., 609 (Dec. 2010)
1128	Hau-cheoil Choi, <i>Deblocking Filter and Adaptive Loop Filter</i> , 15:4 J. Broadcasting Eng. 66 (Dec. 2010) (machine translation and original)
1129	U.S. Application No. 2012/0106640
1130	Japanese Patent Application No. 2012/124591 (machine translation and original)
1131	All Meetings, VCEG Pictel File-Transfer Protocol Website, http://standard.pictel.com/ftp/video-site/ , WayBack Machine (Feb. 5, 2003)

Exhibit	Reference
1132	Osaka Meeting (J) – Document Register, VCEG Pictel File-Transfer Protocol Website, http://standard.pictel.com/ftp/video-site/0005_Osa/ , WayBack Machine (Dec. 25, 2002)
1133	Geneva Meeting (B) – Document Register, VCEG/JVT Pictel File-Transfer Protocol Website, http://standard.pictel.com/ftp/video-site/0201_Gen/ , WayBack Machine (Dec. 25, 2002)

LISTING OF CHALLENGED CLAIMS

Reference	Claim Limitation
Claim 1	
1[pre]	A method for filtering control applicable to a block of multiple pixels in a video frame, each pixel having a respective pixel value, said method comprising:
1[a]	calculating a first filter decision value for said block, by only using pixels p_{2_i}, p_{1_i}, and p_{0_i}, based on: $ p_{2_i} - 2p_{1_i} + p_{0_i} $ wherein p_{0_i} denotes a pixel value of a pixel closest to, in a first line of pixels in said block, a block boundary to a neighboring block of multiple pixels in said video frame, p_{1_i} denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary and p_{2_i} denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;
1[b]	calculating a second filter decision value for said neighboring block, by only using pixels q_{2_i}, q_{1_i}, and q_{0_i}, based on: $ q_{2_i} - 2q_{1_i} + q_{0_i} $ wherein q_{0_i} denotes a pixel value of a pixel in said neighboring block closest to, in a corresponding first line of pixels in said neighboring block, said block boundary, q_{1_i} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary and q_{2_i} denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary;
1[c]	determining how many pixels in said first line of pixels in said block to filter relative to said block boundary based on a comparison of said first filter decision value to a first threshold value; and
1[d]	determining how many pixels in said corresponding first line of pixels in said neighboring block to filter relative to said block boundary based on a comparison of said second filter decision value to a second threshold value.

Reference	Claim Limitation
Claim 2	
2[a]	The method according to claim 1, wherein calculating said first filter decision value comprises calculating said first filter decision value as: $ p_{2_0}-2p_{1_0}+p_{0_0} + p_{2_3}-2p_{1_3}+p_{0_3} $ wherein p_{0_0} denotes said pixel value of said pixel closest to, in said first line of pixels, said block boundary, p_{1_0} denotes said pixel value of said pixel next closest to, in said first line of pixels, said block boundary, p_{2_0} denotes said pixel value of said pixel second next closest to, in said first line of pixels, said block boundary, p_{0_3} denotes a pixel value of a pixel closest to, in a second line of pixels in said block, said block boundary, p_{1_3} denotes a pixel value of a pixel next closest to, in said second line of pixels, said block boundary and p_{2_3} denotes a pixel value of a pixel second next closest to, in said second line of pixels, said block boundary; and
2[b]	calculating said second filter decision value comprises calculating said second filter decision value as: $ q_{2_0}-2q_{1_0}+q_{0_0} + q_{2_3}-2q_{1_3}+q_{0_3} $ wherein q_{0_0} denotes said pixel value of said pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary, q_{1_0} denotes said pixel value of said pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary, q_{2_0} denotes said pixel value of said pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary, q_{0_3} denotes a pixel value of a pixel in said neighboring block closest to, in a corresponding second line of pixels in said neighboring block, said block boundary, q_{1_3} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding second line of pixels, said block boundary and q_{2_3} denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding second line of pixels, said block boundary.

Reference	Claim Limitation
Claim 3	
3[a]	The method according to claim 1, wherein determining how many pixels in said first line of pixels in said block to filter comprises:
3[b]	comparing said first filter decision value to a first threshold value;
3[c]	determining to filter two pixels in said first line of pixels in said block relative to said block boundary if said first filter decision value is below said first threshold value; and
3[d]	determining to filter one pixel in said first line of pixels in said block relative to said block boundary if said first filter decision value is equal to or exceeds said first threshold value; and
3[e]	determining how many pixels in said corresponding first line of pixels in said neighboring block to filter comprises:
3[f]	comparing said second filter decision value to a second threshold value;
3[g]	determining to filter two pixels in said corresponding first line of pixels in said neighboring block relative to said block boundary if said second filter decision value is below said second threshold value; and
3[h]	determining to filter one pixel in said corresponding first line of pixels in said neighboring block relative to said block boundary if said second filter decision value is equal to or exceeds said second threshold value.
Claim 5	
5[pre]	A filtering control device comprising:
5[a]	a first decision value calculator configured to calculate a first filter decision value for a block of multiple pixels in a video frame, by only using pixels p_{2i}, p_{1i}, and p_{0i}, based on: $ p_{2i}-2p_{1i}+p_{0i} $ wherein p_{0i} denotes a pixel value of a pixel closest to, in a first line of pixels in said block, a block boundary to a neighboring block of multiple pixels in said video frame, p_{1i} denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary and p_{2i} denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;

Reference	Claim Limitation
5[b]	a second decision value calculator configured to calculate a second filter decision value for a neighboring block, by only using pixels q_{2_i}, q_{1_i}, and q_{0_i}, based on: $ q_{2_i} - 2q_{1_i} + q_{0_i} $ wherein q_{0_i} denotes a pixel value of a pixel in said neighboring block closest to, in a corresponding first line of pixels in said neighboring block, said block boundary, q_{1_i} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary and q_{2_i} denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary;
5[c]	a first pixel determiner configured to determine how many pixels in a first line of pixels in said block to filter relative to said block boundary based on said first filter decision value calculated by said first decision value calculator; and
5[d]	a second pixel determiner configured to determine how many pixels in a corresponding first line of pixels in said neighboring block to filter relative to said block boundary based on said second filter decision value calculated by said second decision value calculator, wherein
5[e]	further comprising a first comparator configured to compare said first filter decision value calculated by said first decision value calculator to a first threshold value; and
5[f]	a second comparator configured to compare said second filter decision value calculated by said second decision value calculator to a second threshold value.

Reference	Claim Limitation
Claim 6	
6[a]	The device according to claim 5, wherein said first decision value calculator is configured to calculate said first filter decision value as: $ p_{2_0}-2p_{1_0}+p_{0_0} + p_{2_3}-2p_{1_3}+p_{0_3} $ wherein p_{0_0} denotes said pixel value of said pixel closest to, in said first line of pixels, said block boundary, p_{1_0} denotes said pixel value of said pixel next closest to, in said first line of pixels, said block boundary, p_{2_0} denotes said pixel value of said pixel second next closest to, in said first line of pixels, said block boundary, p_{0_3} denotes a pixel value of a pixel closest to, in a second line of pixels in said block, said block boundary, p_{1_3} denotes a pixel value of a pixel next closest to, in said second line of pixels, said block boundary and p_{2_3} denotes a pixel value of a pixel second next closest to, in said second line of pixels, said block boundary; and
6[b]	said second decision value calculator is configured to calculate said second filter decision value as: $ q_{2_0}-2q_{1_0}+q_{0_0} + q_{2_3}-2q_{1_3}+q_{0_3} $ wherein q_{0_0} denotes said pixel value of said pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary, q_{1_0} denotes said pixel value of said pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary, q_{2_0} denotes said pixel value of said pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary, q_{0_3} denotes a pixel value of a pixel in said neighboring block closest to, in a corresponding second line of pixels in said neighboring block, said block boundary, q_{1_3} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding second line of pixels, said block boundary and q_{2_3} denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding second line of pixels, said block boundary.

Reference	Claim Limitation
Claim 7	
7[a]	The device according to claim 5, further comprising: said first pixel determiner is configured to i) determine to filter two pixels in said first line of pixels in said block relative to said block boundary if said first filter decision value is below said first threshold value as determined by said first comparator and
7[b]	ii) determine to filter one pixel in said first line of pixels in said block relative to said block boundary if said first filter decision value is equal to or exceeds said first threshold value as determined by said first comparator; and
7[c]	said second pixel determiner is configured to i) determine to filter two pixels in said corresponding first line of pixels in said neighboring block relative to said block boundary if said second filter decision value is below said second threshold value as determined by said second comparator and
7[d]	ii) determine to filter one pixel in said corresponding first line of pixels in said neighboring block relative to said block boundary if said second filter decision value is equal to or exceeds said second threshold value as determined by said second comparator.
Claim 9	
9[a]	The device according to claim 5, further comprising a threshold determiner configured to i) determine said first threshold value based on a quantization parameter associated with said block, and
9[b]	ii) determine said second threshold value based on a quantization parameter associated with said neighboring block.

Reference	Claim Limitation
Claim 10	
10[a]	The device according to claim 5, further comprising: a first offset calculator configured to calculate a first offset Δ based on $\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16}$ wherein $p0_j$ denotes a pixel value of a pixel closest to, in said first line of pixels, said block boundary, $p1_j$ denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary, $q0_j$ denotes a pixel value of a pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary and $q1_j$ denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary;
10[b]	a first pixel modifier configured to modify said pixel value of said pixel closest to said block boundary in said first line of pixels by adding said first offset calculated by said first offset calculator to said pixel value of said pixel closest to said block boundary in said first line of pixels;
10[c]	a second pixel modifier configured to modify said pixel value of said pixel in said neighboring block closest to said block boundary in said corresponding first line of pixels by subtracting said first offset calculated by said first offset calculator from said pixel value of said pixel in said neighboring block closest to said block boundary in said corresponding first line of pixels;
10[d]	a second offset calculator configured to calculate, if said first filter decision value is below said first threshold value as determined by said first comparator, a second offset based on $\frac{p0_j + p2_j - 2p1_j + 2\Delta}{4}$ wherein $p2_j$ denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;

Reference	Claim Limitation
10[e]	a third pixel modifier configured to modify, if said first filter decision value is below said first threshold as determined by said first comparator, said pixel value of said pixel next closest to said block boundary in said first line of pixels by adding said second offset calculated by said second offset calculator to said pixel value of said pixel next closest to said block boundary in said first line of pixels;
10[f]	a third offset calculator configured to calculate, if said second filter decision value is below said second threshold value as determined by said second comparator, a third offset based on $\frac{q0_j + q2_j - 2q1_j - 2\Delta}{4}$ wherein q2_j denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary; and
10[g]	a fourth pixel modifier configured to modify, if said second filter decision value is below said second threshold as determined by said second comparator, said pixel value of said pixel in said neighboring block next closest to said block boundary in said corresponding first line of pixels by adding said third offset calculated by said third offset calculator to said pixel value of said pixel in said neighboring block next closest to said block boundary in said corresponding first line of pixels.
Claim 11	
11[a]	The device according to claim 5, further comprising a third comparator configured to compare a sum of said first filter decision value and said second filter decision value to a threshold value, wherein
11[b]	said first pixel determiner is configured to determine, if said sum of said first filter decision value and said second filter decision value is below said threshold value as determined by said third comparator, how many pixels in said first line of pixels in said block to filter relative to said block boundary based on said first filter decision value calculated by said first decision value calculator; and

Reference	Claim Limitation
11[c]	said second pixel determiner is configured to determine, if said sum of said first filter decision value and said second filter decision value is below said threshold value as determined by said third comparator, how many pixels in said corresponding first line of pixels in said neighboring block to filter relative to said block boundary based on said second filter decision value calculated by said second decision value calculator.
Claim 12	
12[a]	An encoder comprising a filtering control device according to claim 5.
Claim 13	
13[pre]	A user equipment comprising:
13[a]	a memory configured to store video frames; and
13[b]	an encoder according to claim 12 configured to encode said video frames into encoded video frames, wherein said memory is further configured to store said encoded video frames.
Claim 14	
14[a]	A decoder comprising a filtering control device according to claim 5.
Claim 15	
15[pre]	A user equipment comprising:
15[a]	a memory configured to store encoded video frames;
15[b]	a decoder according to claim 14 configured to decode said encoded video frames into decoded video frames; and
15[c]	a media player configured to render said decoded video frames into video data displayable on a display.
Claim 16	
16[pre]	A non-transitory computer-readable medium comprising, stored thereupon, a computer program for filtering control of a block of multiple pixels in a video frame each pixel having a respective pixel value, said computer program comprising code means which when run on a computer causes the computer to:
16[a]	calculate a first filter decision value for said block, by only using pixels p_{2i} , p_{1i} , and p_{0i} , based on: $ p_{2i}-2p_{1i}+p_{0i} $

Reference	Claim Limitation
	wherein $p0_i$, denotes a pixel value of a pixel closest to, in a first line of pixels in said block, a block boundary to a neighboring block of multiple pixels in said video frame, $p1_i$ denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary and $p2_i$ denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;
16[b]	calculate a second filter decision value for said neighboring block, by only using pixels $q2_i$, $q1_i$, and $q0_i$, based on: $ q2_i - 2q1_i + q0_i $ wherein $q0_i$ denotes a pixel value of a pixel in said neighboring block closest to, in a corresponding first line of pixels in said neighboring block, said block boundary, $q1_i$ denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary and $q2_i$ denotes a pixel value of a pixel in said neighboring block second next closest to, in said corresponding first line of pixels, said block boundary;
16[c]	determine how many pixels in a first line of pixels in said block to filter relative to said block boundary based on said first filter decision value; and
16[d]	determine how many pixels in a corresponding first line of pixels in said neighboring block to filter relative to said block boundary based on said second filter decision value,
16[e]	wherein determine how many pixels in said first line of pixels in said block to filter comprises determining how many pixels in said first line of pixels in said block to filter relative to said block boundary based on a comparison of said first filter decision value to a first threshold value; and
16[f]	determine how many pixels in said corresponding first line of pixels in said neighboring block to filter comprises determining how many pixels in said corresponding first line of pixels in said neighboring block to filter relative to said block boundary based on a comparison of said second filter decision value to a second threshold value.

I. INTRODUCTION

When encoding video into digital data, each image is broken into smaller “blocks” that are processed separately for efficiency. After the image is reconstructed from the digital data, however, the image may look “blocky” as an artifact of this block-based encoding. These “blocking artifacts”—which occur at the boundaries between neighboring blocks—have long been reduced using a “deblocking” filter, which first evaluates the pixels near the boundary to determine whether and how much filtering is needed, and, if needed, filters (blurs) the boundary between adjacent blocks to make them look less “blocky.” But it was also known that, sometimes, what looks like a blocking artifact may actually be part of the original image—such as an edge between two objects—that should not be blurred.

The ’841 Patent purports to avoid filtering true image details by *separately* determining how many pixels to filter on each side of a block boundary. But such “asymmetric” decision making was known, as was the equation the ’841 Patent used to make this decision. Although the ’841 Patent also provides another specific equation for *filtering* (blurring) the pixels once a decision had been made, the prior art taught nearly identical equations and defined the boundaries within which those in the art experimented. Using a trivially different equation for filtering was not inventive either.

Intel Corporation (“Intel” or “Petitioner”) accordingly petitions for *inter partes* review (“IPR”) of claims 1-3, 5-7, and 9-16 (“Challenged Claims”) of the ’841 Patent.

II. GROUNDS FOR STANDING

Petitioner certifies that the ’841 Patent is available for *inter partes* review and Petitioner is not barred or estopped from requesting IPR on the grounds herein.

III. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

A. Prior Art

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

Prior Art
Bjontegaard (Gisle Bjontegaard et al., <i>Definition of the new coding elements from Telenor</i> , Q15-J-28), published May 9, 2000; prior art under §102(b). (Ex-1009)
Bjontegaard’s Code (Gisle Bjontegaard et al., <i>Software Including Proposed New H.26L Coding Elements from Telenor</i> , Q15-J-29), published May 9, 2000; prior art under §102(b). (Ex-1010-1015)
H.241 (ITU-T, <i>H.241: Extended video procedures and control signals for H.300-series terminals</i>), published December 2006; prior art under §102(b). (Ex-1017)
B205 (JCT-VC, <i>Test Model under Consideration</i> , JCTVC-B205 v.7), published October 9, 2010; prior art under §102(a), and Applicant Admitted Prior Art (“AAPA”), ’841 Patent, 2:5-3:5. (Ex-1018)

Aksu (WO 01/54415), filed January 22, 2001, published July 26, 2001; prior art under §102(b). ¹ (Ex-1019)
Gonzalez (Rafael Gonzalez, <i>Digital Image Processing</i> (3d ed.)), published in 2008; prior art under §102(b). (Ex-1020)
Fu (certified translation of WO 2009/033390), filed August 21, 2008, published March 19, 2009; prior art under §102(b). (Ex-1022). The original is provided. (Ex-1021)
B118 (JVT, <i>Working Draft Number 2, Revision 2 (WD-2)</i> , JVT-B118r2), published March 15, 2002; prior art under §102(b). (Ex-1023)

B. Grounds for Challenge

Ground		Claims	References
1A	§103	1, 3, 5, 7, 9, 10, 12-16	Bjontegaard, Bjontegaard's Code
1B	§103	2, 6, 11	Bjontegaard, Bjontegaard's Code, H.241, B205/AAPA
2A	§103	1, 3, 5, 7, 9, 12-16	Aksu, Gonzalez
2B	§103	2, 6, 11	Aksu, Gonzalez, H.241, B205/AAPA
2C	§103	10	Aksu, Gonzalez, Fu, B118

¹ References are to pre-AIA law. Because the '841 Patent is transitional, even if post-AIA law applies, that changes only which §102 provision each reference qualifies as prior art. *Platinum Optics Tech. Inc. v. Viavi Sols. Inc.*, IPR2021-00631, Paper 28, 9 n.5 (Oct. 3, 2022).

IV. TECHNOLOGY BACKGROUND

The '841 Patent concerns “deblocking” filters applied during the encoding and/or decoding of digital videos. To store or transmit digital videos, each image making up the video must be “encoded” into binary digital data. Ex-1003 ¶¶40-43; Ex-1001, Fig. 13, 24:11-51. For efficient processing, “the original video frames are split into blocks which are processed relatively independently,” Ex-1001, 1:12-14; *id.*, 24:11-51; Ex-1003 ¶41, as the example below shows:



Ex-1070, 4.

When a decoder subsequently reconstructs the image from the digital data, the reconstructed image may contain “blocking artifacts” due to its block-based encoding. Ex-1001, 1:12-18; Ex-1003 ¶44. For instance, one encoding step is to “quantize” (or compress) each block, which reduces the amount of data that represents the block’s contents by eliminating, e.g., smaller details. Ex-1003 ¶¶41,

44. This can produce perceptible changes in pixel color at the boundaries between blocks that make the image look “blocky.” *Id.* ¶44.

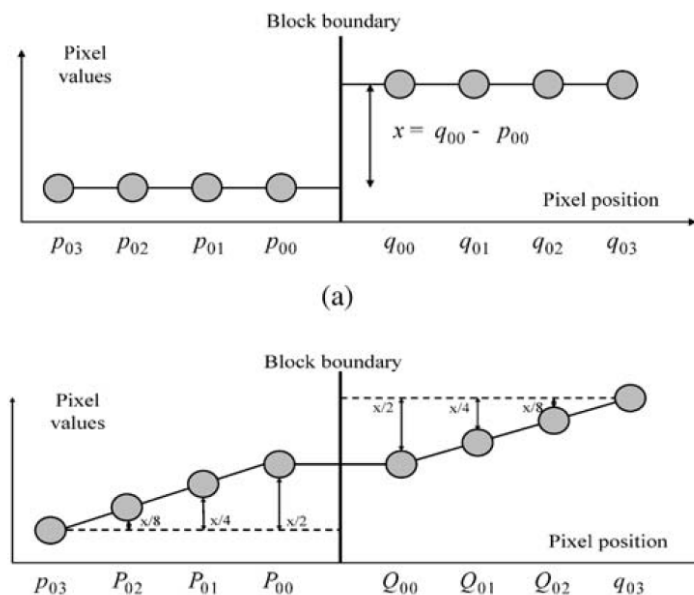
“Deblocking filters are used...to combat blocking artifacts,” Ex-1001, 1:11-12, by blurring the pixels near the boundaries of adjacent blocks, as shown below:



Ex-1052, 1372 (annotations added); Ex-1003 ¶45. Deblocking filters operate on “pixel values”—numerical values that represent the pixel’s color. Ex-1003 ¶46. For instance, in black-and-white images, each pixel can have a value of 0-255, with 0 representing black, 255 representing white, and values in between representing different greys. *Id.* ¶47. For color, each color component (e.g., red, green, and blue) can have a value of 0-255, which combined produce millions of colors. *Id.* ¶¶48-50.

By representing pixel colors as numbers, pixel colors can be changed using “filters” based on mathematical equations.² *Id.* ¶51.

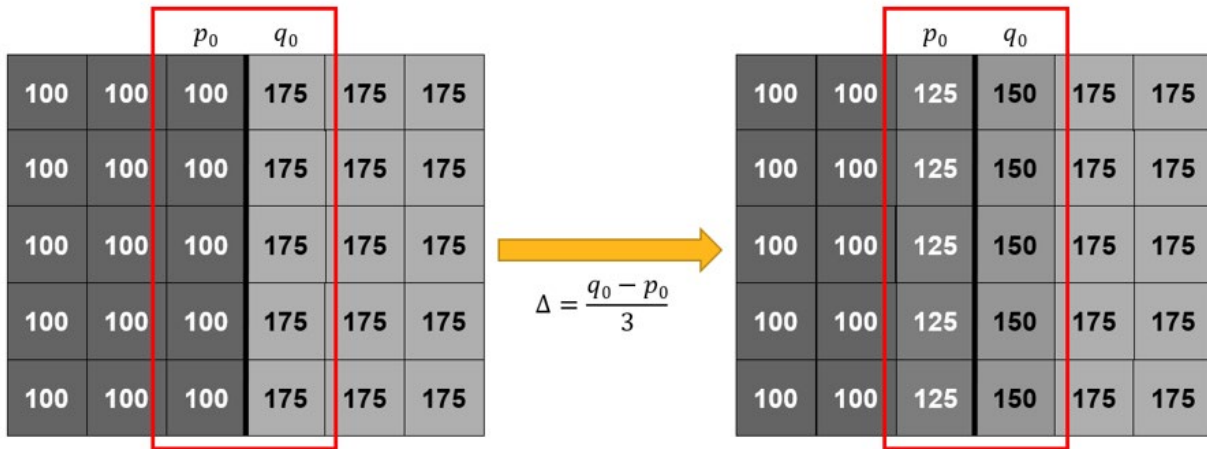
Deblocking filters first decide *whether* to filter a block boundary. *Id.* ¶65. If a decision to filter is made, pixel values nearest the block boundary are then modified using a “filter” equation that works in one of two ways. *Id.* ¶¶52-53, 65. First, a new pixel value can be calculated directly by weighting one or more original-pixel values in a “filter” equation. *Id.* ¶53. Alternatively, an intermediate “offset” can be calculated, which is then used to modify one or more original-pixel value(s), as shown below (the offset shown as a function of “x”):



² For example, “low-pass” filters blur images by reducing color differences between nearby pixels, and “high-pass” filters sharpen images by increasing color differences between nearby pixels. Ex-1003 ¶¶54-57.

Ex-1039, 215; Ex-1003 ¶53.

For instance, if one (“P”) block has pixel values of “100,” and a neighboring (“Q”) block has values of “175,” that produces a perceptible change at the boundary. To blur this boundary, an offset (Δ) can be calculated using pixel values closest to the boundary (e.g., p_0 and q_0), which is then added to the p_0 pixels and subtracted from the q_0 pixels, smoothing the transition from the P to Q block.



Deblocking filters have long been used in the prior art, tending to differ as to: (1) what conditions control whether deblocking is performed; (2) how many pixels are modified during deblocking; and (3) the filter/offset equations used to perform deblocking (how “weak” or “strong” the filter is). *Id.* ¶¶65, 67-94.

V. ’841 PATENT OVERVIEW

The ’841 Patent was filed on August 26, 2013, as a national-stage entry of PCT/SE2011/051526, filed December 16, 2011. It claims priority to provisional application 61/447,862 filed March 1, 2011.

A. Specification

The '841 Patent describes a deblocking filter that first determines *whether* to perform deblocking filtering for two neighboring blocks, and then separately determines *how many* pixels to filter in each block based on each block's contents. Ex-1003 ¶¶98-99. The '841 Patent first describes the prior-art deblocking filter used in the H.264 standard, Ex-1001, 1:19-2:4, Ex-1003 ¶98, and the deblocking filter proposed for H.264's successor: the "HEVC" (or "H.265") standard, Ex-1001, 2:5-3:5; Ex-1003 ¶98. The proposed HEVC standard first decides *whether* to perform deblocking using the following equation:

$$d = |p_{2_2} - 2p_{1_2} + p_{0_2}| + |q_{2_2} - 2q_{1_2} + q_{0_2}| \\ + |p_{2_5} - 2p_{1_5} + p_{0_5}| + |q_{2_5} - 2q_{1_5} + q_{0_5}| < \beta$$

Ex-1001, 2:5-3:5; Ex-1003 ¶92. Using p_{ji} and q_{ji} : j represents the distance of the pixel from the block boundary, and i represents the line of pixels within the block. Ex-1001, 2:12-20. Below shows two blocks (blue, red), with the implicated pixels darkened:

p3 ₀	p2 ₀	p1 ₀	p0 ₀	q0 ₀	q1 ₀	q2 ₀	q3 ₀
p3 ₁	p2 ₁	p1 ₁	p0 ₁	q0 ₁	q1 ₁	q2 ₁	q3 ₁
p3 ₂	p2 ₂	p1 ₂	p0 ₂	q0 ₂	q1 ₂	q2 ₂	q3 ₂
p3 ₃	p2 ₃	p1 ₃	p0 ₃	q0 ₃	q1 ₃	q2 ₃	q3 ₃
p3 ₄	p2 ₄	p1 ₄	p0 ₄	q0 ₄	q1 ₄	q2 ₄	q3 ₄
p3 ₅	p2 ₅	p1 ₅	p0 ₅	q0 ₅	q1 ₅	q2 ₅	q3 ₅
p3 ₆	p2 ₆	p1 ₆	p0 ₆	q0 ₆	q1 ₆	q2 ₆	q3 ₆
p3 ₇	p2 ₇	p1 ₇	p0 ₇	q0 ₇	q1 ₇	q2 ₇	q3 ₇

Ex-1001, Fig. 2A (annotations added); Ex-1003 ¶92. Deblocking is performed when the equation’s output (“d”) is less than a threshold (“β”)—which represents the amount of quantization (compression) applied during encoding. Ex-1001, 2:5-3:5; Ex-1003 ¶92. Depending on a few additional conditions, one of two filters is applied to blur the boundary—a “strong” or “weak” filter. Ex-1001, 2:28-40; Ex-1003 ¶92. The ’841 Patent appears drawn to the weak filter, Ex-1003 ¶103, and the weak filter proposed for HEVC used the following “offset” equation:

$$\Delta = \text{Clip}(-t_c, t_c, (13(q_{0_i} - p_{0_i}) + 4(q_{1_i} - p_{1_i}) - 5(q_{2_i} - p_{2_i}) + 16) \gg 5))$$

Id., 2:44-50. This equation has two parts: (1) the offset expression (red) to calculate the offset; and (2) a “clipping” function (blue) to ensure the offset stays within a prescribed range (between “-t_c” and “t_c”). Ex-1001, 2:58-60; *see* Ex-1003 ¶¶72, 93.

The offset expression (red) can be written in: (1) “bit-shift” form specially intended to run on computer hardware, which always outputs an integer, and (2) division form, which must be rounded to the closest integer. Ex-1003 ¶¶79-84. To convert between the two, the “bit shift” (“>>n”) becomes a denominator of 2ⁿ, and the rounding integer (2ⁿ/2, or “+16” for the draft HEVC standard) is omitted. Ex-1003 ¶¶82-84, 242.

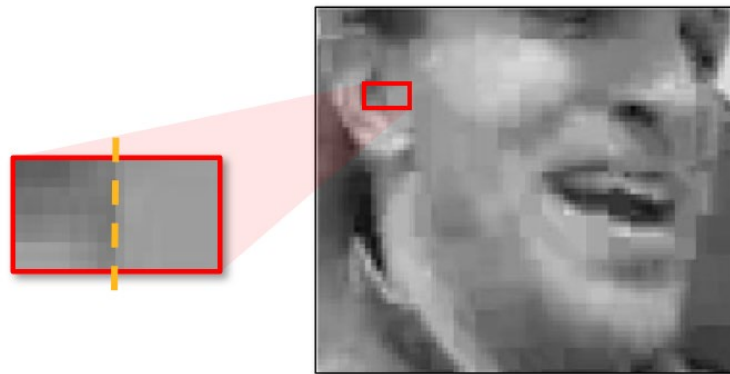
Bit-Shift Form	Division Form
$13(q_{0_i} - p_{0_i}) + 4(q_{1_i} - p_{1_i}) - 5(q_{2_i} - p_{2_i}) + 16 \gg 5$	$\frac{13(q_{0_i} - p_{0_i}) + 4(q_{1_i} - p_{1_i}) - 5(q_{2_i} - p_{2_i})}{32}$

The offset (Δ) is then applied to the “p” and “q” pixels using “filter” equations:

$$\begin{aligned} p0_i &= \text{Clip}_{0-255}(p0_i + \Delta) & q0_i &= \text{Clip}_{0-255}(q0_i - \Delta) \\ p1_i &= \text{Clip}_{0-255}\left(p1_i + \frac{\Delta}{2}\right) & q1_i &= \text{Clip}_{0-255}\left(q1_i - \frac{\Delta}{2}\right) \end{aligned}$$

Ex-1001, 2:50-57. These equations also include a clipping function to ensure each modified-pixel value stays between 0 and 255 (the valid pixel-value range). Ex-1003 ¶¶72, 93. Unlike HEVC’s predecessor H.264, which separately decided *how many* pixels to filter in each block, the draft HEVC standard’s weak filter always filtered two pixels in each block, modifying the pixels closest to the boundary (p0/q0) by the full offset, and the second-closest pixels (p1/q1) by half the offset. Ex-1003 ¶¶87, 93.

According to the ’841 Patent, the draft HEVC standard may filter “local structures” that may be mistaken for blocking artifacts. Ex-1001, 3:6-12; Ex-1003 ¶98. In the image below, the ear canal may fall along a block boundary and be mistaken for a blocking artifact. *Id.*



To address this potential issue, the '841 Patent proposes modifying the draft HEVC standard. First, although the '841 Patent uses a similar “d” equation as the draft HEVC standard (using pixels from both blocks) to decide *whether* to perform deblocking, the '841 Patent also proposes splitting the equation for each block (using pixels from each block, separately) to determine *how many* pixels to filter in each block:

$$d_p = |p2_i - 2p1_i + p0_i| \qquad d_q = |q2_i - 2q1_i + q0_i|$$

Ex-1001, 3:24-50, 6:65-7:20, 8:11-23; Ex-1003 ¶¶99. A lower $d_{p/q}$ value means pixel values near the boundary of that block are not changing significantly, indicating there is likely no detail worth preserving, so more (two) pixels in that block can be filtered. Ex-1001, 9:35-59; Ex-1003 ¶¶101, 61-63. A higher $d_{p/q}$ value means image details may be present, and thus fewer (one) pixels in that block should be filtered. Ex-1001, 9:35-59; Ex-1003 ¶¶101, 61-63. These block-specific, “asymmetric decisions mean[] that the amount of filtering applied to one side of the block boundary can differ from the amount of filtering applied to the other side of the block boundary.” Ex-1001, 4:62-5:2; Ex-1003 ¶¶100-01.

Second, whereas the draft HEVC standard calculates filter-decision values (d) for both blocks using pixels from lines 2 and 5, the '841 Patent states that filter-decision values may be calculated either: (1) line-by-line; or (2) to represent a subset

of lines or the entire block using different lines (e.g., 0 and 3) or a different number of lines (e.g., 0, 2, 3, and 5). Ex-1001, 8:53-9:25, 11:14-12:29; Ex-1003 ¶102.

Finally, the '841 Patent provides different offset equations for filtering. The offset equations for the p0/q0 pixels are provided below in both bit-shift and division form (normalized so both are divided by 16). The red annotations show the differences between the two equations:

Draft HEVC Standard (<i>Id.</i> , 2:47-48)	'841 Patent (<i>Id.</i> , 17:5-6, 18:30-33)
$(13(q_0-p_0)+4(q_1-p_1)-5(q_2-p_2)+16)>>5$	$(9(q_0-p_0)-3(q_1-p_1)+8)>>4$
$\frac{7.5(q_0 - p_0) + 2(q_1 - p_1) - 2.5(q_2 - p_2)}{16}$	$\frac{9(q_0 - p_0) - 3(q_1 - p_1)}{16}$

Ex-1003 ¶103.

B. Prosecution History

The '841 Patent is a national-stage entry of a PCT application. Ex-1002, 93.

The examiner rejected all but one claim—directed to the offset equations that now appear as dependent claim 10, *id.*, 308-10—in view of Bjontegaard (U.S. 2006/0078048, different than Ground 1, Ex-1065) and List (Ex-1024), *id.*, 407-20. The applicant did not materially amend the claims, *id.*, 489-502. and argued that the prior art made a single filter decision *whether* to perform deblocking using pixels on both sides of the block boundary (like the draft HEVC standard), and not

two *separate* decisions to determine *how many* pixels to filter in each block. *Id.*, 503-08.

The examiner agreed—finding applicant’s arguments distinguishing the claims’ “decision to filter a determined number of pixels on one side of a block boundary [as having] no mathematical basis in (no relationship with) the pixels on the other side of the block boundary was persuasive”—and allowed the claims. *Id.*, 518. As discussed below, the prior art disclosed or suggested these exact features of a deblocking filter.

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) would have had a bachelor’s degree in electrical engineering, computer science, or a similar field and two to three years of experience in the design and development of relevant multimedia processing, such as video and/or image coding or compression. Ex-1003 ¶165. Advanced graduate studies with coursework and/or research focusing on topics of video encoding/decoding and processing could substitute for working experience in the field. *Id.*

VII. CLAIM CONSTRUCTION

Claims should be construed “in accordance with the ordinary and customary meaning...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); *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 ’841 Patent. 37 C.F.R. §42.100(b). Nor does Petitioner believe the Challenged Claims need construction to demonstrate unpatentability. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

However, in proceedings before the ITC, the parties there dispute the meaning of several terms of the ’841 Patent. Ex-1026. Only one construction—if adopted—is potentially relevant here. Specifically, the parties dispute what is required of independent claims 1, 5, and 16 to “determin[e] how many pixels” to filter in each block. *Id.*, 6-7.

ITC Respondent Lenovo contends that “determining” how many pixels to filter in one block “has no mathematical basis (no relationship) with the pixel values of the pixels in said neighboring block.” *Id.*, 6-7. Ericsson contends that no construction is necessary, *id.*; Ex-1027, 14-15, and the “plain meaning” does not foreclose some “mathematical basis” existing as long as separate decision values are calculated for each block, Ex-1027, 16-19.

Although Petitioner agrees with Lenovo’s narrower construction because it is consistent with the intrinsic record, *see, e.g.*, Ex-1002, 518, applying Ericsson’s broader constructions would only open the ’841 Patent to additional prior art. For these reasons, Ground 1 adopts Ericsson’s broader construction, and Ground 2

adopts Lenovo's narrower construction. Either way, the Challenged Claims are unpatentable as obvious.

VIII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1A: Obviousness over Bjontegaard and Bjontegaard's Code

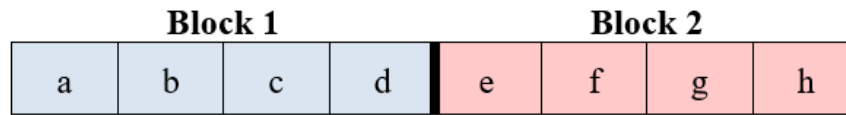
If the “determining” step does not preclude a mathematical basis in pixel values from a neighboring block, as Ericsson argues in the co-pending ITC proceeding, then Bjontegaard and Bjontegaard's Code (and other prior art; Ground 1B) renders the Challenged Claims obvious.

1. Bjontegaard and Bjontegaard's Code

Bjontegaard is a proposal submitted to the ITU Video Coding Experts Group (“VCEG”). Ex-1003 ¶¶109, 111. Bjontegaard was published on May 9, 2000, Ex-1082, and constitutes prior art under §102(b).³

Bjontegaard discloses a “coding model” for a draft standard (what became H.264) that includes a “deblocking filter.” Bjontegaard, 1, 4, 18. Bjontegaard describes the deblocking filter using pixels a-h, spanning two blocks:

³ Bjontegaard was publicly accessible, Ex-1005 ¶¶51-70, and those interested knew where to find it, Ex-1113, 8; Ex-1114, 4; Ex-1115 ¶2; Ex-1116 ¶6.



Id., 19; Ex-1003 ¶¶112-16.

Bjontegaard performs a series of calculations to decide how many pixels to filter in each block:

$$S0 = \mathbf{abs}(4(c-f)+12(e-d)) + \mathbf{abs}(9(c-d-e+f))$$

$$S0_1 = 12\mathbf{abs}(a-2b+c)$$

$$S1_1 = \mathbf{abs}(4(a-d)+12(c-b)) + \mathbf{abs}(9(a-b-c+d))$$

$$S0_2 = 12\mathbf{abs}(e-2f+g)$$

$$S1_2 = \mathbf{abs}(4(e-h)+12(g-f)) + \mathbf{abs}(9(e-f-g+h))$$

Bjontegaard, 19; Ex-1003 ¶117. S0 uses pixels from both blocks, and S0₁/S1₁ and S0₂/S1₂ use pixels only from blocks 1 and 2, respectively. *Id.*

Bjontegaard uses S0₁/S1₁ and S0₂/S1₂ to calculate an initial “activity” value (“Act”) for each block, beginning with block 1, based on the following conditions:

- Act₁ is initially set to 1
- If S0₁ < Lim(qp), Act₁ is incremented by 1
- If S1₁ < Lim(qp), Act₁ is incremented by 1

Bjontegaard, 19; Ex-1003 ¶¶118-19. Act₁ can have a value between 1 and 3 depending on how many conditions are met. Ex-1003 ¶119. This “same procedure is made for Act₂” in block 2, using S0₂/S1₂. Bjontegaard, 19; Ex-1003 ¶120.

Act_1 and Act_2 are then *separately* compared to a measure of *overall* activity across blocks 1 and 2 (“ $Act_{overall}$ ”). Bjontegaard, 19; Ex-1003 ¶121. $Act_{overall}$ can have a value between 0 and 3. Ex-1003 ¶118. Whichever value of $Act_{overall}$ compared to Act_1 or Act_2 is lower is used as the final activity value of each block (“ Act_{11} ” and “ Act_{22} ,” respectively). Bjontegaard, 19; Ex-1003 ¶121.

Bjontegaard then discloses filtering the pixel closest to the block boundary in each block (d, e) based on Act_{11} and Act_{22} :

$$Act_{edge} = (Act_{11} + Act_{22})/2$$

$$\text{If } Act_{edge} \text{ equals 1 or 2:} \quad Diff = (3(c - f) - 8(d - e))/16$$

$$\text{If } Act_{edge} \text{ equals 3:} \quad Diff = (b + c - 8(d - e) - f - g)/16$$

$$d_{modified} = d + Diff \text{ (clipped to stay in the range 0-255)}$$

$$e_{modified} = e - Diff \text{ (clipped to stay in the range 0-255)}$$

Bjontegaard, 20 (§4.5.5); Ex-1003 ¶122.

Bjontegaard then decides whether to filter a second pixel in each block, separately. Bjontegaard, 20 (§§4.5.6.1, 4.5.6.2); Ex-1003 ¶¶123-24. If Act_{11} equals 2 or 3, pixel c in block 1 is filtered:

$$\text{If } Act_{11} \text{ equals 2:} \quad Diff = 5(b - 2c + d)/16$$

$$\text{If } Act_{11} \text{ equals 3:} \quad Diff = 3(a + b - 4c + d + e)/16$$

$$C_{modified} = c + Diff \text{ (clipped to stay in the range 0-255)}$$

Bjontegaard, 20; Ex-1003 ¶123. If Act_{22} equals 2 or 3, pixel f in block 2 is filtered.
Bjontegaard, 20; Ex-1003 ¶124.

Bjontegaard also submitted computer-executable code (“Bjontegaard’s Code”) incorporating Bjontegaard’s deblocking filter into an encoder and decoder. Ex-1010-1015. Bjontegaard’s Code also constitutes prior art under §102(b).⁴ *Id.* Because they were published at the same time, by the same person, as part of one standards proposal, a POSITA would have understood that Bjontegaard and Bjontegaard’s Code were intended to be read together. Ex-1003 ¶¶110, 125-27; *Nvidia Corp. v. Advanced Cluster Sys., Inc.*, IPR2021-00075, Paper 9, 21, 26, 39-42 (May 5, 2021) (endorsing reliance on accompanying code to support paper disclosures).

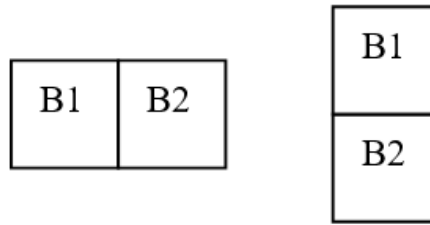
2. Claim 1

a. 1[pre]

If limiting, Bjontegaard discloses 1[pre].

Bjontegaard discloses a method of “[f]iltering of pixels on both sides of a 4x4 block boundary” in a video frame, Bjontegaard, 4, 19, shown below:

⁴ Bjontegaard’s Code was publicly accessible, Ex-1005 ¶¶51-70.



Bjontegaard, 18. Bjontegaard explains that “pixels [a-h] belong to two neighbouring blocks (1 and 2). There is a block boundary between pixels d [and] e.” Bjontegaard, 19.



Ex-1003 ¶¶172-73. Bjontegaard provides equations that operate on the pixels, *id.* ($S0_n$ and $S1_n$), meaning each pixel has a pixel value. Ex-1003 ¶173.

b. 1[a]

Bjontegaard with Bjontegaard’s Code teaches 1[a].

Bjontegaard discloses calculating $S0_1 = 12 \text{abs}(a - 2b + c)$ for block 1. Bjontegaard, 19. “ $S0_1$ ” is a “first filter decision value” because its value is used to determine how many pixels to filter. §VIII.A.1; Ex-1003 ¶¶174-75. $S0_1$ is also “based on” the equation “ $|p_{2i} - 2p_{1i} + p_{0i}|$,” as claimed, because: (1) $S0_1$ uses three pixel values (a, b, c) with the same coefficients ($[1 \ -2 \ 1]$) as the claimed equation; and (2) the vertical bars ($| \dots |$) are a well-known symbol for the “absolute value,” like $S0_1$ ’s “abs()” function. Ex-1003 ¶176; Bjontegaard, 19.

A POSITA would have understood that Bjontegaard discloses calculating $S0_1$ using the three pixels closest to the block boundary, even though Bjontegaard recites pixels a, b, c instead of b, c, d:



Ex-1003 ¶¶177-83.

Deblocking filters conventionally evaluate the pixel values *nearest* the block boundary because that is where blocking artifacts appear, and filter the pixels closest to the boundary to reduce them. *Id.* ¶178. Indeed, for the corresponding equation in block 2 ($S0_2 = 12\text{abs}(e - 2f + g)$), Bjontegaard uses the three pixels closest to the boundary:



Bjontegaard, 20; Ex-1003 ¶179. When Bjontegaard modifies second-pixel “f” in block 2 (based on $S0_2$), Bjontegaard uses the same three pixels as $S0_2$: $\text{Diff}_{[2]} = 5(e - 2f + g)/16$. Bjontegaard, 20 (“Diff” subscripts added to identify block); Ex-1003 ¶180. But when modifying corresponding second-pixel “c” in block 1, Bjontegaard uses the pixels closest to the boundary— $\text{Diff}_{[1]} = 5(b - 2c + d)$ —unlike $S0_1$. Bjontegaard, 20; Ex-1003 ¶180. A POSITA would have expected that the same pixels should be used for both $S0_1$ and $\text{Diff}_{[1]}$, like $S0_2$ and $\text{Diff}_{[2]}$. *Id.* Finally, Bjontegaard’s Code uses

pixels b, c, d to calculate $S0_1$, rather than pixels a, b, c. *See, e.g.*, Ex-1012, 10; Ex-1003 ¶¶125, 181-82. A POSITA therefore would have read Bjontegaard as: $S0_1=12\text{abs}(b-2c+d)$, matching the location of pixels $p2_i$, $p1_i$, and $p0_i$, as claimed. Ex-1003 ¶183.

c. $1[b]$

Bjontegaard discloses or suggests $1[b]$.

Bjontegaard discloses calculating $S0_2=12\text{abs}(e-2f+g)$ for neighboring block 2. Bjontegaard, 19. “ $S0_2$ ” is a “second filter decision value” because its value is used to determine how many pixels to filter. §VIII.A.1; Ex-1003 ¶¶184-85. $S0_2$ is also “based on” the equation “ $|q2_i-2q1_i+q0_i|$ ” because it takes the absolute value, and uses the three pixels closest to the block boundary (e, f, g, corresponding to pixels $q0_i$, $q1_i$, $q2_i$) with the same $[1 \ -2 \ 1]$ coefficients, as claimed. Ex-1003 ¶185.

d. $1[c]$

Bjontegaard discloses or suggests $1[c]$.

Bjontegaard discloses filtering pixel d (closest to the boundary) in block 1. Bjontegaard, 20 (§4.5.5); Ex-1003 ¶¶186-87. A decision is then made whether to also filter second-closest pixel c in block 1. Bjontegaard, 20 (§4.5.6.1); Ex-1003 ¶187. Thus, Bjontegaard determines whether to filter one or two pixels in a line of pixels relative to the boundary in block 1. Ex-1003 ¶187.

Pixel c is filtered only if Act_{11} equals 2 or 3:

4.5.6.1 Modification if c in block 1

Initially set $\text{Diff} = 0$

If Act_{11} equals 2: $\text{Diff} = 5(b - 2c + d)/16$

If Act_{11} equals 3: $\text{Diff} = 3(a + b - 4c + d + e)/16$

$\text{Diff} = \max(-\text{Clip}(\text{qp}, \text{Strength}_1), \min(\text{Clip}(\text{qp}, \text{Strength}_1), \text{Diff}))$

$\text{C}_{\text{modified}} = c + \text{Diff}$ (clipped to stay in the range 0-255)

Bjontegaard, 20; Ex-1003 ¶188. That is because the offset (“ $\text{Diff}_{[1]}$ ”) equals zero by default, meaning pixel c is not modified, and stays zero if Act_{11} is not greater than 1. Bjontegaard, 20; Ex-1003 ¶188.

To determine the value of Act_{11} , first, a preliminary “ Act_1 ” is set to 1, and then increased using the following conditions:

- If $S0_1 < \text{Lim}(\text{qp})$, Act_1 is incremented by 1
- If $S1_1 < \text{Lim}(\text{qp})$, Act_1 is incremented by 1

Bjontegaard, 19; Ex-1003 ¶189. Accordingly, Act_1 will always be greater than 1 if $S0_1$ ($12\text{abs}(b-2c+d)$) is less than a first threshold ($\text{Lim}(\text{qp})$). Ex-1003 ¶189. Act_{11} is then set equal to the smaller of Act_1 and $\text{Act}_{\text{overall}}$:

$$\text{Act}_{11} = \min(\text{Act}_{\text{overall}}, \text{Act}_1),$$

Bjontegaard, 19; Ex-1003 ¶190. When $\text{Act}_1 \leq \text{Act}_{\text{overall}}$, however, Act_{11} simply equals Act_1 . Bjontegaard, 19; Ex-1003 ¶191. In this circumstance, Act_1 determines whether second-pixel c is filtered because, if $S0_1 < \text{Lim}(\text{qp})$, Act_1 will always be greater than 1, and pixel c will be filtered. Ex-1003 ¶191. But if $S0_1 \geq \text{Lim}(\text{qp})$ —and the second

condition does not increment Act_1 either—then Act_1 will equal 1, Act_{11} will not be greater than 1, and second-pixel c will not be filtered. Bjontegaard, 19; Ex-1003 ¶191.

Because Act_{11} equals Act_1 whenever $Act_1 \leq Act_{overall}$, and the value of Act_1 depends on comparing first filter-decision value $S0_1$ to threshold $Lim(qp)$, Bjontegaard discloses determining how many pixels to filter (one or two pixels) in block 1 by comparing first-filter decision value $S0_1$ to first-threshold $Lim(qp)$. Bjontegaard, 19-20; Ex-1003 ¶192; *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”).

e. 1[d]

Bjontegaard discloses or suggests 1[d].

Bjontegaard discloses the same procedure used for block 1 is followed to determine whether to filter second-closest pixel f in neighboring block 2, except using variables $S0_2$, $S1_2$, Act_2 , and Act_{22} associated with block 2 instead. Bjontegaard, 19-20; Ex-1003 ¶¶193-94. Bjontegaard therefore discloses determining how many pixels to filter (one or two) in neighboring block 2 by comparing second filter-decision value $S0_2$ to second-threshold $Lim(qp)$. Bjontegaard, 19-20; §VIII.A.2.d; Ex-1003 ¶195.

If the “second threshold” must have a different value than the “first threshold,” it would have been obvious to select separate threshold values for blocks 1 and 2. Ex-1003 ¶¶196-202. Bjontegaard discloses that threshold $\text{Lim}(\text{qp})$ is a function of the blocks’ “qp” (or “quantization parameter”), and that the “qp of the two blocks may differ.” Bjontegaard, 19; Ex-1003 ¶196. Although Bjontegaard defines threshold $\text{Lim}(\text{qp})$ for both blocks using the “maximum qp for the two blocks,” Bjontegaard, 19, this was a design choice for efficiency. Ex-1003 ¶¶199-200. But when the threshold is being compared to equations that use pixels only from one block, a POSITA would have been motivated to use a threshold based on the qp only of that block (i.e., using $\text{Lim}(\text{qp}_1)$ for S_{01}/S_{11} ; using $\text{Lim}(\text{qp}_2)$ for S_{02}/S_{12}) because it would provide a more-representative measure of how much filtering is required in each block, since the qp represents how much compression each block underwent. Ex-1003 ¶¶197-98. Others had successfully implemented block-specific thresholds to make filter decisions. Ex-1003 ¶¶201-02; Ex-1064, Fig. 17, ¶¶95, 95.

3. Claim 3

a. 3[a]

See §VIII.A.2.d (disclosing “determining” step); Ex-1003 ¶203.

b. 3[b]

Bjontegaard discloses or suggests comparing first filter-decision value $S0_1$ to first-threshold $\text{Lim}(\text{qp})$ (or $\text{Lim}(\text{qp}_1)$). §§VIII.A.2.d, VIII.A.2.e; Ex-1003 ¶204.

c. 3[c]

Bjontegaard discloses or suggests 3[c].

Bjontegaard determines to filter two pixels (c, d) in a line of block 1 if first filter-decision value $S0_1$ is below first-threshold $\text{Lim}(\text{qp})$ because that will increment Act_1 , causing second pixel c to be filtered (when $\text{Act}_1 \leq \text{Act}_{\text{overall}}$). §VIII.A.2.d; Ex-1003 ¶205; *Unwired*, 841 F.3d at 1002.

d. 3[d]

Bjontegaard discloses or suggests 3[d].

Bjontegaard determines to filter one pixel (d) in a line of block 1 if first filter-decision value $S0_1$ equals or exceeds first-threshold $\text{Lim}(\text{qp})$, because that will not increment Act_1 , and second pixel c will not be filtered (when $\text{Act}_1 \leq \text{Act}_{\text{overall}}$ and $S1_1 \geq \text{Lim}(\text{qp})$). §VIII.A.2.d; Ex-1003 ¶206; *Unwired*, 841 F.3d at 1002.

e. 3[e]

See §VIII.A.2.e (disclosing “determining” step); Ex-1003 ¶207.

f. 3[f]

Bjontegaard discloses or suggests comparing second filter-decision value $S0_2$ to second-threshold $\text{Lim}(\text{qp})$ (or $\text{Lim}(\text{qp}_2)$). §VIII.A.2.e; Ex-1003 ¶208.

g. 3[g]

Bjontegaard discloses or suggests 3[g].

Bjontegaard determines to filter two pixels (e, f) in a line in neighboring block 2 if second filter-decision value $S0_2$ is below second-threshold $\text{Lim}(\text{qp})$, because that will increment Act_2 , causing second-pixel f to be filtered (when $\text{Act}_2 \leq \text{Act}_{\text{overall}}$). §§VIII.A.2.d, VIII.A.2.e; Ex-1003 ¶209; *Unwired*, 841 F.3d at 1002.

h. 3[h]

Bjontegaard discloses or suggests 3[h].

Bjontegaard determines to filter one pixel (e) in a line in neighboring block 2 if second filter-decision value $S0_2$ equals or exceeds second-threshold $\text{Lim}(\text{qp})$ because that will not increment Act_2 , and thus second-pixel f will not be filtered (when $\text{Act}_2 \leq \text{Act}_{\text{overall}}$ and $S1_2 \geq \text{Lim}(\text{qp})$). §§VIII.A.2.d, VIII.A.2.e; Ex-1003 ¶210; *Unwired*, 841 F.3d at 1002.

4. Claim 5

a. 5[pre]

If limiting, Bjontegaard with Bjontegaard's Code, teaches 5[pre].

Bjontegaard discloses a “coding model” for video images, Bjontegaard, 4, which would obviously operate on a device (e.g., computer), as confirmed by Bjontegaard’s Code, Ex-1010-1015, and as was conventional, Ex-1040, 1-5 (noting importance of video encoders/decoders to consumer devices); Ex-1061, 9:49-57, 10:9-13; Ex-1062, 18:17-18, 22:23-24; Ex-1003 ¶¶211-15; *Samsung Elecs. v. Lynk Labs, Inc.*, IPR2022-00100, Paper 30, 23-24, 34-35 (June 7, 2023) (conventional teachings supported by background references obvious).

b. 5[a]

Bjontegaard with Bjontegaard’s Code teaches 5[a], §§VIII.A.2.b, VIII.A.4.a, and that the prescribed calculations would be performed on a device’s processor with calculators, Ex-1012, 10 (computer code for calculator); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶216.

c. 5[b]

See §§VIII.A.2.c, VIII.A.4.a, VIII.A.4.b; Ex-1003 ¶217.

d. 5[c]

Bjontegaard with Bjontegaard’s Code teaches 5[c], §VIII.A.2.d, VIII.A.4.a, and that the prescribed determinations would be performed on a device’s processor with determiners, Ex-1012, 11 (computer code for determiner); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶218.

e. 5[d]

See §§VIII.A.2.e, VIII.A.4.a, VIII.A.4.d; Ex-1003 ¶219.

f. 5[e]

Bjontegaard with Bjontegaard's Code teaches 5[e], §VIII.A.2.d, VIII.A.3.b, VIII.A.4.a, and also that the prescribed comparisons would be performed on a device's processor with comparators, Ex-1012, 10-11 (computer code for comparator); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶220.

g. 5[f]

See §§VIII.A.2.e, VIII.A.3.f, VIII.A.4.a, VIII.A.4.f; Ex-1003 ¶221.

5. Claim 7

a. 7[a]

See §§VIII.A.3.c, VIII.A.4.d, VIII.A.4.f; Ex-1003 ¶222.

b. 7[b]

See §§VIII.A.3.d, VIII.A.4.d, VIII.A.4.f; Ex-1003 ¶223.

c. 7[c]

See §§VIII.A.3.g, VIII.A.4.e, VIII.A.4.g; Ex-1003 ¶224.

d. 7[d]

See §§VIII.A.3.h, VIII.A.4.e, VIII.A.4.g; Ex-1003 ¶225.

6. Claim 9

a. 9[a]

Bjontegaard with Bjontegaard's Code teaches 9[a].

Bjontegaard discloses that first-threshold $\text{Lim}(\text{qp})$ (for $S0_1$) depends on the quantization parameter ("qp") of the blocks, providing a table to determine $\text{Lim}(\text{qp})$ based on qp:

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Lim	7	8	9	10	11	12	14	15	17	20	22	25	28	31	35	39	44	49	55	62	69	78	88	98	110	124	139	156	175	197	221	248

Bjontegaard, 9, 19; Ex-1003 ¶¶226-27.

Because Bjontegaard discloses that $\text{Lim}(\text{qp})$ is based on the maximum qp of the two blocks, Bjontegaard, 19, Bjontegaard determines the first threshold based on the qp of block 1 when $\text{qp}_1 > \text{qp}_2$, Ex-1003 ¶227. But it also would have been obvious to determine $\text{Lim}(\text{qp})$ based on the qp of each block, separately. Ex-1003 ¶228; §VIII.A.2.e.

The prescribed determination would obviously be performed on a device's processor with threshold determiners. §VIII.A.4.a; Ex-1012, 10 (computer code for threshold determiner); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶229.

b. 9[b]

Bjontegaard with Bjontegaard's Code teaches 9[b].

Bjontegaard discloses that $\text{Lim}(\text{qp})$ is also used as a second threshold (for S_0), and that it is “based on” the qp of the two blocks by determining which is maximum. Bjontegaard, 19; Ex-1003 ¶¶230-31. However, determining separate thresholds based on the qp of each block, separately, was obvious, §VIII.A.2.e, as was determining the second threshold using a threshold determiner, §VIII.A.6.a; Ex-1003 ¶232.

7. Claim 10

a. 10[a]

Bjontegaard in view of Bjontegaard’s Code and a POSITA’s knowledge teaches 10[a].

Bjontegaard discloses a first-offset equation: $\text{Diff} = (3(c-f) - 8(d-e))/16$, Bjontegaard, 20, re-written using “p” and “q” for a line “j” of pixels:

$$\Delta = \frac{3(p_{1j} - q_{1j}) - 8(p_{0j} - q_{0j})}{16} = \frac{8(q_{0j} - p_{0j}) - 3(q_{1j} - p_{1j})}{16}$$

Ex-1003 ¶¶233-34. The only difference between Bjontegaard and the claimed equation is that the first coefficient is an “8” instead of “9.” Ex-1003 ¶234.

Which coefficient to apply to each set of pixel differences is nothing more than a design choice resulting from routine experimentation. Ex-1003 ¶¶235-53. A POSITA would have understood that the coefficients affect filtering, and the art was replete with deblocking filters with first-offset equations like Bjontegaard’s that demonstrate a workable range of coefficients. Ex-1003 ¶235. Some examples

(normalized to be divided by 16) are provided below, with their differences highlighted red:

'841 Patent	$\frac{9(q_0 - p_0) - 3(q_1 - p_1)}{16}$
Bjontegaard	$\frac{8(q_0 - p_0) - 3(q_1 - p_1)}{16}$
B-118	$\frac{8(q_0 - p_0) - 2(q_1 - p_1)}{16}$
Fu	$\frac{6(q_0 - p_0) - 2(q_1 - p_1)}{16}$
H.264, List	$\frac{8(q_0 - p_0) - 2(q_1 - p_1)}{16}$
VCEG-M21	$\frac{8(q_0 - p_0) - 2.5(q_1 - p_1)}{16}$
H.241 Bjontegaard '048	$\frac{8(q_0 - p_0) + 4(q_1 - p_1) - 4(q_2 - p_2)}{16}$
B205	$\frac{7.5(q_0 - p_0) + 2(q_1 - p_1) - 2.5(q_2 - p_2)}{16}$

Ex-1003 ¶¶236-44; Bjontegaard, 20; B118, 48; Fu, 8; Ex-1037, 207; Ex-1024, 617; Ex-1069, 2; H.241, 29; Ex-1065 ¶47; B205, 131.

A POSITA would have known that these offset equations differed only as to two variables: (1) the coefficient applied to each pixel difference **X**(q0-p0), **Y**(q1-p1), **Z**(q2-p2); and (2) the number of pixel differences used. Ex-1003 ¶235. It was also well-understood that these variables control the filter's frequency response, and thus how much offset would be applied to pixels p0/q0 to blur blocking artifacts without reducing overall image quality. *Id.*; Gonzales, 153, 273.

Because of the prevalence of first-offset equations modifying these two variables, a POSITA would have understood that they were points of routine experimentation. Ex-1003 ¶¶235-45, 252, 263. But a POSITA would have further observed that several design constraints limited the first-offset design:

1. The magnitude of the coefficient applied to $X(q_0-p_0)$ is typically 2-4 times larger than the coefficient applied to $Y(q_1-p_1)$ and, if included, $Z(q_2-p_2)$. Ex-1003 ¶246. This ratio controls the “width” of the filter. *Id.*; Gonzales, 153, 273; Ex-1072, 7.
2. Summing the coefficients in the numerator, and dividing by the denominator, produces a fraction between $\frac{1}{4}$ and $\frac{1}{2}$, meaning the total offset would be approximately $\frac{1}{4}$ to $\frac{1}{2}$ the pixel-value differences across the boundary. Ex-1003 ¶247. This determines how much offset is applied and ensures pixels are not over-filtered. *Id.*; Ex-1039, 215.
3. The “window” used to calculate the offset is conventionally between four (q_0-p_0 ; q_1-p_1) and six (q_0-p_0 ; q_1-p_1 ; q_2-p_2) pixels. Ex-1003 ¶248. Using more than six pixels increases computational complexity, and risks inducing “ringing” artifacts into the filtered image. *Id.*; Ex-1070, 3.
4. The “DC” gain of the offset should equal 1, meaning that applying a “flat” signal to the offset equation produces a value of zero. Ex-1003 ¶249; Gonzalez, 153-54; Ex-1072, 7. The numerator coefficients must therefore sum

to zero. Gonzalez, 696, 699. For Bjontegaard: $\frac{8(q_0-p_0)-3(q_1-p_1)}{16} =$

$\frac{8q_0-8p_0-3q_1+3p_1}{16}$, thus $8-8-3+3=0$. Ex-1003 ¶243. And when the offset is

applied to the original pixel value ($p_0'=p_0+\text{offset}$), summing the coefficients and dividing by the denominator must equal 1. Gonzalez, 153-54. For

Bjontegaard: $p_0' = p_0 + \frac{8(q_0-p_0)-3(q_1-p_1)}{16} = \frac{16p_0}{16} + \frac{8q_0-8p_0-3q_1+3p_1}{16} =$

$\frac{8q_0+8p_0-3q_1+3p_1}{16}$, thus: $\frac{8+8-3+3}{16} = \frac{16}{16} = 1$. Ex-1003 ¶244. Equations that use

pixel differences will always meet this requirement, but it constrains the design of non-difference-based offset equations (like p_1/q_1 offsets). *Id.* ¶249.

5. The value of the offset's right-shift is typically between 3 and 6 (i.e., divided by 8, 16, 32, or 64). *Id.* ¶250. Using a larger right-shift allows for more precise filters because more integer coefficients are available for the numerator, but increases computational complexity. *Id.*
6. Integer coefficients (not decimals) are used to reduce computational complexity. Ex-1003 ¶251; *see* Ex-1001, 18:26-28.

A POSITA would have experimented within these known constraints to optimize Bjontegaard's offset equation. And "where the general conditions of a claim are disclosed in the prior art, it is not inventive to discover the optimum or workable ranges by routine experimentation." *In re Aller*, 220 F.2d 454, 456 (CCPA 1955); *Sony Corp. v. Fujifilm Corp.*, IPR2017-00800, Paper 14 (Aug. 18, 2017)

(obvious to experiment with three variables according to “Wallace’s Law” to design a magnetic recording medium); *Google LLC v. Singular Comput. LLC*, IPR2021-00165, Paper 57 (May 11, 2022) (obvious to experiment with the number of “mantissa bits” to balance power consumption and computational precision); *LG Elecs. Inc. v. Immervision, Inc.*, IPR2020-00195, Paper 6 (May 13, 2020) (obvious to experiment with various lens-design parameters to achieve a “±10%” divergence of certain wavelengths to balance tradeoffs in image quality).

A POSITA experimenting with Bjontegaard’s coefficients (8 and 3) would have appreciated that coefficients 9 and 3 are so close that they would exhibit very similar filtering properties, but that coefficients 9 and 3 would generally provide a *minor* increase in filtering strength, if desired, while still meeting all of the other offset-equation considerations (ratio of 3, approximate offset of $\frac{3}{8}$, four pixels, DC gain of 1). Ex-1003 ¶253; *Friskit, Inc. v. Real Networks, Inc.*, 306 F. App’x 610, 615 (Fed. Cir. 2009) (noting “insignificant differences” between prior art and claims not patentable). Such a modification would not produce any unexpected results, Ex-1003 ¶¶255-62, and even the ’841 Patent recognized that the weights were not critical, as “[d]ifferent such functions ... can be used” in the deblocking filter. Ex-1001, 17:1-8, 18:23-24.

The prescribed calculations would obviously be performed on a device's processor with offset calculators. §VIII.A.4.a; Ex-1012, 11 (computer code for offset calculator); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶254.

b. 10[b]

Bjontegaard with Bjontegaard's Code teaches 10[b].

Bjontegaard discloses " $d_{\text{modified}}=d+\text{Diff}$," meaning the pixel closest to the boundary (d) in block 1 is modified by adding the first offset (Diff). Bjontegaard, 20; Ex-1003 ¶¶264-65.

The prescribed modifications would obviously be performed on a device's processor with pixel modifiers. §VIII.A.4.a; Ex-1012, 11 (computer code for pixel modifier); Ex-1001, 23:13-21 (processor implementation); Ex-1003 ¶266.

c. 10[c]

Bjontegaard with Bjontegaard's Code teaches 10[c].

Bjontegaard discloses " $e_{\text{modified}}=e-\text{Diff}$," meaning the pixel closest to the boundary (e) in neighboring block 2 is modified by subtracting the first offset (Diff). Bjontegaard, 20; Ex-1003 ¶¶267-69; §§VIII.A.4.a, VIII.A.7.b.

d. 10[d]

Bjontegaard in view of Bjontegaard's Code and a POSITA's knowledge of the art, teaches 10[d].

Bjontegaard discloses a second-offset equation for block 1: $\text{Diff}_{[1]} = 5(b - 2c + d)/16$, Bjontegaard, 20. However, a POSITA would have read Bjontegaard's equation as: $\text{Diff}_{[1]} = 5(b - 2c + d_{\text{modified}})/16$, or $\text{Diff}_{[1]} = 5(b - 2c + d + \Delta)/16$ (since $d_{\text{modified}} = d + \Delta$), because Bjontegaard discloses a step-by-step procedure that modifies pixel d *before* filtering other pixels, and Bjontegaard's Code uses d_{modified} in the second-offset equation, Ex-1003 ¶¶270-74; Ex-1012, 11. Alternatively, a POSITA would have been motivated to use d_{modified} to ensure the second offset is related to the first offset to avoid inducing a discontinuity into the filtered block. Ex-1003 ¶275. Many prior-art deblocking filters used the first offset in the second-offset equation to ensure a smooth transition at the block boundary. *Id.*; H.241, 29 (applying $\Delta/2$ for second offset); B205, 131 (same); Ex-1039, 215 (applying $\Delta/2$, $\Delta/4$, $\Delta/8$ when moving further from boundary).

Accordingly, Bjontegaard's equation would be rewritten (using p):

$$\Delta_{p1} = \frac{5(p2 - 2p1 + p0)}{16} + \frac{5\Delta_{p0}}{16}$$

Ex-1003 ¶276. A POSITA would have understood that Bjontegaard's equation has two components, just like the claimed equation: (1) a high-pass filter component

$(\frac{5(p_2 - 2p_1 + p_0)}{16})$; and (2) a first-offset component $(\frac{5\Delta}{16})$. Ex-1003 ¶277. The only difference is the coefficient assigned to each component:

Claimed Equation	Bjontegaard's Equation
$\Delta = \frac{1(p_2 - 2p_1 + p_0)}{4} + \frac{1\Delta}{2}$	$\Delta = \frac{5(p_2 - 2p_1 + p_0)}{16} + \frac{5\Delta}{16}$

Id.

A POSITA would have understood that, like the first-offset equation, these coefficients were known result-effective variables that controlled the frequency response of the filter, and thus how much pixel p_1 would be modified. Ex-1003 ¶¶278-88. Weighting the high-pass component—intended to target high frequencies for additional filtering—by $\frac{1}{4}$ (or $4/16$) instead of $5/16$ would mean *slightly* less high-frequency blurring would be factored into the second offset. *Id.* ¶284. Weighting the first-offset component—intended to blur the second pixel as a function of the first offset—by $\frac{1}{2}$ (or $8/16$) instead of $5/16$ would mean *slightly* more blurring consistent with the first offset would be factored into the second offset. *Id.* ¶285.

A POSITA would have understood that these coefficients were subject to routine experimentation and optimization because others in the art had similarly combined these two components and applied different coefficients. Ex-1003 ¶¶279-

82; Fu, 8:17 (disclosing equivalent of $\Delta_{p1} = \frac{1(p2-3p1+3p0-q0)}{8} + \frac{1\Delta}{2}$); B118, 48 (disclosing equivalent of $\Delta_{p1} = \frac{1(p2-2p1+p0)}{2} + \frac{1\Delta}{2}$). The inventors also acknowledged that different coefficients could be used depending on how much filtering is desired. Ex-1063, 15:42-45 (disclosing $\frac{(p2-2p1+p0)}{2} + \frac{\Delta}{2}$ *or* $\frac{(p2-2p1+p0)}{4} + \frac{\Delta}{2}$); Ex-1001, 18:23-24; Ex-1003 ¶287.

Accordingly, modifying Bjontegaard's coefficients to apply $\frac{1}{4}$ to the high-pass component, and $\frac{1}{2}$ to the first-offset component was obvious, *Friskit*, 306 F. App'x at 615, as was calculating the offset on a device's processor with calculators, §§VIII.A.4.a, VIII.A.7.a; Ex-1003 ¶289.

e. 10[e]

Bjontegaard with Bjontegaard's Code teaches 10[e].

Bjontegaard discloses filtering third-pixel c (second closest to the boundary in block 1) when $S0_1$ is below first-threshold $\text{Lim}(qp)$. §§VIII.A.3.b-VIII.A.3.d.

Bjontegaard also discloses " $c_{\text{modified}} = c + \text{Diff}_{[1]}$," meaning pixel c in block 1 is modified by adding the second offset ($\text{Diff}_{[1]}$). Bjontegaard, 20; Ex-1003 ¶¶290-92.

The prescribed modifications would obviously be performed on a device's processor with pixel modifiers. §§VIII.A.4.a, VIII.A.7.b; Ex-1003 ¶293.

f. 10[f]

Bjontegaard with Bjontegaard's Code teaches 10[f].

Bjontegaard discloses a third-offset equation for neighboring block 2:
 $\text{Diff}_{[2]} = 5(e - 2f + g)/16$. Bjontegaard, 20.

Like the second offset, a POSITA would have obviously read Bjontegaard's equation as $\text{Diff}_{[2]} = 5(e_{\text{modified}} - 2f + g)/16$, and thus the only difference to the third-offset equation claimed is the weights applied to each component:

Claimed Equation	Bjontegaard's Equation
$\Delta = \frac{1(q_2 - 2q_1 + q_0)}{4} - \frac{1\Delta}{2}$	$\Delta = \frac{5(q_2 - 2q_1 + q_0)}{16} - \frac{5\Delta}{16}$

Ex-1003 ¶¶294-300; §VIII.A.7.d. As with block 1, modifying Bjontegaard's weights to apply $\frac{1}{4}$ to the high-pass component and $\frac{1}{2}$ to the first-offset component (instead of $\frac{5}{16}$ to each) was obvious, *Friskit*, 306 F. App'x at 615; §VIII.A.7.d, as was calculating the offset on a device's processor with calculators, §§VIII.A.4.a, VIII.A.7.a; Ex-1003 ¶¶300-01.

g. 10[g]

Bjontegaard with Bjontegaard's Code teaches 10[g].

Bjontegaard discloses filtering fourth-pixel f (second closest to the boundary in neighboring block 2) when S_0_2 is below threshold $\text{Lim}(qp)$. §§VIII.A.3.f-VIII.A.3.h.

Bjontegaard also discloses “ $f_{\text{modified}} = f + \text{Diff}_{[2]}$,” meaning pixel f is modified by adding the third offset ($\text{Diff}_{[2]}$). Bjontegaard, 20; Ex-1003 ¶¶302-04.

The prescribed modifications would obviously be performed on a device’s processor with pixel modifiers. §§VIII.A.4.a, VIII.A.7.b; Ex-1003 ¶305.

8. Claim 12

a. 12[a]

Bjontegaard with Bjontegaard’s Code teaches 12[a].

Bjontegaard’s deblocking filter would obviously be implemented in an encoder because Bjontegaard’s Code implements it in an encoder, Ex-1013, 1; Ex-1015, 1; Ex-1003 ¶¶306-09, and that was conventional to reduce blocking artifacts during image prediction, Ex-1003 ¶308; *see e.g.*, Ex-1040, 171; Ex-1061, 3:20-31; Ex-1062, 18:21-23; *Samsung*, IPR2022-00100, Paper 30, 23-24, 34-35.

See §VIII.A.4 (claim 5).

9. Claim 13

a. 13[pre]

If limiting, Bjontegaard with Bjontegaard’s Code teaches 13[pre].

Bjontegaard’s deblocking filter obviously would be implemented in user equipment because Bjontegaard’s Code was intended for computers, Ex-1010; Ex-1015, 1; Ex-1001, 24:1-2 (noting a computer can be “user equipment”); Ex-1003 ¶¶310-13, and building encoders with deblocking filters into user equipment was

conventional. *See, e.g.*, Ex-1040, 1-5, 247; Ex-1061, 2:56-62; Ex-1003 ¶312; *Samsung*, IPR2022-00100, Paper 30, 23-24, 34-35.

b. 13[a]

Bjontegaard with Bjontegaard's Code teaches 13[a].

Bjontegaard would obviously store video frames in memory before encoding, as Bjontegaard's Code does, Ex-1013, 2 ("Input YUV file"); Ex-1003 ¶¶314-17, and as was also conventional during predictive encoding, which stores video "reference frames." *See, e.g.*, Bjontegaard, 13; Ex-1040, 84, Ex-1061 at 7:33-44; Ex-1062, 10:56-59; Ex-1003 ¶316; *Samsung*, IPR2022-00100, Paper 30, 23-24, 34-35.

c. 13[b]

Bjontegaard with Bjontegaard's Code teaches 13[b].

Bjontegaard would obviously store encoded video frames in memory, as Bjontegaard's Code does, Ex-1013, 2 ("output H.26L bitstream"); Ex-1003 ¶318-21, and as was convention for encoders, Ex-1040, 3, 247; Ex-1061, 10:56-59; Ex-1062 ¶30; §VIII.A.9.b. Ex-1003 ¶320.

See §VIII.A.8 (claim 12).

10. Claim 14

a. 14[a]

Bjontegaard with Bjontegaard's Code teaches 14[a].

Bjontegaard discloses a “coding method by mainly defining the decoder actions,” Bjontegaard, 4, which includes a filtering control device (i.e., a deblocking filter). Bjontegaard, 18 (§4.5); Ex-1003 ¶¶322-24.

Bjontegaard’s Code also implements Bjontegaard’s deblocking filter in a decoder to decode encoded-video frames into decoded-video frames. Ex-1011, 1; Ex-1015, 1; Ex-1003 ¶323.

See §VIII.A.4 (claim 5).

11. Claim 15

a. 15[pre]

If limiting, Bjontegaard with Bjontegaard’s Code teaches 15[pre]. §VIII.A.9.a; Ex-1003 ¶325.

b. 15[a]

Bjontegaard with Bjontegaard’s Code teaches 15[a]. §VIII.A.9.b.

Bjontegaard would obviously retrieve encoded video frames from memory for decoding, as Bjontegaard’s Code does, Ex-1011, 1 (“input H.26L bitstream”), 3; Ex-1003 ¶¶326-29, and as was convention for decoders, Ex-1040, 1-5, 247; Ex-1061, 10:56-59; Ex-1062 ¶30; Ex-1003 ¶328.

c. 15[b]

See §VIII.A.10; Ex-1003 ¶330.

d. 15[c]

Bjontegaard with Bjontegaard's Code teaches 15[c].

Bjontegaard would obviously be built into video-capable devices such as user equipment to decode (and play back) video, §VIII.A.9.a; Ex-1003 ¶¶331-35, as Bjontegaard's Code discloses a decoder to output decoded video frames, §VIII.A.10; Ex-1003 ¶333.

Moreover, it was conventional to implement decoders in dedicated media players or in media-player software. Ex-1003 ¶334; Ex-1040, 1-5 (dedicated media player); Ex-1055, Ex-1071 (media-player software). *Samsung*, IPR2022-00100, Paper 30, 23-24, 34-35.

12. Claim 16

a. 16 [pre]

If limiting, Bjontegaard with Bjontegaard's Code teaches 16[pre].

Bjontegaard's deblocking filter would obviously be run on a computer, including a non-transitory computer-readable medium, such as a computer hard drive, with computer (e.g., software) instructions to perform the decoder functions, including filtering. Ex-1003 ¶¶336-39. Indeed, Bjontegaard's Code includes computer programs with computer-readable instructions for encoding/decoding video frames with a deblocking filter, and requires memory to save the video files and computer programs/instructions, and execute them. *Id.* ¶337; Ex-1010.

Bjontegaard discloses a method for filtering control. Bjontegaard, 19; §VIII.A.2.a; Ex-1003 ¶337.

b. 16[a]

See §VIII.A.2.b; Ex-1003 ¶340.

c. 16[b]

See §VIII.A.2.c; Ex-1003 ¶341.

d. 16[c]

See §VIII.A.2.d; Ex-1003 ¶342.

e. 16[d]

See §VIII.A.2.e; Ex-1003 ¶343.

f. 16[e]

See §§VIII.A.2.d, VIII.A.3.b; Ex-1003 ¶344.

g. 16[f]

See §§VIII.A.2.e, VIII.A.3.f; Ex-1003 ¶345.

B. Ground 1B: Obviousness over Bjontegaard, Bjontegaard's Code, H.241, and B205/AAPA

1. H.241

H.241 is a standard published by the ITU in December 2006. H.241, Cover. H.241 constitutes prior art under pre-AIA §102(b).⁵

H.241 describes a “reduced-complexity decoding process” with a deblocking filter. *Id.*, 1, 27; Ex-1003 ¶¶145-46. H.241 discloses that “[a] determination is made of whether or not to apply filtering across a block edge” based on:

$$d = \left| \begin{array}{c} p_{2_2} - 2 * p_{1_2} + p_{0_2} \\ q_{0_5} - 2 * q_{1_5} + q_{2_5} \end{array} \right| + \left| \begin{array}{c} q_{0_2} - 2 * q_{1_2} + q_{2_2} \\ p_{2_5} - 2 * p_{1_5} + p_{0_5} \end{array} \right| +$$

H.241, 29. If $d < \beta$ (among other conditions), *id.*, two pixels in each block may be filtered. *Id.*; Ex-1003 ¶¶147-51.

2. B205/Applicant-Admitted Prior Art

B205 is a working draft of the HEVC standard, and was published on October 9, 2010. Ex-1094. B205 constitutes prior art under pre-AIA §102(a),⁶ and is Applicant-Admitted Prior Art. Ex-1001, 2:5-8.

⁵ H.241 was publicly accessible, Ex-1005 ¶¶89-103, and those interested knew where to find it, Ex-1123 ¶47; Ex-1124 ¶7; Ex-1125 ¶102; Ex-1126, 2, 14:31-15:5.

⁶ B205 was publicly accessible, Ex-1005 ¶¶104-24, and those interested knew where to find it, Ex-1127, 618; Ex-1128, 11; Ex-1129 ¶84; Ex-1130, 3.

B205/AAPA discloses a deblocking filter with the same filter-decision equation as H.241. B205, 130; Ex-1001, 2:5-27. B205/AAPA also discloses that deblocking is performed when $d < \beta$. B205, 130; Ex-1001, 2:5-27. But B205/AAPA uses a different offset equation than H.241. B205, 131; Ex-1001, 2:44-49; Ex-1003 ¶¶152-55.

3. Claim 2

In co-pending ITC proceedings, Lenovo asserts that claim 2's use of six pixels from two lines contradicts claim 1's requirement of "*only* using" three pixels from one ("first") line, *compare* 1[a], *with* 2[a], but "[t]he indefiniteness of a limitation...precludes a patentability determination only when the indefiniteness renders it logically impossible for the Board to reach...a decision" on the prior-art grounds. *Intel Corp. v. Qualcomm Inc.*, 21 F.4th 801, 813 (Fed. Cir. 2021). Here, no logical impossibility exists because claims 1 and 2 can be assumed not inconsistent, and the prior art taught using one *or* two lines of three pixels-per-line, and thus both claims are obvious even if the claims are inconsistent. Ex-1027, 23-28 (PO contending that claim 1 encompasses multiple lines).

a. 2[a]

Bjontegaard in view of Bjontegaard's Code, H.241, and B205/AAPA teaches 2[a].

Bjontegaard discloses calculating filter-decision value $S0_1$ using pixels from one line in a 4x4 block. Bjontegaard, 18-19. Bjontegaard does not disclose using pixels from two lines in block 1, or using lines 0 and 3.

First, a POSITA would have been motivated to calculate $S0_1$ using two lines to make a filter decision for the entire block—rather than make decisions line-by-line—to increase computational efficiency. Ex-1003 ¶¶348-55, 359. H.241 and B205/AAPA, for instance, teach calculating a filter-decision value using two lines (2 and 5) to decide whether to filter all lines in the block:

$$d = \left| \begin{array}{c} p_{2_2} - 2*p_{1_2} + p_{0_2} \\ q_{0_5} - 2*q_{1_5} + q_{2_5} \end{array} \right| + \left| \begin{array}{c} q_{0_2} - 2*q_{1_2} + q_{2_2} \\ p_{2_5} - 2*p_{1_5} + p_{0_5} \end{array} \right| +$$

H.241, 29; B205, 130-31; Ex-1001, 2:5-27 (AAPA same); Ex-1003 ¶350.

A POSITA would have understood that using two lines as representative of the entire block significantly reduces the computational complexity of making filter decisions—including both *whether* (like H.241/B205) and *how many* pixels to filter (like Bjontegaard). Ex-1003 ¶351; Ex-1053, 27-28; Ex-1056, 3; Ex-1059, 1693. Using two lines also safeguards against incorrect filter decisions if one line has noise or unrepresentative pixel values. Ex-1003 ¶350. Calculating line-by-line or representative filter-decisions values was an obvious design choice, balancing accuracy and efficiency. *Id.*; Ex-1065 ¶¶41-44.

Moreover, a POSITA would have understood that $S0_1$'s expression $(b-2c+d)$ is multiplied by 12 so that it has the same order of magnitude as the threshold $\text{Lim}(qp)$. Ex-1003 ¶¶353-54. But that this is mathematically the same as comparing $S0_1$'s expression to a threshold divided by 12: $(b-2c+d) < \text{Lim}(qp)/12$. *Id.* ¶355. When evaluating multiple lines of pixels for filter decisions, as taught by H.241 and B205/AAPA, dividing the threshold once would be slightly more efficient than multiplying the expression twice (once per line), and therefore a POSITA would have been motivated to sum the expression $(b-2c+d)$ for each line, as taught by H.241 and B205/AAPA, and compare it to an appropriately scaled threshold. *Id.*; H.241, 29; B205, 130; Ex-1001, 2:5-27 (AAPA same).

Accordingly, it would have been obvious to calculate $S0_1$ using two lines of pixels $(b-2c+d)$ from block 1 to determine how many pixels to filter in all lines of block 1. Ex-1003 ¶352. This modification would have merely applied a known technique (using two lines to represent a block) to a known device (deblocking filter) ready for improvement (computational efficiency) to yield predictable results (more-efficient deblocking decisions). *Id.*; *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417-18 (2007).

Second, a POSITA would have understood that which two lines were selected is immaterial and would have been an obvious design choice, as the '841 Patent recognizes. Ex-1003 ¶¶356-59; Ex-1001, 10:14-16 (lines 2 and 5), 11:14-17 (lines 3

and 4), 11:61-64 (lines 0 and 3), 12:5-11 (lines 4 and 7). Those in the art simply selected lines that were spaced apart to ensure the lines were more-representative of the block. Ex-1003 ¶356. And because Bjontegaard discloses filtering 4x4 blocks, Bjontegaard, 19, only three, two-line options with spacing would be available (lines 0-2, 0-3, 1-3). Ex-1003 ¶357. A POSITA therefore would have found it obvious to use lines 0 and 3 for Bjontegaard's filter decisions. Ex-1003 ¶¶357-58.

b. 2[b]

Bjontegaard in view of Bjontegaard's Code, H.241, and B205/AAPA teaches 2[b].

It would have been obvious to use multiple lines to calculate $S0_2$ as representative for neighboring block 2, and select lines 0 and 3, for the same reasons as block 1. Ex-1003 ¶360; §VIII.B.3.a.

4. Claim 6

The same inconsistency between claims 1 and 2 exists for claim 5 and 6. §VIII.B.3. Petitioner assumes no inconsistency for the prior-art challenge. *Id.*

a. 6[a]

See §VIII.B.3.a, VIII.A.4.b; Ex-1003 ¶363.

b. 6[b]

See §§VIII.B.3.a-VIII.B.3.b, VIII.A.4.c; Ex-1003 ¶364.

5. Claim 11

a. 11[a]

Bjontegaard in view of Bjontegaard’s Code, H.241, and B205/AAPA teaches 11[a].

Bjontegaard discloses several conditions to decide *whether* to perform deblocking filtering. Bjontegaard, 19 (deciding whether to filter based on “Strength” conditions); Ex-1003 ¶¶365-66. Bjontegaard does not disclose comparing the sum of first and second filter-decision values to a threshold to determine *whether* to perform deblocking.

A POSITA would have been motivated to use the sum of $S0_1$ and $S0_2$ as an additional condition for *whether* to perform deblocking based on the activity of the pixels near the boundary. Ex-1003 ¶367. H.241 and B205 disclose summing a series of expressions just like Bjontegaard’s $S0_1$ and $S0_2$ ($b-2c+d$; $e-2f+g$) to determine *whether* to perform deblocking filtering:

$$\begin{aligned} d = & |p2_2 - 2p1_2 + p0_2| + |q2_2 - 2q1_2 + q0_2| \\ & + |p2_5 - 2p1_5 + p0_5| + |q2_5 - 2q1_5 + q0_5| < \beta \end{aligned}$$

in addition to the same “Strength” conditions Bjontegaard discloses. *Id.*; compare H.241, 29 (determining *whether* to filter if block “is intra coded,” “contain[] non-

zero transform coefficients,” and “absolute difference between ... motion vectors,” in addition to checking whether “ $d \geq \beta$ ”); B205, 130 (same), *with* Bjontegaard, 19 (same, but no evaluation of “d”).

A POSITA would have been motivated to sum Bjontegaard’s S_{01} and S_{02} equations, as taught by H.241 and B205, to use a measure of activity on both sides of the boundary to control the deblocking decision, as opposed to just the “Strength” conditions that do not look at the pixel values themselves. Ex-1003 ¶¶368-69. H.241 and B205 showed that adding this additional condition to decide *whether* to perform deblocking would work. *Id.* And the comparison would obviously be performed on a device’s processor with a comparator. Ex-1003 ¶370.

b. 11[b]

See §§VIII.A.2.d, VIII.A.4.b, VIII.A.4.d, VIII.B.5.a; Ex-1003 ¶371.

c. 11[c]

See §§VIII.A.2.e, VIII.A.4.c, VIII.A.4.e, VIII.B.5.a; Ex-1003 ¶372.

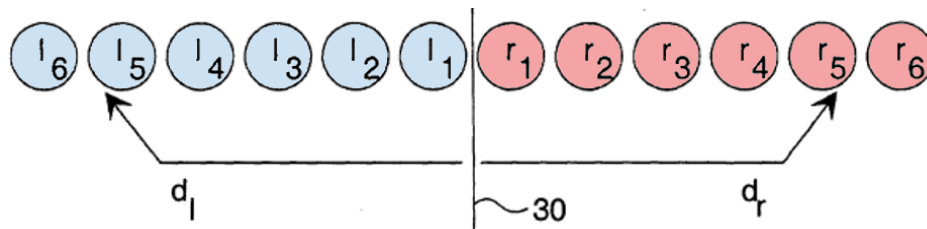
C. Ground 2A: Obviousness over Aksu and Gonzalez

If the “determining” step precludes having any “mathematical basis” in pixel values in a neighboring block, then the Challenged Claims are unpatentable as obvious over Aksu and Gonzalez (and other prior art; Grounds 2B, C).

1. Aksu

Aksu is an International Patent Application filed January 22, 2001, and published July 26, 2001. Aksu constitutes prior art under pre-AIA §102(b).

Aksu discloses a “method and associated device for filtering digital video images,” Aksu, 1:1, for “reducing blocking artefacts,” *id.*, 5:26-29. According to Aksu, past deblocking filters “tend to remove lines that belong to real features of the image.” *Id.*, 5:22-24. To preserve those “real features,” Aksu decides both *whether* and *how many* pixels to filter near the boundary of two blocks—an “l” (left) block and an “r” (right) block.



Aksu, Fig. 2 (coloring added); Ex-1003 ¶¶128-31.

Before making these decisions, Aksu first determines how many pixels (up to six) in each block to evaluate. Aksu, 9:27-10:11. Each selected pixel (r_{2-6}) is then compared to the pixel closest to the boundary (r_1) to calculate the difference, which is compared to a threshold. *Id.*, 13:15-21. These series of calculations produce a single value (“ d_r ”), *id.*, 13:10-20, which represents the activity of the pixels near the boundary in block “r,” Ex-1003 ¶¶132-34.

This process is repeated for block “l” using the “l” pixels. Aksu, 13:15-19, 13:21-23. As Aksu explains, “[i]t is advantageous that the values of the parameters d_r and d_l are calculated independent of one another.” *Id.*, 13:30-33.

If both d_r and d_l are greater than “1,” filtering is performed, and the values of d_r and d_l control how many pixels to filter in each block, separately. Aksu, 15:17-16:4; Ex-1003 ¶¶135-37. The higher the $d_{r/l}$ value, the less activity is present in that block, and thus more pixels in that block will be filtered. Ex-1003 ¶136.

2. Gonzalez

Gonzalez is a textbook titled “Digital Image Processing,” published in 2008. Gonzalez, Cover, Copyright. Gonzalez constitutes prior art under pre-AIA §102(b).⁷

Gonzalez discloses that identifying image features is critical to image processing, Gonzalez, 689-90, and is accomplished by identifying “sharp, *local* changes in [pixel] intensity” using mathematics, *id.*, 692. A “first-order derivative” (like velocity)—which calculates the difference between pixels ($p_1 - p_2$)—is the simplest way to identify pixel changes. *Id.*, 693; Ex-1003 ¶¶138-42. Alternatively, a “second-order derivative” (like acceleration) such as the “Laplacian”—which calculates the difference-of differences between three pixels ($p_1 - 2p_2 + p_3$)—is a more-sophisticated method. Gonzalez, 160, 693, 696; Ex-1003 ¶142. Gonzalez explains

⁷ Gonzalez was publicly accessible. Ex-1007 ¶¶6-10.

that the Laplacian “is much more aggressive than a first-order derivative in enhancing sharp changes,” and can identify “fine detail...much more than first-order derivatives.” Gonzalez, 694-95. Accordingly, the Laplacian can detect “isolated points” or thin “lines” much better than first-order derivatives. *Id.*, 696-97; Ex-1003 ¶143.

3. Claim 1

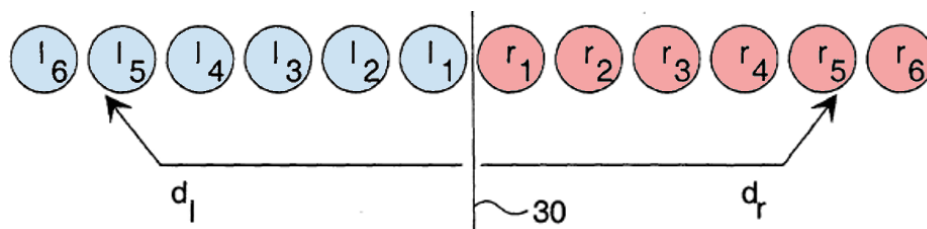
a. 1[pre]

If limiting, Aksu discloses 1[pre].

Aksu discloses “a method for reducing visual artefacts in a frame of a digital video” such as “artefacts due to a block boundary.” Aksu, 1:3-7, Abstract.

Aksu discloses that “[t]he frame is divided into adjacent blocks having a size of e.g. 8 x 8 pixels.” Aksu, 4:24-25; Ex-1003 ¶¶374-76.

Aksu explains that each pixel has a pixel value. Aksu, 4:29-32, 5:16-20, 15:4-8, 7:16-20, 8:1-6, 8:11-13, 13:10-13. Aksu further depicts a line of pixels spanning two neighboring blocks (l_{1-6} and r_{1-6}):

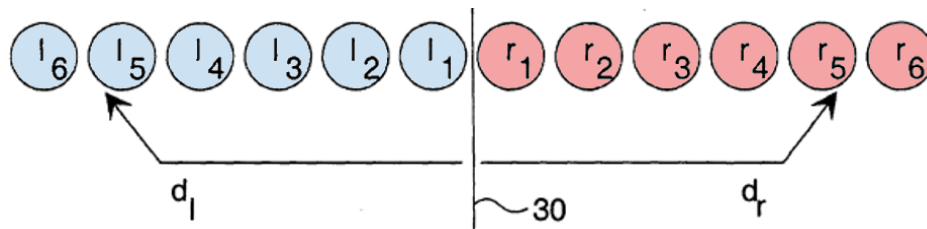


Aksu, Fig. 2 (coloring added throughout); Ex-1003 ¶¶377-78.

b. 1[a]

The Aksu-Gonzalez combination teaches 1[a].

Aksu discloses calculating a first filter-decision value “ d_r ” which represent[s] activity” in block r. Aksu, 13:10-19. To calculate “ d_r ,” Aksu performs multiple first-order difference calculations between pixel “ r_1 ” (closest to the boundary) and up to five pixels (r_{2-6}) in a line further from the boundary ($|r_1-r_2|$, $|r_1-r_3|$, etc.), comparing the absolute value of each difference to a threshold based on β . Aksu, 13:10-19; Ex-1003 ¶¶379-80.



Aksu, Fig. 2.

Aksu discloses that d_r is sometimes calculated using three pixel values r_1 , r_2 , r_3 —corresponding to pixel values $p0_i$, $p1_i$, $p2_i$, Ex-1003 ¶381—but Aksu does not disclose the calculation is “based on” $|p2_i-2p1_i+p0_i|$.

However, Gonzalez teaches that a “second-order” Laplacian—which applies coefficients $[1, -2, 1]$ to three pixel values—is superior to first-order difference calculations for detecting noise and fine details in an image. Gonzalez, 160, 693-96. This equation represents multiple difference calculations (similar to Aksu’s) wrapped into one. *Id.*, 154; Ex-1003 ¶¶382-88.

<u>Aksu</u>	<u>Aksu + Gonzalez</u>
d_r determined based on, e.g.:	
$ r_1 - r_2 $	
$ r_1 - r_3 $	
$ r_1 - r_4 $	$d_r = r_3 - 2r_2 + r_1 $
$ r_1 - r_5 $	
$ r_1 - r_6 $	

A POSITA would have been motivated to replace Aksu’s multiple difference calculations with a single Laplacian, as taught by Gonzalez, for several reasons. Ex-1003 ¶¶384-96.

First, Gonzalez explains that the Laplacian is superior at detecting image details with greater precision, while avoiding responding to *gradual* pixel-value changes (“ramps”) like Aksu’s first-order difference calculations, which are not the type of image details that preclude filtering. Gonzalez, 694-95; Ex-1003 ¶385.

Second, Aksu suggests a cumbersome, *pre-decision* process to determine how many pixels (0-6) are needed to calculate d_r , Aksu, 9:31-10:21. This process could be eliminated because a Laplacian always uses three pixels, reducing the computational resources needed to calculate d_r . Ex-1003 ¶386; Aksu, 10:12-14 (noting other methods can be used for pre-decision process). Relevant to claim construction, eliminating this pre-decision process—which looks to the “difference of pixel values ... across the block boundary” to determine how many pixels are needed to calculate d_r —means Aksu’s filter decision would have “no mathematical basis” with pixel values from a neighboring block. Aksu, 9:27-10:8; Ex-1003 ¶387.

Third, rather than perform up to five calculations to determine the differences between r_1 and r_{2-6} , using up to six pixels as inputs, a Laplacian (like Gonzalez's) would require one calculation with just three pixels, further reducing computational complexity and resource demands. Ex-1003 ¶388. Although Aksu evaluates up to six pixels, Aksu filters up to three pixels closest to the boundary. Aksu, 15:17-35. Accordingly, evaluating only those three pixels with a single Laplacian would provide a lower-complexity assessment of their activity to determine how many of them to filter. Ex-1003 ¶388. Indeed, a POSITA would have known that other high-efficiency deblocking filters applied Laplacians to the three closest pixels near a boundary to measure activity. *Id.* ¶¶395-96; H.241, 29; B205, 130; Ex-1065 ¶¶42-44.

Most critically, however, a POSITA would have known that similar modifications to Aksu were made during subsequent prior-art standards developments, confirming these modifications are improvements that would work. Ex-1003 ¶¶389-94.

A POSITA would have known that, when replacing Aksu's difference calculations with a Laplacian (like Gonzalez's), the Laplacian would measure the three pixels closest to the block boundary (r_1, r_2, r_3), because these are the pixels closest to the boundary subject to filtering, and the prior art routinely applied a Laplacian to these three pixels. Ex-1003 ¶397; H.241, 29; B205, 130; Ex-1065 ¶¶42-

44. Furthermore, a POSITA would have known to take the “absolute value” of the Laplacian—represented by vertical bars ($|r_3 - 2r_2 + r_1|$) that always outputs a positive value of a calculation (e.g., $|1| = |-1| = 1$)—because Aksu (and others) measured activity using the absolute value, and it is the *magnitude* of activity that is important. Ex-1003 ¶398; Aksu, 13:15-19 (absolute value of difference calculations); H.241, 29 (absolute value of Laplacian); B205, 130; (same) Ex-1065, ¶¶42-44 (same).

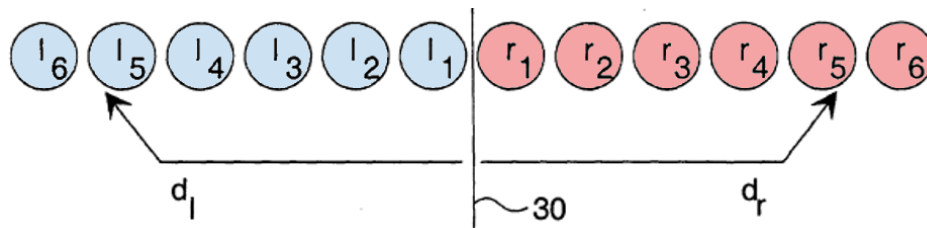
Moreover, incorporating a Laplacian into Aksu involves a simple substitution of one known element (first-order activity measures) with another (a second-order activity measure) to obtain predictable results (a measure of pixel activity). Ex-1003 ¶399; *KSR*, 550 U.S. at 416. Additionally, the modification applies a known technique (using a Laplacian to measure activity) to a known device (deblocking filter) ready for improvement (in computational efficiency) to yield predictable results (a lower-complexity measure of activity). Ex-1003 ¶399; *KSR*, 550 U.S. at 417-18.

A POSITA would have had a reasonable expectation of success in making this modification because the prior art demonstrated that similar modifications had already been made to Aksu, and others had successfully used Laplacians as an activity measure in deblocking filters. Ex-1003 ¶¶395, 400.

c. 1[b]

The Aksu-Gonzalez combination teaches 1[b].

Aksu discloses calculating for neighboring block l a value of “ d_l ...which represent[s] activity” in block l. Aksu, 13:10-19. To calculate d_l , Aksu performs the same series of difference calculations as d_r , “except that all r’s are replaced by l’s....” (using l_1, l_2, l_3 , etc.). Aksu, 13:21-23; Ex-1003 ¶¶401-02.



Aksu, Fig. 2.

As with block r, §VIII.C.3.b, a POSITA would have found it obvious to replace Aksu’s d_l difference calculations with a single Laplacian, as taught by Gonzalez, using pixels l_1, l_2, l_3 —corresponding to pixels $q0_i, q1_i, q2_i$ —such that the decision value would be “based on” $|q2_i - 2q1_i + q0_i|$, as claimed. Ex-1003 ¶403.

d. 1[c]

The Aksu-Gonzalez combination teaches 1[c].

Aksu determines how many pixels in a line in block r to filter by making comparisons to a threshold. Specifically, Aksu discloses that d_r is determined by comparing each of Aksu’s first-order difference calculations to a threshold based on

β , Aksu, 13:15-19 (explaining $d_r=i$ where $|r_1-r_{i+1}|>\beta/i$, and $|r_1-r_j|\leq\beta/j$, where j includes all values between 1 and i), and d_r controls how many pixels are filtered:

d_r ($d_i > 1$)	r_1	r_2	r_3
1	X	X	X
2	1	X	X
3	1	1*	X
4	2	2	X
5	2	2	2**
6	3 or 2***	3	3

Aksu, 15:1-36 (X means no filtering; a number means filtering of that pixel and defines the filter equation used). When $d_r=1$, no pixels in the line in block r are filtered; when $d_r=2$, only one pixel (r_1) is filtered; and when $d_r=3$, two pixels are filtered (r_1 and r_2). Aksu, 15:21-22; Ex-1003 ¶405.

When using a second-order Laplacian (like Gonzalez's) to measure activity, §VIII.C.3.b, a POSITA would have understood that the Laplacian's output would be compared directly to a threshold based on β —just like Aksu's difference calculations—to determine how many pixels to filter. Aksu, 13:15-19; Ex-1003 ¶¶406-10. Aksu assigns d_r an integer between 1 and 6 to track Aksu's *multiple* (up to 5) difference calculations measuring activity, which are each compared to a threshold. Aksu, 13:10-28, 15:1-36; Ex-1003 ¶¶406-08. With a single Laplacian, however, there would be no need for these integers because only one activity-measure calculation would be performed. Ex-1003 ¶408. And because Aksu discloses filtering more pixels when the activity measure is below the threshold (by

making the value of d_r bigger, indicating no image details to preserve), and filter less when its activity measure equals or exceeds the threshold (by making the value of d_r smaller, indicating an image detail to preserve), a POSITA would have followed this same approach when using the output of the Laplacian as the value of d_r . Ex-1003 ¶¶406-10.

A POSITA would have also known that when Aksu's teachings were modified during subsequent standards developments, the single activity-measure calculation was compared directly to a threshold to control *how many* pixels to filter. Ex-1003 ¶¶409-10; B118, 48 (filtering two pixels if $a_p = |p_2 - p_0| < \beta$ and one pixel if $a_p \geq \beta$).

Finally, even if claim construction requires that the first threshold must be associated only with block r , that would have been obvious. Aksu discloses a threshold based on the minimum "QP" (or "quantization parameter") of blocks r and l . Aksu, 10:9-12. Thus, Aksu discloses that the threshold uses only the QP of block r when $QP_r < QP_l$. *Unwired*, 841 F.3d at 1002. But it would have also been obvious to always use the QP associated with block r when selecting a threshold for d_r (effectively, β_r) for the same reasons discussed above. Ex-1003 ¶¶411-13; §VIII.A.2.e.

e. 1[d]

The Aksu-Gonzalez combination teaches 1[d].

Aksu discloses that “[t]he [modified] values of the pixels l_1 , l_2 and l_3 are determined in the same manner according to the value of the parameter d_l .” Aksu, 15:19-21. A POSITA would have found it obvious to use d_l to determine how many pixels in a line in neighboring block l to filter by comparing d_l to a threshold based on β . Ex-1003 ¶¶414-15; §VIII.C.3.d. If the second threshold for block l must be different than the first threshold for block r , selecting block-specific thresholds would have also been obvious. Ex-1003 ¶415; §VIII.A.2.e.

4. Claim 3

a. 3[a]

See §VIII.C.3.d (disclosing “determining” step); Ex-1003 ¶416.

b. 3[b]

The Aksu-Gonzalez combination teaches comparing a first filter-decision value (d_r) to a first threshold based on β (or β_r). §VIII.C.3.d; Ex-1003 ¶417.

c. 3[c]

The Aksu-Gonzalez combination teaches 3[c].

Aksu discloses that more pixels are filtered when more difference calculations are below a threshold, because a pixel difference below the threshold means there is no significant activity between those pixels, and therefore they can be filtered. Aksu, 13:15-19; §VIII.C.3.d; Ex-1003 ¶¶418-19. Aksu simply represents these multiple difference calculations in a single variable (d_r) that arbitrarily *increases* for each

difference calculation below the threshold. Aksu, 13:15-19; §VIII.C.3.d; Ex-1003 ¶¶406-08. Accordingly, Aksu discloses a value of d_r ($d_r=3$) such that two pixels are filtered, Aksu, 15:24-36, meaning two difference calculations (r_1-r_2 ; r_2-r_3) are below the threshold, but the next calculation (r_1-r_4) exceeds the threshold, *id.*, 13:15-19, indicating some activity between pixels r_3 and r_4 , and thus only pixels r_1 and r_2 should be filtered, Ex-1003 ¶419.

Upon incorporating a Laplacian as taught by Gonzalez into Aksu, a POSITA would have known to compare the Laplacian's activity measure to a threshold based on β . §VIII.C.3.d. A POSITA would have been further motivated to establish a threshold based on β below which two pixels would be filtered to preserve Aksu's control over how many pixels are filtered based on activity, because Aksu discloses filtering more pixels in a line in block r when activity measures are below a threshold. Ex-1003 ¶¶406-08, 420-23. This exact change had been made post-Aksu. *Id.* ¶422; B118, 48 (filtering two pixels if $a_p=|p_2-p_0|<\beta$). Determining the exact threshold would have been an empirical exercise within the skill of a POSITA. Ex-1003 ¶¶421-23.

d. 3[d]

The Aksu-Gonzalez combination teaches 3[d].

Aksu discloses that fewer pixels are filtered when fewer difference calculations are below a threshold based on β , because a pixel difference exceeding

the threshold means there is activity between those pixels, and therefore they should not be filtered. Aksu, 13:15-19; §VIII.C.3.d; Ex-1003 ¶¶424-26. Aksu also discloses a value of d_r ($d_r=2$) such that one pixel is filtered, Aksu, 15:24-36, meaning one difference calculation (r_1-r_2) is below the threshold, but the subsequent calculation (r_1-r_3) exceeds the threshold, *id.*, 13:15-19, indicating activity between pixels r_2 and r_3 , and thus only pixel r_1 should be filtered, Ex-1003 ¶425.

Upon incorporating a Laplacian into Aksu, a POSITA would have known to compare the Laplacian's activity measure to a threshold based on β . §VIII.C.3.d. A POSITA would have been further motivated to establish a threshold above which one pixel would be filtered to preserve Aksu's control over how many pixels are filtered based on activity, because Aksu discloses filtering less pixels in a line in block r when an activity measure exceeds a threshold. Ex-1003 ¶¶406-08, 425. This exact change had been made post-Aksu. *Id.* ¶422; B118, 48 (filtering one pixel if $a_p \geq \beta$). Determining the exact threshold would have been an empirical exercise within the skill of a POSITA. Ex-1003 ¶426.

e. 3[e]

See §VIII.C.3.e (disclosing “determining” step); Ex-1003 ¶427.

f. 3[f]

The Aksu-Gonzalez combination teaches comparing a second filter-decision value (d_l) to a second threshold based on β (or β_l). §§VIII.C.3.d, VIII.C.3.e; Ex-1003 ¶428.

g. 3[g]

The Aksu-Gonzalez combination teaches 3[g].

Aksu discloses filtering neighboring block l following the same procedure as block r. §§VIII.C.3.d, VIII.C.3.e; Ex-1003 ¶¶429-30. Accordingly, there would be a threshold below which two pixels in a line in neighboring block l would be filtered, as with block r. §§VIII.C.3.d, VIII.C.4.c; Ex-1003 ¶430.

h. 3[h]

The Aksu-Gonzalez combination teaches 3[h].

Aksu discloses filtering neighboring block l following the same procedure as block r. §§VIII.C.3.d, VIII.C.3.e; Ex-1003 ¶¶431-32. Accordingly, there would be a threshold at or above which one pixel in a line in neighboring block l would be filtered, as with block r. §§VIII.C.3.d, VIII.C.4.d; Ex-1003 ¶432.

5. Claim 5

a. 5[pre]

If limiting, Aksu discloses or suggests 5[pre].

Aksu discloses “[a] method and associated device for filtering digital video images,” Aksu, 1:1, and that the device “implement[s] the method,” *id.*, 6:12-16; Ex-1003 ¶¶433-34.

b. 5[a]

The Aksu-Gonzalez combination teaches 5[a]. §VIII.C.3.b. Aksu also discloses a “digital signal processor” programmed “to apply predetermined processing functions” to implement the method, Aksu, 17:19-23, which would obviously include calculators to make the calculations, §VIII.C.3.b; Ex-1003 ¶435.

c. 5[b]

The Aksu-Gonzalez combination teaches 5[b]. §§VIII.C.3.c, VIII.C.5.b; Ex-1003 ¶436.

d. 5[c]

The Aksu-Gonzalez combination teaches 5[c]. §VIII.C.3.d. Aksu’s “digital signal processor” programmed “to apply predetermined processing functions” to implement the method, Aksu, 17:19-23, would also obviously include determiners to make the determinations, Ex-1003 ¶437.

e. 5[d]

The Aksu-Gonzalez combination teaches 5[d]. §VIII.C.3.e, VIII.C.5.d; Ex-1003 ¶438.

f. 5[e]

The Aksu-Gonzalez combination teaches 5[e]. §§VIII.C.3.d, VIII.C.4.b. Aksu’s “digital signal processor” programmed “to apply predetermined processing functions” to implement the method, Aksu, 17:19-23, would also obviously include comparators to make the comparisons, Ex-1003 ¶439.

g. 5[f]

The Aksu-Gonzalez combination teaches 5[f]. §§VIII.C.3.e, VIII.C.4.f, VIII.C.5.f; Ex-1003 ¶440.

6. Claim 7

a. 7[a]

See §§VIII.C.4.c, VIII.C.5.d, VIII.C.5.f; Ex-1003 ¶441.

b. 7[b]

See §§VIII.C.4.d, VIII.C.5.d, VIII.C.5.f; Ex-1003 ¶442.

c. 7[c]

See §§VIII.C.4.g, VIII.C.5.e, VIII.C.5.g; Ex-1003 ¶443.

d. 7[d]

See §§VIII.C.4.h, VIII.C.5.e, VIII.C.5.g; Ex-1003 ¶444.

7. Claim 9

a. 9[a]

The Aksu-Gonzalez combination teaches 9[a].

The Aksu-Gonzalez combination teaches comparing a Laplacian to a threshold based on β . §VIII.C.3.d. Aksu also discloses determining the threshold using a quantization parameter associated with the block. Aksu, 13:21 (“ $\beta=4\log(QP)$ ”), 9:31-35 (“QP” defined as “quantization step”), 10:9-12 (selecting QP_r if $QP_r < QP_l$); §§VIII.C.3.d, VIII.A.2.e (block-specific thresholds). Aksu also discloses a “digital signal processor” programmed “to apply predetermined processing functions” to implement the method, Aksu, 17:19-23, which would also obviously include a threshold determiner to make the determination. Ex-1003 ¶¶445-48.

b. 9[b]

See §§VIII.C.3.d, VIII.A.2.e (block-specific thresholds), VIII.C.3.e, VIII.C.7.a; Ex-1003 ¶449.

8. Claim 12

a. 12[a]

The Aksu-Gonzalez combination teaches 12[a].

Aksu discloses “a video encoder comprising...a filter for reducing visual artefacts due to a block boundary.” Aksu, 1:13-18; Ex-1003 ¶¶450-51.

See §VIII.C.5 (claim 5).

9. Claim 13

a. 13[pre]

If limiting, Aksu discloses or suggests 13[pre].

Aksu discloses “a mobile terminal 46 intended for use as a portable video telecommunication device and applying the method according to the invention.” Aksu, 18:31-19:3, 17:19-23, Fig. 6. A POSITA would have understood that a “mobile terminal” is “user equipment,” as claimed, because user equipment is a generic term for a user device like a mobile terminal that implements an encoder and/or decoder. Ex-1003 ¶¶452-54; Ex-1001, 24:1-2, 25:27-30.

b. 13[a]

Aksu discloses or suggests 13[a].

Aksu discloses a typical video-encoding process, which includes decoding a “prediction error frame.” Aksu, 3:5-33; Ex-1003 ¶¶455-57. Aksu explains that the “frame is saved in a frame memory 16.” Aksu, 3:5-21.

c. 13[b]

The Aksu-Gonzalez combination teaches 13[b].

Aksu discloses the encoded video frames are transmitted to a decoder, Aksu, 2:5-8, Fig. 1, but Aksu obviously stores the encoded video frames in memory prior to transmission, as was conventional, Ex-1003 ¶¶458-60; §§VIII.C.9.b, VIII.A.9.c.

See §VIII.C.8 (claim 12).

10. Claim 14

a. 14[a]

The Aksu-Gonzalez combination teaches 14[a].

Aksu discloses “a video decoder comprising...a filter for reducing visual artefacts due to a block boundary.” Aksu, 1:13-18; Ex-1003 ¶¶461-62.

See §VIII.C.5 (claim 5).

11. Claim 15

a. 15[pre]

If limiting, Aksu discloses or suggests 15[pre]. §VIII.C.9.a; Ex-1003 ¶463.

b. 15[a]

Aksu discloses or suggests 15[a].

Aksu discloses a video-decoding process, which includes receiving encoded video frames from an encoder, Aksu, 4:7-10, and utilizing memory when decoding video frames, Aksu, 4:7-18; Ex-1003 ¶¶464-66. Aksu’s encoded video frames would obviously be stored in memory before decoding, as was conventional. Ex-1003 ¶466; §VIII.A.11.b.

c. 15[b]

See §§VIII.C.10, VIII.C.11.b; Ex-1003 ¶467.

d. 15[c]

Aksu teaches 15[c].

Aksu discloses “the mobile terminal 46 comprises at least display means 47 for displaying images” and is intended as “a portable video telecommunication device.” Aksu, 18:31-19:3. Because Aksu discloses displaying decoded video, the mobile terminal constitutes a “media player.” Ex-1003 ¶¶468-71.

Alternatively, Aksu’s decoder would have obviously been implemented in a media player, as was conventional. Ex-1003 ¶470; §VIII.A.11.d.

12. Claim 16

a. 16 [pre]

Aksu discloses or suggests 16[pre].

Aksu discloses “[a] storage medium can be used for storing a software program comprising machine executable steps for performing the method according to the invention,” and that “the software program is read from the storage medium to a device comprising programmable means, *e.g.* a processor, for performing the method of the invention.” Aksu, 18:10-15. The “method” Aksu provides is for filtering control. §VIII.C.3.a; Ex-1003 ¶¶472-73.

b. 16[a]

See §VIII.C.3.b; Ex-1003 ¶474.

c. 16[b]

See §VIII.C.3.c; Ex-1003 ¶475.

d. 16[c]

See §VIII.C.3.d; Ex-1003 ¶476.

e. 16[d]

See §VIII.C.3.e; Ex-1003 ¶477.

f. 16[e]

See §§VIII.C.3.d, VIII.C.4.b; Ex-1003 ¶478.

g. 16[f]

See §§VIII.C.3.e, VIII.C.4.f; Ex-1003 ¶479.

D. Ground 2B: Obviousness over Aksu, Gonzalez, H.241, and B205 (AAPA)

1. Claim 2

Petitioner assumes no inconsistency between claims 1 and 2. §VIII.B.3.

a. 2[a]

Aksu in view of Gonzalez, H.241, and B205/AAPA teaches 2[a].

Aksu discloses calculating filter-decision value d_r using a single line in block r , Aksu, 13:10-19, and then repeating the calculation line-by-line, *id* at 16:15-18. The Aksu-Gonzalez combination discloses calculating d_r using a Laplacian, §VIII.C.3.d, but does not expressly disclose applying a Laplacian to two lines, or using lines 0 and 3, in block r .

As with Ground 1, a POSITA would have found it obvious to: (1) use two lines to calculate d_r to make filter decisions for the entire block r to increase computational efficiency, as taught by H.241 and B205; and (2) to select lines 0 and 3 as a design choice. Ex-1003 ¶¶482-86; §VIII.B.3.a. Although Aksu disclosed 8x8 blocks as exemplary, a POSITA would have known that deblocking filters were commonly applied to 4x4 blocks, Ex-1037, 197-98; Ex-1053, 26; Ex-1056, 3, that Aksu was not limited to 8x8 blocks, and choosing lines 0 and 3 would have been especially obvious for 4x4 blocks, Ex-1003 ¶¶485-86.

b. 2[b]

Aksu in view of Gonzalez, H.241, and B205/AAPA teaches 2[b]. It would have been obvious to use multiple lines to calculate d_l as representative for neighboring block l , and select lines 0 and 3, for the same reasons as block r . §§VIII.D.1.a, VIII.B.3.a; Ex-1003 ¶487.

2. Claim 6

Petitioner assumes no inconsistency between claims 5 and 6. §§VIII.B.3, VIII.D.1.

a. 6[a]

See §§VIII.D.1.a, VIII.C.5.b; Ex-1003 ¶490.

b. 6[b]

See §§VIII.D.1.a, VIII.D.1.b, VIII.C.5.c; Ex-1003 ¶491.

3. Claim 11

a. 11[a]

Aksu in view of Gonzalez, H.241, and B205/AAPA teaches 11[a].

Aksu discloses that “for filtering to be applied to any pixel, parameters d_r and d_l must both have a value greater than 1.” Aksu, 15:24-25. Aksu therefore uses d_r and d_l to decide both *whether* to filter, and also *how many* pixels to filter in each block. Ex-1003 ¶¶492-93. Aksu does not disclose summing d_r and d_l and comparing it to a threshold to decide *whether* to filter.

When using a Laplacian like Gonzalez’s to calculate d_r and d_l , §§VIII.C.3.b, VIII.C.3.c, a POSITA would have been motivated to sum d_r and d_l and compare the sum to a threshold, as taught by H.241 and B205, to determine *whether* to perform deblocking:

$$d = |p2_2 - 2p1_2 + p0_2| + |q2_2 - 2q1_2 + q0_2| \\ + |p2_5 - 2p1_5 + p0_5| + |q2_5 - 2q1_5 + q0_5| < \beta$$

H.241, 29; B205, 130; Ex-1001, 2:5-27 (AAPA same); Ex-1003 ¶494. A POSITA would have understood that, unlike Aksu’s *separate* evaluations of d_r and d_l , summing d_r and d_l would better represent the *overall* activity near the boundary, as had been often used to decide *whether* to perform deblocking. Ex-1003 ¶¶494, 496.

Moreover, summing the Laplacians for d_r and d_l is a substitution of one known element (separate evaluations of d_r and d_l) with another (sum of d_r and d_l) to obtain

predictable results (a decision *whether* to filter). Ex-1003 ¶495; *KSR*, 550 U.S. at 416. Such a modification would have also been an application of a known technique (summing of Laplacians representing overall activity) to a known device (deblocking filter) ready for improvement (decision efficiency) to yield predictable results (a yes/no filter decision). Ex-1003 ¶495; *KSR*, 550 U.S. at 417-18.

Once a decision is made to perform deblocking, a POSITA would have known to then evaluate d_r and d_l *separately* to decide *how many* pixels to filter in each block, as taught by the Aksu-Gonzalez combination. Ex-1003 ¶494.

b. 11[b]

See §§VIII.C.3.d, VIII.C.5.b, VIII.C.5.d, VIII.D.3.a; Ex-1003 ¶497.

c. 11[c]

See §§VIII.C.3.e, VIII.A.4.c, VIII.C.5.e, VIII.D.3.a; Ex-1003 ¶498.

E. Ground 2C: Obviousness over Aksu in view of Gonzalez, Fu, and B118

1. Fu

Fu is an International Patent Application filed on August 21, 2008 and published on March 19, 2009. Fu constitutes prior art under §102(b).

Fu discloses a method and device for “reducing block distortion” using a filter. Fu, Cover, 1:15-38. Fu provides a deblocking process that applies specific offset equations for the pixels closest to the boundary (p_0/q_0):

$$\begin{aligned}\text{delta} &= \text{Clip3}(-C, C, (((q_0 - p_0) \times 3 + (p_1 - q_1) + 4) \gg 3)) \\ P_0 &= \text{Clip1}(p_0 + \text{delta}) \\ Q_0 &= \text{Clip1}(q_0 - \text{delta}),\end{aligned}$$

Fu, 8:5-9. Fu also provides offset equations for the pixels second closest to the boundary (p_1/q_1):

$$\begin{aligned}P_1 &= \text{Clip1}(p_1 + \text{Clip3}(-C, C, ((P_0 - p_1) \times 3 + (p_2 - Q_0) + 4) \gg 3)) \\ Q_1 &= \text{Clip1}(q_1 - \text{Clip3}(-C, C, ((q_1 - Q_0) \times 3 + (P_0 - q_2) + 4) \gg 3))\end{aligned}$$

Id. (offset underlined); Ex-1003 ¶¶156-57.

2. B118

B118 is a working draft of a Joint Video Team (“JVT”) standard (what became H.264), and was published on or around March 15, 2002. B118, 1. B118 constitutes prior art under at least §102(b).⁸

B118 discloses a deblocking filter, B118, 46, and provides the offset equations used for filtering the p_0/q_0 pixels (which B118 calls p_1/q_1):

$$\begin{aligned}\Delta &= \text{Clip}(-C, C, ((q_1 - p_1) \ll 2 + (p_2 - q_2) + 4) \gg 3) \\ P_1 &= \text{Clip}(0, 255, (p_1 + \Delta)) \\ Q_1 &= \text{Clip}(0, 255, (q_1 - \Delta))\end{aligned}$$

B118, 48; Ex-1003 ¶¶158-61.

⁸ B118 was publicly accessible, Ex-1005 ¶¶71-88, and those interested knew where to find it, Ex-1118, 20:2-6; Ex-1119 ¶7; Ex-1120, 2:42-45; Ex-1121 ¶3; Ex-1122 ¶4.

B118 then determines whether to filter a second pixel in each block, and, if so, applies the following offset equations for the p1/q1 pixels (which B118 calls p2/q2):

$$\begin{aligned} P_2 &= p_2 + \text{Clip}(-C0, C0, \underline{(p_3 + P_1 - p_2 \ll 1) \gg 1}) \\ Q_2 &= q_2 + \text{Clip}(-C0, C0, \underline{(q_3 + Q_1 - q_2 \ll 1) \gg 1}) \end{aligned}$$

B118, 48; Ex-1003 ¶¶162-63.

3. Claim 10

a. 10[a]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[a].

Aksu discloses that “filtering is carried out using a suitable filter. The invention does not limit the kind of filter that can be used,” but Aksu provides an averaging filter as one example. Aksu, 15:1-8; Ex-1003 ¶¶499-500.

Because Aksu leaves the selection of a particular filter to a POSITA, a POSITA would have been naturally motivated to explore alternatives. Ex-1003 ¶503; *Apple Inc. v. Masimo Corp.*, IPR2022-01291, Paper 75, 39 (Jan. 30, 2024) (leaving choice to POSITA, coupled with prior-art options, provides motivation). A POSITA would have found that the art predominantly used weighted-offset equations in deblocking filters, and would have appreciated that these offset equations were efficient and provided better control over the filter's frequency

response than Aksu's exemplary averaging filter. Ex-1003 ¶¶501-03; Gonzalez, 153, 273; §VIII.A.7.a.

Indeed, a POSITA would have known that Aksu's averaging filter was replaced with a weighted-offset equation during subsequent standards developments. Ex-1003 ¶¶389-92, 503; B118, 48.

A POSITA also would have appreciated that these prior-art, first-offset equations varied in two ways: (1) the coefficients assigned to each pixel difference; and (2) the number of pixel differences used, to control the frequency response. Ex-1003 ¶¶235, 503; §VIII.A.7.a. Given the prevalence of prior-art equations that differed only as to these two variables, a POSITA would have known that these equations were a point of routine experimentation. Ex-1003 ¶¶235, 503; §VIII.A.7.a.

A POSITA would have therefore experimented with a weighted-offset equations like Fu's or B118's (or a similar equation, §VIII.A.7.a), and optimized those two variables within the known design constraints to arrive at a first-offset equation that balances the desired deblocking performance (by controlling the filter's frequency response) with the computational resources needed. Ex-1003 ¶¶504-05; §VIII.A.7.a. A POSITA would have found the claimed offset equation obvious because it is within the workable range established by the prior art; the claimed weights (9 and 3) simply scaled Fu's weights (6 and 2) by 50%, meaning *slightly* more filtering would be applied, or were minimally different than

Bjontegaard's (changing an 8 to a 9) and H.264's (changing 8 and 2 to 9 and 3) weights, while still meeting all of the other offset-design considerations (ratio of 3, approximate offset of $\frac{3}{8}$, four pixels, DC gain of 1). Ex-1003 ¶¶505-06; §VIII.A.7.a.

Aksu's "digital signal processor" programmed "to apply predetermined processing functions" to implement the method, Aksu, 17:19-23, would also obviously include calculators to calculate the offset, §VIII.C.5.b; Ex-1003 ¶507.

b. 10[b]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[b].

A POSITA would have followed the well-known technique of applying an offset to a first pixel by adding the offset. Fu at 8:3-10 ("P0=Clip1(**p0**+delta)"); B118, 48 (same); H.241, 29 (same); B205, 131 (same); Ex-1037, 207 (same); Ex-1003 ¶¶508-09.

Aksu's "digital signal processor" programmed "to apply predetermined processing functions" to implement the method, Aksu, 17:19-23, would also obviously include pixel modifiers to apply the offsets. Ex-1003 ¶510.

c. 10[c]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[c].

A POSITA would have followed the well-known technique of applying an offset to a first pixel in a neighboring block by subtracting the offset. Fu at 8:3-10 (“Q0=Clip1(**q0**-delta)”); B118, 48 (same); H.241, 29 (same); B205, 131 (same); Ex-1037, 207 (same); Ex-1003 ¶¶511-13; §VIII.E.3.b.

d. 10[d]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[d].

Like the first offset, Aksu leaves the selection of the filter to a POSITA, but describes an averaging filter applied to all pixels as exemplary. Aksu, 15:1-8; Ex-1003 ¶¶514-15.

A POSITA have been motivated to experiment with prior-art, second-offset equations to use in Aksu's deblocking filter. Ex-1003 ¶515; §VIII.E.3.a. A POSITA would have appreciated that several prior-art deblocking filters had combined two well-known components into a second-offset equation: (1) a high-pass component; and (2) a first-offset component, which differed only as to: (1) the exact high-pass component used; and (2) the coefficient assigned to each component, as shown (with the high-pass component identified in blue and the coefficients identified in red).

Ex-1003 ¶516; Fu, 8:17 (disclosing equivalent of $\Delta_{p1} = \frac{1(p2-3p1+3p0-q0)}{8} + \frac{1\Delta_{p0}}{2}$); B118, 48 (disclosing equivalent of $\Delta_{p1} = \frac{1(p2-2p1+p0)}{2} + \frac{1\Delta_{p0}}{2}$); Bjontegaard, 20, §VIII.A.7.d (disclosing equivalent of $\Delta_{p1} = \frac{5(p2-2p1+p0)}{16} + \frac{5\Delta_{p0}}{16}$). Incorporating a first-offset component ($\Delta/2$) into the second offset was extremely well-known, but a POSITA would have understood that incorporating an additional high-pass filter component would provide further control over filtering high frequencies near the block boundary, with the coefficients simply controlling how much each component contributes to the second offset. Ex-1003 ¶516; §VIII.A.7.d.

A POSITA would have been motivated to experiment with different high-pass-filter equations for the high-pass component, and the coefficients assigned to each of the two components, because these were known result-effective variables to control the frequency response of the second offset. Ex-1003 ¶¶517-20. A POSITA would have found it obvious to use a Laplacian as the high-pass component (like B118 and Bjontegaard) because the Laplacian was an extremely well-known and simple high-pass filter. Ex-1003 ¶520. A POSITA also would have also found it obvious to weight the Laplacian by $\frac{1}{4}$ depending on how much additional high frequency a POSITA sought to remove, because such a coefficient produced outputs substantially the same as Fu's high-pass component, yet used one less pixel and was thus more efficient. Ex-1003 ¶¶517-20. Moreover, a POSITA would have found it

obvious to weight the first-offset component by $\frac{1}{2}$, because that was common in the art for the second-offset equation, including when combined with a high-pass component. Fu, 8:17 (weighting $\Delta/2$); B118, 48 (same); H.241, 29 (same); B205, 131 (same). At base, a POSITA would have found a second-offset equation $\Delta_{p1} = \frac{1(p2-2p1+p0)}{4} + \frac{1\Delta}{2} = \frac{(p2-2p1+p0)+2\Delta}{4}$, as claimed, an obvious result of routine experimentation. Ex-1003 ¶520.

Aksu's "digital signal processor" would also obviously include calculators to calculate the offset. §VIII.C.5.b; Ex-1003 ¶521.

e. 10[e]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[e].

The Aksu-Gonzalez combination discloses filtering third-pixel r_2 , second closest to the boundary in block r , when d_r is less than a threshold based on β (or β_r). §§VIII.C.4.b-VIII.C.4.d.

A POSITA would have followed the well-known technique of applying an offset to second-closest pixel r_2 by adding the offset. Fu, 8:15-17 ("P1=Clip1(**p1**+[offset]"); B118, 48 (same); H.241, 29 (same); B205, 131 (same); Ex-1037, 207 (same); Ex-1003 ¶¶522-25; §VIII.E.3.b (pixel modifiers).

f. 10[f]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[f].

A POSITA would have understood that the third-offset equation is simply the second-offset equation corresponding to the neighboring (q) block, and that the prior-art taught using corresponding equations with the two above-identified high-pass and first-offset components. Ex-1003 ¶¶526-28; Fu, 8:18 (disclosing equivalent of $\Delta_{q1} = \frac{1(q2-3q1+3q0-p0)}{8} - \frac{1\Delta}{2}$); B118, 48 (disclosing equivalent of $\Delta_{q1} = \frac{1(q2-2q1+q0)}{2} - \frac{1\Delta}{2}$); Bjontegaard, 20, §VIII.A.7.f (disclosing equivalent of $\Delta_{q1} = \frac{5(q2-2q1+q0)}{16} - \frac{5\Delta_{q0}}{16}$). As with the second-offset equation, a POSITA would have been motivated to experiment with the high-pass component and coefficients assigned to each component, and would have found the claimed third-offset equation $\Delta_{q1} = \frac{1(q2-2q1+q0)}{4} - \frac{1\Delta}{2} = \frac{(q2-2q1+q0)-2\Delta}{4}$, as claimed, obvious. Ex-1003 ¶529; §VIII.E.3.d.

Aksu's "digital signal processor" would also obviously include calculators to calculate the offset. §VIII.C.5.b; Ex-1003 ¶530.

g. 10[g]

Aksu in view of Gonzalez, Fu, B118, and a POSITA's knowledge teaches 10[e].

The Aksu-Gonzalez combination discloses filtering fourth-pixel l_2 , second closest to the boundary in neighboring block l , when d_l is less than a threshold based on β (or β_l). §§VIII.C.4.f-VIII.C.4.h.

A POSITA would have followed the well-known technique of applying an offset to the second-closest pixel l_2 in the neighboring block. Fu, 8:15-17 (“ $Q1 = \text{Clip1}(q1 - [\text{offset}])$ ”); B118, 48 (“ $Q2 = q2 + [\text{offset}]$ ”); H.241, 29 (same); B205, 131 (same); Ex-1037, 208 (same); Ex-1003 ¶¶531-32. Although Fu *subtracted* the second offset, this is the mathematical equivalent of *adding* its negative, which simply depends on how the equation is written but does not change the result. Ex-1003 ¶¶528, 532.

Aksu’s “digital signal processor” would also obviously include pixel modifiers to apply the offsets, §VIII.E.3.b; Ex-1003 ¶533.

IX. Discretionary Factors Favor Institution

A. *Fintiv*

The Board should not deny institution on the basis of *Fintiv*.

The ’841 Patent is being litigated before the district court and ITC. *Fintiv* does not apply to ITC proceedings. K. Vidal, Memorandum, U.S. PAT. AND TRADEMARK OFFICE, 2-3 (June 21, 2022).

For the district court, the relevant *Fintiv* factors weigh heavily against a discretionary denial. Those proceedings were initiated in December 2023,

Telefonaktiebolaget LM Ericsson v. Lenovo (United States), Inc., No. 5-23-cv-00714, ECF No. 1 (EDNC Dec. 12, 2023), no schedule has been entered, and the case has been stayed pending resolution of the ITC proceedings, ECF No. 40 (EDNC Feb. 16, 2024). Any district court trial would therefore be well after the FWD deadline (likely December 2025). Further, it is unclear whether any overlap exists between this Petition and any district court issues. Discretionary denial under *Fintiv* is therefore unwarranted.

B. 325(d)

The Board should 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).

Considering *Advanced Bionics* part one (Becton factors (a), (b), (d)), apart from B205—which is relied upon in the Petition for a narrow purpose—no other reference presented here was previously before the Office. Moreover, the examiner never considered a basis of invalidity that relied primarily upon either Bjontegaard or Aksu. Nor is either reference cumulative of those before the examiner, because the prior art the examiner relied upon lacked block-specific decisions that both primary references disclose. Thus, the same or substantially same art or arguments

were not previously presented to or considered by the Office; the Board need not reach *Advanced Bionics* part two.

X. Mandatory Notices

A. Real Party in Interest

In addition to itself, Petitioner identifies Lenovo (United States), Inc., Lenovo (Shanghai) Electronics Technology Co. Ltd., Lenovo Beijing, Ltd., Lenovo PC HK Limited, Lenovo Group, Ltd., Lenovo Information Products (Shenzhen) Co., Ltd. as real parties in interest.

B. Related Matters

To the best of Petitioner's knowledge, the '841 Patent has been involved in the following matters:

- *Telefonaktiebolaget LM Ericsson v. Lenovo (United States), Inc.*, No. 5-23-cv-00714 (EDNC);
- Certain Electronic Computing Devices, and Components and Modules Thereof; Inv. No. 337-TA-1387 (ITC).

C. Notice of Counsel and Service Information

Lead Counsel: Jessica C. Kaiser Reg. No. 58,937 PERKINS COIE LLP 1900 Sixteenth St., Suite 1400 Denver, CO 80202-5255 (303) 291-2300 (telephone) (303) 291-2400 (fax) kaiser-ptab@perkinscoie.com	Backup Counsel: Vinay Sathe Reg. No. 55,595 PERKINS COIE LLP 11452 El Camino Real, Suite 300 San Diego, CA 92130-2080 (858) 720-5700 (telephone) (858) 720-5799 (fax) Sathe-ptab@perkinscoie.com
	Matthew Lembo Reg. No. 75,633 PERKINS COIE LLP 1155 Avenue of the Americas 22nd Floor New York, NY 10036-2711 (212) 262-6900 (telephone) (212) 977-1649 (fax) Lembo-ptab@perkinscoie.com Tyler Kendrick Reg. No. 73,930 PERKINS COIE LLP 1201 Third Avenue, Suite 4900 Seattle, WA 98101-3099 (206) 359-8000 (telephone) (206) 359-6507 (fax) Kendrick-ptab@perkinscoie.com Anna Onley Reg. No. 73,876 PERKINS COIE LLP 110 N Wacker Dr Suite 3400, Chicago, IL 60606 (312) 324-8400 (telephone) (312) 324-9400 (fax)

	Onley-ptab@perkinscoie.com Louis Yang Reg. No. 79,056 PERKINS COIE LLP 110 North Wacker Drive, Suite 3400 Chicago, IL 60606-1511 (312) 324-8400 (telephone) (312) 324-9400 (fax) yang-ptab@perkinscoie.com
--	--

D. Service Information

Please address all correspondence to lead and back-up counsel at the addresses shown above, and as follows. Petitioners consent to electronic service by e-mail.

Intel-Ericsson-IPRs@perkinscoie.com

XI. Conclusion

Inter Partes Review of the challenged claims is respectfully requested.

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: June 14, 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 9,641,841, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,984. 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: June 14, 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. 9,641,841 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:

MARCONI
1717 McKinney Avenue
Suite 1050
Dallas, TX 75202

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

Adam D. Swain - Adam.swain@alston.com
John D. Haynes - John.haynes@alston.com
Shri Abhyankar - Shri.abhyankar@alston.com
David S. Frist - David.frist@alston.com
Katherine M. Donald - Katie.donald@alston.com
Stephen G. McNiff - Steve.mcniff@alston.com
Shawn Gannon - Shawn.gannon@alston.com
Sloane Kyrakis - Sloane.kyrakis@alston.com
Stephen R. Lareau - Stephen.lareau@alston.com
Karlee N. Wroblewski - Karlee.wroblewski@alston.com
Theodore Stevenson - ted.stevenson@alston.com
Elliott C. Riches - elliottriches@alston.com
Madeline E. Byrd - maddy.byrd@alston.com
Ryan Koppelman - Ryan.koppelman@alston.com
Steven M. Anzalone - steven.anzalone@fisherbroyles.com
Gary Rinkerman - gary.rinkerman@fisherbroyles.com
Michael M. Murray - mike.murray@fisherbroyles.com
Kevin D. Jablonski - kevin.jablonski@fisherbroyles.com
Nicholas Mathews - nmathews@McKoolSmith.com
Alexander J. Chern - achern@McKoolSmith.com

Casey L. Shomaker - cshomaker@McKoolSmith.com
Kevin P. Hess - khess@McKoolSmith.com
R. Mitch Verboncoeur - mverboncoeur@McKoolSmith.com
Joshua J. Newcomer - jnewcomer@McKoolSmith.com
Blake H. Bailey - bbailey@mckoolsmith.com

Respectfully submitted,

/ Jessica Kaiser /

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

Date: June 14, 2024