

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,

Petitioners

v.

TELEFONAKTIEBOLAGET LM ERICSSON,

Patent Owner.

IPR2024-01008

U.S. PATENT NO. 9,641,841

DECLARATION OF DR. IAIN RICHARDSON

TABLE OF CONTENTS

I.	INTRODUCTION	16
II.	QUALIFICATIONS	17
III.	MATERIALS CONSIDERED	19
IV.	SCOPE OF OPINIONS	20
V.	LEGAL STANDARDS	20
A.	Presumption of Validity	20
B.	Anticipation	22
C.	Obviousness.....	23
D.	Motivations to Combine	25
E.	Objective Indicia	26
VI.	LEVEL OF ORDINARY SKILL IN THE ART	26
VII.	CLAIM CONSTRUCTION	27
VIII.	TECHNOLOGY BACKGROUND.....	28
A.	Introduction to Video Coding	28
B.	Introduction to Deblocking Filters	34
IX.	U.S. PATENT NO. 9,641,841	43
A.	'841 Patent Invention – Ericsson’s Deblocking Filter	43
B.	Additional Details Regarding Deblocking Filtering in the Draft H.265 Standard	67
C.	Priority Date	70
D.	Prosecution History	71
X.	ASSERTED REFERENCES	76
A.	Bjontegaard (Ex-1009).....	76
B.	Bjontegaard’s Code (Ex-1010-1015)	82
C.	H.241 (Ex-1017).....	91
D.	B205/AAPA	93
E.	Aksu (Ex-1019)	96
F.	Gonzalez (Ex-1020)	102
G.	Fu (Ex-1022)	103
H.	B118 (Ex-1023).....	107

I.	VCEG-M21	111
J.	Bjontegaard-048	111
XI.	SIMULATION OF DEBLOCKING FILTER DESIGNS.....	112
A.	Testing Methodology	112
B.	Matlab Scripts.....	116
C.	Analysis of Test Simulation Results	126
-	Using Thr1/Thr2 from the '841 patent and using the Q15-J-28 disclosed method for calculating Diff values:	128
-	Using Thr1/Thr2 from the '841 patent and the 'image.c' method for calculating Diff values:	131
-	Using S01/S02 from Q15-J-28 as the 'threshold' comparisons and the Q15-J-28 disclosed method for calculating Diff values:	131
-	Using S01/S02 from Q15-J-28 as the 'threshold' comparisons and the 'image.c' method for calculating Diff values:	131
XII.	PETITIONER'S GROUNDS FOR CANCELLATION DO NOT RENDER THE CHALLENGED CLAIMS INVALID BECAUSE THEY ARE NOT OBVIOUS.....	133
D.	Petitioner's Ground 1A Fails Because The Challenged Claims Are Not Obvious In View of Bjontegaard And Bjontegaard's Code.....	133
1.	Claim 1, 5, 16.....	137
2.	Claim 10 (Ground 1A).....	153
E.	Petitioner's Ground 1B Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Bjontegaard, Bjontegaard's Code, H.241, and B205/AAPA	171
F.	Petitioner's Ground 2A Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Aksu and Gonzalez	172
1.	No Motivation to Combine Aksu and Gonzalez.....	172
G.	Claims 1, 5, 16.....	173
H.	Petitioner's Ground 2B Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Aksu, Gonzalez, H.241, and B205/AAPA	186

I.	Petitioner’s Ground 2C Fails Because The Challenged Claims Are Not Obvious in View of the Combination of Aksu, Gonzalez, Fu, and B118	186
1.	No Motivation to Combine Aksu with Gonzalez, Fu, and/or B118.....	186
J.	Claim 10 (Ground 2C).....	188
XIII.	SUMMARY AND OTHER REMARKS	200

TABLE OF EXHIBITS

Exhibit	Description
Ex-2002	<i>Curriculum Vitae</i> of Dr. Iain Richardson
Ex-2003	ITU Webpage showing ITU-T Rec. H.261, first published in 1988, https://www.itu.int/rec/T-REC-H.261-198811-S/en
Ex-2004	PDF Print of FiltScript2.m
Ex-2005	PDF Print of TestFilt2.m
Ex-2006	PDF Print of FiltBlocksAltThr 2a.txt
Ex-2007	PDF Print of FiltBlocksAltThr 2b.txt
Ex-2008	PDF Print of FiltBlocksThr1Thr2_2a.txt
Ex-2009	PDF Print of FiltBlocksThr1Thr2_2a.txt
Ex-2010	PDF Print of FiltStatsAltThr 2a.txt
Ex-2011	PDF Print of FiltStatsAltThr 2b.txt
Ex-2012	PDF Print of FiltStatsThr1Thr2_2a.txt
Ex-2013	PDF Print of FiltStatsThr1Thr2_2b.txt
Ex-2014	PDF Print of TestUint8.c
Ex-2015	Sullivan et al., <i>HEVC: Algorithms and Architectures</i> (2014)

CLAIM LISTING

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_{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;
1[b]	calculating a second filter decision value for said neighboring block, by only using pixels q_{2i}, q_{1i}, and q_{0i}, based on: $ q_{2i}-2q_{1i}+q_{0i} $ wherein q_{0i} 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_{1i} 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_{2i} 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

Reference	Claim Limitation
	boundary based on a comparison of said second filter decision value to a second threshold value.
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_{20}-2p_{10}+p_{00} + p_{23}-2p_{13}+p_{03} $ wherein p_{00} denotes said pixel value of said pixel closest to, in said first line of pixels, said block boundary, p_{10} denotes said pixel value of said pixel next closest to, in said first line of pixels, said block boundary, p_{20} denotes said pixel value of said pixel second next closest to, in said first line of pixels, said block boundary, p_{03} denotes a pixel value of a pixel closest to, in a second line of pixels in said block, said block boundary, p_{13} denotes a pixel value of a pixel next closest to, in said second line of pixels, said block boundary and p_{23} 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_{20}-2q_{10}+q_{00} + q_{23}-2q_{13}+q_{03} $ wherein q_{00} denotes said pixel value of said pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary, q_{10} 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_{20} 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_{03} 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_{13} 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_{23} denotes a pixel value of a pixel in said neighboring block second

Reference	Claim Limitation
	next closest to, in said corresponding second line of pixels, said block boundary.
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]	second decision value calculator configured to calculate a second filter decision value for a neighboring block, by only using pixels q_{2i}, q_{1i}, and q_{0i}, based on: $ q_{2i}-2q_{1i}+q_{0i} $ wherein q_{0i} 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_{1i} 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_{2i} 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.
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_{20}-2p_{10}+p_{00} + p_{23}-2p_{13}+p_{03} $ wherein p_{00} denotes said pixel value of said pixel closest to, in said first line of pixels, said block boundary, p_{10} denotes said pixel value

Reference	Claim Limitation
	of said pixel next closest to, in said first line of pixels, said block boundary, p_{20} denotes said pixel value of said pixel second next closest to, in said first line of pixels, said block boundary, p_{03} denotes a pixel value of a pixel closest to, in a second line of pixels in said block, said block boundary, p_{13} denotes a pixel value of a pixel next closest to, in said second line of pixels, said block boundary and p_{23} 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_{20}-2q_{10}+q_{00} + q_{23}-2q_{13}+q_{03} $ wherein q_{00} denotes said pixel value of said pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary, q_{10} 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_{20} 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_{03} 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_{13} 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_{23} 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.
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

Reference	Claim Limitation
	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.
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 $9 \times ((q_{0j} - p_{0j}) - 3 \times (q_{1j} - p_{1j})) / 16$ wherein p_{0j} denotes a pixel value of a pixel closest to, in said first line of pixels, said block boundary, p_{1j} denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary, q_{0j} denotes a pixel value of a pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary and q_{1j} 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

Reference	Claim Limitation
	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 $(p_{0j} + p_{2j} - 2p_{1j} + 2\Delta)/4$ wherein p_{2j} denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;
10[e]	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 $(q_{0j} + q_{2j} - 2q_{1j} - 2\Delta)/4$ wherein q_{2j} 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	

Reference	Claim Limitation
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
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]	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	

Reference	Claim Limitation
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} $ 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;
16[b]	calculate a second filter decision value for said neighboring block, by only using pixels q_{2i}, q_{1i}, and q_{0i}, based on: $ q_{2i}-2q_{1i}+q_{0i} $ wherein q_{0i} 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_{1i} 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_{2i} 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

Reference	Claim Limitation
	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.

DECLARATION OF DR. IAIN RICHARDSON

I, Dr. Iain Richardson, do hereby declare and say as follows:

I. INTRODUCTION

1. My name is Dr. Iain Richardson, and I have been retained as a technical expert by counsel for Patent Owner Telefonaktiebolaget LM Ericsson (“Ericsson” or “Patent Owner”) to investigate and opine as to the validity of U.S. Patent No. 9,641,841 (“the ’841 Patent”) in view of Petitioner Intel Corp.’s (“Petitioner” or “Intel”) Petition for *Inter Partes* Review (the “Petition”) (Paper 1).

2. This Declaration is based on information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, and thus may expand or modify my opinions as my investigation and study continues. I may also supplement my opinions in response to any additional information that becomes available to me, any matters raised by Respondents and/or opinions rendered by Respondents’ experts, or in light of any relevant orders from the Administrative Law Judge.

3. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials. A list of materials considered is included as Exhibit 2002 to my Declaration.

4. My company Vcodex BV is being compensated at the rate of \$900 per hour for my work in this case, including time spent testifying. I am also being

reimbursed for reasonable fees and expenses, including hotel and travel expenses, incurred as a result of my work in this case. Neither I nor my company has received any additional compensation for my work in this Investigation. Neither Vcodex BVs compensation nor any personal remuneration I receive from my company Vcodex BV or from my company Vcodex Ltd is tied in any way to the substance of my testimony or the outcome of this Investigation. I have no other financial interest in this Investigation or to the parties. I am available to offer these opinions at deposition and at trial if called upon to do so.

II. QUALIFICATIONS

5. I have summarized in this section my educational background, work experience, and other relevant qualifications. A true and correct copy of my curriculum vitae is attached as Exhibit 2001 to Patent Owner's Preliminary Response which includes a list of all other cases in which, during the previous four years, I have testified at trial or by deposition.

6. I am Director and co-founder of Vcodex BV and Vcodex Limited. I am an internationally recognized expert on video and image compression coding with 30 years of experience in the field of video, image, multimedia, and general-purpose data encoding, decoding, compression, decompression, and transmission.

7. I obtained a Masters of Engineering from Heriot-Watt University, Edinburgh, UK, in 1990. I received a Doctor of Philosophy (Ph.D.) in video

compression coding from Robert Gordon University, Aberdeen, UK in 1999. I served as a full professor at Robert Gordon University for 17 years, where I taught undergraduate and graduate engineering courses on topics, including signal processing, communications engineering, image and video processing, and multimedia compression.

8. In 1993, I assumed a post as a Lecturer, then Reader and eventually a Full Professor in the field of image and video compression in the School of Engineering at Robert Gordon University. In 2009, I was awarded the position of Honorary Professor with the Robert Gordon University.

9. I am the author of five books and over 70 journal and conference papers on video compression, including two widely cited books on the H.264/MPEG-4 standards. I am regularly invited to give lectures on video compression coding, including at the Smithsonian Institution and the US Patent and Trademark Office.

10. I am a former member of British Standards Institute expert group IST/37, the committee which represents the UK in ISO/IEC MPEG standardization. I am currently a member of NEN, the Netherlands equivalent group.

11. I have many years of experience in researching, developing and implementing video compression codecs in software and hardware. In 1995, during my employment at Robert Gordon University, I co-founded and became Technical Director, equivalent of Chief Technology Officer, at 4i2i Communications Limited.

I held this position until 2000. In my role at 4i2i, I was responsible for and engaged in the design of software and hardware modules for compression and communication systems, including the design of video processing algorithms in software and hardware for implementation on processing devices and the design of compression algorithms such as block-based transforms, run-length coding and variable-length coding in software and hardware. A modified version of one of my video compression hardware designs was successfully used by NASA to stream video from the International Space Station.

12. In 2007, I founded Vcodex Limited and currently serve as the CEO, through which I provide technical and strategic advice to technology companies particularly relating to video, image, and data compression for streaming.

13. In 2009, I founded Onecodec Limited and served as CEO, leading the company's development of innovative video, image, and multimedia data compression and storage software and systems. Onecodec's technology and business activities were merged with Vcodex Limited in 2015, and Vcodex B.V. in 2019.

14. I am a named inventor on 11 patents or patent applications, including U.S. and non-US patents and applications.

15. I believe I am qualified to render the opinions set forth here.

III. MATERIALS CONSIDERED

16. The materials that I considered and relied upon in preparing my Declaration and forming my opinions include all exhibits cited in this Declaration and the Patent Owner's Response, as well as all exhibits included in the Petition (Paper 1) and Dr. Schonfeld's Declaration (Ex-1003), including the '841 Patent, the '841 file history, all of the relevant prior art, and the simulations I created (Ex-2004-2014).

17. I have further relied on my academic and professional experience in reaching the opinions expressed in this Declaration.

IV. SCOPE OF OPINIONS

18. I set forth my opinions throughout this Declaration.

19. It is my opinion that the Asserted Claims of the '841 Patent are not anticipated and/or obvious in view of any of the asserted references, alone or in combination, discussed herein.

V. LEGAL STANDARDS

20. For purposes of this Declaration, I use the legal principles below as a guide in formulating my opinions. I am not an attorney, and I am not offering any opinions regarding legal matters; however, I have been informed by Ericsson's counsel of the legal principles relevant to the issues herein and have the following understanding.

A. Presumption of Validity

21. I understand that in deciding whether to issue a patent, the U.S. Patent and Trademark Office (“USPTO”) examines the patent specification, its claims, and relevant prior art references to determine whether the patent application and its claims meet the requirements for patentability. I understand that if the USPTO then issues the patent, it has determined that the claims are valid. I understand that the law recognizes that the USPTO possesses special expertise to conduct such determinations, and that, as a result, each claim of a patent issued by the USPTO is presumed valid by law. This presumption of validity can be overcome only if the party seeking to invalidate a claim proves invalidity by clear and convincing evidence, which I understand to mean evidence that convinces you that it is highly probable that the particular proposition is true. I understand that clear and convincing evidence is more than a preponderance of the evidence but less than beyond a reasonable doubt. I understand that the clear and convincing evidence standard applies to all aspects of an allegation of anticipation or obviousness, including the prior art status of the relevant reference(s).

22. I understand the first step in determining whether or not a patent claim is valid is to properly construe the claims. I also understand that the claims must be construed the same way in determining validity/invalidity and infringement/non-infringement. I also understand that, as a general matter, a claim should not be limited to a preferred embodiment, but that in certain cases, the scope of the right to

exclude may be limited by a narrow disclosure or by positions taken, such as by statements made during patent prosecution. I also understand the claims must be supported by the specification. Also, to the extent that a patent claims priority to an earlier filed application, the claims must be supported by the disclosure in that application. For terms that have not been construed I understand that they should be afforded their plain and ordinary meaning to one of ordinary skill in the art.

B. Anticipation

23. I have been informed by counsel and understand that a patent is invalid on the basis of anticipation (under 35 U.S.C. § 102) if a single prior art reference discloses, either expressly or inherently, each and every limitation of the claimed invention. If the single prior art reference fails to disclose even one claim limitation, it does not anticipate the claim.

24. I understand that a single prior art reference may anticipate without disclosing a feature of the claimed invention if such feature is necessarily present, or inherent, in that reference

25. I have been informed that under the principles of inherency, if the prior art necessarily functions in accordance with or includes the claimed limitations, it anticipates. However, the prior art reference must necessarily include the non-disclosed limitation. Mere probabilistic inherency, nor the presence of an

unrecognized *de minimis* quantity of a claimed substance in the prior art, cannot anticipate a later patent or patent application.

26. I understand that a prior art reference alleged to be anticipatory must also enable one of ordinary skill in the art to make the claimed invention without undue experimentation. There exists a rebuttable presumption prior art patents are enabled.

C. Obviousness

27. I further understand that a claimed invention is not patentable under 35 U.S.C. § 103 if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. Obviousness, as I understand it, is based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and secondary indicia of obviousness to the extent they exist.

28. I understand that whether there are any relevant differences between the prior art and the claimed invention is to be analyzed from the view of a person of ordinary skill in the art at the time of the invention. A person of ordinary skill in the art is a hypothetical person who is presumed to be aware of all of the relevant art at the time of the invention. The person of ordinary skill is not an automaton, and may be able to fit together the teachings of multiple patents employing ordinary

creativity and the common sense that familiar items may have obvious uses in another context or beyond their primary purposes.

29. I understand that if a proposed combination of references does not teach or suggest or suggest all of the elements of a claim, the combination cannot render the claim obvious. I also understand that in order to combine prior art references to show obviousness, a person of ordinary skill in the art, without knowledge of the claimed invention, or without the use of hindsight, must have been motivated to combine the prior art by some suggestion or teaching in the prior art, the knowledge of one of skill in the art, or the nature of the problem to be solved.

30. An invention is obvious if a designer of ordinary skill in the art, facing the wide range of needs created by developments in the field, would have seen an obvious benefit to the solutions tried by the applicant. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, it would be obvious to a person of ordinary skill to try the known options. If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique would have been obvious.

31. I understand that obviousness of a patent claim cannot properly be established through hindsight, and that elements from different prior art references, or different embodiments of a single prior art reference, cannot be selected to create

the claimed invention using the invention itself as a roadmap. I understand that the claimed invention as a whole must be compared to the prior art as a whole, and courts must avoid aggregating pieces of prior art through hindsight that would not have been combined absent the inventors' insight.

D. Motivations to Combine

32. I understand that in deciding obviousness, it is appropriate to consider whether some teaching, suggestion, or motivation would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that it is appropriate to look to a variety of factors when evaluating reasons to combine references in an obviousness analysis, and that it is important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the prior art elements in the manner claimed. I understand that in determining whether a combination of prior art references renders a claim obvious, it may be helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in doing so. I understand, however, that the teaching, suggestion, or motivation to combine inquiry is not required and may not be relied upon in lieu of the obviousness analysis outlined above. I understand that a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. I also understand that this is so because

inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known.

E. Objective Indicia

33. I understand that in evaluating whether patent claims are invalid as obvious, objective indicia of non-obviousness can be considered. Such objective indicia can include commercial success, long-felt but unsolved needs, copying, praise, unexpected results, industry acceptance, failure of others, and skepticism by experts.

VI. LEVEL OF ORDINARY SKILL IN THE ART

34. I understand that patent claims are to be interpreted from the viewpoint of one of ordinary skill in the relevant art. To determine the level of skill that would be “ordinary,” I understand that a person of ordinary skill (“POSITA”) must generally have the capability of understanding the general principles that are applicable to the pertinent art.

35. In my opinion, a POSITA would have had a bachelor’s degree in electrical engineering, computer engineering, or a related field, with about two years of experience in video decoding or encoding technologies. More work or practical experience may qualify one not having the requisite education as a POSITA art while a higher level of education could offset less experience. I was at least a person of

ordinary skill in the art as of the effective filing date of the '841 Patent under this definition.

36. I have reviewed Dr. Schonfeld's Declaration (Ex-1003). I understand that Dr. Schonfeld contends that a person of ordinary skill in the art for the '841 Patent "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. However, advanced graduate studies with coursework and/or research focusing on topics of video and/or image coding and processing could substitute for working experience in the field." (Ex-1003 ¶165.)

37. In my opinion, two years of experience in video decoding or encoding technologies is an appropriate level of experience in the context of the '841 Patent. I was at least a person of ordinary skill in the art as of the effective filing dates of the Asserted Patents under their definition.

38. My opinions set forth below would not change regardless of whether my proposed definition or Dr. Schonfeld's definition of a person of ordinary skill in the art is adopted.

VII. CLAIM CONSTRUCTION

39. I understand that the Petitioner and Dr. Schonfeld are not advancing any proposed constructions in the Petition and therefore apply the plain and ordinary meaning of the claims. (See Ex-1003, ¶¶167-168.)

40. However, in the ITC proceedings, I understand the Complainant and Respondents dispute several terms, including the “determining” limitations of the independent claims. I understand that Respondent Lenovo proposes that the claims should be construed such that determining how many pixels to filter in a block can have “no mathematical basis in (i.e. no relationship with)” the pixels in the neighboring block. Ex. 1-26, 6-7. While I disagree that that this proposed construction is proper because it is not consistent with the intrinsic record or with the plain meaning of the claims, in my opinion, none of the prior art discloses or renders obvious the limitations of the Challenged Claims under any proposed construction.

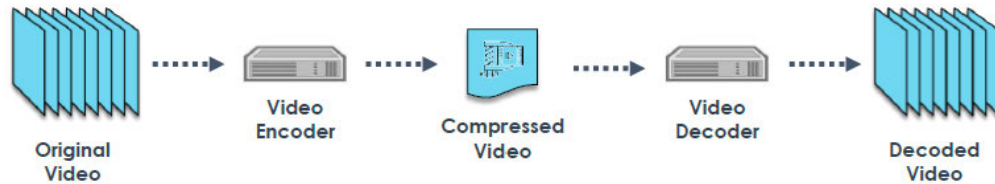
VIII. TECHNOLOGY BACKGROUND

41. Below, I provide a brief overview of video coding technology and the relevant processes involved in encoding and decoding video.

A. Introduction to Video Coding

42. Video compression coding, often abbreviated to video coding, is the process of compressing (encoding) video prior to transmission or storage and decompressing (decoding) compressed video so that it can be displayed. A video

encoder converts digital video into a compressed form known as coded or compressed video, suitable for transmission or storage, and a video decoder converts coded or compressed video into uncompressed digital video. The pair of video encoder and video decoder is known as a video codec:



43. Uncompressed video typically consists of a series of frames, each of which is an image of the visual scene at a particular time instant. Popular formats for uncompressed video include:

- Standard Definition (SD): Around 704 x 576 pixels per frame, displayed at 25 or 30 frames per second.
- High Definition (HD): 1920 x 1080 pixels per frame, displayed at 25 – 60 frames per second.
- Ultra High Definition (UHD) or 4K: 3840 x 2160 pixels per frame, displayed at 50-60 frames per second.

44. Video coding is important because uncompressed video takes up a significant amount of storage space and would require an impractical amount of bandwidth to transmit in real time. For example, 500Gbyte hard disk can only store around 40 minutes of uncompressed HD video. The same hard disk can store around

600 hours of compressed video, depending on the video content and the compression format.

45. Video data takes up the majority of storage and transmission capacity in today's consumer devices and networks, even in its compressed form. As display resolutions get bigger and consumers expect better video quality and higher resolution images when viewing stored video or video streamed over the internet, efficient and effective video compression continues to be of vital importance.

46. The performance of a video codec can be measured in at least three ways, namely compression, quality, and computation performance.

47. Compression describes the extent to which data is reduced in size. This can be expressed as a ratio, *e.g.*, 100:1 compression means that the video, if stored in a file, occupies 1/100th of the space when compressed, compared with the uncompressed video. Alternatively, this can be expressed as a bitrate, *i.e.*, the transmission bitrate required to transmit the compressed video data in real time. For example, a typical compressed HD video clip might require between 2 and 4 Mbps (2-4 million bits per second) of transmission bitrate, to send the video in real time.

48. Quality describes how the compressed and decompressed video appears to a user. Video compression is typically "lossy," which means that the quality of the decoded video is not as good as the original, uncompressed video. For many

consumer video applications, the aim is to provide the viewer with pleasing video quality, whether subjectively evaluated (i.e., how does a human observer rate the quality of the video) or objectively evaluated (i.e., an automatically calculated approximation of visual quality).

49. Computation describes how much processing power is required to encode (compress) or decode (decompress) the video. Video coding uses complicated algorithms and methods that typically require significant computing time and resources and may require to be carried out in real time, for example to ensure continuous playback of decoded video.

50. These three variables—compression, quality, and computation—are usually interrelated. For example, we can often compress more and/or achieve higher video quality by using more computationally intensive compression processes or algorithms.

51. Many different techniques for implementing video coding designs have been researched and proposed. Each year, new and innovative technologies for improving compressions are described in many hundreds of research papers. However, in general, it is common for commercial video coding applications to use a limited number of standardized techniques for video compression. Standardized video coding formats have several potential benefits when compared to non-standard formats including for example, interoperability. Standards for video coding simplify

inter-operability between encoders and decoders from different manufacturers. This is critical in applications where each ‘end’ of the system (e.g., encoder and decoder) could be manufactured or provided by a different company. For example, a provider of streaming video content need not be (and is unlikely to be) the same company that manufactures the laptop or other device that receives, decodes and plays the streaming video content. As long as both the encoder and decoder operate in conformance with the same standard, any conforming decoder should be able to decode a compressed bitstream that was encoded in conformance with the standard.

52. Two specifications for video compression are H.264 (advanced Video Coding or AVC) and H.265 (High Efficiency Video Coding or HEVC). ITU-T Recommendation H.264 is an industry standard for video compression that was first published in 2003. ITU-T Recommendation H.265, also referred to as High Efficiency Video Coding (HEVC) is an industry standard for video compression that was first published in 2013.

53. The H.265 standard can provide around twice the compression performance of the older H.264 standard. This means that an H.265 video encoder (i.e. a video encoder that produces a coded video bitstream that conforms to the H.265 standard) can compress video into a file that is about half the size of the file produced by an H.264 video encoder, for the same decoded video quality. An H.265 video

encoder or decoder is typically more computationally complex than a similar H.264 video encoder or decoder.

54. Both H.264 and H.265 were developed by committees of industry experts. H.264 was developed by the Joint Video Team (JVT) and H.265 was developed by the Joint Collaborative Team on Video Coding (JCT-VC), each comprising experts from the Video Coding Experts Group (a subgroup of ITU-T) and from the Moving Picture Experts Group (a sub-group of ISO/IEC). Both standards were co-published by ITU-T and ISO/IEC.

55. In each case (H.264 and H.265), the process of developing the standard took a number of years. Typically, such a process begins with an identification of a requirement for a new video coding standard and a Call for Proposals. Depending on the response to the Call for Proposals, the working group (JVT or JCT-VC) selects certain technologies to form the basis for further development (sometimes described as a Test Model) and then holds a series of meetings with the aim of continuing to refine the selected technologies. Once the Test Model reaches a certain level of maturity and performance, the working group proposes a Draft Standard which ultimately becomes a published international standard.

56. I provide an overview of Deblocking Filters below and I provide explanations of further aspects of video coding as necessary throughout this Declaration.

B. Introduction to Deblocking Filters

57. Current video codecs, including H.265, utilize hybrid coding schemes. Sullivan et al., *HEVC: Algorithms and Architecture* (2014) (Ex-1004) at 113. These hybrid video coding schemes utilize video sample prediction, transformation of the prediction error, and then entropy coding of the prediction information. *Id.* There are two types of prediction relevant here, intra-prediction, which predicts a block in a current picture based on nearby samples, and inter-prediction, which predicts a block in a current picture that occur at different moments in time. *Id.* at 114. This is known as motion-compensated prediction. An illustration of the inter-picture prediction concept is shown below:

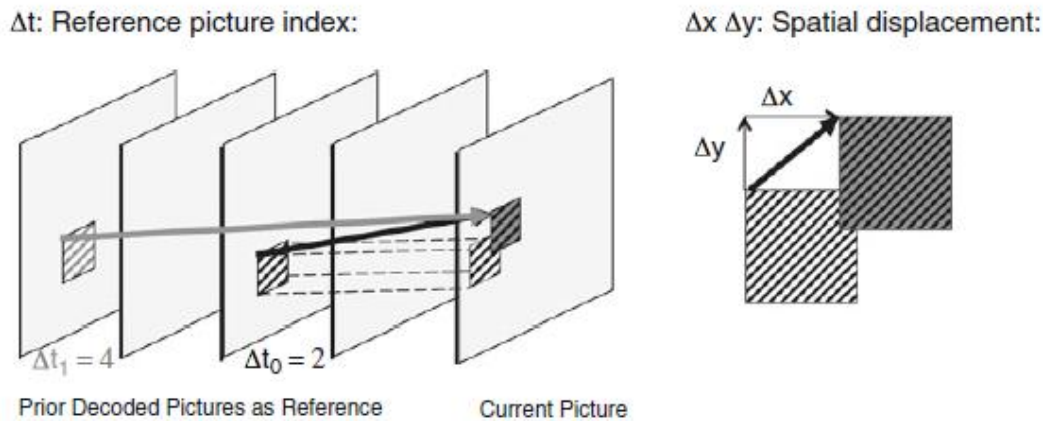


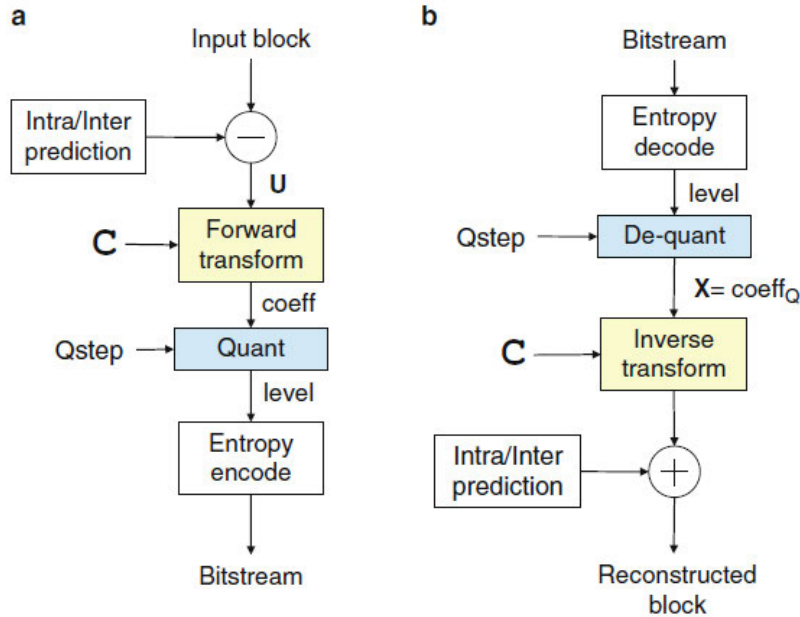
Fig. 5.1 Inter-picture prediction concept and parameters using a translational motion model

Ex-1004 at 114.

58. When using inter-prediction, as shown above, prior decoded pictures can be used as prediction references for subsequent pictures. *Id.* This motion

compensated prediction can break a picture up into blocks, such that the predicted translational motion (i.e. spatial displacement) of a given block can be predicted by a motion vector comprised of $(\Delta x, \Delta y)$. This motion vector represents the horizontal and vertical displacement, respectively, of the picture relative to the current block. Inter-picture prediction also uses reference pictures from different time instances, such that the place in time of each picture relative to another is defined by Δt , where Δt is a reference index into a reference picture list, where pictures at different time instances will have different Δt values.

59. As noted above, this video sample prediction is used in conjunction with transformation of the error associated with the prediction. *Id.* To that end, transforms are applied to the residual, or error, that is left over after video sample prediction. *Id.* at 142. HEVC uses two-dimensional transforms that can vary in size from 4x4 to 32x32 and are finite precision approximations to the discrete cosine transform (DCT). The prediction error, or residual, is input into the forward transform at the encoder, with the output being transform coefficients that are subject to quantization. *Id.* The decoder then needs to determine the associated residuals, and therefore performs the opposite process i.e. perform entropy decoding, rescale by a quantization step size, and then inverse transform the coefficients, to output the reconstructed block. *Id.* this is shown below in figure 6.1:



Ex-1004 at 142.

60. In H.265/HEVC, and other video codecs that utilize inter and/or intra-prediction along with transform coding of residuals, there can be a mismatch between the input block and the reconstructed block after transforming, quantization, dequantization, and inverse-transforming. *Id.* Because of these mismatches, along with other errors related to block prediction, there can be discontinuities at prediction and transform block boundaries. This can have a detrimental effect on the subjective quality of decoded video pictures and also on the effectiveness of motion compensated prediction, which in turn can lead to reduced compression efficiency. *Id.* at 171.

61. In HEVC/HEVC, both the motion prediction and transform coding are block-based, such that blocks are coded with some independence from the

neighbouring blocks and approximate the original signal with some degree of similarity. Since coded blocks only approximate the original signal, the difference between the approximations may cause discontinuities at the prediction and transform block boundaries. For example, motion prediction of the adjacent blocks may come from the non-adjacent areas of a reference picture (see Fig. 7.2) or even from different reference pictures. In case of non-overlapping block transforms, used in HEVC, coarse quantization can also create discontinuities at the block boundaries.” *Id.* at 173-174. An example of these discontinuities is shown below in Figs. 7.2 and 7.3 of the Sullivan Textbook.

Fig. 7.2 Block artifact may be created when adjacent blocks are predicted from non-adjacent areas in the reference picture

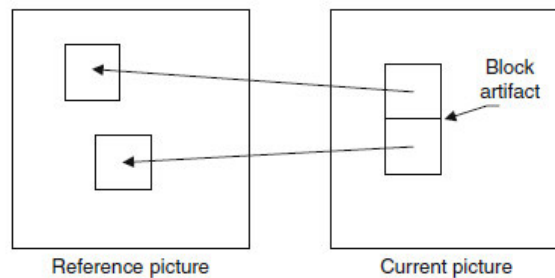
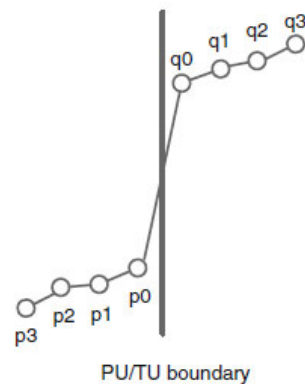


Fig. 7.3 Example of block artifact in one dimension [37]



Ex-1004 at Figs. 7.2-7.3.

62. These discontinuities between the blocks of pixels can result in the final picture output appearing blocky because of the generated “blocking artifacts.” See Ex-1003 ¶ 44. A real-world example of these discontinuities is shown below.



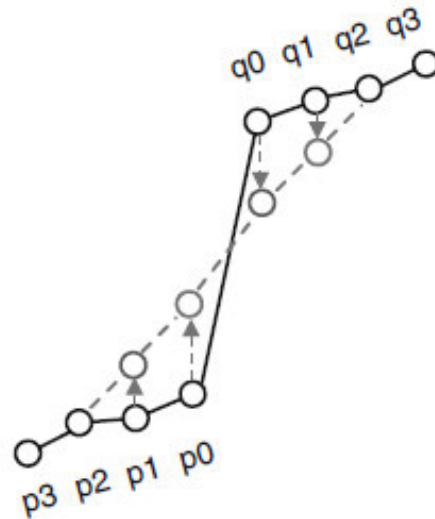
Ex-1003 ¶ 44.

63. Deblocking filters are intended to “attenuate[] the artifacts in the areas, where they are mostly visible, i.e. in the smooth, uniform areas. The excessive filtering in the highly detailed areas should be avoided since it can cause undesirable blurring.” Ex-1004 at 174. Therefore, “an important part of the deblocking filter is the deblocking filtering decisions, which determine whether a particular part of a block boundary is to be filtered. In these decisions, the HEVC deblocking filter uses the mode and motion information from the decoded bitstream as well as analyzes the values of reconstructed samples on the sides of the block boundary.” *Id.* Deblocking filtering takes the form of modifying the pixel value(s) that are filtered.

64. These decisions regarding whether to modify pixels within a deblocking filter, and the amount to modify each pixel, are an important

determination in optimizing deblocking filter performance. For example, because “the deblocking [filter] needs to attenuate visible artifacts in smooth areas, the deblocking decisions evaluate whether the signal at the sides of the block boundary is smooth, i.e. if the signal is flat or has a shape of an inclined plane (ramp).” *Id.* at 176. Based on this determination, some pixels will be filtered, while others will not. When pixel values are filtered, they are modified so that they will more closely resemble the smooth line that appears smoother and less visible to the human eye.

Fig. 7.9 Illustration of normal filtering mode operations: original block boundary (solid black line) and modified block boundary (dashed gray line)

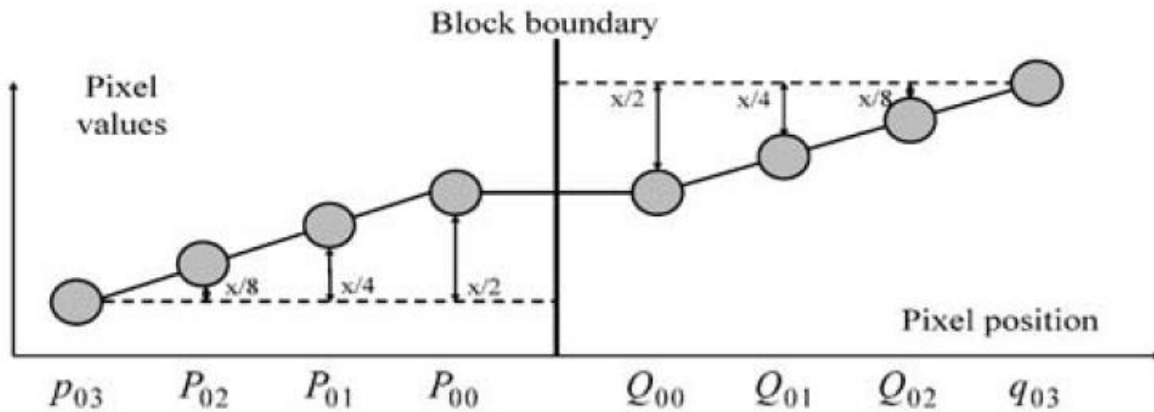
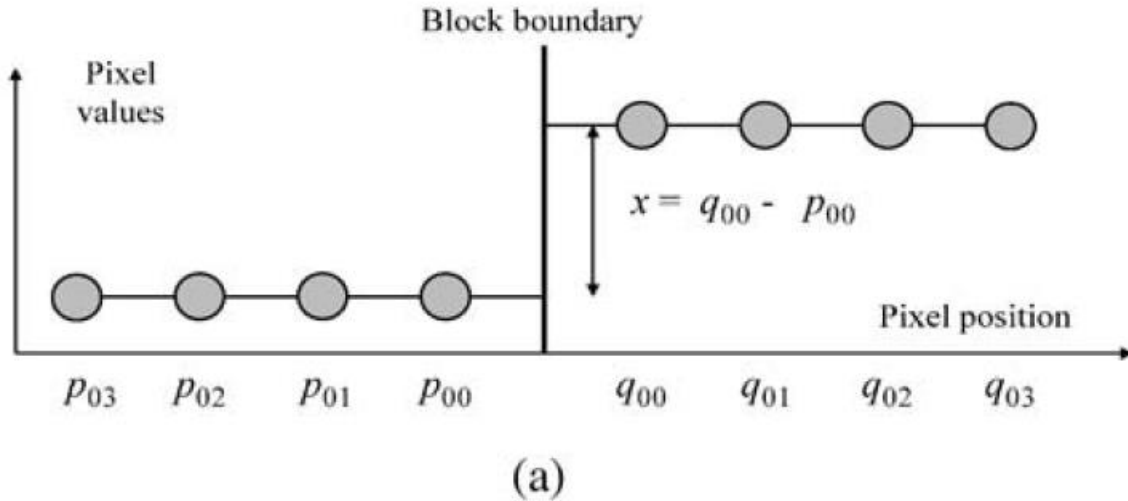


Ex-1004 at 181.

65. One way to accomplish the filtering is to calculate an offset, which determines how much a pixel will be modified. In Figure 7.9 above for example, the calculated offset will determine how much or how little to move pixel p0 upwards towards the more ideal straight ramp shape. Likewise, the same offset (or different

calculated offsets) can be used to modify other pixels as well, such as p1, q0, and q1 above.

66. As Dr. Schonfeld describes, the modification of pixel values can be depicted graphically, wherein the figure below shows how multiple pixels are filtered to smooth the effects of blocking artifacts over a block boundary to obtain a signal shape represents the natural image over a block boundary. This occurs by calculating the difference “x” across the block boundary and using the difference to create a slope for a smooth transition:



Ex-1039, 215; Ex-1003, ¶53.

67. The real-world result of this pixel modification shown above, is a smoother looking picture that is more visually appealing and which may provide a better source for constructing inter predicted blocks. (*Id.* At 171). In general, deblocking filters have a goal of accuracy, where they should be designed such that natural images are not over filtered, while blocking artifacts are smoothed.

<u>No Deblocking</u>	<u>After Deblocking</u>
	

Ex-1003, ¶¶ 44-45.

68. A challenge with deblocking filters is that they are recognized by persons of ordinary skill in the art to be complex to design, and can be computationally expensive when implemented. For example, it has been recognized that the loop filter in H.264 (deblocking is an in-loop filter), “can easily account for one-third of the computational complexity of a decoder.” Ex-1024 at 614.

69. Due to the complexity of designing deblocking filters and the goal of reducing computational expense in practical video codecs, there has not been a uniform approach to deblocking that those skilled in the art have agreed upon. Instead, as illustrated by the diverse approaches in the references cited by Petitioner and Dr. Schonfeld, many deblocking filters have been designed over the course of the decades of video codec development. *See, e.g.*, Ex-1009; Ex-1017; Ex-1018; Ex-1022; Ex-1024; Ex-1023; Ex-1025 (disclosing different deblocking filter designs).

The prior art demonstrates that many researchers and engineers working on deblocking over time did not agree on a uniform approach. In my view this is because the delicate balance between the need for eliminating blocking artifacts, while reducing computational complexity results in complex design processes that require extensive subjective and objective testing, in order to balance the goals of minimizing implementation complexity, improving subjective quality and optimizing inter prediction efficiency. The nature of standardization bodies that develop and publish video coding standards is that development is incremental and continual, and alternative proposals for improvements to functionality including deblocking filter design are continually submitted to and developed by the standards bodies. Deblocking filters in particular have been the subject of research and development for over 20 years prior to the '841 Patent. Ex-2003.

IX. U.S. PATENT NO. 9,641,841

A. '841 Patent Invention – Ericsson's Deblocking Filter

70. The '841 Patent, titled “Deblocking filtering control,” is directed to controlling deblocking filtering control over block boundaries in a video frame. In particular, the '841 Patent provides asymmetric filtering decisions over a block boundary.

71. The '841 Patent explains that deblocking filters are used in video coding standards to combat blocking artifacts, which arise because the original video

frames are split into blocks which are processed relatively independently. For example, neighboring blocks can be subject to different intra predictions, quantization effects and motion compensation. Ex-1001 at 1:11-17.

72. A deblocking filter may be specified to be applied after prediction and residual reconstruction and before storage of the reconstruction (i.e. the decoded video frame) for later reference. The deblocking filter can therefore reduce the effects of blocking artifacts on decoded video frames that may be displayed and also stored for later prediction, which can lead to improved prediction and therefore improved compression of further coded video frames, as the filtered version of the decoded frame (with reduced blocking artifacts) can be used to construct a more accurate prediction of a block in a future coded frame. Ex-1001 at 1:19-23.

73. The '841 Patent explains that the deblocking filtering decisions in the prior video coding standards, including the then-draft HEVC design, “can lead to inaccurate deblocking filtering over block boundaries for certain blocks.” Ex-1001 at 3:6-8. As an example, the '841 Patent highlights that neighboring blocks having different levels of local structure could be filtered incorrectly according the draft HEVC design disclosed in B205. The inventors of the '841 Patent were involved in the development of the H.265 Standard and recognized that the design in B205 could introduce errors by filtering one of the blocks too much because of the local structure of the neighboring block. Ex-1001 at 3:8-12. Further, the '841 Patent describes the

fundamentally different approaches for making filtering decisions. For example, the '841 Patent discloses that the H.264 provided for a strong filter and a normal or weak filter. Ex. 1001 at 1:49-52, 1:65-67. In the strong filter, the same number of pixels (i.e. 3 pixels on each side of the block boundary) are always filtered. The weak filter of H.264, however, allows for asymmetric filtering and thus it is possible to filter different number of pixels on each side of the block boundary (i.e. 1-2 pixels on each side of the block boundary). The decision as to how many pixels to modify on each side of the block boundary is calculated based on the difference in pixel value between two pixels (e.g. $p_2 - p_0$). *Id.*

74. The HEVC draft also provided for a strong filter and a weak filter. In the strong filter, three pixels are always filtered on each side of the block boundary. Ex. 1001 at 2:61-3:5. Contrary to the H.264 Standard, the HEVC Draft does not allow asymmetric filtering in its weak filter. Instead, when weak filtering is applied, HEVC will always filter two pixels on each side of the block boundary. Ex. 1001 at 2:43-60.

75. The '841 Patent improves upon the deblocking filter in H.264 and in the Draft H.265 Standard described in B205 by achieving asymmetric deblocking decisions that control deblocking filtering such that it is adaptive to the structure on each side of the block boundary. Ex-1001 at 4:62-64. The '841 Patent explains that “asymmetric” refers to the fact that the amount of filtering applied to one side of the

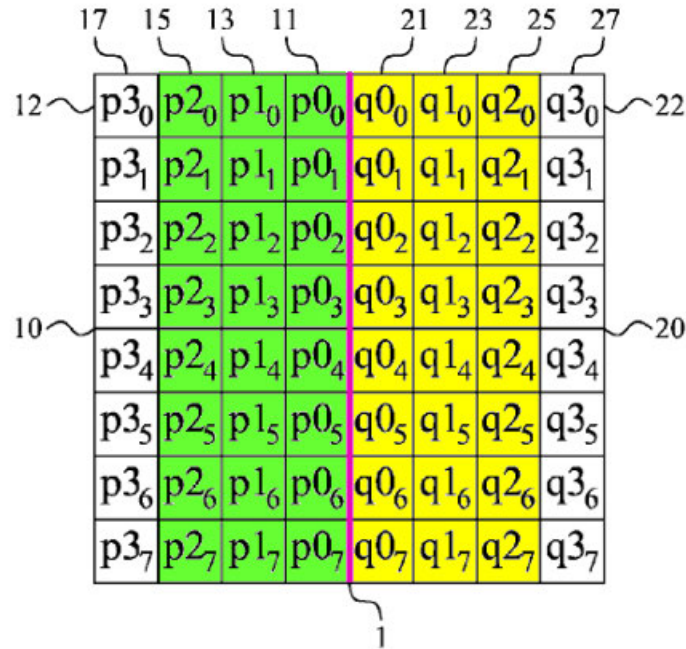
block boundary can differ from the amount of filtering applied to the other side of the block boundary. Ex-1001 at 4:64-5:2. This adaptive filtering control makes filtering decisions and applies the amount of filtering based on the localized structure on each side of the block boundary, which is dependent on the pixel values on each side of the block boundary. In addition, the '841 Patent describes and claims a first, second, and third offset calculated based on the asymmetric filtering decisions described and claimed in the '841 Patent, that are used to modify the pixel values when carrying out the actual filtering. *See, e.g.*, Ex-1001, cl. 10.

76. The asymmetric filtering described in the '841 Patent is achieved by calculating “a first filter decision value” for a block of pixels “using only pixels p_{2i} , p_{1i} , and p_{0i} , based on: $|p_{2i} - 2p_{1i} + p_{0i}|$ ” and calculating “a second filter decision value” for a neighboring block of pixels “using only pixels q_{2i} , q_{1i} , and q_{0i} , based on: $|q_{2i} - 2q_{1i} + q_{0i}|$ ”. *See, e.g.*, '841 Patent, cl. 1, 5, 16. Separate determinations of how many pixels to filter in the block of pixels (the “p” block) are made based on a comparison of the first filter decision value to a first threshold value and of how many pixels to filter in the neighboring block of pixels (the “q” block) based on a comparison of the second filter decision value to a second threshold value.

77. For example, claim 1 recites “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 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.” ’841 Patent, cl. 1. Though I reference claim 1 as exemplary, independent claims 5 and 16 include corresponding requirements. *See* Ex-1001 at cl. 5, 16.

78. Thus, the ’841 Patent requires separate filter decision values calculated separately for each side of a block boundary. The filter decision values are based on the absolute value of an equation that measures or estimates the degree of ramp of the pixel values using the pixel value of the pixels first, second, and third closest to the block boundary on each side of the block boundary, which are illustrated below in connection with Figure 2A of the ’841 Patent. I have annotated the block boundary labeled ‘1’ in purple. I have annotated the pixels in the three columns to the left side of the block boundary are annotated in green, and I have annotated the pixels in the three columns to the right side of the block boundary are annotated in yellow.



Ex-1001 at Fig. 2A (annotated)

79. The '841 Patent describes that pixels in green can be used to measure the ramp of the p block, while the pixels in yellow may be used to calculate the ramp of the q block.

80. The '841 Patent discloses separate determinations of how many pixels to filter on each side of the block boundary based on the comparison of each of the first and second filter decision values to a first and second threshold value, respectively. As a result, different filtering can be applied to each side of the block boundary. For example, using the asymmetric filtering design claimed in the '841 Patent, no pixels could be filtered in the “p” block, while 2 pixels could be filtered in the “q” block. As another example, 1 pixel could be filtered in the “p” block, and

2 pixels could be filtered in the “q” block. As another example, 2 pixels could be filtered in the “p” block, and no pixels could be filtered in the “q” block.

81. While the independent claims focus on how to independently determine the number of pixels to be filtered on each side of the block boundary, certain dependent claims recite very specific requirements for how the pixels should be filtered. For example, the ’841 Patent’s dependent claim 10 recites the first, second, and third offset used to modify the pixel values as part of the filter. Specifically, claim 10 recites:

a first offset calculator configured to calculate a first offset Δ based on

$$\frac{9 \times (q0_j - p0_j) - 3 \times (q1_j - p1_j)}{16},$$

...

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;

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;

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

$$\frac{p_0j+p_2j-2p_1j+2\Delta}{4},$$

...

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

$$\frac{q_0j+q_2j-2q_1j-2\Delta}{4},$$

...

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.

'841 Patent, cl. 10.

82. These offsets are an important part of the overall deblocking filter design presented in the '841 Patent. Specifically, to understand the full behaviour and impact of a deblocking filter, a POSITA would understand that it is necessary to consider the interplay between the determinations of which pixels to filter and how the pixels are then filtered. In the case of the '841 Patent, the specific offset equations are designed to achieve a very specific frequency response and behavior in response to certain structures formed by the pixel values across a block boundary. Specifically, the filter is designed not to modify pixel values over a block boundary

if the pixel values form a ramp, but smooth the pixel values if they instead are in the form of a step. In a related application (U.S. Patent No. 10,142,659) filed by the same inventors, which also proposes improvements to a deblocking filter, the inventors explained that the weak deblocking filter in the HEVC Draft does not have “good low-pass characteristics,” “cannot remove high frequencies that can appear near the block border and cannot remove ringing,” and “does not adapt well to the local structure of the side of the block boundary.” Ex-1063 at 3:18-33. The inventors solved these problems in the prior art through a more complex, but more adaptable, deblocking filter that also improved compression efficiency. Ex. 1001 at 1:10-2:39.

83. Specifically, the '841 Patent is directed to a novel deblocking filter that (i) calculates separate filter decision values for each side of the block boundary; (ii) separately determines how many pixels for the current block and its neighboring block; and (iii) provides for specific modifications or filtering of pixels near the block boundary such that local structures that form a ramp are not modified but structures in the form of a step are smoothed. *Id.* at 86-98.

84. As set forth above, the '841 Patent provides for calculating a first filter decision value using only pixels from the p block based on $|p_{2i}-2p_{1i}+p_{0i}|$, and calculating a second filter decision value using only pixels from q block based on: $|q_{2i}-2q_{1i}+q_{0i}|$. Thus, the filter decision values provide information regarding the local structures on each side of the block boundary. Ex. 1001 at 3:16-5:2. For

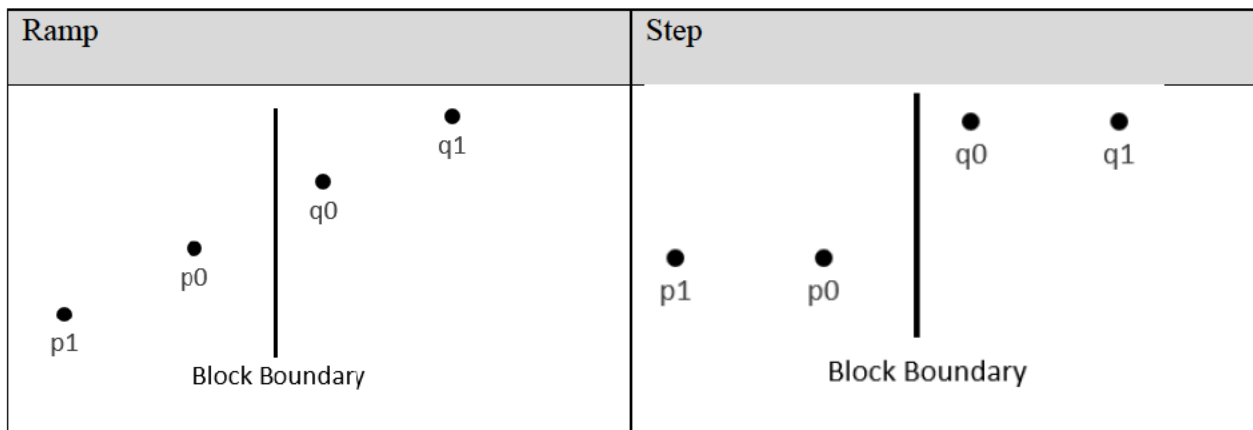
example, when the pixels p_{2i} , p_{1i} , and p_{0i} form a ramp, the filter decision value will be equal to 0. The '841 Patent provides for calculations of a filter decision value that indicate whether a local structure is a ramp or a step.

85. The '841 Patent also provides for separately determining how many pixels to filter for the current block (p block) and its neighboring block (q block). The determination of how many pixels to filter for the current block must be based on, but not exclusively use, a comparison of the first filter decision value to a threshold, and the determination of how many pixels to filter for the neighboring block must be based on, but not exclusively use, a comparison of the second filter decision value to a threshold. As a result, the '841 Patent provides for asymmetric filtering decisions, which allows the filter to be adaptive to local structures on both sides of the block boundary. Ex. 1001 at 4:62-5:2.

86. The combination of the asymmetric decisions of how many pixels to filter with the use of the specific filter decision value calculations (e.g. $|p_{2i}-2p_{1i}+p_{0i}|$) improves the deblocking filter's ability to identify block artifacts created by the decoding process and distinguish naturally occurring edges. Specifically, the filter decision value calculations (e.g. $|p_{2i}-2p_{1i}+p_{0i}|$) will identify local structures in the form of a ramp, small steps, and/or lines and allow the deblocking filter to distinguish those structures from irregular structures or large differences in pixel

values. This information allows the deblocking filter more accurately determine how many pixels to filter to address specific local structures.

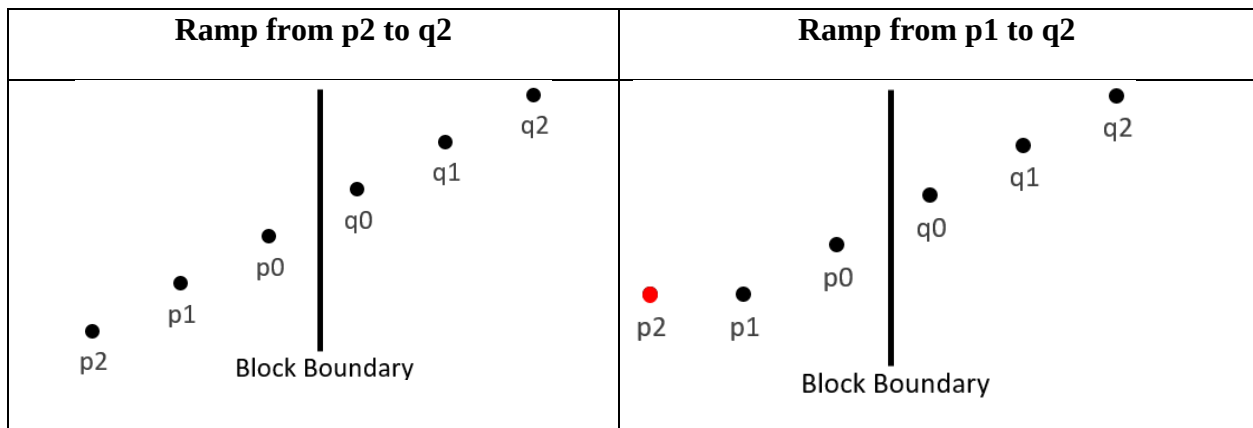
87. In addition, '841 Patent provides for the use of specific offsets in the filtering process. Specifically, the '841 Patents will modify the pixel values over a block boundary if these pixel values form a ramp but smooth the pixel values if they instead are in the form of a step. Ex-1001, 181-23; Ex-1063, 8:57-63. This results in a deblocking filter that has “good lowpass characteristics and can remove high frequencies that can appear near the block boundary.” Ex-1063 at 8:57-63. As illustrated below, a ramp is a series of linearly increasing or decreasing pixels. Ex-1063 at 13:21-24. A step occurs when there is a sudden or instantaneous change in the values of adjacent pixels from one constant value to another in the values of adjacent pixels (i.e. a step increase or decrease). Ex-1063 at 13:26-27.



88. The '841 Patent achieves this behaviour of not filtering ramp structures but smoothing step structures in a particular way by the particular expressions

claimed as the first, second, and third offsets. *See, e.g.*, Ex-1001, cl. 10. The '841 Patent describes utilizing a first offset based on $9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) / 16$. Ex-1001 at 18:30-35. Once this improved first offset value is determined, the deblocking filter will modify the pixel values of the pixels closest to the block boundary (p_0 and q_0) using the first offset. *Id.* at 3:55-63. In modifying the pixel values, the deblocking filter will physically alter the pixels of a video frame making the block boundary less noticeable. *Id.*

89. It is important to the design of the '841 Patent that the pixels p_0 and q_0 be modified based on the structure of p_1 , p_0 , q_1 and q_0 . The following structures illustrate two ramps:

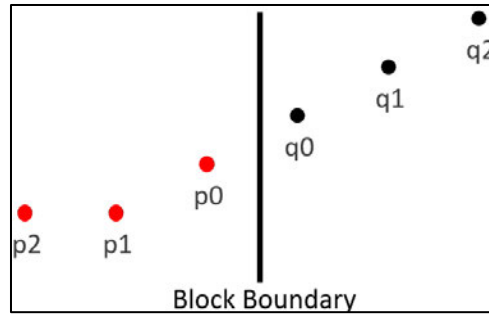


90. The figure on the left includes a ramp that extends from pixel p_2 through q_2 , but the ramp on the figure on the right only extends from pixels p_1 through q_1 . The ramp in the figure on the right is more limited because the transition from pixel p_2 and p_1 does not form a ramp. Despite the difference between the

figures, the offset for p0 and q0 will be 0 in both examples because the first offset of the '841 Patent ignores the pixel p2.

91. The claimed second and third offsets provide localized modification of pixels p1 and q1 of a video frame. The '841 Patent also provides an alternative embodiment where pixels p1 and q1 can be modified by the deblocking filter. The inventors explained that the prior art HEVC Draft “show slight amplifications of higher frequencies when filtering the second pixel (p1_i and q1_i) from the block boundary, which can result in worse rate-distortion (RD) performance of the coder.” Ex-1063 at 3:24-33. The inventors also criticized the HEVC Draft for not allowing offsets of different magnitudes for different sides of the block boundary. *Id.*

92. The claimed second and third offsets allow for local adaption to the block boundary. Ex-1063 at 14:4-10. Specifically, the 841 Patent disclose that the pixel p1 should be modified by a second offset calculated based on $(p2+p0-2 \times p1+2 \times \Delta)/4$, and the pixel q1 should be modified by a third offset calculated based on $(q2+q0-2 \times q1-2 \times \Delta)/4$. Because different pixels are used for each offset, the two offsets can take into account the local structures on each side of the block boundary. For example, when considering the figure below, the second and third offsets will not be the same because the localized structures on each side of the block boundary are not the same.



93. Specifically, the second offset, which is based on the p-pixels highlighted in red, will be non-zero (i.e. a positive value) because the local structure is not a ramp. The third offset, however, will be zero because the localized structure on the right side of the block boundary is still a ramp. Thus, the 841 Patent provides for a specific deblocking filter that accounts for localized structures on each of the block, does not filter ramps within the localized structure, but smooths steps.

94. This design is fundamentally different, in my view, to any approach disclosed in the prior art. Notably, as demonstrated by the prior art asserted and referenced by Petitioner and Dr. Schonfeld, there was no uniform approach in the prior art as to when and how to modify the same pixels.

95. For example, prior art deblocking filters do not take a uniform approach to modifying pixels positioned next to the block boundary (e.g. pixels p0 and q0) based on the same filtering window and same weights and thus the smoothing achieved by these deblocking filters varies dramatically. In my opinion, the prior art demonstrates that there are multiple factors that a POSITA would consider in designing a deblocking filter that filters p0 and q0, including (i) whether to use the

same offset to modify p0 and q0; (ii) which neighboring pixel values to use to calculate an offset; and (iii) how to weight those values. Notably, as shown below, Petitioner has identified at least 8 additional references that disclose deblocking filters that use different offsets for pixels p0 and q0 and thus remove and/or smooth visual artifacts differently. Each of these approaches will have a different effect on both the visual appearance of decoded video frames and the efficiency of inter prediction of future frames based on filtered decoded frames.

96. As an initial example, the strong and weak filters of H.264 use fundamentally different offsets for pixels p0 and q0:

Modified Pixel	H.264 Strong Filter Offset	H.264 Weak Filter Offset
p0	$(\textcolor{red}{p}2+2p1+2p0+2q0+q1+4)>>3$	$(4(q0 - p0) + (p1 - q1) + 4)>>3$
q0	$(p1+2p0+2q0+2q1+\textcolor{red}{q}2+4)>>3$	

97. As shown here, one difference between the strong and weak filter is that the weak filter applies the same offset to p0 and q0, whereas the strong filter uses a different offset for p0 and q0. Further, the pixels values used in the weak filter offset are different than the pixels values used in the strong filter offset. For example, the weak filter does not use pixels p2 or q2 (highlighted in red), but the strong filter does. Ex. 1001 at 1:49-2:4. The operations required for each offset are fundamentally different. The weak filter uses differences between similarly positioned pixels (i.e.

differences between q0 and p0 and p1 and q1) whereas the strong filter uses a combination of five weighted pixel values to calculate the offset.

98. As another example, the HEVC Draft uses different pixel values (e.g. q2 and p2), uses different weights or coefficients, and a different shift operation to calculate two different offsets for filtering pixel p0. The differences between the offsets are shown in the table below:

Modified Pixel	H.264 Weak Filter Offset	HEVC Draft Weak Filter Offset
p0	$(4(q_0 - p_0) + (p_1 - q_1) + 4) \gg 3$	$(13(q_0 - p_0) + 4(q_1 - p_1) - 5(q_2 - p_2) + 16) \gg 5$

Ex. 1001 at 2:43-50.

99. The filter design claimed in the '841 Patent, including the asymmetric filtering decisions for the blocks of pixels on either side of a block boundary and the offset calculations optimized for this localized and adaptable filter, improved upon the deblocking filter designs in H.264 and the draft H.265 Standard described in B205 that the '841 Patent specifically addresses as background and state of the art in its specification. Ex-1001 at 1:10-2:39.

100. As explained above and in the background portion of the '841 Patent specification, deblocking filtering decisions in the prior H.264 standard were based on measuring the absolute difference between pixel values and comparing the

absolute difference to a given threshold value., H.264 specifies the two threshold variables as follows:

Let a_p and a_q be two threshold variables specified by

$$a_p = \text{Abs}(p_2 - p_0) \quad (8-339)$$

$$a_q = \text{Abs}(q_2 - q_0) \quad (8-340)$$

H.264 at 152; Ex-1003, ¶77.

101. H.264's thresholds a_p and a_q are based the absolute value of the difference calculated between two pixels on either side of the block boundary (e.g., p_2 and p_0 for the P side of the block and q_2 and q_0 for the Q side of the block). This is fundamentally different from, and will yield different filtering results than, the calculation of the first filter decision value in the '841 Patent, which uses three pixel values and is based, not on the calculation of the absolute difference, but on the absolute value of the ramp estimation using pixels p_1 , p_2 , and p_0 if using the notation from H.264 Standard. Likewise, the calculation of the absolute difference of two pixels for a_q is fundamentally different from, and will yield different filtering results than, the calculation of the second filter decision value claimed in the '841 Patent, which is based on the absolute value of the ramp estimation calculated using three pixels q_1 , q_2 , and q_0 if using the notation from H.264 Standard.

102. Further, the filtering outcome will be affected by the overall filter design. Thus, isolating the fact that a_p uses only "p" pixels and a_q uses only "q" pixels

or the fact that the equation used to calculate delta (Δ) in H.264 includes the difference between q_0 and p_0 and q_1 and p_1 not only fails to acknowledge that these equations are based on a completely different metric than the filter decisions in the '841 Patent (e.g., measuring or estimating flatness of the pixels values versus ramp), but also fails to consider the effect that the difference in the overall filter design using different metrics and using different filter coefficients would have on the output block of pixels H.264 and Ericsson's deblocking filter design disclosed in the '841 Patent. Indeed, the threshold variables a_p and a_q are a part of the overall filter design in H.264 specified below. Further, different filter coefficients are used to calculate the value of delta (Δ) in the H.264 Standard if p_0 and q_0 are filtered. If p_1 and q_1 are filtered (e.g., a_p is less than β and a_q is less than β , the calculation of the values that are respectively added to p_1 and q_1 to result in the filtered pixel value p_1' and q_1' are different from the second and third offset disclosed and claimed in the '841 Patent. *See, e.g.,* Ex-1001 at cl. 10.

8.7.2.3 Filtering process for edges with bS less than 4

Inputs to this process are the input sample values p_i and q_i ($i = 0..2$) of a single set of samples across an edge that is to be filtered, chromaEdgeFlag, bS, β , and indexA, for the set of input samples, as specified in 8.7.2.

Outputs of this process are the filtered result sample values p'_i and q'_i ($i = 0..2$) for the set of input sample values.

The filtered result samples p'_0 and q'_0 are derived by

$$\Delta = \text{Clip3}(-t_c, t_c, ((q_0 - p_0) < 2) + (p_1 - q_1) + 4) >> 3) \quad (8-334)$$

$$p'_0 = \text{Clip1}(p_0 + \Delta) \quad (8-335)$$

$$q'_0 = \text{Clip1}(q_0 - \Delta) \quad (8-336)$$

where the threshold t_c is determined as follows.

- If chromaEdgeFlag is equal to 0,

$$t_c = t_{c0} + ((a_p < \beta) ? 1 : 0) + ((a_q < \beta) ? 1 : 0) \quad (8-337)$$

- Otherwise (chromaEdgeFlag is equal to 1),

$$t_c = t_{c0} + 1 \quad (8-338)$$

The threshold t_{c0} is specified in Table 8-15 depending on the values of indexA and bS.

Let a_p and a_q be two threshold variables specified by

$$a_p = \text{Abs}(p_2 - p_0) \quad (8-339)$$

$$a_q = \text{Abs}(q_2 - q_0) \quad (8-340)$$

The filtered result sample p'_1 is derived as follows

- If chromaEdgeFlag is equal to 0 and a_p is less than β ,

$$p'_1 = p_1 + \text{Clip3}(-t_{c0}, t_{c0}, (p_2 + ((p_0 + q_0 + 1) >> 1) - (p_1 < 1)) >> 1) \quad (8-341)$$

- Otherwise (chromaEdgeFlag is equal to 1 or a_p is greater than or equal to β),

$$p'_1 = p_1 \quad (8-342)$$

The filtered result sample q'_1 is derived as follows

- If chromaEdgeFlag is equal to 0 and a_q is less than β ,

$$q'_1 = q_1 + \text{Clip3}(-t_{c0}, t_{c0}, (q_2 + ((p_0 + q_0 + 1) >> 1) - (q_1 < 1)) >> 1) \quad (8-343)$$

- Otherwise (chromaEdgeFlag is equal to 1 or a_q is greater than or equal to β),

$$q'_1 = q_1 \quad (8-344)$$

The filtered result samples p'_2 and q'_2 are always set equal to the input samples p_2 and q_2 :

$$p'_2 = p_2 \quad (8-345)$$

$$q'_2 = q_2 \quad (8-346)$$

Ex-1040 at § 8.7.2.3.

103. Thus, I disagree with any of Dr. Schonfeld's opinions that B205 and H.264 disclosed similar deblocking designs to the '841 Patent. *See* Ex-1003 ¶¶ 87, 98-99, 103.

104. Dr. Schonfeld's opinion that deblocking filters typically followed a two-step framework of analyzing the pixels near a block boundary to determine whether they need to be filtered, and if determined, filter or modify the pixel values. Ex-1003, ¶65. However, this general framework does not make the claimed deblocking design obvious, because, as acknowledged by Dr. Schonfeld, the prior art deblocking filters differed (e.g., did not provide any sort of uniform approach) with respect to (i) what conditions were used to decide whether deblocking filtering is performed, (ii) how many pixels to filter as part of deblocking, and (iii) the equations used to filter/modify the pixels values, which can further be split up into a vast array of possible considerations and factors that contribute to the deblocking filter design. As described above, at least the number of pixels used, the metric for determining whether to filter, and the offset equations are different from those described and claimed in the '841 Patent. These differences are very important, as they result in a filter that makes different decisions and produces different results.

105. As further described above, the '841 Patent also discusses the draft H.265 Standard in B205. Notably, the draft H.265 Standard in B205 moved away

from the thresholds specified in H.264, which used only “p” pixels or “q” pixels, to using pixels on both sides of the block boundary in order to make filtering decisions that were equally applicable to pixels on both sides of the block boundary. For example, as described above, B205 specified that the decision to filter was simplified compared with H.264 to include only one step where the calculated value of was compared to a threshold value, β :

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

Ex-1001 at 2:20-29.

106. Thus, the development of the then-draft H.265 Standard disclosed in B205 was moving away from the multiple step, separated filter decisions in H.264 to a one-step calculation, which would be applied to the current and neighboring blocks. Therefore, the asymmetric filtering decisions disclosed in the '841 Patent as a first filter decision value and a second filter decision value were contrary to the direction that the industry was moving, as evidenced by the then-draft H.265 Standard that was under development at the time.

107. Further, Ericsson's deblocking filter design in the '841 Patent improved upon the offset expressions in the draft H.265 Standard in B205. The calculation of the first offset Δ in B205 uses the values of six pixels instead of four pixels, as was

previously disclosed in H.264. That is, pixels p0, q0, p1, q1, p2, and q2 are used in the offset calculation in B205, in contrast to H.264 in which p2 and q2 were not used in an offset calculation. In addition, as shown in Dr. Schonfeld’s “Normalized First Offset Expression” table, the filter coefficients used in B205 were 7.5/16, 2/16, and 2.5/16. See Ex-1003, ¶¶ 103. Thus, the Ericsson deblocking design disclosed in the ’841 Patent was contrary to the direction of the development of the H.265 Standard because of the use of four pixels instead of six pixels and because of the use of different filtering coefficients that would yield different filtering outcomes than those in B205.

108. Moreover, the second and third offsets claimed in the ’841 Patent also moved away from the direction of the industry as evidenced by the development of the H.265 Standard. For example, as acknowledged by Dr. Schonfeld, the second and third offsets in B205 were $\frac{\Delta}{2}$ and $\frac{-\Delta}{2}$, respectively, which are less computationally complex to calculate than the equations in H.264. Ex-1003 ¶93. In the B205 method, the value of Δ is already calculated and can therefore simply be scaled by $\frac{1}{2}$ and then re-used. In H.264, the offsets are calculated by the clipped expression in the equations for calculation p1’ and q1’ below:

$$p'_1 = p_1 + \text{Clip3}(-t_{c0}, t_{c0}, (p_2 + ((p_0 + q_0 + 1) \gg 1) - (p_1 \ll 1)) \gg 1) \quad (8-341)$$

$$q'_1 = q_1 + \text{Clip3}(-t_{c0}, t_{c0}, (q_2 + ((p_0 + q_0 + 1) \gg 1) - (q_1 \ll 1)) \gg 1) \quad (8-343)$$

H.264 Standard at § 8.7.2.3.

109. Thus, the claimed second and third offsets are a significant departure from the computationally less complex design in B205, which reflected the direction of the H.265 Standard Development at the time and which departed from the more computationally complex methods for deblocking in H.264.

110. As described above, the '841 Patent acknowledges that H.264 included a deblocking filter which consisted of filter decisions, filtering operations, a clipping function and changes of pixel values. Ex-1001 at 1:19-25. The decision to filter in H.264 depended on macroblock type motion, motion vector difference between neighboring blocks, whether neighboring blocks have coded residuals and on the local structure of the current and/or neighboring blocks. Ex-1001 at 1:25-31. Then, the amount of filtering for a pixel depends on the position of that pixel relative to the block boundary and on the quantization parameter used for residual coding. Ex-1001 at 1:32-35.

111. Specifically, in H.264, the filter decision is based on comparing three-pixel differences with thresholds, which are adapted to a quantization parameter. The '841 Patent sets forth an example that assumes a vertical block boundary represented with pixels below, where pixels a, b, c, and d are pixels on the left side of the block boundary in a line of pixels in the current block, and pixels e, f, g, and h are pixels in the corresponding line of pixels in the neighboring block. Ex-1001 at

1:38-45. The block boundary between the current block and the neighboring block is represented in red below.

a b c d | e f g h

112. The H.264 deblocking filter makes filtering decisions based on the absolute difference in the pixel values *across the block boundary* compared to threshold values. For example, the filtering decision is positive (e.g., a decision to perform filtering on at least some of the pixels a-h) if the following conditions are met (Ex-1001 at 2:45-48):

$$abs(d - e) < thr1|$$

$$abs(c - d) < thr2$$

$$abs(e - f) < thr2$$

113. In addition, in H.264 deblocking filtering, if there is a positive filtering decision, then the amount of filtering to apply is based on one of two filtering modes: normal filtering or strong filtering. The '841 Patent explains that normal filtering in H.264 employs a delta value to change the current pixel value of the pixels closest to each side of the block boundary resulting in pixel values d' and e' by adding the delta value to d and subtracting the delta value from e, wherein delta has been clipped to a threshold $\pm thr3$, i.e., to a value that is constrained by the quantization parameter. Thus, more filtering occurs with high quantization parameters than with low quantization parameters. Ex-1001 at 1:49-56. Clipping the delta value in H.264 is

described in the '841 Patent as $\text{delta_clipped} = \max(-\text{thr3}, \min(\text{thr3}, \text{delta}))$, where thr3 controls the filter strength. If the value of thr3 is higher, then the filtering is stronger which results in a stronger low-pass filtering effect. Ex-1001 at 1:56-60. Clipping the delta value can adjust the filter strength up or down. That is, the '841 describes that in H.264, the filter strength can be adapted by reducing the clipping of delta , thereby allowing for more variation.

114. The second filtering mode in H.264 described in the '841 Patent is strong filtering. The '841 Patent explains that strong filtering is applied on intra macroblock boundaries only, when the condition below is fulfilled:

$$\text{abs}(d - e) < \text{thr1}/4$$

Ex-1001 at 1:65-67.

115. The '841 Patent refers to List et al., Adaptive Deblocking Filter, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7 (July 2003) for further information on H.264 deblocking.

B. Additional Details Regarding Deblocking Filtering in the Draft H.265 Standard

116. The '841 Patent describes the further development of the deblocking filter in the then-draft H.265 Standard. The '841 Patent specifically references the draft HEVC (High Efficiency Video Coding) specification “Test Model under Consideration,” ITU-T SG16 WP3 document, JCTVC-B205 (“B205” (Ex-1018))

deblocking filter, which works in a different way to the deblocking filter in H.264. Prior to the invention described in the '841 Patent, the deblocking filter process in B205 was performed “if at least one of the block on the side of the boundary is intra, or has non-zero coefficients, or the difference between the motion vector components of the blocks is greater or equal to one integer pixel.” Ex-1001 at 2:5-14.

117. The '841 Patent describes an example of deblocking filtering according to B205. This example uses different pixel notation compared with the H.264 example described above, but the same concept of a line of pixels in a current block and corresponding line of pixels in a neighboring block, separated by a block boundary, is set forth. This is illustrated below, wherein p_{ji} denotes the pixel value of pixel number j of row number i in the current block and q_{ji} denotes the pixel value of pixel number j of row number i in the neighboring block, where $i = 0 \dots 7$ and $j = 0 \dots 3$. The block boundary between p_{0i} and q_{0i} is shown in red. Ex-1001 at 2:12-20.

$p_{3i} \ p_{2i} \ p_{1i} \ p_{0i} \ | \ q_{0i} \ q_{1i} \ q_{2i} \ q_{3i}$

118. In B205, the decision whether to filter pixels on both sides of the block boundary is based on the pixel values on both sides of the block boundary.

Specifically, in B205, if the following condition is met, then filtering is applied to the pixels p of the current block and the pixels q of the neighboring block.

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

Ex-1001 at 2:20-29.

119. In B205, the value of β used in the equation above is based on a quantization parameter QP. A quantization parameter controls the amount of lossy compression applied during encoding. A larger QP typically results in more compression and a more distorted decoded image, which may have more obvious blocking artifacts. A smaller QP typically results in less compression, a higher-quality decoded image and less obvious blocking artifacts. If the above condition is met (i.e. $d < \beta$) and there is a positive filtering decision, then either weak filtering or strong filtering will be applied to the current block and the neighboring block. Ex-1001 at 2:28-31. The choice between strong and weak filtering is made for each line of pixels, but the decision whether or not to filter applies to pixels in corresponding lines of pixels in the current block and the neighboring block. Ex-1001 at 2:31-59. In B205, for each line $i = 0 \dots 7$, the strong filtering is performed if all of the following conditions are met:

$$d < (\beta \gg 2) \\ (|p3_i - p0_i| + |q0_i - q3_i|) < (\beta \gg 3)$$

$$|p0_i - q0_i| < ((5 \times t_c + 1) \gg 1),$$

Ex-1001 at 2:33-40.

120. In the above equations, t_c and β depend on quantization parameters and the symbol $\gg$ denotes a binary right shift operator. In B205, if the above conditions are not met, then weak filtering applies. Ex-1001 at 2:34-42.

121. In B205, the actual filtering works by computing an offset (Δ), adding the offset to the original pixel value and clipping the sum to a filtered output pixel value as set forth in the background section of the specification of the '841 Patent:

$$\Delta = \text{Clip}(-t_c, t_c, (13 \times (q0_i - p0_i) + 4 \times (q1_i - p1_i) - 5 \times (q2_i - p2_i) + 16) \gg 5))$$

$$p0_i = \text{Clip}_{0-255}(p0_i + \Delta)$$

$$q0_i = \text{Clip}_{0-255}(q0_i - \Delta)$$

$$p1_i = \text{Clip}_{0-255}(p1_i + \Delta/2)$$

$$q1_i = \text{Clip}_{0-255}(q1_i - \Delta/2)$$

122. Ex-1001 at 2:47-58

C. Priority Date

123. U.S. Patent No. 9,641,841 (the '841 Patent) is entitled to a priority date of March 1, 2011 based on U.S. Provisional Application No. 61/447,862. Dr. Schonfeld has not set forth any opinion otherwise. Ex-1003, ¶96.

D. Prosecution History

124. The '841 Patent issued from U.S. Application No. 14/001,627, which was filed on August 16, 2013 and claims priority to U.S. Provisional Application No. 61/447,862, filed March 1, 2011. Below is a summary of events during the prosecution of the application that led to the issuance of the '841 Patent.

125. U.S. Patent Application No 14/001,627 is the national stage entry of PCT/SE2011/051526. The applicant made claim amendments prior to examination before the USPTO to remove multiple claim dependencies and ensure the claims comported with U.S. prosecution practice. '841 File History at preliminary claim amendments, dated August 26, 2013 and October 30, 2013).

126. The examiner rejected all claims in an office action dated January 21, 2016. First, the examiner rejected pending claims 49 and 50 as directed to non-statutory subject matter under § 101 because the claims were drawn to computer programs per se. Thus, the Examiner considered whether the claims were directed to an abstract idea during prosecution. In addition, the Examiner also rejected claims 34-42 and 44-50 as obvious under § 103 over Bjontegaard '048 (US 2006/0078048) (Ex-1025) in view of List ("Adaptive Deblocking Filter," IEEE Transactions on Circuits and systems for Video Technology, Vol. 13, No. 7, July 2003) (Ex-1024). See '841 File History (Ex-1002) at Office Action dated January 21, 2016. The cited

portions of Bjontegaard '048 taught the equation used for filtering decisions based on the sum of Laplacian equations compared to a threshold. Ex-1002 at Office Action dated January 21, 2016 at pp. 409-417. For example, Bjontegaard '048 disclosed that filtering across a block boundary is performed if the sum of two Laplacian equations (each with the form $x - 2y + z$) was less than a threshold value β :

[0042] Then, filtering of all 4 edge lines are performed if:

$$d=(|a1-2b1+c1|+|d1-2e1+f1|)<\beta \text{ (QP)}$$

Otherwise no filtering is performed.

a0	b0	c0	d0	e0	f0
a1	b1	c1	d1	e1	f1
a2	b2	c2	d2	e2	f2
a3	b3	c3	d3	e3	f3
a4	b4	c4	d4	e4	f4
a5	b5	c5	d5	e5	f5
a6	b6	c6	d6	e6	f6
a7	b7	c7	d7	e7	f7

127. Ex-1025, ¶ [0042] and Fig. 3. Thus, Bjontegaard '048, which was cited against the '841 Patent during prosecution, disclosed a single filtering decision based

on Laplacian equations that applied on both sides of the block boundary based on the calculation of d .

128. The '841 Patent explicitly references List in its specification as describing additional information relating to deblocking filtering in H.264. Ex-1001 at 2:1-4. For example, List describes the same spatial activity condition on either side of the block boundary for filtering decisions that are disclosed in the H.264 standard. Specifically, List discloses that the two equations below are used to determine the extent of the filtering for luma pixel samples:

$$|p_2 - p_0| < \beta(\text{Index}_B) \quad (8)$$

$$|q_2 - q_0| < \beta(\text{Index}_B). \quad (9)$$

Ex-1024 at 617.

129. Likewise, H.264 discloses these same equations given by a_p and a_q , which are used to determine the filtering strength if $a_p < \beta$ and $a_q < \beta$:

Let a_p and a_q be two threshold variables specified by

$$a_p = \text{Abs}(p_2 - p_0) \quad (8-339)$$

$$a_q = \text{Abs}(q_2 - q_0) \quad (8-340)$$

The filtered result sample p'_1 is derived as follows

- If chromaEdgeFlag is equal to 0 and a_p is less than β ,

$$p'_1 = p_1 + \text{Clip3}(-t_{c0}, t_{c0}, (p_2 + ((p_0 + q_0 + 1) \gg 1) - (p_1 \ll 1)) \gg 1) \quad (8-341)$$

- Otherwise (chromaEdgeFlag is equal to 1 or a_p is greater than or equal to β),

$$p'_1 = p_1 \quad (8-342)$$

The filtered result sample q'_1 is derived as follows

- If chromaEdgeFlag is equal to 0 and a_q is less than β ,

$$q'_1 = q_1 + \text{Clip3}(-t_{c0}, t_{c0}, (q_2 + ((p_0 + q_0 + 1) \gg 1) - (q_1 \ll 1)) \gg 1) \quad (8-343)$$

Ex-1037 at 207-08.

130. Thus, the deblocking decision disclosed in Bjontegaard '048 that uses Laplacian equations for filtering decisions as discussed above was considered by the examiner during prosecution, in combination with the filtering decisions applied to the “p” block and the “q” block shown in List and H.264.

131. In response to this office action, applicant argued that the cited references failed to disclose the limitations of the pending claims. First, applicant distinguished the claims by summarizing the independent claims as “(a) calculating a first filter decision value, based on three pixels in a row on side of a block boundary, (b) calculating a second filter decision value, based on three pixels in a row on the opposite side of the block boundary, (c) determining how many pixels to filter on the first side of the block boundary based on a comparison of the first filter

decision value to a first threshold value, and (d) determining how many pixels to filter on the other side of the block boundary based on a comparison of the second filter decision value to a second threshold value.” Ex-1002 at April 20, 2016 Office Action Response at 503-507. Therefore, the applicant explained that the pending claims required two separate “determinings” of how many pixels to filter, based on two separate comparisons, which thereby allows asymmetric filtering on either side of the block boundary. Ex-1002 at April 20, 2016 Office Action Response at 503-507. The applicant distinguished the invention over the prior art, explaining that Bjontegaard ’048, for example, disclosed a single calculation and a single comparison to determine whether filtering should be applied. Ex-1002 at April 20, 2016 Office Action Response at 503-507. Nothing in the Bjontegaard ’048 application demonstrates two separate determinations of how many pixels in the first block should be filtered and how many pixels in the second block should be filtered. Ex-1002 at April 20, 2016 Office Action Response at 503-507. That is, the applicant distinguished its asymmetric filtering decisions to Bjontegaard ’048’s techniques in which a single filter/no-filter decision is applied to both neighboring blocks, and in which the subsequent filtering operations are likewise symmetrical in that they involve the same number of pixels on either side of the block boundary. This contrasts with the invention because then-pending claim 34 recited a determination of how many pixels on one side of the block boundary are to be filtered, and a

separate determination of how many pixels on the other side of the block boundary are to be filtered, thus disclosing asymmetric filtering. Ex-1002 at April 20, 2016 Office Action Response at 503-507.

132. The examiner issued a notice of allowance on December 27, 2016 and the '841 Patent issued on April 12, 2017. Ex-1002 at 512, 583.

X. ASSERTED REFERENCES

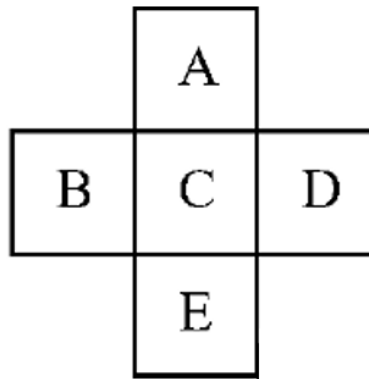
133. Dr. Schonfeld asserts that the '841 Patent is disclosed and/or obvious in light of the foregoing prior art references. For the reasons I set forth in detail below, I disagree. Below I provide a description of the references. In my opinion, Petitioner has identified a number of references with varying approaches to deblocking filter design.

A. Bjontegaard (Ex-1009)

134. Bjontegaard is H.26L Submission Document Q15-J-28 titled "Definition of the new coding elements form Telenor," ITU-T-Q.15/SG16 dated May 16-18, 2000. The H.26L Test Model to which Bjontegaard was submitted was under development as part of the process that led to the development and publication of H.264. Bjontegaard describes the coding model proposed by Telenor in May 2000 for VCEG's Tenth Meeting in Osaka, Japan. In Bjontegaard, a 4x4 block of pixels given by the A, B, C, D, or E in the figure below is assigned a Strength value ranging from 0 to 3, which is based on different criteria. The maximum value is chosen, and

it is assumed that one criterion, set forth below will result in a “value” to be assigned.

The new assignment is $\text{Strength} = \max(\text{Strength}, \text{value})$.



Ex-1009, 18.

135. The Strength value is determined for each of the above exemplary blocks and begins with an initially assigned $\text{Strength}=0$. This is followed by three criteria denoted as Criteria 1, Criteria 2, and Criteria 3, which depend on the attributes of the blocks illustrated above. Ex-1009, 19. If the criteria are not met, the $\text{Strength} = 0$. Filtering will only apply if the Strength value for blocks 1 and 2 are not equal to 0. Consequently, whether filtering applies to the blocks is dependent on whether any of Criteria 1-3 are present. *Id.*

136. In the scenario where Strength1 and Strength2 are both non-zero, the filtering takes place and the following parameters are calculated. In these expressions, a line of 8 pixel values comprising 4 pixels from each of two blocks either side of a block boundary, is denoted ‘abcd | efgh’, which corresponds to the

'841 patent's notation of 'p3p2p1p0 | q0q1q2q3', where '|' denotes the block boundary:

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

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

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

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

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

Ex-1009 at 19. These values may then be used to calculate the value of parameters Act_1 , Act_{11} , Act_2 , Act_{22} , and $Act_{overall}$. the overall activity for both blocks is calculated as $Act_{overall} = (3-i)$ where i is the smallest number such that $S0 < Lim(qp) \times 2^i$. Next, the activity on each side is calculated as Act_1 and Act_2 based on the following:

- Act_1 is initially set to 1
- If $S0_1 < Lim(qp)$, Act_1 is incremented by 1
- If $S1_1 < Lim(qp)$, Act_1 is incremented by 1

The same procedure is made for Act_2 .

Ex-1009 at 19. Thus, Act_1 and Act_2 have a value of either 1, 2, or 3 based on the comparison of $S0_1$, $S1_1$ and $S0_2$, $S1_2$, to $Lim(qp)$, respectively.

137. Subsequently, Act_{11} is calculated as the minimum value between the overall activity value $Act_{overall}$ and the activity value for block 1 (Act_1) and Act_{22} is

calculated as the minimum value as between $Act_{overall}$ and Act_2 . Thus, the values of Act_{11} and Act_{22} could be based either on the activity of one side of the block (e.g., using pixels only on the left side of the boundary for block 1 and pixels on the right side of the block boundary for block 2), or on the calculated overall activity, $Act_{overall}$, which, as shown above, depends on equation S0, which is calculated using pixels on both sides of the block boundary. Ex-2001 ¶ 160.

138. Once these preliminary values are calculated, the Bjontegaard algorithm operates to always calculate Diff and always filter pixels c, d, e, and f. To do so, the Bjontegaard algorithm first sets out the steps for modifying pixels d and e, which are the closest pixels to the block boundary on the left and right side, respectively. Next, Act_{11} and Act_{22} are used to calculate the value of Act_{edge} by dividing the sum of Act_{11} and Act_{22} by 2. A clipping function is then used to calculate $Clip_{edge}$ as depicted below:

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

$$Clip_{edge} = (Clip(qp, Strength_1) + Clip(qp, Strength_2) + Act_1 + Act_2 - 2)/2$$

139. The offset (i.e., the variable “Diff”) is then calculated based on the value of Act_{edge} and $Clip_{edge}$, and Diff is added to the pixel value of d to yield $d_{modified}$, and subtracted from the pixel value e to yield $e_{modified}$. The specific steps for calculating Diff are shown below. Like Petitioner’s analysis of the deblocking

process of other references, such as that of Fu, Petitioner's and Dr. Schonfeld's analysis of Bjontegaard suffers the same shortcoming of selecting one offset (*e.g.*, $\text{Diff} = 3(c - f) - 8(d - e)/16$) out of a number of possible offsets, without any explanation as to why that offset would be selected by a POSITA, or whether or how often the particular offset would be used in the deblocking process. For example, Diff may be calculated three different ways based on the value of Act_{edge} as shown below, and Petitioner fails to address why or how often $\text{Diff} = (3(c - f) - 8(d - e))/16$ would be used as opposed to other possible Diff values.

Initially set $\text{Diff} = 0$

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

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

$\text{Diff} = \max(-\text{Clip}_{\text{edge}}, \min(\text{Clip}_{\text{edge}}, \text{Diff}))$

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

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

140. As shown, the value of Diff will be calculated differently depending on whether Act_{edge} is equal to 1, 2, or 3.

141. Bjontegaard further describes the steps for modifying pixel c, which is the second closest pixel to the block boundary on the left side, as depicted below:

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)

Ex-1009 at 20.

142. Act_{11} is the minimum as between Act_1 and $\text{Act}_{\text{overall}}$. The value of Act_1 may be set to 1, 2, or 3 based on the comparison of $S0_1$ and $S1_1$ to $\text{lim}(\text{qp})$, and $\text{Act}_{\text{overall}}$, which is based on pixels on both sides of the block boundary. Likewise, the value of Act_2 may be set to 1, 2, or 3 based on the comparison of $S0_2$ and $S1_2$ to $\text{lim}(\text{qp})$, and $\text{Act}_{\text{overall}}$, which is based on pixels on both sides of the block boundary. Thus, the equations for Act_1 and Act_2 , are not always used for filtering decisions, for example, the values of Act_1 and/or Act_2 are ignored if $\text{Act}_{\text{overall}}$ is less than Act_1 and/or Act_2 .

143. The value of Diff is determined based on the value of Act_{11} for the modification of pixel c . Diff changes based on Act_{11} being equal to 2 or 3. Diff is then clipped and added to c to yield $\text{C}_{\text{modified}}$ as shown below:

Initially set $\text{Diff} = 0$

If Act_{22} equals 2: $\text{Diff} = 5(e - 2f + g)/16$

If Act_{22} equals 3: $\text{Diff} = 3(d + e - 4f + g + h)/16$

$\text{Diff} = \max(-\text{Clip}(qp, \text{Strength}_2), \min(\text{Clip}(qp, \text{Strength}_2), \text{Diff}))$

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

144. Ex-1009 at 20. The operation for modifying pixels f are similar to pixel c , but Diff changes based on Act_{22} being equal to 2 or 3. Diff is then added to f to yield f_{modified} . As set forth above, the value of Diff is always applied to pixels d , e , c , and f , and therefore these pixels are always filtered. While the deblocking design in Bjontegaard was submitted to H.26L, it was eventually rejected in favor of the deblocking design published in the H.264 Standard that is described in List. Ex. 1024.

B. Bjontegaard's Code (Ex-1010-1015)

145. In his opinions with respect to the '841 Patent, Dr. Schonfeld relies on the "Bjontegaard's Code" as a prior art reference which includes the following references:

Printout of Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29 (q15j29.zip) (Ex-1010);

- Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29, ldecod.c file (q15j29.zip\ldecod\src\ldecod.c) (Ex-1011);
- Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29, image.c file (q15j29.zip\ldecod\src\image.c) (Ex-1012);
- Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29, lencod.c file (q15j29.zip\lencod\src\lencod.c) (Ex-1013);
- Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29, image.c file (q15j29.zip\lencod\src\image.c) (Ex-1014); and
- Gisle Bjontegaard et al., ITU - Telecommunications Standardization Sector, Study Group 16 VCEG (Question 15), Document Q15-J-29, Readme (q15j29.zip\readme.txt) (Ex-1015)

(collectively, “Bjontegaard’s Code”). Bjontegaard’s Code is actually a compilation of six different references executing the Q15-J-29 algorithm.

146. As a preliminary matter, I do not believe it is appropriate to refer to these references as the “Bjontegaard’s Code” as it implies that the source code is the work of Bjontegaard alone. As the files indicate, the draft TML codec (Exs-1011-15) included contributions from other participants in the H.26L standardization process. Exs-1011-15. However, for clarity, I will refer to the references executing the Q15-J-29 algorithm as “Bjontegaard’s Code.” The draft TML codecs comprising

“Bjontegaard’s Code” are parts of the Test Model Software that was developed during the development of H.26L. As I explained earlier, the development of H.26L eventually led to the publication of the H.264 Standard. Thus, the deblocking design described in Bjontegaard’s Code was ultimately rejected in favor of the deblocking design in the published H.264 Standard that is described in List. Ex-1024.

147. **Bjontegaard’s Code Q15-J-29 Algorithm.** In Q15-J-29, image.c (Ex-1012) at lines 430-433, the overall activity of neighboring blocks is measured. This is done using the calculation of i0, i1, and iaa. Specifically, $i0 = 4 * (img \rightarrow li[2] - img \rightarrow li[5]) - 12 * (img \rightarrow li[3] - img \rightarrow li[4])$, $i1 = 9 * (img \rightarrow li[2] + img \rightarrow li[5] - img \rightarrow li[3] - img \rightarrow li[4])$. The calculation of i0 and i1 in this section of the source code is based on pixels on both sides of the block boundary. Corresponding to a line of pixels p3p2p1p0 | q0q1q2q3 (see ¶ 107 above), this section of source code denotes a line of pixels across a block boundary as li[0]li[1]li[2]li[3] | li[4]li[5]li[6]li[7]. Using these parameters, iaa is calculated as $iaa = \text{abs}(i0) + \text{abs}(i1)$. The software then performs a loop to derive the value of the parameter ia as set forth below.

```

435     if((ibl==3) || (ibr==3))
436         ia=1;
437     else
438         ia=0;
439     for(ii=3;ii>0;ii--)
440     {
441         iii=(int)pow(2,ii-1);
442         if(iaa<(iii*lim))
443             ia=4-ii;
444     }

```

148. Ex-1012, lines 435-444. Then, the parameters *il3*, *i0*, *i1*, and *il4* are calculated (for the left side of the block boundary) and the parameters *ir3*, *i0*, *i1*, and *ir4* are calculated (for the right side of the block boundary) as set forth below. Dr. Schonfeld relies on the calculation of *il3* and *ir3*, but, as described below, Dr. Schonfeld ignores that the value *il3* *as well as* *i0* and *i1*, are used to then determine the value of *icl*. The same is true on the right side (e.g., *ir3* as well as *i0* and *i1* are used). The relevant source code is shown below.

```

il3=12*abs(img->li[1]+img->li[3]-2*img->li[2]);
i0=4*(img->li[0]-img->li[3])-12*(img->li[1]-img->li[2]);
i1=9*(img->li[0]+img->li[3]-img->li[1]-img->li[2]);
il4=abs(i0)+abs(i1);
icl=1;
if(il3<lim)
    icl++;
if(il4<lim)
    icl++;

ir3=12*abs(img->li[4]+img->li[6]-2*img->li[5]);
i0=4*(img->li[4]-img->li[7])-12*(img->li[5]-img->li[6]);
i1=9*(img->li[4]+img->li[7]-img->li[5]-img->li[6]);
ir4=abs(i0)+abs(i1);
icr=1;
if(ir3<lim)
    icr++;
if (ir4<lim)
    icr++;

```

149. Ex-1012, lines 446-464. For the left block, *icl* is initially set equal to 1, and is incremented by 1 if *il3* is less than *lim*. If *il4* is less than *lim*, then *icl* is incremented by 1. The same procedure happens for the right block—*icl* is initially set equal to 1, and is incremented by 1 if *ir3* is less than *lim*. If *ir4* is less than *lim*, then *icr* is incremented by 1.

150. Then, the value of idl, idr and idlr are defined. The variable idl is defined as the minimum of ia and icl. The variable idr is defined as the minimum of ia and icr. And the variable idlr is defined as the average of those values, i.e., defined as $(idl+idr)/2$. Ex. 1012, lines 466-468.

151. Then the comments in the code indicate “start filtering” for the edge as shown in the code below:

```

475      /* start filtering */
476      /* edge */
477      id =0;
478      if(idlr==2 || idlr==1)
479          id=(3*(img->li[2]-img->li[5])-8*(img->li[3]-img->li[4]))/16;
480      if(idlr==3 )
481          id=(img->li[1]+img->li[2]-8*(img->li[3]-img->li[4])-img->li[5]-img->li[6])/16;
482      id =max(-islr,min(id,islr));
483      il1=img->li[3]+id;
484      ir1=img->li[4]-id;
485      img->li[3]=il1; /* set back left edge */
486      img->li[4]=ir1; /* set back right edge */

```

Ex-1012, lines 475-486.

152. Here, the variable id is initially set equal to 0. If the value of idlr calculated as explained is equal to 1 or 2, then the variable id is defined in line 479. Otherwise, if idlr is equal to 3, then the value of id is defined in line 481. Thus, the value of id is always calculated, but it may take on different values based on the value of idlr. Then the variable id is further calculated as the maximum of (i) the negative of the variable islr and (ii) the minimum of id and islr. Then the variable il1 (pixel il1) is calculated as pixel li[3] plus the value of id and the variable ir1 (pixel ir1) is calculated as pixel li[4] minus the value of id. These variables il1 and ir1 are

temporarily copied into image sample locations `img->li[3]` (corresponding to `p0`) and `img->li[4]` (corresponding to `q0`), respectively.

153. The filtering algorithm for the center left and center right pixels is defined in lines 488-503 (Ex-1012).

```

488  /* center left */
489  id = 0;
490  if(idl==2)
491      id=5*(img->li[1]-2*img->li[2]+img->li[3])/16;
492  if(idl==3 )
493      id=3*(img->li[0]+img->li[1]-4*img->li[2]+img->li[3]+img->li[4])/16;
494  id =max(-isl,min(id,isl));
495  il2=img->li[2]+id;
496  /* center right */
497  id = 0;
498  if(idr==2)
499      id=5*(img->li[4]-2*img->li[5]+img->li[6])/16;
500  if(idr==3 )
501      id=3*(img->li[3]+img->li[4]-4*img->li[5]+img->li[6]+img->li[7])/16;
502  id =max(-isr,min(id,isr));
503  ir2=img->li[5]+id;

```

154. For center left filtering, the value of `id` is initially set equal to 0. If the value of `idl`, defined above, is equal to 2, then the value of `id` is defined as per line 491. If the value of `idl` is equal to 3, then the value of `id` is defined as per line 493, shown above. The resulting value of `id` is then taken as the maximum value of (i) the negative value of `isl`, and (ii) the minimum of `id` and `isl`. The pixel value `il2` (corresponding to `p1`) is calculated at line 495 as `li[2]` plus `id`.

155. For center right filtering, the value of `id` is initially set equal to 0. If the value of `idr`, defined above, is equal to 2, then the value of `id` is calculated as per line 499 shown above. If the value of `idr` is equal to 3, then the value of `id` is calculated

as per line 501, shown above. The resulting value of *id* is then calculated as the maximum value of (i) the negative value of *isr*, and (ii) the minimum of *id* and *isr*. The pixel value *ir2* (corresponding to *q1*) is then calculated at line 503 as *li*[5] plus *id*.

156. In Bjontegaard's Code (e.g., Ex-1012), there are several incompatibilities with the algorithm described in Q15-J-28 (Bjontegaard, Ex-1009). For example, the equation for S_{01} in Section 4.5.6.1 of Bjontegaard is different from the calculation of *il3* in Bjontegaard's Code and the use of *d* and *e* in four equations in Bjontegaard in Sections 4.5.6.1 and 4.5.6.2 are different from the use of *d*+Diff or *e*-Diff in Bjontegaard's Code (e.g., Ex-1012, lines 479, 481, 491, 493). The inconsistent use of these values result in different filtering results when executed or implemented, as I illustrate in my "testing deblocking filter" simulations discussed below.

157. I analyzed the portions of C language_code (from Bjontegaard's Code) that Dr. Schonfeld relied on and determined that the C code comprises calculations that (a) are inconsistent with the algorithm described in Q15-J-28 in several ways and (b) result in incorrectly calculated filtered values, due to an inherent flaw in the design of the software. In particular, Dr. Schonfeld relies on C source code file *image.c* at lines 485 and 486. Under Dr. Schonfeld's analysis, the value of *li*[3] (corresponding to *d* in Bjontegaard or *p0* in the '659 Patent) is modified (value *il1*,

corresponding to $d + \text{Diff}$ in Bjontegaard) and “redefine[d]” prior to reusing $li[3]$ in the calculation of $c + \text{Diff}$ (image.c at lines 491, 493). Ex-1003, ¶210. Similar steps apply for the corresponding operations on the other side of the block boundary for $li[4]$, $ir1$, and the calculation of $f + \text{Diff}$. *Id.*

158. However, the Bjontegaard Code is written in such a way as to cause overflow and underflow errors when processing pixel values that are at or near the minimum or maximum values (0 or 255).

159. Specifically, the variables $il1$ and $ir1$ are defined as integers, i.e. type `int` (Ex-1012 at 425), whereas variables $li[3]$ and $li[4]$ are defined as unsigned 8-bit values, i.e. type `unsigned char` (Ex-1010, `global.h` at 23). Unsigned char variables can only take values in the range 0-255 (as discussed in relation to my test simulation above). When $li[3]$ or $li[4]$ comprise values at or around the minimum or maximum (0 or 255), adding or subtracting Diff will sometimes lead to a value of $il1$ or $ir1$ that is less than 0 or greater than 255. The software in Bjontegaard’s Code (Ex-1012 at lines 485 and 486) assigns an `int` ($il1$ or $ir1$) that may be, and sometimes will be, outside the range 0-255, to an unsigned char ($li[3]$ or $li[4]$) which cannot take on values outside the range 0-255. This results in what a POSITA would recognise as an underflow or overflow error.

160. I provide an example of the disparate results obtained by Q15-J-28 and Q15-J-29 as a result of this error:

Assume that $li[1]=2$, $li[2]=0$, $li[3]=0$, (i.e. $b=2$, $c=0$, $d=0$), and $Diff = -1$

$il1 = d_{modified} = d + Diff = -1$

Assign $li[3] = il1$

$li[3]$ (which has type unsigned char) cannot take the value of -1

$li[3]$ instead becomes **255** because of underflow error.

Comparing the results of this calculation according to the two methods:

Q15-J-28 Method: Diff equation is $5*(b - (2*c) + d)/16 = 5*(2 - 0 + 0)/16 = 1$ (with rounding) (Section 4.5.6.1).

Q15-J-29 Method: Diff equation is $(5*(b - (2*c) + li[3])/16 = 5*(2 - 0 + 255)/16 = 80$ (with rounding)

161. In this example, the inherent error in the source code of Q15-J-29 results in a calculated Diff of 80 instead of 1.

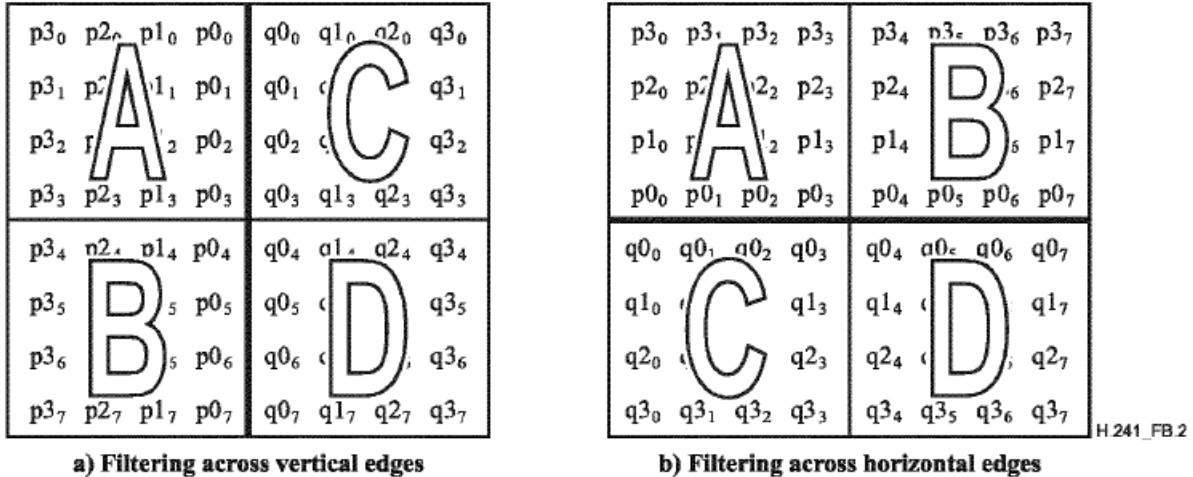
162. I provide further examples of the effect of such overflow or underflow errors in my simulation file, TestUint8.c, described above.

163. The result of such an overflow or underflow error in the filtering of one video frame can and most likely will affect the processing and encoding/decoding of further video frames. As I explained earlier, inter-predicted blocks of pixels in a video frame are predicted from previously coded video frames. Thus, a filtering error in one video frame will result in further coded video frames being predicted from incorrectly-calculated values, i.e. each instance of an overflow or underflow error is likely to propagate through subsequent coded video frames.

C. H.241 (Ex-1017)

164. ITU-T's H.241 Recommendation titled "Telecommunication Standardization Sector of ITU, *ITU Recommendation H.241: Extended video procedures and control signals for H.300-series terminals* (May 29, 2006) was published in May 2006. H.241 states that it defines the use of advanced video codecs, including ITU-T Rec. H.264, in ITU-T Recs H.310, H.320, H.321, H.322, H.323 and H.324 terminals. Ex-1017 at Summary.

165. H.241 specifies a filtering process for luma samples. Figure B.2 illustrates the organization and notations of exemplary 8x8 blocks of luma samples, which can be organized for filtering across vertical or horizontal edges.



Ex-1017 at 28.

166. H.241 describes that the variable d, as specified below, is used for deciding whether samples within an 8 x 8 block are modified or not. The variable d is specified based on whether the block edge is a part of a macroblock edge or not:

- If the block edge is part of a macroblock edge:
$$d = |p_{1_2} - p_{0_2}| + |q_{0_2} - 2*q_{1_2} + q_{2_2}| + |p_{1_5} - p_{0_5}| + |q_{0_5} - 2*q_{1_5} + q_{2_5}|$$
- Otherwise (the block edge is not part of a macroblock edge):
$$d = \left| \begin{array}{c} p_{2_2} - 2*p_{1_2} + p_{0_2} \\ q_{0_5} - 2*q_{1_5} + q_{2_5} \end{array} \right| + |q_{0_2} - 2*q_{1_2} + q_{2_2}| + |p_{2_5} - 2*p_{1_5} + p_{0_5}| +$$

Ex-1017 at 29.

167. Additional conditions must be met for determining whether to apply filtering across a block edge as follows:

- The variable filterBlockEdge shall be equal to 1 if one or more of the following conditions are true:
 - Block A is intra coded or Block C is intra coded.
 - One or more of the blocks A, B, C and D contains non-zero transform coefficients.
 - The absolute difference between the horizontal or vertical component of the motion vectors used for block A and block C is greater than or equal to 4 in units of quarter luma frame samples.
 - Block A and block C are predicted from different reference frames.
- Otherwise, the value of filterBlockEdge shall be equal to 0.
- If one or more of the following conditions are true, no filtering across a block edge takes place.
 - disable_deblocking_filter_idc for the slice containing block D is equal to 1.
 - disable_deblocking_filter_idc for the slice containing block D is equal to 2 and block B and D belong to different slices.
 - $d \geq \beta$.
 - filterBlockEdge = 0.

Ex-1017 at 29.

168. Otherwise, filtering takes places as further specified. For block edges across which filtering is to be performed, the values p_{ni} , q_{ni} , with $n = 0, 1$ and $i = 0, 7$ are modified as follows:

- If the block edge is not part of a macroblock edge:
$$\Delta = \text{Clip3}(-t_c, t_c, ((q_{0i} + ((p_{2i} + q_{1i}) \gg 1)) \gg 1) - ((p_{0i} + ((q_{2i} + p_{1i}) \gg 1)) \gg 1))$$
- Otherwise (the block edge is part of a macroblock edge):
$$\Delta = \text{Clip3}(-t_c, t_c, ((q_{0i} + (q_{1i} \gg 1)) \gg 1) - ((p_{0i} + (q_{2i} \gg 1)) \gg 1))$$

When the filtering is applied, the filtered samples p_{1i} , p_{0i} , q_{0i} , q_{1i} are derived by:

$$\begin{aligned} p_{0i} &= \text{Clip1}_Y(p_{0i} + \Delta) \\ q_{0i} &= \text{Clip1}_Y(q_{0i} - \Delta) \\ p_{1i} &= \text{Clip1}_Y(p_{1i} + \Delta/2) \\ q_{1i} &= \text{Clip1}_Y(q_{1i} - \Delta/2) \end{aligned}$$

169. Ex-1017 at 29.

D. B205

170. The ITU-T's and ISO/IEC's Joint Collaborative Team on Video Coding (JCT-VC) published B205, titled "Test Model Under Consideration," as a draft of its High-Efficiency Video Coding (HEVC) Standard. Ex-1018, 1. As explained above, B205 is mentioned in the Background of the '841 Patent as the prior HEVC draft standard that the inventions of the '841 Patent improves upon. Ex-1001, 2:19-22. The '841 Patent explicitly states that it improves upon the deblocking filter in B205 by achieving asymmetric deblocking decisions that control deblocking filtering such that it is adaptive to the structure on each side of the block boundary. Ex-1001 at 4:62-64. The '841 Patent explains that "asymmetric" refers to the fact 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 at 4:64-5:2.

171. In addition, in a related application with the same inventors and also directed to the same deblocking filter, U.S. Patent No. 10,142,659 states that a problem in the deblocking filter disclosed in B205 is that “the weak filter does not have good low-pass characteristics” and has a frequency response that can cause “worse rate-distortion performance” (i.e. poorer compression performance) in the coder Ex-1063at 3:18-33. The filter disclosed in the '841 Patent explicitly improves upon B205 to address these shortcomings.

172. The B205 reference on its face lists “2nd Meeting: Geneva, CH, 21-28 July, 2010.” Ex-1018, 1. The B205 reference discloses a deblocking filter based on an 8x8 block structure with the below pixel notation:

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
A p3 ₃	p2 ₃	p1 ₃	p0 ₃	q0 ₃	q1 ₃	q2 ₃	q3 ₃ B
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

Notation of an 8 pixel part of a vertical edge for deblocking

Ex-1018 at 133.

173. B205 also includes luma filtering, where a part of the luma edge can be filtered if a number of alternative conditions are met, such as if block A or B shown above has a prediction mode equal to MODE_INTRA *and* the below equation is true. Ex-1018, 133.

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

174. If the edge between blocks is filtered, one of two types of filtering (weak or strong filtering) is performed. The choice between the strong and the weak filtering is done separately for each line depending on certain conditions. If the three below conditions are true, strong filtering is performed. Otherwise, weak filtering is performed.

- $d < (\beta \gg 2)$
- **and** $(|p3_i - p0_i| + |q0_i - q3_i|) < (\beta \gg 3)$
- **and** $|p0_i - q0_i| < ((5 * t_c + 1) \gg 1),$

175. In the strong filtering case, pixels $p0$, $p1$, $p2$, $q0$, $q1$, and $q2$ are all filtered as shown below. Ex-1018, 134.

Strong filtering

Strong filtering mode is performed with the following set of operations.

$p0_0 = \text{Clip}_{0-255}((p2_i + 2 * p1_i + 2 * p0_i + 2 * q0_i + q1_i + 4) \gg 3);$	$i = 0, 7$
$q0_0 = \text{Clip}_{0-255}((p1_i + 2 * p0_i + 2 * q0_i + 2 * q1_i + q2_i + 4) \gg 3);$	$i = 0, 7$
$p1_0 = \text{Clip}_{0-255}((p2_i + p1_i + p0_i + q0_i + 2) \gg 2);$	$i = 0, 7$
$q1_0 = \text{Clip}_{0-255}((p0_i + q0_i + q1_i + q2_i + 2) \gg 2);$	$i = 0, 7$
$p2_0 = \text{Clip}_{0-255}((2 * p3_i + 3 * p2_i + p1_i + p0_i + q0_i + 4) \gg 3);$	$i = 0, 7$
$q2_0 = \text{Clip}_{0-255}((p0_i + q0_i + q1_i + 3 * q2_i + 2 * q3_i + 4) \gg 3);$	$i = 0, 7$

176. In the weak filtering case, a Δ value is calculated, and clipped within a range of $-t_c$ and t_c . The Δ value is calculated using six pixels, $p0$, $p1$, $p2$, $q0$, $q1$, and $q2$. The clipped Δ value is then added or subtracted to the $p0$ and $q0$ pixels to form filtered pixel values, that are likewise clipped within a range of 0 and 255. The calculated Δ value is then divided by two, before being added or subtracted to the $p1$ and $q1$ pixels respectively to form filtered pixel values that are likewise clipped within a range of 0 and 255. Ex-1018, 134.

Weak filtering

Weak filtering mode filtering is performed based on the above conditions,

$$\begin{aligned} \Delta &= \text{Clip}(-tc, tc, (13*(q0_i - p0_i) + 4*(q1_i - p1_i) - 5*(q2_i - p2_i) + 16) >> 5)) & i = 0,7 \\ p0_i &= \text{Clip}_{0-255}(p0_i + \Delta) & i = 0,7 \\ q0_i &= \text{Clip}_{0-255}(q0_i - \Delta) & i = 0,7 \\ p1_i &= \text{Clip}_{0-255}(p1_i + \Delta/2) & i = 0,7 \\ q1_i &= \text{Clip}_{0-255}(q1_i - \Delta/2) & i = 0,7 \\ q2_0 &= \text{Clip}_{0-255}((p0_i + q0_i + q1_i + 3*q2_i + 2*q3_i + 4) >> 3); & i = 0,7 \end{aligned}$$

E. Aksu (Ex-1019)

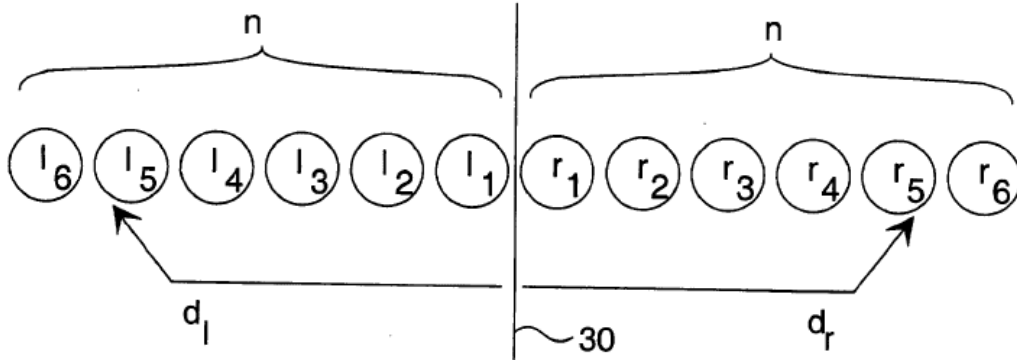
177. PCT/FI01/00049 (“Aksu”) is an International Patent Application filed on January 22, and published on July 26, 2001.

178. Aksu is directed to a “method and associated device for filtering digital video images”, and discloses that it relates to a method for reducing visual artefacts in a frame of a digital video signal, which is coded by blocks and then decoded. Ex-1019 at 1:1-5. Aksu references prior art video transmission systems, for example according to the ITU-T H.261/H.263 recommendations. Thus, the disclosure of Aksu is presumably directed at the development of the then-next generation codec, H.264. In a similar way to the other references discussed herein reflecting contributions to the development of H.264, the design of Aksu was rejected in favor of the design included in the published H.264 Standard, described in List. Ex-1024.

a

179. Figure 2 shows the location of pixels $r_1 - r_6$ and $l_1 - l_6$ in relation to a vertical block boundary, 30, wherein r_1 is the closest pixel to the block boundary in

the right block and l_1 is the closes pixel to the block boundary in the left block. Ex-1019 at 9:27.



Ex-1019, Fig. 2.

180. To implement the deblocking design in Aksu, the parameter n is specified, which is the largest number of pixels to be examined from the block boundary in one direction. Ex-1019 at 9:29-32. In the case of Figure 2 above, the value of n is 6. The parameter n is defined as follows in Aksu for a preferred embodiment of an 8×8 block of pixels, wherein $\alpha = QP \log(QP)$, which is a function of a quantization of the block:

$$n = \begin{cases} 0 & \Delta \geq 2.00\alpha \\ 1 & 1.50\alpha \leq \Delta < 2.00\alpha \\ 2 & 1.00\alpha \leq \Delta < 1.50\alpha \\ 3 & 0.66\alpha \leq \Delta < 1.00\alpha \\ 4 & 0.40\alpha \leq \Delta < 0.66\alpha \\ 5 & 0.25\alpha \leq \Delta < 0.40\alpha \\ 6 & 0 \leq \Delta < 0.25\alpha \end{cases}$$

Ex-1019 at 10:5-10.

181. In the next step of the filtering method in Aksu, the region type information of the two neighboring blocks is examined. According to this information, the value of n may be truncated to provide better deblocking results, which is given by Table 1. Ex-1019 at 10:23-11:15.

type of the Block on the First side of a boundary	type of the Block on the Second side of a boundary							
	INTRA		COPY		CODED		NOT_CODED	
INTRA	n	n	2	2	n	4	n	2
COPY	2	2	2	2	2	4	2	2
CODED	4	n	4	2	4	4	4	2
NOT_CODED	2	n	2	2	2	4	2	2

TABLE 1

Ex-1019 at Table 1.

182. The value on the left of each region type in the truncation value $trval$ for the first side of the block boundary, and the value on the right gives the truncation value $trval$ for the second side of the boundary. The truncated value n is $n_{tr} = \min(trval, n)$

183. The next step of the filtering design in Aksu is to determine the values of d_l and d_r , which represent activity, or the differences of pixel values between pixels on one side of the block boundary. Ex-1019 at 13:10-14. A definition d_r is given by:

$$\begin{aligned} d_r &= 6, \text{ if } |r_1 - r_j| \leq \beta/j \text{ with all } j \in [1, 6], \\ \text{otherwise: } d_r &= i, \text{ where } i \text{ fulfils the conditions} \\ i &\in [1, n_{tr}], \quad (4) \\ |r_1 - r_{i+1}| &> \beta/i, \text{ and} \\ |r_1 - r_j| &\leq \beta/j \text{ with all } j \in [1, i]. \end{aligned}$$

Ex-1019 at 13:15-20.

184. Aksu discloses that the value of d_l is determined similarly, except all r 's are replaced with l 's, and a corresponding truncated n_{tr} must be used. Ex-1019 at 13:20-23. In the above definition, d_r is set equal to six if each of the absolute difference values between r_1 and each of the succeeding pixels in a line until r_6 is less than or equal to β/j , which can be written out as: (1) $|r_1 - r_2| \leq \beta/2$; (2) $|r_1 - r_3| \leq \beta/3$; (3) $|r_1 - r_4| \leq \beta/4$. Ex-1003 ¶133.

185. d_r is set equal to i , where the conditions shown above are filled. The corresponding calculation occurs for d_l .

186. Aksu discloses that when the values of n , n_{tr} , d_r and d_l have been determined, filtering is carried out using a suitable filter, but does not teach or suggest a particular filter that is applied. Ex-1019 at 15:1-5. Table 2 of Aksu shows the determination of the width of the filtering window for the pixels r_1 , r_2 , and r_3 according to the value of d_r . The values of l_1 , l_2 , and l_3 are determined in the same manner according to the value of the d_l . In Table 2, Aksu explains that "X" means that the pixel in question is not filtered at all, and the number of means that the

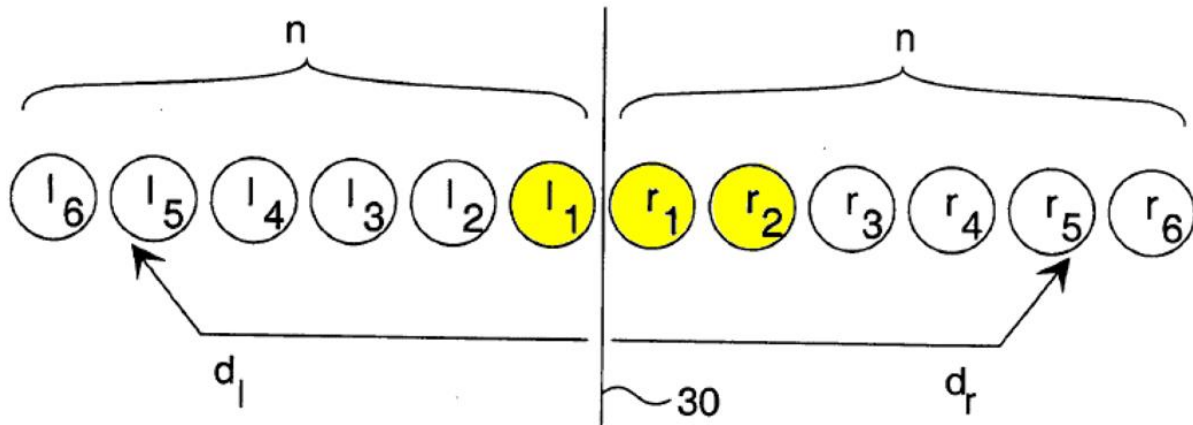
filtering window includes a number of pixels shown by the number from each side of the pixel being examined. Ex-1019 at 15:15-24. Aksu further specifies that as shown in Table 2, for filtering to apply at all to any pixels, d_r and d_l must both have a value greater than 1.

$d_r (d_l > 1)$	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

Ex-1019, Table 2.

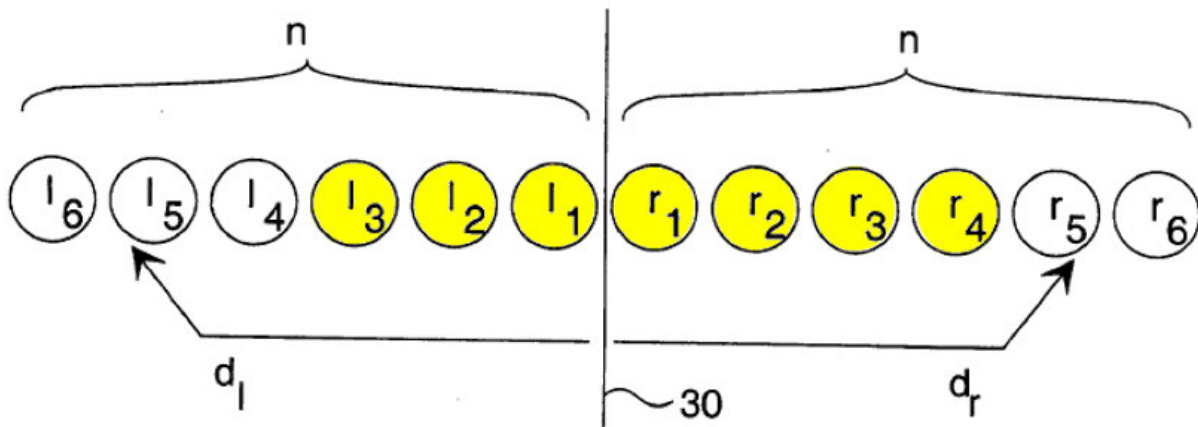
187. Based on the foregoing disclosure of the deblocking design in Aksu, I set forth the following examples of how to determine the way the pixels are filtered according to Aksu's design. If d_r is equal to 2, then the filtering window includes 3 pixels because the number 1 in the table for r_1 "means that the filtering window includes a number of pixels shown the number from each side of the pixel being examined." Ex-1019 at 15:22-24. Thus, if r_1 is in the right block and closest to the block boundary, then the pixel to the left of r_1 (r_2) and the pixel to the right of r_1 (l_1), which is the pixel closest to the block boundary in the neighboring block, will be in the filtering window used to calculate the filter. I illustrate this example below using

Figure 2 of Aksu, which can be developed using the other values in Table 2. In the example below, the filtered pixels are highlighted in yellow.



Ex-1019, Fig. 2 (annotated). In this case, the filter is calculated using Aksu's averaging filter as $\frac{l_1+r_1+r_2}{3}$.

188. Illustrating another example, if $d_r = 6$, then 7 pixels are in the filtering window as shown below because there are 3 pixels from each side of pixel r_1 . In this case, the filter for modifying r_1 is calculated using Aksu's averaging calculator as $\frac{l_3+l_2+l_1+r_1+r_2+r_3+r_4}{7}$.



Ex-1019, Fig. 2 (annotated).

189. Aksu discloses that filtering is carried out using a “suitable filter.” Though it does not limit the filter applied, Aksu discloses a preferred embodiment wherein the filtering window is symmetrical with regard to the pixel to be filtered and a new pixel value (e.g., filtered pixel value) is determined by calculating the mean of the pixel values that appear in the filtering window. Ex-1019 at 15:2-15. This is sometimes referred to by Dr. Schonfeld as an “averaging filter.” See, e.g., Ex-1003 ¶500.

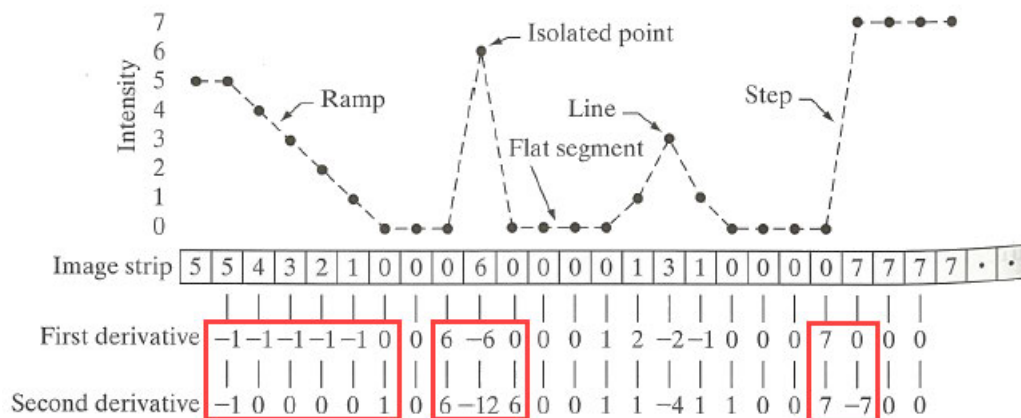
F. Gonzalez (Ex-1020)

190. Digital Image Processing is a textbook by Rafael Gonzalez and Richard E. Woods (“Gonzalez”) was published in 2008. Gonzalez discloses high level concepts directed to types of equations and their behavior. For example, Gonzales includes a sub-chapter titled “The Laplacian in the Frequency Domain,” which discusses the Laplacian and the results when implemented in the spatial and

frequency domains and gives basic equations to illustrate the basic form of these high-level concepts, such as low-pass filtering and high-pass filtering. Ex-1020 at 152-162. For example, Gonzales discloses functions for first and second order derivatives along with visualization aid of the first and second derivative of a 1-D digital function representing a section of a horizontal intensity profile from an image.

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x)$$



Ex-1020 at 694; Ex-1003, ¶143.

G. Fu (Ex-1022)

191. The Fu reference is an International Patent Application that on its face identified an international filing date of August 21, 2008, and a publication date of March 19, 2009. Ex-1022 at cover. The date of Fu, in my view is relevant here, because Petitioner and Dr. Schonfeld assert that a POSITA would have looked at Fu

as the relevant state of the art. However, there were many months of H.265 development that occurred between Fu and the filing of the '659 Patent. In my view, a POSITA would have looked to the then-current draft H.265 Standard described in B205 as the relevant state of the art, which disclosed very different offsets in its description of the deblocking filter compared with Fu. Compare B205 (Ex-1018) with Fu (Ex-1022). A POSITA would have been familiar with the standardization process and understood the continual and incremental nature of improvement of a developing standard and thus, in my view would have worked from the current draft (B205) and would not have considered Fu to be an up-to-date and therefore credible reflection of the state of the art. Dr. Schonfeld opines that Fu discloses a deblocking filter process through “a method and device for ‘reducing block distortion’” via the following offset-filter equation for the pixels (p0 and q0) closest to the block boundary:

$$\begin{aligned}\text{delta} &= \text{Clip3}(-C, C, (((q0 - p0) \times 3 + (p1 - q1) + 4) \gg 3)) \\ P0 &= \text{Clip1}(p0 + \text{delta}) \\ Q0 &= \text{Clip1}(q0 - \text{delta}),\end{aligned}$$

Ex-1003 ¶157. Dr. Schonfeld’s description and analysis of Fu fail to capture the scope of the disclosed deblocking process, which is relevant because there are three possible processes, each using *different* offsets that are applied to *different* pixels, disclosed in the deblocking process described in Fu. Petitioner and Dr. Schonfeld

select a particular offset from Fu without providing any basis as to why a POSITA would select this particular offset from among the offsets used in the three possible deblocking processes disclosed in Fu to arrive at the claimed invention.

192. Specifically, Fu discloses deblocking “Process 1,” “Process 2,” and “Process 3,” the use of which depends on the value of the block boundary filtering strength (BS), which is further based on the macroblock type and “region flatness.” Ex-1022 at 4:3-20. Thus, the determination of when and how to filter pixels in Fu is based on “region flatness,” which Fu describes being measured by calculating the number of “similarity pairs around the block boundary.” Ex-1022 at 2:1-9.

193. Process 1 of Fu is used when BS is equal to 3. In this case, the pixels on both sides of the luma block boundary, p_0 , p_1 , p_2 , q_0 , q_1 , and q_2 are filtered according to the following:

$$\begin{aligned} P0 &= ap ? (2 \times p1 + p2 + 2 \times (p0 + q0) + q1 + 4) \gg 3 : (2 \times p0 + p1 + q0 + 2) \gg 2; \\ P1 &= ap ? (p0 + 2 \times p1 + q0 + 2) \gg 2 : p1; \\ P2 &= ap ? (p0 + p1 + 4 \times p2 + 2 \times p3 + 4) \gg 3 : p2; \\ Q0 &= aq ? (2 \times q1 + q2 + 2 \times (q0 + p0) + p1 + 4) \gg 3 : (2 \times q0 + q1 + p0 + 2) \gg 2; \\ Q1 &= aq ? (q0 + 2 \times q1 + p0 + 2) \gg 2 : q1; \\ Q2 &= (q0 + q1 + 4 \times q2 + 2 \times q3 + 4) \gg 3 : q2; \end{aligned}$$

194. Fu also discloses a second filtering process (Process 2) when the boundary strength is calculated as being equal to 2. Ex-1022 at 7.

Process 2: filter the samples on both sides of the luma block boundary when BS = 2, and only filter the p_0 , p_1 , q_0 and q_1 in Figure 2, and expressed by pseudocode as follows:

$$\begin{aligned} P0 &= ap ? (2 \times p1 + p2 + 2 \times (p0 + q0) + q1 + 4) \gg 3 : (2 \times p0 + p1 + q0 + 2) \gg 2; \\ P1 &= ap ? (p0 + 2 \times p1 + q0 + 2) \gg 2 : p1; \\ Q0 &= aq ? (2 \times q1 + q2 + 2 \times (q0 + p0) + p1 + 4) \gg 3 : (2 \times q0 + q1 + p0 + 2) \gg 2; \\ Q2 &= aq ? (q0 + 2 \times q1 + p0 + 2) \gg 2 : q1; \end{aligned}$$

195. Fu also discloses a third filtering process (Process 3), when BS is calculated as being equal to 1. Ex-1022 at 8. As part of Process 3, a delta value is calculated using a particular equation, where the output is clipped to a range of -C to C. *Id.* This clipped delta value is then either added or subtracted from the sample pixel value (p0 and q0) to create filtered pixel values P0 and Q0. *Id.* The filtered pixel values P0 and Q0 are likewise clipped to be within a range of 0 and 255. *Id.*

Process 3: filter the samples on both sides of the luma block boundary when BS = 1, specifically as follows:

first, filter p0 and q0 (P0 and Q0 are respectively filtered values of p0 and q0), which may be expressed as follows:

$$\text{delta} = \text{Clip3}(-C, C, (((q0 - p0) \times 3 + (p1 - q1) + 4) \gg 3))$$

$$P0 = \text{Clip1}(p0 + \text{delta})$$

$$Q0 = \text{Clip1}(q0 - \text{delta}),$$

wherein Clip3 (a,b,c) is expressed as follows:

$$\text{Clip3}(a, b, c) = \begin{cases} a & ; \quad c < a \\ b & ; \quad c > b \\ c & ; \quad \text{Other} \end{cases}$$

Clip1 (a) is expressed as: Clip1 (x) = Clip3 (0, 255, x)

Id.

196. Once the filtered pixel values P0 and Q0 are calculated, Process 3 determines whether pixel values p1 and q1 also need filtered. If it is determined that the p1 or q1 pixels need filtered, they will be filtered using the equations below. In particular, the filtering of the p1 and q1 pixels requires the filtered pixels P0 and Q0 to determine how much to filter the p1 and q1 pixels. This calculated value is then clipped within the same range as the previously calculated delta, between the values of -C and C. The calculated value is then either added to the p1 pixel, or subtracted

from the q1 pixel, respectively. The output value is then clipped within a range of 0 and 255, and output as filtered pixel values P1 and Q1, respectively.

then, determine whether there is a need to filter p1 and q1, and when there is a need to filter p1 and q1, the calculation of P0 and Q0 may be divided into three cases:

first case: when the block boundary is a chroma block boundary, do not filter p1 and q1.

second case: when ap is greater than 0 at the luma block boundary, filter p1, and the filtered value is as follows:

$$P_1 = \text{Clip1}(p_1 + \text{Clip3}(-C, C((P_0 - p_1) \times 3 + (p_2 - Q_0) + 4) \gg 3))$$

third case: when aq is greater than 0 at the luma block boundary, filter q1, and the filtered value is as follows:

$$Q_1 = \text{Clip1}(q_1 - \text{Clip3}(-C, C((q_1 - Q_0) \times 3 + (P_0 - q_2) + 4) \gg 3))$$

During the filtering process of process 3, the definition of ap and aq is the same as that of process 1 and process 2, and ap is obtained from formula (3) and aq is obtained from formula (4). Moreover, a and P in formula (3) and formula (4) are also the same as in processes 1 and 2. C is called the filtering and clipping parameter, and the relationship between C and IndexA is as shown in Table 3. Table 3 shows the relationship between the filtering and clipping parameter C and IndexA:

Id. at 10.

H. B118 (Ex-1023)

197. The B118 reference is the Working Draft Number 2 (WD-2) released by the Joint Video Team after the JVT meeting occurring in Geneva, Switzerland, from January 29 through February 1, 2002. Ex-1023 at 1. The B118 references shows a “last saved” date of March 15, 2002. *Id.* As part of the current working draft for JVT in 2002, the B118 reference discloses a deblocking filter used on the boundaries of a 4x4 block structure. Ex-1023 at 46.

198. The B118 deblocking filter will first determine the boundary strength for the block boundary between two neighboring blocks. *Id.* at 47.

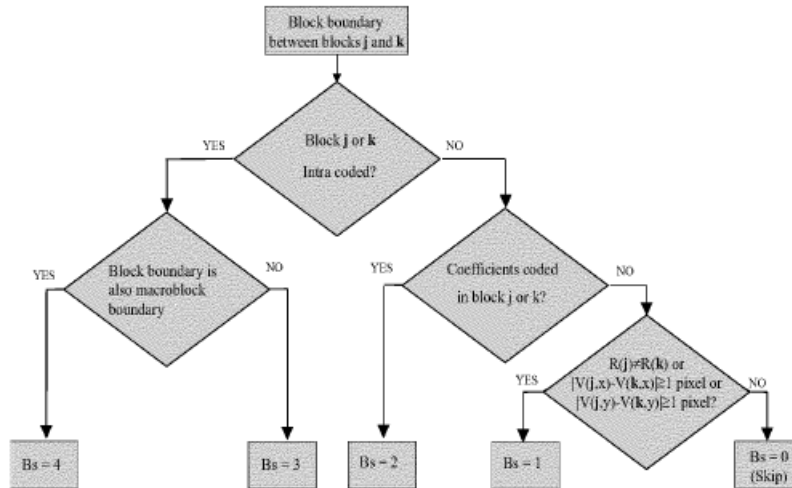


FIGURE 14

Flow chart for determining boundary strength (Bs), for the block boundary between two neighbouring blocks j and k.

Ex-1023 at 47.

199. B118 also notes that “block boundaries of chroma blocks always correspond to a block boundary of luma blocks” such that the Bs calculation can be used for both luma and chroma block boundaries. *Id.* B118 also discloses a slightly different naming convention than the ’841 Patent, with the pixels closest to the block boundary denoted p1 and q1, rather than p0 and q0.

The set of eight pixels across a 4x4 block horizontal or vertical boundary is denoted as

$p_4, p_3, p_2, p_1 \mid q_1, q_2, q_3, q_4$

Id.

200. B118 also notes that in default mode, only two pixels are updated on each side of the block boundary, with the decision whether to filter pixels being based on the boundary strength calculation.

with the actual boundary between p1 and q1. In the default mode up to two pixels can be updated as a result of the filtering process on both sides of the boundary (that is at most p2, p1, q1, q2). Filtering across a certain 4x4 block boundary is skipped all together if the corresponding Bs is equal to zero. Sets of pixels across this edge are only filtered if the condition

$$Bs \neq 0 \text{ AND } |p_1 - q_1| < \alpha \text{ AND } |p_2 - p_1| < \beta \text{ AND } |q_2 - q_1| < \beta$$

is true. The QP dependant thresholds α and β can be found in table 5.

Id.

201. Based on these boundary strength calculations, one of two types of filtering is performed – filtering for the default case, or stronger filtering for intra coded blocks. *Id.* at 48. In the default case, a first offset is calculated using the equation shown below. *Id.*

5.4.2 The Filtering Process

Two sorts of filter are defined. In the default case the following filter will be used for p₁ and q₁

$$\Delta = \text{Clip}(-C, C, ((q_1 - p_1) << 2 + (p_2 - q_2) + 4) >> 3)$$

202. Once the Δ is calculated, it is clipped within a range of -C to C, where, “Clip() denotes a clipping function with the parameters Clip(Min, Max, Value)” and “C is set to C0 and then incremented by one if p₂ will be filtered and again by one if q₂ will be filtered. *Id.* C0 is defined using the equation C0 = ClipTable[QP, Bs] (see TABLE 5) with Table 5 shown below.

TABLE 5

QP and Bs dependent Clipping thresholds

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	
ClipTable[QP,1]	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	4	4	
ClipTable[QP,2]	0	0	0	0	0	0	0	0	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	6	6	
ClipTable[QP,3]	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	6	6	6	8	8	9	10
ClipTable[QP,4]	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	6	6	6	8	8	9	10

Id. at 48.

203. Once the Δ is clipped within a range of $-C$ to C , it is either added or subtracted to the pixel values p_1 and q_1 , to create modified pixel values P_1 and Q_1 . This modified value is similarly clipped within a range of 0 and 255.

$$\begin{aligned} P_1 &= \text{Clip}(0, 255, (p_1 + \Delta)) \\ Q_1 &= \text{Clip}(0, 255, (q_1 - \Delta)) \end{aligned}$$

Id. at 48.

204. The B118 deblocking filter will then determine two intermediate values a_p and a_q , that are used to determine whether to filter p_2 and q_2 . *Id.* If it is determined to filter p_2 and q_2 , they are filtered using the equations below.

The two intermediate threshold variables

$$a_p = |p_3 - p_1|$$

$$a_q = |q_3 - q_1|$$

are used to decide whether p_2 and q_2 are filtered. These pixels are only processed for luminance.

If (Luma && $a_p < \beta$) p_2 is filtered by:

$$P_2 = p_2 + \text{Clip}(-C_0, C_0, (p_3 + P_1 - p_2 < 1) > 1)$$

If (Luma && $a_q < \beta$) q_2 is filtered by:

$$Q_2 = q_2 + \text{Clip}(-C_0, C_0, (q_3 + Q_1 - q_2 < 1) > 1)$$

Id.

205. In some cases, the deblocking filter will instead perform stronger filtering, rather than the default filtering. The stronger filtering determinations and equations are shown below.

5.4.3 Stronger filtering for intra coded blocks

In case of intra coding and areas in the picture with little texture the subjective quality can be improved by stronger filtering across block boundaries (that is: *not* 4x4 block boundaries). This stronger filtering is performed if:

$$(Bs == 4) \text{ AND } (a_p < \beta) \text{ AND } (a_q < \beta) \text{ AND } (1 < |p_1 - q_1| < QP / 4)$$

In this case filtering is performed with the equations shown below.

$$P_1 = (p_3 + 2*p_2 + 2*p_1 + 2*q_1 + q_2 + 4) / 8$$

$$P_2 = (p_4 + 2*p_3 + 2*p_2 + 2*p_1 + q_1 + 4) / 8$$

$$Q_1 = (p_2 + 2*p_1 + 2*q_1 + 2*q_2 + q_3 + 4) / 8$$

$$Q_2 = (p_1 + 2*q_1 + 2*q_2 + 2*q_3 + q_4 + 4) / 8$$

Only for luma p3 and q3 are filtered:

$$P_3 = (2*p_4 + 3*p_3 + 2*p_2 + p_1 + 4) / 8$$

$$Q_3 = (2*q_4 + 3*q_3 + 2*q_2 + q_1 + 4) / 8$$

Id.

I. VCEG-M21 (Ex-1069)

206. Though not an asserted reference, Petitioner and Dr. Schonfeld rely on VCEG-M21 for its disclosure of an offset that is part of the prior art. VCEG-M21 is a Technical Proposal to Video Coding Expert Group Study Group 16 of the Core Experiment Results on Low-Complexity Loop Filtering dated March 27, 2001. As with other references, VCEG-M21, based on its dated and content was a proposal to the development of H.264. Ex-1069. Petitioner relies on VCEG-M21 for an example of a first offset given by $(8(q_0 - p_0) - 2.5(q_1 - p_1))/16$. See Petition at 115. In my opinion, VCEG-M21 undermines Petitioner and Dr. Schonfeld's assertion that the claimed first offset would have been obvious because there were design constraints on coefficients requiring integer values. See petition at 28; Ex-1003, ¶280.

J. Bjontegaard-048 (Ex-1025)

207. U.S. Patent Application Publication US2006/0078048 (Ex-1025)

(“Bjontegaard ’048”) was filed October 13, 2005. While Petitioner does not assert Bjontegaard ’048 as a reference against the ’841 Patent, Petitioner relies on Bjontegaard ’048 for an example of a first offset given by that can be written as $(8(q_0 - p_0) + 4(q_1 - p_1) - 4(q_2 - p_2))/16$ See Ex-1003, ¶272. Bjontegaard ’048 describes that it discloses an “improvement over prior art method for removal of blocking and quantization noise . . . described in the [H.264] specification.” Ex-1025 at ¶[0022]. Thus, Bjontegaard ’048 proposes an improvement over the published H.264 Standard. In addition, as I described in further detail herein, Bjontegaard ’048 was a cited reference in the prosecution history of the ’841 Patent. In my opinion, Bjontegaard ’048 undermines Petitioner and Dr. Schonfeld’s assertion that the claimed first offset would have been obvious because Bjontegaard ’048 discloses a fundamentally different offset design that uses six pixels instead of four pixels, which teach away from the claimed first offset.

XI. SIMULATION OF DEBLOCKING FILTER DESIGNS

A. Testing Methodology

208. As I explain above, in his opinions with respect to the ’841 Patent, Dr. Schonfeld relies on Bjontegaard and Aksu as primary references. Dr. Schonfeld alleges that the Bjontegaard, “disclosed calculating a first offset for filtering the

pixels at the edge,” that Dr. Schonfeld asserts maps to the claimed first offset in the ’841 Patent. Ex-1003 ¶ 137.

209. To illustrate, analyze, and compare certain aspects of Bjontegaard/Bjontegaard’s Code and the deblocking filter design claimed in the ’841 Patent, I created a test simulation based on pseudo-code and description provided in the specification and claims of the ’841 Patent and a test simulation that is based on the deblocking filtering design disclosed in Q15-J-28 and/or on portions of the so-called Bjontegaard Code.

210. I developed these simulations to illustrate and analyse the different filtering results when applying different deblocking filter designs to the same blocks of pixel values, and to support my opinion that the difference between the deblocking design disclosed in the ’841 Patent and the references that Dr. Schonfeld relies on for his opinions, including Bjontegaard, Bjontegaard Code, H.264/List, Fu, B118, VCEG-M21, H.241, Aksu, and B205 cannot be oversimplified to a mere difference in the value of filter coefficients. The difference in filtering results based on my test simulation, which I summarise below, illustrates the complex differences in design that results in significantly different filtering decisions, different filter frequency response, and different filtering results of the ’841 Patent compared with (for example) Bjontegaard, that in my opinion would not have been obvious to a POSITA as a part of routine experimentation.

211. I created simulation scripts for execution in the Matlab development environment. I created four Matlab scripts: FiltScript2.m and TestFilt2.m. TestFilt2.m implements relevant aspects of the deblocking filter the process disclosed in Bjontegaard and/or Bjontegaard Code (e.g., Q15-J-28, Ex-1009) and/or Bjontegaard's Code (e.g., the software in Exs-1010-1015) for Ground 2. TestFilt2.m implements deblocking filter processes disclosed in the '841 Patent. Ex-2004-2014 attached to the Patent Owner's Preliminary Response includes FiltScript2.m and TestFilt2.m and corresponding results files described below.

212. I implemented portions of "Bjontegaard," "Bjontegaard's Code" and the claims of the '841 Patent as Matlab script files because Matlab provides a convenient, 'rapid prototyping' environment that allows for the testing and comparison of processes and algorithms. Matlab is familiar to a POSITA and is a commonly used tool for analysing and comparing algorithms in the image and video processing and coding field. Matlab scripts are executed in a managed environment (the Matlab application environment) and the supplied Matlab scripts will run in the same way on any compatible platform (e.g. Windows, MacOS), which in my experience is not necessarily the case for a C-language implementation.

213. I chose not to compile and execute the lcodec C-language versions referenced in the Petition (in particular, the C language source code files included in Ex-1010-1015 (part of the Bjontegaard Code)). The source code files supplied on

the VCEG site are not compatible with modern-day software compilers and platforms, as they use a number of out-of-date and deprecated C-language software constructs such as `min()`, `max()` and certain time measurement methods that I found not to be supported by present-day compilers. For example, MacOS's Xcode compiler will not compile the Bjontegaard Code on a present-day Apple computer. I determined that identifying and setting up a legacy environment such as a version of MS Windows from around the time of the Bjontegaard Code, would be time consuming and would potentially restrict accessibility, for example making it difficult for other parties in this matter to repeat my analysis. Further, software corresponding exactly to Bjontegaard (Q15-J-28) is not available on the VCEG site (see my analysis herein, identifying discrepancies between the disclosure of Q15-J-28 and the lcodec software versions). I therefore determined it would be more efficient and effective to implement certain portions of Q15-J-28, lcodec4.0 and the '841 Patent's claimed methods in Matlab, for the purposes of illustration, analysis and comparison. Further, as Matlab is a readily available tool which would be familiar to a POSITA, I consider such an implementation to be accessible for further experimentation and evaluation in this matter.

214. I implemented a simple simulation of C language operation (TestUint8.c) in response to certain assertions in Lenovo's contentions on Invalidity in the ITC proceeding and Bjontegaard', which I discuss below.

B. Matlab Scripts

215. I created Matlab script files, namely FiltScript2.m and TestFilt2.m.

216. The test script FiltScript2 calls TestFilt2 1000 times, each time collecting statistics regarding the outcome of a filtered pair of blocks, with randomly-generated variation between successive pairs of blocks. FiltScript2 generates an output file which records statistics for each pair of blocks, including the number of filtered pixels, the difference values, and the offset values with and without clipping, and records statistics relating to the set of 1000 blocks. FiltScript2 generates a further output file which contains a printout of each pair of blocks, including the input blocks, the output blocks processed based on Q15-J-28 and the result of the deblocking filter functionality claimed in the '841 Patent.

217. These script files generate a number of pairs of input blocks and process them according to aspects of the filtering designs disclosed in Bjontegaard's Code and in the claims of the '841 Patent. Two output files are produced, FiltStats[suffix].txt and FiltBlocks[suffix].txt, where [suffix] may be Thr1Thr2 or AltThr, as explained in the tables below. FiltStats[suffix].txt contains printouts of parameters (a) relating to each pair of input blocks and (b) relating to the complete set of input blocks. FiltBlocks[suffix].txt contains printouts of (a) each generated pair of input blocks, (b) a first filtered version of each pair of blocks, based on the disclosure of Q15-J-28, (c) a second filtered version of each pair of blocks, based on

the disclosure of the '841 Patent, without the use of certain clipping operations, and (d) a third filtered version of each pair of blocks, based on the disclosure of '841 Patent and including certain clipping operations, as explained in the tables below.

218. The variable “pixin()” contains the value of blocks of pixel values used as input to the filtering processes. Each pair of blocks contained in ‘pixin()’ comprises a set of pixel values that in my view are representative of image blocks that may be filtered in a typical video coding scenario. The average pixel value, a gradient across the block boundary and the individual pixel values themselves are all generated with pseudorandom variation. The notation of the indices of the array of input pixels determined by pixin() corresponds to the notation of the '841 Patent, Q15-J-28 (“Bjontegaard”), and lcodec4.0 (included in “Bjontegaard’s Code”) as follows, wherein the red line indicates the block boundary between neighboring blocks:

pixin() array	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
'841 Patent	p3	p2	p1	p0	q0	q1	q2	q3
Q15-J-28	a	b	c	d	e	f	g	h
lcodec4.0 li[]	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

219. Below, I explain the implementation of the functionality in my test simulation scripts on a line-by-line basis to show how the scripts implement the

relevant aspects of Q15-J-28 and the '841 Patent using simulated sample blocks of pixels, in order to compare the filtering results obtained using the selected aspects of Q15-J-28 (the "Q15-J-28 method") versus the selected aspects of the '841 Patent (the "'841 method").

FiltScript2.m line numbers	Discussion
6	Set 'randomseedOn' to 1 in order to generate a repeatable sequence of random numbers, i.e. to generate an identical series of pseudorandom block data each time FiltScript is executed. Set to 0 in order to generate a different series of block data each time FiltScript is executed.
7	Set 'numruns', the number of pairs of blocks to generate (e.g. 1000 pairs)
8	Set UseAltThreshold to 0 in order to use the thresholds Thr1 and Thr2 specified in the '841 patent. Set to 1 in order to use an alternative threshold, discussed herein.
9	Set Uselcodec4Diff to 0 in order to use the Diff equations disclosed in Q15-J-28 sections 4.5.6.1 and 4.5.6.2. Set to 1 in order to use the Diff calculation alleged to be disclosed in lcodec4.0 image.c at lines 491 and 499.
11-45	Create and open files 'FiltStats[suffix].txt' and 'FiltBlocks[suffix].txt', where [suffix] changes depending on UseAltThreshold and on Uselcodec4Diff .
47-64	Initialise and set to zero certain arrays used in the calculations to follow.
66	Main loop will repeat 'numruns' times.
67-69	Seed Matlab's random number generator such that the same random number sequences are generated for the same value of run.
70-92	Generate a random quantization parameter QP in the range 4-20. Run 'TestFilt.m' to generate and process a pair of blocks. Record certain output values.

94-116	Calculate certain statistical values from the data recorded during the main loop processes. Print these values in file FiltStats[suffix].txt.
118-220	Calculate further statistical values from the data recorded during the main loop process, including the frequency of occurrence of Act11, Act22 and ActEdge values and the frequency of occurrence of filtered difference values (i.e. the difference between the original and filtered pixel values) for pixels c, d, e and f (Q15-J-28) and pixels p1, p0, q0 and q1 (841/841). Print the results of the calculations in FileStats[suffix].txt.

TestFilt2.m line numbers	Discussion
3-7	Note (Matlab comment) on array notations in Matlab, C, the '841 patent and Q15-J-28.
9-28	Set QP, UseAltThreshold, UseIcodec4Diff , FIDblocks and FIDstats to default values if not already defined (e.g. if TestFilt.m is run without running FiltScript.m)
31-50	Generate and print a pair of blocks of sample values. These are intended to be representative of typical image sample blocks, with a randomly generated starting value, a randomly generated gradient up or down (moving across the blocks from left to right) and random variations between pixel values. The generated blocks are contained in array pixin(8,8) and represent the right-hand 4 columns of a first 8x8 block and the left-hand 4 columns of a second 8x8 block.
54-67	Set up parameters in preparation for simulating the "Q15-J-28 method", including Strength1 and Strength2 parameters (fixed at 2 for the purposes of the simulation) and the array Lim(qp) (Q15-J-28 at 4.5.2)
71-114	Repeat the following for the 8 lines of pixels in pixin. For each line of pixels, calculate parameters according to Q15-J-28 4.5.2, 4.5.3 and 4.5.4. Note: (a) This functionality assumes Dr Schonfeld's assignment of $S0_1$ (Ex-1003 ¶ 238, 644), namely $12abs(b-2c+d)$

	rather than the equation for $S0_1$ disclosed in J28, i.e. $12abs(a-2b+c)$ (b) This functionality also assumes that Act2 is generated based on $S0_2$ and $S1_2$.
118-137	Calculate dmodified and emodified (variables dresult and eresult) based on Q15-J-28 4.5.5, using the function Clip(qp,strength) defined in 4.5.4, implemented as a function ClipQP within TestFilt.m. Calculate dresult0 and eresult0, which comprise d+Diff or e+Diff respectively, without clipping, as disclosed in lcodec4.0 image.c at lines 485 and 486 respectively.
140-158	If Uselcodec4Diff is 0, calculate cmodified based on Q15-J-28 4.5.6.1. Note that the calculation uses original samples d and e, rather than dmodified and emodified. If Uselcodec4Diff is 1, calculate cmodified based on the disclosure of lcodec4.0 image.c at lines 491-493.
161-179	If Uselcodec4Diff is 0, calculate fmodified based on Q15-J-28 4.5.6.2. Note that the calculation uses original samples d and e, rather than dmodified and emodified. If Uselcodec4Diff is 1, calculate fmodified based on the disclosure of lcodec4.0 image.c at lines 499-501.
142-162	Print out each line of filtered pixels corresponding to a, b, cmodified, dmodified, emodified, fmodified, g and h.
181-209	Print out parameters and calculated statistics for the block of pixels pixin, processed by the filtering process of Q15-J-28. Note: The parameters printed at line 202 are the parameters for all the lines of pixels in the pair of blocks.
215-275	Set up parameters in preparation for simulating aspects of the claims of the '841 Patent, the "'841 method". Set up (a) the Beta table (Table 1, '841 Column 16) and (b) the Tc and Tc2 arrays ('841 Column 11 and 14:20). Set up a modified QP, QPmod, by adding 12 to the QP calculated prior to simulating the method of Q15-J-28. This is done in order to ensure approximately equal quantization step sizes when simulating each method. See QP discussion below for further details.
277-290	For each of the 8 lines of input samples, set up p3...p0 and q0...q3, dpi, dqi and First, Second and Third Offsets as per

	claims 5 and 10 of the '841 patent and claim 4 of the '841 patent.
292-296	Filter pixels p0 and q0 as per '841 patent claim 4 and '841 patent claim 10.
298-307	Determine whether or not to filter p1 and q1, either (a) by comparing decision values for p1 and q1 to thresholds Thr1 and Thr2 respectively, if UseAltThreshold is 0, or (b) by comparing S01 and S02 respectively to Lim(qp) as per Q15-J-28, using the 'original' value of QP (i.e. not using QPmod).
309-316	Filter pixel p1 if the above determination is met.
317-324	Filter pixel q1 if the above determination is met.
325-335	Calculate the clipping functions of '841 claims 2 and 6. Save the (possibly) filtered and unclipped and the (possibly) filtered and clipped output pixels.
337-372	Print out (possibly) filtered block values and statistics for the simulated '841 method.
377-401	A function ClipQP implementing the Clip(qp, strength) function specified in Q15-J-28 at page 20.
404-408	A function Clip8 utilised by ClipQP.

220. **QP and thresholds discussion.** For consistency and accuracy in the test simulation discussed above, I calculated QP in order to ensure consistency between the Quantization Step Size in Q15-J-28 and the methods of the '841 patents as set forth in detail below.

221. In a video coding method such as Q15-J-28 (Ex-1009) or the methods of the H.265 draft standard JCTVC-B205 (Ex-1018), a Quantization Parameter QP is employed in the encoding or decoding of each block of pixel data. QP ranges between a minimum and maximum value, QPmin to QPmax, with each value of QP corresponding to a Quantization Step Size. During video encoding, transformed

coefficients of a block of pixel are quantized, such that their values are limited to a set of discrete values separated by the Quantization Step Size. This has the effect of reducing the precision of the data, resulting in both compression and a loss of visual quality when the video is decoded. In general, a larger value of QP maps to a larger Quantization Step Size, producing more compression but also a more distorted decoded video image. The amount of distortion in the decoded image, particularly at the boundaries between blocks, depends on the Quantization Step Size and therefore on the value of QP.

222. Q15-J-28 and B205 each specify a different mapping between QP and Quantization Step Size. A portion of this mapping is provided in the following Table:

Q15-J-28 QP	Q15-J-28 Quantization Step Size	B205 QP	B205 Quantization Step Size
0	2.5	12	2.5
1	2.81	13	2.75
2	3.15	14	3.25
3	3.53	15	3.5
4	3.97	16	4
5	4.46	17	4.5
6	5.0	18	5.0

...		...	
12	10.0	24	10.0
...		...	
18	20.0	30	20.0
...		...	

223. The Q15-J-28 QP ranges from 0 to 31 and as the Table illustrates, an increase in QP of 6 results in a doubling of the Quantization Step Size. A Quantization Step Size of 10, for example, means that transform coefficients (transformed pixel values), when decoded, are effectively rounded to the nearest multiple of 10.

224. The B205 QP ranges from 0 to 51. In B205, QP values less than 12 result in Quantization Step Sizes less than 2.5. As the Table illustrates, B205 QP values (which I denote QP_{mod}) from 12-43 result in approximately the same Quantization Step Sizes as Q15-J-28 QP values (which I denote QP) from 0-31. I verified this relationship as follows.

225. First, I derived the array B(QP) based on Q15-J-28 at pages 16-18 using the following steps:

- (1) Barray = [3881, 4351, 4890, 5481, 6154, 6914, 7761, 8718, 9781, 10987, 12339, 13828, 15523, 17435, 19561, 21873, 24552, 27656, 30847, 34870, 38807, 43747, 49103, 54683, 61694, 68745, 77615, 89113, 100253, 109366, 126635, 141533];
- (2) Derive the relationship between B(QP) and quantizer step size:
 - (a) Dequantized level $k = \text{level} * B(QP)$ (Q15-J-28 at 18)
 - (b) Inverse transform scales values by 676 (Q15-J-28 at 16)
 - (c) The pixel values are downscaled by 2^{20} after the inverse transform (Q15-J-28 at 18)
- (3) Thus, Quantizer Step Size = $(\text{Barray}(QP) * 676) / 2^{20}$ and the result is that for $QP=0, 1, 2, 3, 4, 5, 6, \dots$ the Quantizer Step Size = 2.5, 2.81, 3.15, 3.53, 3.97, 4.46, 5..., with the Step Size doubling for every $QP+6$.

226. Next, I derived the QP_{mod} that provided an equivalent Quantizer Step Size for the methods of the '841 Patent based on the draft H.265 Standard B205:

- (1) Quantizer Parameter ranges from 0-51
- (2) The basic principles of H.264 apply
- (3) In H.264, the mapping of the QP range from 0-51 to QStep means that for QP=12, 13, 14, 15, 16, 17, 18, ..., the approximate Quantizer Step Size = 2.5, 2.75, 3.25, 3.5, 4, 4.5, 5...., doubling every $QP_{mod} + 6$.
(Richardson at 191)

227. Therefore JCTVC-B205 QP_{mod} in the range 12-43 produces approximately the same Quantizer Step Size as Q15-J-28 QP in the range 0-31. Accordingly, in TestFilt.m, I calculate a modified QP, QP_{mod} , in order to simulate the methods of the '841 Patent, by adding 12 to the QP applied when simulating the method of Q15-J-28. Each of the simulated methods depends on QP because QP affects the expected magnitude of blocking artefacts.

228. I compare the behaviour of the Q15-J-28 filtering method and the '841 Patent filtering method wherein the Q15-J-28 filter output is calculated as disclosed in Q15-J-28, using the Q15-J-28 QP (denoted simply QP) and applying the disclosed steps including comparing $S0_1$ to $Lim(QP)$, and Diff initially being set to 0 under the assumptions listed above. The filtering decision for the '841 Patent method is calculated by calculating thresholds Thr1 and Thr2 based on Beta, '841 patent 16:59-60, where Beta depends on the Quantization Parameter which in this case is the B205 QP, i.e. QP_{mod} as discussed above.

C. Analysis of Test Simulation Results

229. Here I consider selected results from the simulation scripts. These results are obtained from FiltScript2.m and TestFilt2.m.

230. Consider a sample input, the very first pair of blocks generated and saved in FiltBlocksThr1Thr2_2a.txt:

Block 1:

Input blocks:

110 102 94 86 | 79 72 66 60

111 103 99 90 | 85 79 73 65

109 105 102 95 | 90 86 82 74

106 98 94 86 | 80 77 71 66

105 100 96 87 | 83 80 75 68

107 99 93 89 | 82 75 67 58

108 100 93 87 | 78 72 64 59

109 101 94 89 | 82 73 67 62

231. The difference between Block 1 and the version filtered using the method of Q15-J-28 is as follows:

0	0	0	0	0	1	0	0
0	0	0	1	-1	0	0	0
0	0	0	1	-1	0	0	0

0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	0
0	0	1	0	0	0	0	0
0	0	0	-1	1	-1	0	0
0	0	1	0	0	0	0	0

- i.e. the simulated Q15-J-28 filtering process has adjusted some of the pixels c, d, e and f, in this case by +/-1.

232. The difference between Block 1 and the version filtered using the simulated method of the Patents (without the Clip functionality of the '841 patent claims 2 and 6) is as follows, showing that this process has also adjusted some of the pixels p1, p0, q0 and q1 by +/-1:

0	0	0	0	0	0	0	0
0	0	0	1	-1	-1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	0
0	0	0	-1	1	0	0	0
0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	0

233. In the following copy of the input blocks, I have highlighted the pixels for which the **same** adjustment was made in **blue** and the pixels for which a **different** adjustment was made in **red** (excluding the pixels for which no adjustment was made in either method):

110 102 94 86 | 79 **72** 66 60

111 103 99 **90** | **85** **79** 73 65

109 105 102 **95** | **90** 86 82 74
106 98 94 86 | 80 77 71 66
105 100 96 **87** | **83** 80 75 68
107 99 **93** **89** | **82** 75 67 58
108 100 93 **87** | **78** **72** 64 59
109 101 **94** 89 | 82 73 67 62

234. In this example block, seven pixels are filtered in the **same** way by the two methods, for example the d and e pixels in line 2 are filtered in the same way by adding or subtracting 1, in both methods. Eight pixels are filtered **differently** by the two methods, for example d and e in line 3 are filtered by the Q15-J-28 method and are not filtered by the Patent method.

235. Next, I consider the behaviour of each filtering process for the complete dataset of 1000 pairs of blocks. In the following discussion, I analyse the simulated Q15-J-28 method and the simulated '841 method, without the clipping functionality of '841 claims 2 and 6.

- *Using Thr1/Thr2 from the '841 patent and using the Q15-J-28 disclosed method for calculating Diff values:*

236. Act11 is equal to 2 for **2196** of the 8000 rows of pixels. Thus for the test data set, using the Q15-J-28 method, pixels c are filtered using the equation $\text{Diff} = 5(b-2c+d)/16$ around 30% of the time.

237. Act22 is equal to 2 for **2152** of the 8000 rows of pixels. Thus for the test data set, using the Q15-J-28 method, pixels f are filtered using the equation $\text{Diff} = 5(e-2f+g)/16$ around 30% of the time.

238. We can compare how often the '841 Patent's method and the Q15-J-28 method result in the **same** or **different** filtering choices. Consider for example the filtering of pixel p1 ('841 method) and c (Q15-J-28 method). For the test data and parameters simulated, p1 and c are each adjusted by either -2, -1, 0, 1 or 2, i.e. the difference between p1 and c before and after filtering can be -2, -1, 0, 1 or 2. The Matlab script counts the number of times each method adjusts p1 or c by the same amount (e.g. Q15-J-28 adjusts c by -1 and the '841 method also adjusts p1 by -1), or by a different amount (e.g. Q15-J-28 adjusts c by -1 and the '841 method adjusts p1 by a different amount such as -2, 0, 1, or 2):

Number of occurrences of each pair of differences,
in the range (-2, +2):

p1diff:	-2	-1	0	1	2
cdiff -2:	1	7	8	0	0
cdiff -1:	2	387	665	14	0
cdiff 0:	0	559	4687	553	0
cdiff 1:	0	16	705	378	7
cdiff 2:	0	0	4	6	1

239. For example, there are 387 rows for which the Q15-J-28 method adjusts c by -1 (i.e. filters c by subtracting 1) and the '841 method also adjusts p1 by -1. However, there are (2+665+14+0)=681 rows for which the Q15-J-28 method adjusts c by -1 and the '841 method adjusts p1 by a **different** value. Excluding the case where neither method makes any adjustment at all (i.e. the 0,0 case), we can colour all the entries for which the two methods make the same decision in **blue** and all the entries for which the methods make different decisions in **red**:

Number of occurrences of each pair of differences, in the range (-2, +2):

p1diff:	-2	-1	0	1	2
cdiff -2:	1	7	8	0	0
cdiff -1:	2	387	665	14	0
cdiff 0:	0	559	4687	553	0
cdiff 1:	0	16	705	378	7
cdiff 2:	0	0	4	6	1

240. Adding up the blue and the red entries, the simulation script calculates the following:

Excluding (0,0), p1 and c were filtered the same way **767** times and different ways **2546** times

241. In other words, for the pixels p1 or c which were modified by one or both methods during filtering, the two methods made **different** decisions for 2546 rows of pixels and the two methods made the **same** decision for 767 rows of pixels. The corresponding calculations for the pairs of pixels p0 and d, q0 and e and q1 and f are as follows:

Excluding (0,0), p0 and d were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q0 and e were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q1 and f were filtered the same way 767 times and different ways 2547 times

- *Using Thr1/Thr2 from the '841 patent and the 'image.c' method for calculating Diff values:*

Excluding (0,0), p1 and c were filtered the same way 924 times and different ways 2241 times
 Excluding (0,0), p0 and d were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q0 and e were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q1 and f were filtered the same way 941 times and different ways 2240 times

- *Using S01/S02 from Q15-J-28 as the 'threshold' comparisons and the Q15-J-28 disclosed method for calculating Diff values:*

Excluding (0,0), p1 and c were filtered the same way 1171 times and different ways 2627 times
 Excluding (0,0), p0 and d were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q0 and e were filtered the same way 3045 times and different ways 2141 times
 Excluding (0,0), q1 and f were filtered the same way 1123 times and different ways 2716 times

- *Using S01/S02 from Q15-J-28 as the 'threshold' comparisons and the 'image.c' method for calculating Diff values:*

Excluding (0,0), p1 and c were filtered the same way 1287 times and different ways 2411 times

Excluding (0,0), p0 and d were filtered the same way
3045 times and different ways 2141 times

Excluding (0,0), q0 and e were filtered the same way
3045 times and different ways 2141 times

Excluding (0,0), q1 and f were filtered the same way
1247 times and different ways 2510 times

242. **Summary.** The results shown above demonstrate the following.

243. For the pixels which were modified by one or both of the simulated Q15-J-28 and '841 Patent's methods during filtering:

- Pixels p0 ('841 method) and d (Q15-J-28 method) were filtered the same way around 60% of the time and differently around 40% of the time.
- Pixels q0 ('841 method) and e (Q15-J-28 method) were filtered the same way around 60% of the time and differently around 40% of the time.
- Pixels p1 and c, and pixels q1 and f, were filtered differently most of the time, i.e. the two methods made different filtering decisions for most of the rows of pixels.

244. These results are also conservative based on the range of pixel values I used to generate the simulated blocks. I used conservative values to attempt to accurately simulate a real-world example while also avoiding critique of my results by over-inflated inputs. If the range of pixel values was increased by increasing the scale, then the variance explained in my summary of results would increase. A higher dynamic range is representative of real-world examples, such as video displayed on new displays.

**XII. PETITIONER'S GROUNDS FOR CANCELLATION DO NOT
RENDER THE CHALLENGED CLAIMS INVALID BECAUSE THEY
ARE NOT OBVIOUS**

**D. Petitioner's Ground 1A Fails Because The Challenged
Claims Are Not Obvious In View of Bjontegaard And
Bjontegaard's Code.**

245. In my opinion, Bjontegaard alone or in combination with Bjontegaard's Code, does not teach or suggest the Asserted Claims of the '841 Patent.

246. First, I understand that Dr. Schonfeld asserts that a POSITA would have read Bjontegaard in conjunction with Bjontegaard's Code and/or found them obvious to combine. I disagree because Bjontegaard's Code does not teach or suggest a single coherent, or even compatible algorithm or process. For example, Dr. Schonfeld states that a POSITA would have considered the May 2000 Telenor Proposal Q15-J-28 (referred to as "Bjontegaard") and the Software Including H.26L Coding Elements from Telenor, Q15-J-29 (referred to as "Bjontegaard's Code") to collectively describe a single system. I disagree because there are aspects of Bjontegaard's Code and Bjontegaard that are inconsistent with each other. For example, the equation for $S0_1$ in Section 4.5.2 of Bjontegaard and the equation for $il3$ (Bjontegaard's Code, image.c at line 446) are inconsistent.

247. I incorporate herein my discussion of Bjontegaard and Bjontegaard's Code in Section X.A-B above, including my description of the Q15-J-28 and Q15-J-29 algorithms. As described in there, Act_1 and Act_2 have a value of either 1, 2, or

3 based on the comparison of $S0_1$, $S1_1$ and $S0_2$, $S1_2$, to $\text{Lim}(qp)$, respectively. Importantly, $S1_1$ and $S1_2$ which affect the eventual value of Act_{11} and Act_{22} used to calculate the different values of Diff , which Dr. Schonfeld maps to the claimed first, second, and third offsets, are not calculated by an equation with a Laplacian form. Act_{11} is calculated as the minimum value between the overall activity value $\text{Act}_{\text{overall}}$ and the activity value for block 1 (Act_1) and Act_{22} is calculated as the minimum value as between $\text{Act}_{\text{overall}}$ and Act_2 . Thus, the values of Act_{11} and Act_{22} may be based either on the activity of one side of the block (e.g., using pixels only on the left side of the boundary for block 1 and pixels on the right side of the block boundary for block 2), or they may be based on the calculated overall activity, $\text{Act}_{\text{overall}}$, which, as shown above, depends on equation $S0$, which is calculated using pixels on both sides of the block boundary and without using a Laplacian equation. In all of Dr. Schonfeld's asserted combinations which I address below, Dr. Schonfeld fails to acknowledge this part of the algorithm in Bjontegaard where a three-pixel Laplacian is not always used to make filtering decisions. In fact, as shown above in Section XI, my test simulation demonstrates that in some cases, the value of $S0_1$ (or $S0_2$) are ignored because the value of $\text{Act}_{\text{overall}}$ was lower than the value of Act_1 (or Act_2). For example, my simulation results showed that Act_{11} is equal to 2 for **2196** of the 8000 tested rows of pixels. Thus, for the test data set, using Bjontegaard's method, pixels c were filtered using the Laplacian equation $\text{Diff} = 5(b-2c+d)/16$ only around 30%

of the time. This is a critical difference between Bjontegaard and Ericsson's filtering design claimed in the '841 Patent because in the claimed design of the '841 Patent, if the claimed first filter decision value is not less than the first threshold value, then the second offset is not calculated. '841 Patent, cl. 10. Likewise, if the claimed second filter decision value is not less than the second threshold, then the third offset is not calculated. '841 Patent, cl. 10. In contrast, as described in my description above in Section VIII.G.1, the calculations for Diff that Dr. Schonfeld asserts are the second and third offsets are *always* performed. Moreover, the value of Diff is always applied to pixel *c* to obtain c_{modified} . Even if Diff is equal to 0, the Diff would still be applied to pixel *c*.

248. As I mention previously and describe in more detail below in my element-by-element analysis, this design is very different from the design disclosed and claimed in the '841 Patent. Moreover, my test simulation results described above further demonstrate the disparate filtering results when the design in Bjontegaard and the design in the '841 Patent are implemented for the same blocks of pixels and compared. Section X.A.1.a.

249. As I explained above in Section X.A.1.b, in Bjontegaard's Code, there are several differences from the deblocking design described in Bjontegaard.

250. In addition, as I explained above in Section X.A.1.b, Bjontegaard's Code is written in such a way as to cause overflow and underflow errors when

processing pixel values that are at or near the minimum or maximum values (0 or 255). Thus, in my opinion, a POSITA would recognize these errors merely by inspecting the source code, and would further obtain incorrect results if running the software. Thus, I disagree with Dr. Schonfeld's assertions that a POSITA would allegedly look to Bjontegaard's code to address errors, missing pieces or inconsistencies in the written proposal Q15-J-28 ("Bjontegaard"). In my opinion, a POSITA would look to the description in Bjontegaard to correct the errors in the faulty software design of Bjontegaard's Code.

251. Based on the inconsistencies between Bjontegaard and Bjontegaard's Code, the inherent overflow/underflow errors due to the faulty software design of Bjontegaard's Code and the known practice of the VCEG group to rely on the Test Model document as the primary source of information rather than the accompanying software, in my view a POSITA would rely on the disclosures of Bjontegaard and discount the disclosures of Bjontegaard's Code. Therefore, I do not concur with Dr. Schonfeld's analysis of Bjontegaard, which I address in more detail below, that in my view relies on carefully selected elements of Bjontegaard's Code to adjust and fix alleged typographical errors in the calculation of $S0_1$ and the alleged use of unclipped " d_{modified} " and e_{modified} " in the calculation of Diff for pixels c and f. See, e.g., Ex-1003 at ¶¶ 126, 266. I further note that the later Q15-J-59 document, titled 'Test Model TML-5', created in the course of a subsequent standardization meeting

(August 2000) and incorporating elements of the Bjontegaard Q15-J-28 proposal, specifies the **same** equations for S_{01} , modification of c in block 1 and modification of f in block 2 (Ex-1009 at 71-73) as Q15-J-28 (“Bjontegaard”). Thus, even though the VCEG group had the opportunity to “correct” what Dr Schonfeld describes as errors or omissions in Bjontegaard in this later Test Model document, they did not do so. I note that the Q15-J-59 **does** include ‘bug fixes’ (noted on Ex-1009 at 52) but chooses not to make the changes that Dr Schonfeld proposes.

252. In addition, I disagree that it would have been obvious to assess a later contribution to the H.264 Standard and corresponding software would not be motivated to look back at prior iterations because the general practice in standardization is to incrementally build on the most developed and most recent version of the proposed changes to the codec.

1. Claim 1, 5, 16

a. $1[a]$, $5[a]$, $16[a]$:

253. In my opinion, Bjontegaard in view of Bjontegaard’s Code does not teach or suggest this limitation.

254. Dr. Schonfeld has not shown that Bjontegaard discloses a “first filter decision value.” For example, Petitioner and Dr. Schonfeld assert that Bjontegaard discloses and/or renders obvious this limitation by the calculation of the parameter S_{01} . However, S_{01} is given by $12\text{abs}(a-2b+c)$, which, when interpreted using the

notation of the '841 Patent for pixel location relative to the block boundary, is $12\text{abs}(p_{3i}-2p_{2i}+p_{1i})$. Thus, S_{01} is not calculated using only pixels p_{0i} , p_{1i} , and p_{2i} , and is not based on $|p_{2i}-2p_{1i}+p_{0i}|$, as required by this claim limitation. Dr. Schonfeld attributes this difference between the Bjontegaard's Code and the '841 Patent to a "typographical error." Ex-1003 ¶ 174-78. I disagree with Dr. Schonfeld that a POSITA would find it obvious to use Dr. Schonfeld's corrected equation for S_{01} using pixels closest to the block boundary instead of using a pixel window offset by 1 pixel from the block boundary.

255. First, Dr. Schonfeld reasons that a POSITA would use his corrected equation for S_{01} because the equations for S_{01} and S_{02} were "counterparts" of one another. Ex-1003 ¶ 179. Even were this the case, this does not explain why one of skill in the art would not assume the equation for S_{01} was correct and therefore adjust the equation for S_{02} instead.

256. To this end, Dr. Schonfeld asserts that a POSITA would be motivated to use his correct S_{01} equation because "deblocking filters applied to the pixels closest to the block boundary." Ex-1003 ¶ 178. However, as explained above in the algorithm walk-through for Q15-J-28 and below in further detail, S_{01} is simply a calculated parameter which is combined with several other parameters to determine the Diff values used to filter not just pixel c, but also d and e.

257. In addition, Dr. Schonfeld asserts that a POSITA would have been motivated to use his corrected $S0_1$ equation because “these equations were intended to be centered on filtering the second pixel c.” I disagree with this, because $S0_1$ likewise affects the filtering applied to at least c, d, and e, as described above so there is no “centering” on pixel c. Ex-1003 ¶180.

258. Further, Dr. Schonfeld asserts that a POSITA would have been motivated to use his corrected $S0_1$ equation due to a POSITA being motivated to refer to “the corresponding source code” in Bjontegaard’s Code (Ex-1003 ¶¶181-82), but does not acknowledge the inherent overflow errors I describe in Section X.B above. Thus, I disagree that Dr. Schonfeld has established that it would have been obvious to a POSITA to correct the equation in $S0_1$ to Dr. Schonfeld’s proposed equation on this basis.

259. Even using a corrected equation for $S0_1$, it is still not a filter decision value as recited in the ’841 Patent. In the ’841 Patent, the independent claims require a determination of how many pixels to filter based on comparison of the first filter decision value to a first threshold value. *See, e.g.*, Ex-1001, cl. 1, 5, 16. The ’841 Patent further requires calculating a first offset that is used to modify $p0$, and further calculating a second offset that is used to modify $p1$ if the first filter decision value is below said first threshold. *See* Ex-1001, cl. 10. Thus, the ’841 contemplates (i) a

determination that pixels will be filtered; and (ii) separately requires calculating the offsets that are used to modify those pixels.

260. In contrast, $S0_1$ is one of several parameters calculated to determine value of the offset used to filter the pixel; not the number of pixels to filter. *Id.* ¶325. That is, Bjontegaard fails to disclose or render obvious the determination of how many pixels to filter that is separate from the calculation of the offsets used to modify (e.g., filter) the pixels. While the value of $S0_1$ could impact the calculated value of Diff (e.g., the value used to modify pixels d and c), the Diff value in Bjontegaard will always be calculated despite $S0_1$ being above or below a threshold $\text{Lim}(qp)$. The same number of pixels is filtered, $S0_1$ would only influence the value used to modify those filtered pixels. In addition, Diff being equal to 0 or a non-zero value is not enough to meet the claim limitations requiring the first filter decision value being used to determine how many pixels to filter because this is fundamentally different from the teachings of the '841 Patent. Diff being equal to zero or a non-zero number corresponds only to the second module of the offset calculation taught by the '841 Patent—there is no first module of a determination of how many pixels to filter taught by Bjontegaard. Critically, $\text{Diff} = 0$ is still applied to the pixels c. The c pixel is always filtered, contrary to Petitioner's and Dr. Schonfeld's assertions. A filter offset of 0 is not a determination not to filter the pixel. In contrast, the '841 Patent teaches that if the first filter decision value is greater than or equal to the first

threshold value, then Diff that is applied to modify pixel p_1 would not be calculated. Bjontegaard teaches that the Diff that is applied to pixel c is always calculated. This is a fundamental difference between the two deblocking designs.

261. For example, using a hypothetical where ramp is present (e.g. $p_1 = 10$, $p_2 = 20$, $p_3 = 30$, and $p_4 = 40$), the first filter decision value of the '841 Patent will be 0. Although the first filter decision value is 0, the '841 Patent provides examples where the determination of the number of pixels to filter would be to filter 2 pixels. However, because of the presence of a ramp, each of the offsets of the '841 Patent would also be 0. Thus, the '841 Patent contemplates scenarios where the deblocking filter would decide to filter a number of pixels but the filter offset would be 0. Thus, a filter offset of 0 is not a decision not to filter a pixel.

262. In my view, Petitioner acknowledges this distinction because it asserts that pixel d is always filtered, but Dr. Schonfeld nor the Petitioner analyze this distinction.

263. Moreover, under certain circumstances, S_{0_1} may not even be used to calculate Diff. In other words, S_{0_1} can, but does not always influence Diff—the values used to modify pixels d and c . *Id.*

264. The comparison of S_{0_1} to the threshold given by $\text{Lim}(qp)$ thus may influence the Diff value added to pixels c , d , and e , (i.e., corresponding to p_0 , p_1 , and q_0 in the notation of the '841 Patent) if the resulting Activity value of block 1

given by Act_1 (incremented by 1 if $S0_1 < Lim(qp)$) is smaller than the calculated overall activity given by $Act_{overall}$. For example, depending on the value of Act_{11} , which is based on Act_1 and $Act_{overall}$, the value of Act_{edge} is calculated, which is then used to determine whether $Diff = (3(c-f) - 8(d-e))/16$ (if Act_{edge} equals 1 or 2) or $Diff = (b+c-8(d-e)-f-g)/16$ (if Act_{edge} equals 3) or $Diff = 0$ (if Act_{edge} equals 0). This $Diff$ value is then used to modify pixels d and e (e.g., pixels $p0$ and $q0$, closest to the block boundary on either side). Further, the comparison of $S0_1$ may also influence the $Diff$ value added to pixel c where $Diff = 5(b-2c+d)/16$ (if Act_{11} equals 2) or $Diff = 3(a+b-4c+d+e)/16$ (if Act_{11} equals 3) or $Diff = 0$ (if Act_{11} equals 0 or 1). However, this demonstrates that $S0_1$ may influence the offset value for pixels c , d , and e . Further, in some cases, as shown above in Section XI, the value of $Act_{overall}$ may be less than the value of Act_1 , and therefore $S0_1$ is not used in the determination of $Diff$.

265. Dr. Schonfeld opines that $S0_1$ is a first filter decision value because “ $S0_1$ is used to determine Act , which is used to decide how many pixels to filter.” Ex-1003 ¶ 175, 178. I disagree with Dr. Schonfeld’s characterization, which oversimplifies the algorithm, apparently using hindsight based on the claims of the ’841 Patent. Specifically, the value of Act_1 is incremented by 1 if $S0_1$ is less than $Lim(qp)$, but Act_1 can also be incremented by 1 when $S1_1$ is less than $Lim(qp)$, which could result in Act_{11} being equal to 2 and thereby modifying a second pixel c by

adding Diff. $S1_1$ is calculated as $\text{abs}(4(a-d) + 12(c-b)) + \text{abs}(9(a-b-c+d))$, which is not based on a Laplacian form such as $|p2_i - 2p1_i + p0_i|$ (e.g., $\text{abs}(b-2c+d)$), nor is this calculation limited to using only the three pixels closest to the block boundary (e.g., d, c, b) for calculating the first filter decision value. Further, as discussed above, Act_{11} used in the calculation of Diff to determine C_{modified} based on $\text{Act}_{\text{overall}}$, when $\text{Act}_{\text{overall}}$ is lower than Act_1 . $\text{Act}_{\text{overall}}$, which is based on $S0$, which is calculated as $\text{abs}(4(c-f) + 12(e-d)) + \text{abs}(9(c-d-e+f))$. Thus, Dr. Schonfeld has not shown that $S0_1$ would be a first filter decision value, because there can be cases where Act_1 , based on $S0_1$ is ignored, as I show in my testing. Dr. Schonfeld has not performed any testing to show if or how often $S0_1$ would be used to determine how many pixels to filter.

266. An example from my simulation results helps to illustrate this point. My simulation results illustrate that Dr Schonfeld's alleged filtering decision threshold in Bjontegaard, i.e. the test for whether $S0_1 < \text{Lim}(\text{QP})$ or $S0_2 < \text{Lim}(\text{QP})$, can in some cases have no influence on whether certain pixels are filtered. For example, consider Row 5 in the very first pair of blocks. The calculated Bjontegaard parameters are listed in my simulation output file FiltStatsThr1Thr2.txt:

Row:5 S0:70 S01:60 S11:60 S02:24 S12:36 ActOverall:1 Act1:1 Act2:2 Act11:1
Act22:1 ActEdge:1 ClipEdge:2

Ex-2012-2013.

267. In this row of pixels, $S0_1$ is calculated to have a value of 60 and $S0_2$ has a value of 24. Act_1 is calculated (in part influenced by $S0_1$) to be 1. Act_2 is calculated (in part based on $S0_2$) to be 2. $Act_{overall}$, which does not depend at all on $S0_1$ or $S0_2$, is calculated to have a value of 1. As a result, Act_{11} and Act_{22} are both set to 1, regardless of the values of Act_1 or Act_2 .

268. The filtering choices for pixels c and f are influenced by the value of Act_{11} and Act_{22} respectively (Ex-1009 (Bjontegaard) sections 4.5.6.1 and 4.5.6.2). See Ex-1009 at 20. But for this row, because $Act_{overall}$ is less than or equal to both Act_1 and Act_2 , the alleged threshold comparisons (comparing $S0_1$ with $Lim(QP)$ and comparing $S0_2$ with $Lim(QP)$) have no influence whatsoever on the filtering choices for pixels c and f and therefore have no influence on the alleged decision as to whether to filter c and f. In other words, whether or not $S0_1$ is less than $Lim(QP)$ or $S0_2$ is less than $Lim(QP)$, for this row of pixels, pixels c and f will be filtered the same way. Thus for this and many other rows of pixels in my test data set, the “determination of how many pixels to filter” is not based on $S0_1$ or $S0_2$ at all.

269. Thus, Dr. Schonfeld has not shown that Bjontegaard alone or in combination with Bjontegaard’s Code in combination, teach or suggest this limitation.

b. 1[b], 5[b], 16[b]

270. In my opinion, Dr. Schonfeld has not shown that the Bjontegaard's Code, alone or in combination, teach or suggest this limitation.

271. As I described above, the deblocking design disclosed in Bjontegaard are very different from the deblocking design disclosed in the '841 Patent, including with respect to this limitation. Dr. Schonfeld has not shown that Bjontegaard teaches or suggests a "second filter decision value" as recited in the '841 Patent because, as described above, the independent claims require a determination of how many pixels to filter based on comparison of the second filter decision value to a second threshold value. *See, e.g.*, Ex-1001, cl. 1, 5, 16. The '841 Patent further requires calculating a first offset that is used to modify q_0 , and further calculating a third offset that is used to modify q_1 if the second filter decision value is below said second threshold. *See* Ex-1001, cl. 10. Thus, the '841 contemplates (i) a determination that pixels will be filtered; and (ii) separately requires calculating the offsets that are used to modify those pixels.

272. In contrast, like S_{01} , S_{02} is one of several parameters calculated to determine value of the offset used to filter the pixel; not the number of pixels to filter. *Id.* That is, Bjontegaard fails to disclose or render obvious the determination of how many pixels to filter that is separate from the calculation of the offsets used to modify (e.g., filter) the pixels. While the value of S_{02} could impact the calculated value of Diff (e.g., the values used to modify pixels e and f), the Diff value in

Bjontegaard will always be calculated despite $S0_2$ being above or below a threshold $Lim(qp)$. That is, the same number of pixels is filtered, $S0_2$ would only influence the value used to modify those filtered pixels. In addition, Diff being equal to 0 or a non-zero value is not enough to meet the claim limitations requiring the second filter decision value being used to determine how many pixels to filter because this is fundamentally different from the teachings of the '841 Patent. Diff being equal to zero or a non-zero number corresponds only to the second module of the offset calculation taught by the '841 Patent—there is no first module of a determination of how many pixels to filter taught by Bjontegaard. Critically, $Diff = 0$ is still applied to the pixel f . The f pixel is always filtered, contrary to Petitioner's and Dr. Schonfeld's assertions. A filter offset of 0 is not a determination not to filter the pixel. In contrast, the '841 Patent teaches that if the first filter decision value is greater than or equal to the first threshold value, then the third offset that is applied to modify pixel q_1 would not be calculated. Bjontegaard teaches that the Diff that is applied to pixel f is always calculated. This is a fundamental difference between the deblocking designs. Similar to the prior element and contrary to the Petition, Bjontegaard discloses that the pixel e is always filtered, and thus, in my view, the Petitioner has not shown that $S0_2$ is a filter decision value or that a determination of how many pixels to filter is made based on that value.

273. The Petition does not address this distinction and does not argue obviousness in light of the correct interpretation of Bjontegaard and Bjontegaard's Code

274. Moreover, under certain circumstances, $S0_2$ may not even be used to calculate Diff. In other words, $S0_2$ can, but does not always influence Diff—the values used to modify pixels e and f. *Id.*

275. The comparison of $S0_2$ to the threshold given by $\text{Lim}(qp)$ could influence the Diff value added to pixels e, f, and d, (i.e., $q0$, $q1$, and $p0$ in the '841 Patent) if the resulting Activity value of block 2 given by Act_2 (incremented by 1 if $S0_2 < \text{Lim}(qp)$) is smaller than the calculated overall activity given by $\text{Act}_{\text{overall}}$. For example, depending on the value of Act_{22} , which is based on Act_2 and $\text{Act}_{\text{overall}}$, the value of Act_{edge} is calculated, which is then used to determine whether $\text{Diff} = (3(c-f) - 8(d-e))/16$ (if Act_{edge} equals 1 or 2) or $\text{Diff} = (b+c-8(d-e)-f-g)/16$ (if Act_{edge} equals 3) or $\text{Diff} = 0$ (if Act_{edge} equals 0). This Diff value is then used to modify pixels d and e (e.g., pixels $p0$ and $q0$, closest to the block boundary on either side). See Ex-1009 at 19. Further, the comparison of $S0_2$ could also influence the Diff value added to pixel f where $\text{Diff} = 5(b-2c+d)/16$ (if Act_{22} equals 2) or $\text{Diff} = 3(a+b-4c+d+e)/16$ (if Act_{22} equals 3) or $\text{Diff} = 0$ (if Act_{22} equals 0 or 1). However, this demonstrates that $S0_2$ may influence the offset value for pixels e, f and d. Further, in some cases, as shown above in Section XI, the value of $\text{Act}_{\text{overall}}$ may be less than the value of Act_2 ,

and therefore $S0_2$ using the Laplacian calculation is not used in the determination of Diff.

276. Dr. Schonfeld opines that $S0_2$ is a second filter decision value because “the value is then used to make filter decisions.” determination of how many pixels to filter (1 versus 2) in the line of pixels in the right block is based on Act_{22} , which is based on Act_2 , which is based on comparing $S0_2$ (the second filter decision value) to a second threshold value ($Lim(QP)$).” Ex-1003 ¶ 120-21, 185. I disagree with Dr. Schonfeld’s characterization, which oversimplifies the deblocking design, apparently using hindsight based on the claims of the ’841 Patent.

277. Specifically, the value of Act_2 is incremented by 1 if $S0_2$ is less than $Lim(qp)$, but Act_2 is alternatively incremented by 1 when $S1_2$ is less than $Lim(qp)$, which could result in Act_{22} being to 2 and thereby modifying a second pixel f by adding Diff. $S1_2$ is given by $abs(4(a-d) + 12(c-b)) + abs(9(a-b-c+d))$. Further, as discussed above, Act_{22} used in the calculation of Diff to determine $f_{modified}$ based on $Act_{overall}$, when $Act_{overall}$ is lower than Act_2 . $Act_{overall}$ is based on $S0$, which is calculated as $abs(4(c-f)+12(e-d)) + abs(9(c-d-e+f))$. Thus, Dr. Schonfeld has not shown that $S0_2$ would be a second filter decision value, because there can be cases where Act_2 , based on $S0_2$ is ignored, as I show in my testing, described above. Dr. Schonfeld has not performed any testing to show if or how often $S0_2$ would be used to determine how many pixels to filter.

278. In my opinion, Dr. Schonfeld has not shown that a POSITA would find this limitation obvious in light of Bjontegaard alone or in combination with Bjontegaard's Code.

c. 1[c], 5[c], 5[e], 16[c];;

279. In my opinion, Dr. Schonfeld has not shown that the combination of Bjontegaard and Bjontegaard's Code teach or suggest this limitation.

280. As stated above in my analysis with respect to limitations 1[a], 5[a], and 16[a], $S0_1$ in Bjontegaard is not a "first filter decision value." For the same reasons described above, the comparison of $S0_1$ to the threshold $\text{Lim}(\text{qp})$ does not determine how many pixels to filter in a line of pixels in the block of pixels. I incorporate that analysis herein by reference. Instead, $S0_1$ is part of the algorithm for calculating the offsets (Diff values), which are always added to pixels d and c. Specifically, Dr. Schonfeld's and Petitioner's allegation that the value of $\text{Act}_{\text{overall}}$ satisfies these limitations. In particular, Dr. Schonfeld opines identifies Bjontegaard's disclosure that "if $\text{Act}_{\text{overall}}=0$, no pixels will be filtered," and if $\text{Act}_{\text{overall}}=1, 2$, or 3, at least pixel d in block 1" will be filtered. Ex-1003, ¶ 187. Dr. Schonfeld further opines that there is a separate decision as to whether to also filter pixel c, wherein "pixel c in block 1 is filtered only if Act_{11} equals 2 or 3, which in turn depends on a comparison of $S0_1$ to a threshold value $\text{Lim}(\text{qp})$." Ex-1003, ¶ 188. I disagree that the identified disclosure of Bjontegaard satisfies the claim limitations

requiring determining how many pixels to filter. This is because the value of $Act_{overall}$ does not influence whether or not Diff is calculated and applied to pixel d. Diff is calculated and applied to pixel d (e.g., pixel d is filtered) regardless of whether $Act_{overall}=0, 1, 2,$ or 3 . Diff is merely the offset value applied to the filter and it can be zero or non-zero, but there is no determination in Bjontegaard of whether to apply the offset or not. The same is true for the value of Act_{11} and pixel c. Namely, the value of Diff that is applied—whether it is zero or calculated according one of the defined Diff expressions—is influenced by Act_{11} but there is no determination in Bjontegaard of whether to apply Diff (the offset) to pixel c. The Diff offset is calculated and applied to filter the pixels, even if that filter value is 0.

281. In addition, while the comparison of $S0_1$ to the threshold value $\lim(qp)$ may influence the value of $Act_{overall}$ and Act_{11} , it also may not, as explained above in connection with limitations 1[a], 5[a], and 16[a], which is incorporated herein by reference. Neither Dr. Schonfeld, nor Petitioner, provide any explanation for the fact that equation $S0_1$ is not always used to calculate $Act_{overall}$ or Act_{11} value. Thus, in my opinion, Dr. Schonfeld and Petitioner have not shown that Bjontegaard discloses or renders obvious these limitations.

d. 1[d], 5[d], 5[f], 16[d]

282. In my opinion, Bjontegaard alone or in combination does not teach or suggest these limitations.

283. As stated above in my analysis with respect to limitations 1[b], 5[b], and 16[b], S_{02} in Bjontegaard is not a “second filter decision value.” For the same reasons described above, the comparison of S_{02} to the threshold $\text{Lim}(\text{qp})$ does not determine how many pixels to filter in a line of pixels in the block of pixels. I incorporate that analysis herein by reference. Instead, S_{02} is part of the algorithm for calculating the offsets (Diff values), which are always added to pixels e and f. Specifically, Dr. Schonfeld incorporates his analysis for the preceding limitation based on the calculation of Act_{22} instead of Act_{11} . As explained above, Dr. Schonfeld identifies Bjontegaard’s disclosure that “if $\text{Act}_{\text{overall}}=0$, no pixels will be filtered,” and if $\text{Act}_{\text{overall}}=1, 2$, or 3 , at least pixel [e] in block [2]” will be filtered. Ex-1003, ¶ 187. Dr. Schonfeld further opines that there is a separate decision as to whether to also filter pixel f which follows the same procedure as for pixel c in block 1. Thus, I understand Dr. Schonfeld’s opinions to be that pixel f in block 2 is filtered only if Act_{22} equals 2 or 3, which in turn depends on a comparison of S_{02} to a threshold value $\text{Lim}(\text{qp})$. See Ex-1003, ¶ 188, 194. As I explain above, I disagree that the identified disclosure of Bjontegaard satisfies the claim limitations requiring determining how many pixels to filter. This is because the value of $\text{Act}_{\text{overall}}$ does not influence whether or not Diff is calculated and applied to pixel e. Diff is calculated

and applied to pixel e (e.g., pixel e is filtered) regardless of whether $Act_{overall}=0, 1, 2, \text{ or } 3$. Diff is merely the offset value applied to filter the pixel, and it can be zero or non-zero, but there is no determination in Bjontegaard of whether to apply the offset or not. The same is true for the value of Act_{22} and pixel f. Namely, the value of Diff that is applied—whether it is zero or calculated according one of the defined Diff expressions—is influenced by Act_{22} but there is no determination in Bjontegaard of whether to apply Diff (the offset) to pixel f. The Diff offset is calculated and applied to filter the pixels, even if that filter value is 0.

284. In addition, as I explain above, while the comparison of $S0_2$ to the threshold value $\lim(qp)$ may influence the value of $Act_{overall}$ and Act_{22} , it also may not, as explained above in connection with limitations 1[b], 5[b], and 16[b], which is incorporated herein by reference. Neither Dr. Schonfeld, nor Petitioner, provide any explanation for the fact that equation $S0_2$ is not always used to calculate $Act_{overall}$ or Act_{22} value. Thus, in my opinion, Dr. Schonfeld and Petitioner have not shown that Bjontegaard discloses or renders obvious these limitations.

285. Further, I understand that Dr. Schonfeld provides opinions that a POSITA would have found it obvious to modify the disclosure of Bjontegaard such the $S0_1$ and $S0_2$ are compared to different threshold values. Ex-1003, ¶¶196-202. In my view, the claims do not require the claimed first threshold value and second threshold value to be different values, because the '841 Patent contemplates that the

thresholds could be different or the same. Ex-1001, 16:59-62. Under any interpretation, Bjontegaard fails to disclose or render obvious these limitations.

2. Claim 10 (Ground 1A)

286. Claim 10 depends from Claim 5, therefore, Bjontegaard and Bjontegaard's Code does not teach or suggest Claim 10 for at least the same reasons described above for Claim 5. See supra §XII.D.1.a-d. In addition for the reason stated below, Bjontegaard and Bjontegaard's Code do not render obvious the additional limitations of claim 10.

- a. 10[a] *a first offset calculator configured to calculate a first offset Δ based on $\frac{9 \times (q_{0j} - p_{0j}) - 3 \times (q_{1j} - p_{1j})}{16}$, wherein p_{0j} denotes a pixel value of a pixel closest to, in said first line of pixels, said block boundary, p_{1j} denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary, q_{0j} denotes a pixel value of a pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary and q_{1j} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary;*"

287. In my opinion, Bjontegaard in view of Bjontegaard's Code and a POSITA's knowledge of the art does not teach or suggest element 10[a].

288. It is undisputed that Bjontegaard, even in combination with any other reference does not disclose the first offset recited in this limitation. Instead, Dr. Schonfeld alleges that the change of the coefficient from an 8 in Bjontegaard to the 9 that is claimed in the '841 Patent would have been obvious to a POSITA because Bjontegaard's coefficients (8 and 3) "are very close to those claimed (9 and 3)" and

a POSITA “would have expected them to have very similar deblocking filtering properties.” Ex-1003, ¶234. I disagree with this proposition. In my opinion a POSITA would not have expected the Diff equation in Bjontegaard and the first offset in claim 10 to have similar deblocking filtering properties based on their “similar” coefficients because a POSITA would have understood that adjusting the coefficients would result in significantly different filtering outcomes because of the sensitivity of deblocking filter design. This is supported and illustrated by my test simulation results. For example, my testing shows that when comparing the deblocking design disclosed in Bjontegaard and that of the ’841 Patent as applied to the same set of pixels, the filtering outcomes are different. In particular, the results show that the pixels closest to the block boundary on either side of the block were filtered differently about 40% of the time, and the pixels next closest to the block boundary in either neighboring block were filtered differently most of the time. These results illustrate that Dr. Schonfeld’s proposition is incorrect. Moreover, Dr. Schonfeld’s own simulation of the frequency response of the filter using the asserted first offset of Bjontegaard and that of the ’841 Patent demonstrate the significant differences in the “frequency response”. Significant differences in frequency response to certain structures are illustrated in Dr. Schonfeld’s figure below. Ex-1003, ¶253.

p2	p1	p0	q0	q1	Type	$\frac{9(q0 - p0) - 3(q0 - p0)}{16}$	$\frac{8(q0 - p0) - 3(q0 - p0)}{16}$
200	150	150	150	150	Step (p1)	0	0
200	200	150	150	150	Step (p0)	9	9
200	200	200	150	150	Step (q0)	-19	-16
150	160	170	180	190	Ramp (10)	0	-1
125	150	175	200	225	Ramp (25)	0	-2
200	250	200	250	200	Alternating	38	34
255	50	50	255	255	Line	77	64
255	255	50	50	255	Line	0	0
175	200	150	150	175	Random	5	5
50	125	100	150	125	Random	28	25
75	100	150	75	125	Random	-47	-42

289. Critically, the claimed first offset expression does not filter ramp, which is an important and specific design choice because a ramp structure is more likely to be the shape of the actual image signal as opposed to a blocking artifact, which could take the form of a line or a step. In contrast, Bjontegaard's first offset does filter ramp, and does not smooth other structures the same way as the '841 Patent. This demonstrates the careful design of a filter, including the coefficients that react in certain ways to particular structures, and smooth other structures a specific amount. For example, in only four of Dr. Schonfeld's eleven examples in the figure above

was the filter response the same, which reflects specific design choices that in my view would not have been obvious using routine experimentation.

290. Indeed, as Dr. Schonfeld acknowledges, “the prior art was filled” with other offset equations that used different coefficients than those claimed. Ex-1003, ¶235. In my opinion, the prior art reflects the failure, over the course of a decade of those skilled in the art to arrive at the claimed first offset using the specific coefficients.

Table 1: Condensed Offset Equation	
'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	$\frac{7.5(q_0 - p_0) + 2(q_1 - p_1) - 2.5(q_2 - p_2)}{16}$

291. Petitioner argues that, by normalizing to a common denominator of 16, the prior art offset-filter equations demonstrate a “pattern that differ[] only with

respect to (1) the coefficient applied to each set of pixel differences . . . and (2) the number of pixels used to calculate the offset.” Petition, 26. However, despite using 7 different equations, none of the prior art equations developed through “routine” experimentation use a coefficient value equal to or larger than the 9/16 value used with the ’841 Patent’s first offset equation.

292. Petitioner’s citation to the several design constraints that would have allegedly limited the first-offset filter design also undermine its obviousness argument in light of the prior art. For example, Petitioner alleges that a POSITA would have experimented with six alleged design constraints to optimize Bjontegaard’s offset equation, including (i) the magnitude of the coefficients applied that controls the “width” of the filter, (ii) the sum of the coefficients in the numerator and division by the denominator, (iii) the “window” used to calculate the offset, (iv) the “DC” gain of the offset equaling one; (v) the value of the offset’s right-shift being between three and six; and (vi) the use of integer coefficients instead of decimal coefficients. Under Petitioner’s own analysis of the design constraints that a POSITA would have been aware of and experimented with, no reference, spanning over the period of a decade, disclosed the particular offset claimed in the ’841 Patent, with its specific combination of coefficients, window, DC gain, and right shift. Thus, Petitioner’s argument that these design constraints would have been used by a POSITA to modify the first offset in Bjontegaard to arrive at the claimed amounts

to nothing more than hindsight bias, in my view. For example, the design constraints could yield over 100 possibilities, and Dr. Schofeld fails to explain why a POSITA would pick one constraint over another, or a value within the constrained range over another.

293. In addition, I have seen no analysis with respect to how Petitioner's proposed modification to Bjontegaard would affect the overall filter design. In my opinion, a POSITA would not have expected to be able to modify one aspect of the filter design and create a successful filter design because of the resultant change in the frequency response.

294. Despite Petitioner's attempt to show a "known workable range" of coefficients (Petition, 30), Petitioner fails to explain what that range is. Notably, the alleged "range" does not appear to include the coefficients $9/16$ and $3/16$. For example, a coefficient of $9/16$ is outside the range of any of the coefficients described by Dr. Schonfeld, which only range from $2/16$ to $8/16$. Petitioner ignores that, under its position, because a POSITA may consider other coefficients in an increased range compared to prior methods, the number of possible coefficients that a POSITA would have to evaluate or consider would increase. As the possible number of coefficients increase, the idea that a POSITA would have necessarily arrived at or tested claim 10's proposal actually decreases, since the likelihood of a POSITA 'finding' an optimal set of coefficients becomes smaller. Moreover, the claimed

offsets of $9/16$ and $3/16$ sum to $12/16$ or $3/4$, which is outside the range that Petitioner states is workable range. Petition, 32. Tellingly, no art cited by Petitioner arrived at or even mentioned the coefficients used by the '841 Patent (or mentioned "routine" experimentation to arrive at the coefficients used by the '841 Patent).

295. Indeed, Petitioner argues that "a minor difference in filter coefficients would not have produced new or unexpected results." Petition, 30. However, this is not the case. My simulations with only Petitioner's purported "minor" changes to filter coefficients are actually significant changes, and the filtering results are significantly different.

296. As described above, the invention of the '841 Patent was carefully designed to have specific qualities, including not filtering ramp while smoothing other structures, such as a step, in a particular way. No prior art reference achieved these same qualities. For example, as demonstrated by Dr. Schonfeld's analysis, Bjontegaard's filters ramp and it does not smooth step, line or random in the same way that the '841 Patent first offset does. These differences illustrate that the design of the offset equations cannot be minimized to their component parts such as coefficients and filtering window. This is because the filters are designed with the ultimate goal of achieving specific responses to certain pixel values, as shown below, they are not designed with the ultimate goal of having a certain coefficient value or filtering window.

- b. 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 $\frac{p_{0j}+p_{2j}-2p_{1j}+2\Delta}{4}$, wherein p_{2j} denotes a pixel value of a pixel second next closest to, in said first line of pixels, said block boundary;

297. In my opinion, Bjontegaard does not disclose or suggest element 10[d] alone or in combination.

298. First, I have seen no analysis from Dr. Schonfeld or Petitioner that the second offset (Diff) in Bjontegaard is calculated if the accused first filter decision value, S_{01} is below the threshold $Lim(qp)$. As I explain above, this comparison may influence the value of Diff, but in my opinion, Dr. Schonfeld and Petition have not shown that this claim requirement is satisfied.

299. In addition, I do not agree that a POSITA would have read the accused Diff equation to use the value of $d_{modified}$ instead of d as the equation appears in Bjontegaard. As I explain above, this proposition is contrary to the text of Bjontegaard because the authors of Bjontegaard, having defined $d_{modified}$ in Section 4.5.5, chose to use d in 4.5.6.1 and 4.5.6.2, even though the defined $d_{modified}$ was available to use in the Diff equations.

300. Assuming that Dr. Schonfeld is correct and $d_{modified}$ is used, Dr. Schonfeld shows the difference between the Second Offset of the '841 Patent and that of Bjontegaard:

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

Ex-1003, ¶ 277.

301. Dr. Schonfeld opines that for the same reasons he provided with respect to the coefficients applied to each pixel difference in the first offset equations, a POSITA would have also recognized that the coefficients applied to each of the components in the equations above would have been obvious. Thus, for the same reasons I provide in connection with element 1[a], it is my opinion that the coefficients would not have been obvious. *See* Ex-1003, ¶ 278.

302. While Dr. Schonfeld points to the use of different weights applied in other references such as Fu (which also uses a different “high-pass component” as explained above) and B118, these differences (both in “high-pass components” and coefficients) are critical to the differing frequency responses of the filters and the resulting filtering outcomes. Thus, in my view, these differences cannot be minimized as Dr. Schonfeld does in his analysis such that the equation of the '841 Patent could have been understood by a POSITA by “optimizing” the equation of Bjontegaard using routine experimentation. Instead, each offset expression reflects a very specific design with an intentionally designed frequency response. In particular, the '841 second offset uses a Laplacian that only uses p pixels for the

“high-pass component.” This was a specific design choice of the overall ’841 deblocking design such that the (i) consider the local structure near the block boundary and (ii) offsets do not filter ramp. Moreover, the frequency response is specific to other types of structures, and the coefficients ensure that other structures, such as lines and steps are smoothed in a particular way.

303. The differences between these two offsets is shown when the Δ , the first offset, is considered. The following table reflects the actual deblocking filters normalized based on the value of Δ .

	Second Offset
841 Patent	$\frac{64(p_2 - 2p_1 + p_0) + 72(q_0 - p_0) - 24(q_1 - p_1)}{256}$
Bjontegaard	$\frac{80(p_2 - 2p_1 + p_0) + 40(q_0 - p_0) - 15(q_1 - p_1)}{256}$

304. Dr. Schonfeld provides no analysis to address these differences. In my view, this significantly undermines Dr. Schonfeld’s “routine experimentation” opinion because the range of possible coefficients when normalized is substantial and varied. When considering the inclusion of Δ , Dr. Schonfeld cannot, in my opinion, show that the coefficients in the prior art offset equations were within a known workable range, nor has Petitioner or Dr. Schonfeld provided any analysis as to how changing the first offset Δ would affect the overall filter design. As explained

above and illustrated in my testing, deblocking filter designs are sensitive to any change, and the frequency response will be affected by any change. I therefore do not agree that it would have been obvious to a POSITA to reach the claimed second offset expressions using routine experimentation because a POSITA would not have a reasonable expectation of success in obtaining the particular frequency response obtained using the claimed second offset. Dr. Schonfeld then engages in a hindsight driven analysis with respect to how much the weights of Bjontegaard's coefficient would need to be modified (i.e., from $5/16$ to $1/2$), which Dr. Schonfeld asserts would have resulted in "slightly more blurring." Ex-1003, ¶285. However, Dr. Schonfeld does not explain one way or the other why a POSITA would want to adjust the second offset of Bjontegaard to achieve slightly more blurring such that a POSITA would arrive at the specific frequency response produced by the '841 Patent. In my opinion, Dr. Schonfeld cannot make such a showing. Dr. Schonfeld's table illustrates my point that these differences cannot be minimized to a mere difference in coefficients because the frequency response is substantially different. Critically, the '841 Patent second offset does not filter ramp, while Bjontegaard's second offset does filter ramp. Moreover, Bjontegaard's second offset does not smooth the other structures, such as a step, in the same way that the '841 Patent's offset does.

p2	p1	p0	q0	q1	Type	$\frac{1(p2 - 2p1 + p0)}{4} + \frac{1\Delta}{2}$	$\frac{5(p2 - 2p1 + p0)}{16} + \frac{5\Delta}{16}$
200	150	150	150	150	Step (p1)	13	16
200	200	150	150	150	Step (p0)	-8	-13
200	200	200	150	150	Step (q0)	-10	-5
150	160	170	180	190	Ramp (10)	0	0
125	150	175	200	225	Ramp (25)	0	-1
200	250	200	250	200	Alternating	-6	-21
255	50	50	255	255	Line	90	84
255	255	50	50	255	Line	-51	-64
175	200	150	150	175	Random	-16	-22
50	125	100	150	125	Random	-11	-23
75	100	150	75	125	Random	-17	-5

Ex-1003, ¶283.

305. Further, in my view, Dr. Schonfeld has not provided any motivation whatsoever for why a POSITA would have combined the elements of any of the referenced offsets in any of the prior art above and modified them to reach the claimed second offset. In my view there is evidence to the contrary that is illustrated by my testing: changing filter designs leads to significant impacts on the operation of the filter. For example, when a simulation of Bjontegaard's filter design was compared to a simulation of the '841 filter design, the differences in the design led

to the pixel next closest to the block boundary being filtered differently by the two filters a majority of the time.

306. Further, in my opinion, the non-obviousness of the claimed second offset, and the apparent hindsight that Dr. Schonfeld is engaging in in his analysis is illustrated by the fact that, over decades of deblocking development, those skilled in the art came up with many solutions to the same problem of calculating a second offset using greatly differing approaches. For example, the normalized versions of the second offsets disclosed in the asserted references vary substantially in their coefficients used, which and how many pixels are used, the measure used (e.g., slope versus difference values) and whether they used previously calculated offsets in their expressions. Thus, Petitioner's argument that the claimed second offset expression would have been obvious falls apart in light of the many approaches that collectively teach away from the notion that particular expression would have been achieved by a POSITA using routine experimentation to optimize an entire equations with many different possible components.

- c. 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 $\frac{q_{0j}+q_{2j}-2q_{1j}-2\Delta}{4}$, wherein q_{2j} 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;*

307. In my opinion, Bjontegaard does not disclose or suggest element 10[f] alone or in combination.

308. First, although Dr. Schonfeld points to other offset equations in the prior art, he has not articulated any motivation to combine Bjontegaard with other references to reach the claimed third offset. For the reasons below, it is my opinion that Dr. Schonfeld cannot make this showing.

309. First, I have seen no analysis from Dr. Schonfeld or Petitioner that the third offset (Diff) in Bjontegaard is calculated if the accused second filter decision value, $S0_2$ is below the threshold $Lim(qp)$. As I explain above, this comparison may influence the value of Diff, but in my opinion, Dr. Schonfeld and Petition have not shown that this claim requirement is satisfied.

310. In addition, I do not agree that a POSITA would have read the accused Diff equation to use the value of e_{modified} instead of e as the equation appears in Bjontegaard. As I explain above, this proposition is contrary to the text of Bjontegaard because the authors of Bjontegaard, having defined e_{modified} in Section 4.5.5, chose to use e in 4.5.6.1 and 4.5.6.2, even though the defined e_{modified} was available to use in the Diff equations.

311. Like the second offset, the difference between Dr. Schonfeld's normalized third offset of the '841 Patent and that of Bjontegaard (with his own modification to Bjontegaard to include the extra parameter Δ):

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

312. Dr. Schonfeld opines that for the same reasons he provided with respect to the second offset, a POSITA would have also recognized that the coefficients applied to each of the components in the equations above would have been obvious. Ex-1003, ¶394-301. Thus, for the same reasons I provide in connection with the same elements it is my opinion that the coefficients would not have been obvious.

313. For example, while Dr. Schonfeld points to the use of different weights applied in other references such as Fu (which also uses a different “high-pass component” as explained above) and B118, these differences (both in “high-pass components” and coefficients) are critical to the differing frequency responses of the filters and the resulting filtering outcomes. Thus, in my view, these differences cannot be minimized as Dr. Schonfeld does in his analysis such that the equation of the '841 Patent could have been understood by a POSITA by optimizing the equation of Bjontegaard using routine experimentation. Instead, each offset expression reflects a very specific design with an intentionally designed frequency response. In particular, the '841 third offset uses a Laplacian that only uses q pixels for the “high-pass component.” This was a specific design choice of the overall '841 deblocking design such that the (i) consider the local structure near the block boundary and (ii)

offsets do not filter ramp. Moreover, the frequency response is specific to other types of structures, and the coefficients ensure that other structures, such as lines and steps are smoothed in a particular way.

314. The differences between these two offsets is further illustrated when the Δ is considered. As set forth in the preceding limitation, a similar normalization can be performed for the third offset that includes and propagates the Δ equation, and demonstrates that substantially different filtering coefficients.

315. Dr. Schonfeld provides no analysis to address these differences. In my view, this significantly undermines Dr. Schonfeld's "routine experimentation" opinion because the range of possible coefficients when normalized is substantial and varied. When considering the inclusion of delta, Dr. Schonfeld cannot, in my opinion, show that the coefficients in the prior art offset equations were within a known workable range, nor has Petitioner or Dr. Schonfeld provided any analysis as to how changing the first offset Δ would affect the overall filter design. As explained above and illustrated in my testing, deblocking filter designs are sensitive to any change, and the frequency response will be affected by any change. I therefore do not agree that it would have been obvious to a POSITA to reach the claimed third offset expressions using routine experimentation because a POSITA would not have a reasonable expectation of success in obtaining the particular frequency response obtained using the claimed third offset. Indeed the analysis the Dr. Schonfeld must

conduct with respect to how much the weights of Bjontegaard's coefficient would need to be modified (i.e., from $5/16$ to $1/2$), which Dr. Schonfeld asserts would have resulted in "slightly more blurring," is apparently driven by hindsight Ex-1003, ¶285. Dr. Schonfeld does not explain one way or the other why a POSITA would want to adjust the third offset of Bjontegaard to achieve slightly more blurring such that a POSITA would arrive at the specific frequency response produced by the '841 Patent. In my opinion, Dr. Schonfeld cannot make such a showing. Dr. Schonfeld's table illustrates my point that these differences cannot be minimized to a mere difference in coefficients because the frequency response is substantially different. Critically, the '841 Patent third offset does not filter ramp, while Bjontegaard's third offset does filter ramp. Moreover, Bjontegaard's third offset does not smooth the other structures, such as a step, in the same way that the '841 Patent's offset does.¹

¹ While the table below is a demonstrating using the second offset expression, a corresponding table using the third offset expression and corresponding pixels could be created with the same results.

p2	p1	p0	q0	q1	Type	$\frac{1(p2 - 2p1 + p0)}{4} + \frac{1\Delta}{2}$	$\frac{5(p2 - 2p1 + p0)}{16} + \frac{5\Delta}{16}$
200	150	150	150	150	Step (p1)	13	16
200	200	150	150	150	Step (p0)	-8	-13
200	200	200	150	150	Step (q0)	-10	-5
150	160	170	180	190	Ramp (10)	0	0
125	150	175	200	225	Ramp (25)	0	-1
200	250	200	250	200	Alternating	-6	-21
255	50	50	255	255	Line	90	84
255	255	50	50	255	Line	-51	-64
175	200	150	150	175	Random	-16	-22
50	125	100	150	125	Random	-11	-23
75	100	150	75	125	Random	-17	-5

Ex-1003, ¶283.

316. Further, in my view, Dr. Schonfeld has not provided any motivation whatsoever for why a POSITA would have combined the elements of any of the referenced offsets in any of the prior art above and modified them to reach the claimed third offset. In my view there is evidence to the contrary that is illustrated by my testing: changing filter designs leads to significant impacts on the operation of the filter. For example, when a simulation of Bjontegaard's filter design was compared to a simulation of the '841 filter design, the differences in the design led

to the pixel next closest to the block boundary being filtered differently by the two filters a majority of the time.

317. Further, in my opinion, the non-obviousness of the claimed third offset, and the apparent hindsight that Dr. Schonfeld is engaging in in his analysis is illustrated by the fact that, over decades of deblocking development, those skilled in the art came up with many solutions to the same problem of calculating a second offset using greatly differing approaches. For example, the normalized versions of the second offsets disclosed in the asserted references vary substantially in their coefficients used, which and how many pixels are used, the measure used (e.g., slope versus difference values) and whether they used previously calculated offsets in their expressions. Thus, Petitioner's argument that the claimed second offset expression would have been obvious falls apart in light of the many approaches that collectively teach away from the notion that particular expression would have been achieved by a POSITA using routine experimentation to optimize an entire equations with many different possible components.

E. Petitioner's Ground 1B Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Bjontegaard, Bjontegaard's Code, H.241, and B205/AAPA

318. I understand that Petitioner alleges that claims 2, 6, and 11 are obvious over Bjontegaard, Bjontegaard's Code, H.241, and B205. In my opinion, Petitioner's Ground 1B fails because claims 2, 6, and 11 depend from claim 1 (claim 2) and claim

5 (claims 6 and 11). Thus, for the same reasons that Bjontegaard alone or in combination fails to disclose or render obvious claims 1 and 5 discussed above, the dependent claims are also not obvious.

F. Petitioner's Ground 2A Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Aksu and Gonzalez

319. It is my opinion that Aksu in view of Gonzalez fails to teach the limitations of the Challenged Claims under Ground 2A for the reasons discussed below.

1. No Motivation to Combine Aksu and Gonzalez

320. I disagree with Dr. Schonfeld's opinion that a POSITA would have been motivated to combine Aksu's deblocking filter with Gonzalez's "second-order calculations." Ex-1003 ¶ 433. At the time of the '841 Patent in 2011, a POSITA looking at Aksu, which is a 2001 reference, would not have been motivated to look at a textbook that was generally directed to image compression techniques from 2008 to modify Aksu. Rather, a POSITA would in my view have looked more narrowly at other, and more recent, deblocking filtering designs that use fundamentally different deblocking filters.

321. In addition, Aksu's use of a series of 2-pixel difference calculations, which considers up to six pixels to determine the alleged "filter decision values", is a completely different measurement to detect a specific type of image property (e.g.,

flatness) compared with Gonzalez's second-order derivative calculations that are designed to detect a different type of image property (e.g., slope). A POSITA looking at Aksu's use of multiple 2-pixel difference calculations and trying to achieve computational efficiency would not be motivated to use a completely different metric in a completely different way (e.g., using one calculation using three pixels instead of using up to 6 calculations with up to six pixels) and expect to achieve a successful deblocking filtering design without extensive and undue testing.

322. Thus, in my opinion, a POSITA would not have been motivated to combine Aksu with Gonzalez.

G. Claims 1, 5, 16

a. 1[a]/5[a]/16[a]:

323. In my opinion, Aksu in combination with Gonzalez does not teach or suggest these limitations.

324. Dr. Schonfeld asserts that Aksu's d_r is a filter decision value because it represents activity and Aksu determines the number of pixels to filter based on d_r . Ex-1003, ¶ 380.

325. Aksu does not disclose that d_r is calculated using only pixels p_0 , p_1 , and p_2 , as required by the claimed. Instead, Aksu filter will measure the activity in 1 to 6 pixels (based on parameter n and n_{tr}) on each side of the block boundary to

calculate a decision value for a block and its neighboring block. The decision value is stored in d_r and d_l . An example of the specific calculations for d_r are shown below:

$$\begin{aligned} d_r = 6, & \text{ if } |r_1 - r_j| \leq \beta/j \text{ with all } j \in [1, 6], \\ \text{otherwise: } d_r = i, & \text{ where } i \text{ fulfils the conditions} \\ & i \in [1, n_{tr}], \quad (4) \\ & |r_1 - r_{i+1}| > \beta/i, \text{ and} \\ & |r_1 - r_j| \leq \beta/j \text{ with all } j \in [1, i]. \end{aligned}$$

Ex-1019 at 13:15-20.

326. The values of d_r and d_l are then used to identify the number of pixels to use in the filtering window. For example, for d_r , Table 2 of Aksu shows the determination of the width of the filtering window for the pixels r_1 , r_2 , and r_3 according to the value of d_r .

$d_r (d_l > 1)$	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

Ex-1019 at Table 2.

327. I disagree with Dr. Schonfeld that a POSITA would have been motivated to modify the teaching of Aksu by replacing the calculation of d_r in Aksu

with a three-pixel Laplacian equation using Gonzalez to arrive at the claimed calculation of a first filter decision value recited in these limitations.

328. First, Dr. Schonfeld's proposed modification requires a significant number of changes that would not have been obvious, including at least (i) changing the metric by which image activity is measured, (ii) changing and limiting the number of pixels considered for measuring that activity, (iii) changing the measurement from a series of equations to one equation, (iv) comparing the modified singular equation to a single threshold, instead of comparing each series of equations to a threshold, (v) adding an absolute value operator to the equation of Aksu, and (vi) applying the filtering decision to multiple lines in an 4x4 block instead of an 8x8 block to arrive at the claimed first filter decision value. I incorporate by reference my walk-through of the deblocking filter design in Aksu discussed above in Section X.E. This extensive modification would not have been obvious to a POSITA reading Aksu, alone or in combination without the benefit of hindsight because the modifications are in my view inconsistent with the overall deblocking design disclosed in Aksu.

329. Aksu's use of a series of 2-pixel difference calculations, with up to six pixels in the eventual filtering window (Ex-1019 at 13:10-20; Table 2), is a completely different measurement to detect a different type of image property (e.g., flatness) than the Laplacian equation in Gonzalez, which measures or detects slope.

See supra ¶260. As I explained above, a POSITA looking at Aksu’s use of multiple 2-pixel difference calculations would not be motivated to substitute this method using a completely different metric in a completely different way (e.g., using one calculation using three pixels instead of using up to 6 calculations with up to six pixels) and expect to achieve a successful deblocking filtering design, without extensive and undue testing. Ex-1019 13:10-20.

330. In addition, I disagree that a POSITA would have recognized that a Laplacian would be an improvement over first order differences such as those disclosed in Aksu for purposes of computing a measurement that can identify and respond to “regions of rapid intensity change” as Dr. Schonfeld alleges. Ex-1003, ¶383.

331. Further, a POSITA working on the development of H.265 at the time of the ’841 Patent would not have been motivated to reach back to technical proposals, such as those identified by Dr. Schonfeld that correspond to Aksu, which were proposed and rejected by the prior generation standard.

332. Further, in my view, the reasons given by Dr. Schonfeld for why a POSITA would be motivated to replace Aksu’s difference calculations of d_r with a one-dimensional Laplacian appear to be motivated by hindsight. For example, the alleged benefit of detecting rapid pixel-value changes with greater precision undermined by the fact that pixel changes would be detectable using Aksu’s

extremely granular difference calculations, that calculates the difference between neighboring pixels in up to a six-pixel window. Thus, Aksu's calculation would likely lead to more accuracy. Second, the pre-decision process disclosed in Aksu provides accuracy and granularity that is apparently compatible with Aksu's overall filter design. Changing the decision process to eliminate this process would, in my view, have a (most likely detrimental) effect on the overall performance of the filtering process in Aksu. Dr. Schonfeld fails to address this in his analysis, and a POSITA would not have a reasonable expectation of success in simply replacing an entire portion of a deblocking filter design, since these are very sensitive and carefully designed systems as I explained and illustrated earlier. The same reasoning applies to Dr. Schonfeld's third reason. Ex-1003, ¶388. The computational complexity allegedly saved by replacing Aksu's six calculations with one single calculation must be balanced with the desire for an accurate and effective deblocking filter. Dr. Schonfeld has not provided any analysis that the result filter would be successful with this replacement.

333. In addition, I disagree with Dr. Schonfeld's assertion that a POSITA would have known that Gonzalez's second-order difference of the form of $|p_2 - 2p_1 + p_0|$ "would provide a better approach to calculate the activity in a line of pixels" because such "algorithms were superior." Ex-1003, ¶383. In particular, even if using a second order derivative produced a stronger response to fine lines, isolated points

and noise than first order derivatives, robustness of the filter would necessarily be affected by the pixels used in measuring activity on either side of the block. Because the calculation of d_r in Aksu relies on a series of first order derivative equations for up to six pixels (Aksu, Ex-1019, 13:10-20), a POSITA optimizing for robustness would not recognize that a three-pixel Laplacian would, in isolation, produce a more “robust” filter (e.g., not as susceptible to over- or under-filtering) than the equation used to calculate d_l in Aksu, which can provide a more detailed measure of the image activity in the block and potentially therefore lead to more accurate deblocking. Thus, a POSITA would not necessarily expect that substituting Aksu’s equations with a Laplacian would improve visual quality, as Dr. Schonfeld suggests. Further, first-order difference calculations like those used in Aksu detect image details that preclude filtering (e.g., flatness) because a flat line is more likely to be a blocking artifact while ramp structures which are measured by Laplacians are more likely to be the actual image signal. Thus, a POSITA would understand that the first-order difference calculations of Aksu could in fact be used to preclude filtering.

334. In addition, Dr. Schonfeld asserts that it would have been obvious to remove Aksu’s process for determining how many pixels to consider, which could range from 1 to 6 (Ex-1019 at 13:10-20), in favor of fixing the computation to only consider 3 pixels. I disagree that a POSITA would modify Aksu in this way without the benefit of hindsight. The value of “n” in Aksu gives an overall activity parameter

that is related to both the difference of the pixel values across the block boundary and to the size of the quantization step QP of the coefficients applied during image coding to result in more accurate deblocking filter by determining which pixels to examine for filtering. Ex-1019 at 9-10. Thus, it would not have been obvious to modify Aksu to simply replace the step of calculating both n and d_l with a fixed three-pixel Laplacian equation in the pursuit of reducing computational complexity as Dr. Schonfeld asserts, because the overall deblocking design that Aksu teaches, including the calculation of n , leads to a more accurate and responsive filtering decision. Ex-1019 at 10:14-21 (“it is advantageous that its value is generally higher when the difference of pixel values Δ across the block boundary is small in comparison with the size of the quantization step QP of the coefficients received as the result of the coding transformation. If the difference between the pixel values Δ is larger, there is a high probability that there is a real image edge at the block boundary, and in this case the pixels are preferably not examined for filtering at all ($n=0$)”). Therefore, it would not be obvious to a POSITA to simply eliminate the calculation of n in Aksu as Dr. Schonfeld asserts (see Ex-1003 ¶¶ 381-89, 396-99) because it would undermine the overall deblocking design disclosed in Aksu and most likely lead to filtering inaccuracies and poorer overall deblocking performance.

335. For the foregoing reasons I explain that it would not have been obvious to simply substitute all of the series of absolute difference equations used to calculate

d_l in Aksu with a singular equation that uses a different activity metric and a different number of pixels, it would also not have been obvious to a POSITA to compare a three-pixel Laplacian to a single threshold. In Aksu, each of the calculated differences for i are compared to a threshold as set forth below.

$$\begin{aligned}
 & d_r = 6, \text{ if } |r_1 - r_j| \leq \beta/j \text{ with all } j \in [1, 6], \\
 & \text{otherwise: } d_r = i, \text{ where } i \text{ fulfils the conditions} \\
 & \quad i \in [1, n_{tr}], \quad (4) \\
 & \quad |r_1 - r_{i+1}| > \beta/i, \text{ and} \\
 & \quad |r_1 - r_j| \leq \beta/j \text{ with all } j \in [1, i].
 \end{aligned}$$

Ex-1019 at 13:10-20.

336. As set forth above, the overall filtering mechanism disclosed in Aksu is a granular approach that calculates the value of d_l and d_r based on a comparison of each of the series of differences in pixel values to a threshold. Ex-1019, 13:10-20. As described above, simply replacing all of the series of equations in Aksu with one equation that uses a different metric to measure the properties of the block based on a different number of pixels and likewise replacing all of the threshold comparisons with one threshold comparison, would significantly alter the calculation of the value of d_l and d_r that is used later in the filtering in Aksu, and would thus be inconsistent with the overall filtering design. Dr. Schonfeld has not shown otherwise.

337. Thus, a mere substitution of all of the elements explained herein would not lead a POSITA to a reasonable expectation of success in the resulting filtering design because the substituted components are inconsistent with the overall filter design in Aksu, and as explained above in Section XI, simply taking one component of a filter and inserting it into another filter design requires extensive and undue experimentation and testing, and further modification and optimization of the overall filter design because the filter value is applied to the underlying video signal and because even apparently ‘modest’ changes in the filtering design can have significant effects on the filtering results and therefore on subjective quality and inter prediction performance.

338. Finally, Dr. Schonfeld ignores that the trend in the industry taught away from using a Laplacian equation in this context. For example, early references in the early 2000’s such as Bjontegaard’s Code that were directed to the development of the H.264 Standard disclosed Laplacian equations as a measure to detect image activity. However, those proposals were not incorporated into the published H.264 standard, which uses a wholly different metric of absolute difference calculations to measure image activity (*see* the calculation of a_p and a_q above in Section IX.A). In addition, Dr. Schonfeld relies on Aksu for its calculations of d_l and d_r that are based on absolute difference values to show a first filter decision and a second filter decision. However, this same concept of measuring the absolute difference for each

side of the block boundary is included in the H.264 deblocking process, which is explicitly distinguished on the face of the '841 Patent and was considered during prosecution of the '841 Patent.

339. Thus, it would not have been obvious, in my opinion, to modify Aksu to use a three-pixel Laplacian equation to arrive at this limitation.

b. 1[b], 5[b], 16[b]

340. In my opinion, Aksu in combination with Gonzalez does not teach or suggest these limitations.

341. Dr. Schonfeld asserts that Aksu's d_l is a filter decision value because it represents activity and because Aksu determines the number of pixels to filter based on d_l . Ex-1003, ¶¶401-03. I disagree with Dr. Schonfeld that a POSITA would have been motivated to modify the teaching of Aksu by replacing the calculation of d_l with a three-pixel Laplacian equation using Gonzalez to arrive at the claimed calculation of a second filter decision value. Ex-1003, ¶¶402-403.

342. I further disagree with Dr. Schonfeld's opinion that Aksu performs the same series of difference calculations that were used for d_l , "except that all r 's are replaced by l 's." Ex-1003, ¶ 403 (citing Ex-1019 at 13:21-23). That is because Aksu teaches the calculation and use of d_l and d_r in that are applied for the left and right side of the block boundary, respectively. Ex-1019 at 12:21-23, 13:10-20, Fig. 2.

343. Accordingly, Dr. Schonfeld's proposed modification would entail d_l undergoing the same significant change as d_r described in the preceding section for limitation 1[a]. Thus, for all the reasons set forth above in limitation 1[a] in the preceding section, Aksu in combination with Gonzalez does not teach or suggest 1[b]. *See supra* §XII.D.2.b.

c. 1[c], 5[c], 5[e], 16[c]

344. In my opinion, Aksu, alone or in combination with Gonzalez, does not teach or suggest these limitations.

345. In my opinion, the combination does not disclose or render obvious these limitations because the determination of how many pixels in Aksu is not based on the comparison of the alleged first filter decision value, d_r , to the thresholds based on β . Instead, d_r , which Petitioner alleges is the claimed first filter decision value, is assigned an integer value, i , but d_r is not compared to the thresholds based on β .

346. In Aksu the number of pixels to filter is dependent on whether d_r is equal to 0, 1, 2, 3, 4, 5, or 6 as set forth in Table 2 below, and filtering only occurs if both d_r and d_l are greater than 1.

$d_r (d_i > 1)$	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

Paper 1 at 59-60.

347. I disagree that this fundamental difference between Aksu and the '841 Patent can be alleviated using the combination of Aksu with Gonzalez's second-order Laplacian to measure activity and alleging that a POSITA would understand that there would be no need for the integer values between 1 and 6 because there would only be one activity measure calculation performed. Petition, 60. Dr. Schonfeld and Petitioner fail to explain how the proposed combination would satisfy the limitations requiring determining *how many* pixels to filter *based on a comparison of the first filter decision value to a first threshold value*, but merely state that the combination would generally filter more pixels when the activity level is below a threshold, and generally filter less pixels when the activity is equal to or exceeds the threshold. *Id.* Further, making the integer value of d_r bigger does not indicate filtering more pixels when the activity measure is *below* a threshold because Aksu only teaches setting $d_r = i$ where $|r_1 - r_{i+1}|$ is *greater than* β/i . Ex-1019, Further,

making the integer value of d_r smaller is not filtering less pixels when the activity is equal to or exceeds a threshold because Aksu only teaches setting $d_r = i$ where $|r_1 - r_j|$ is *equal to or less than* β/j and j is all numbers from 1 to i . A POSITA would not have understood Aksu to disclose filtering more or fewer pixels based on Petitioner's interpretation of Aksu's threshold comparisons because the value of d_r is not compared to any threshold in Aksu.

348. Therefore, in my opinion, Aksu does not teach or suggest these limitations.

d. 1[d], 5[d], 5[f], 16[d].

349. In my opinion, Aksu, alone or in combination does not teach or suggest these limitations.

350. Dr. Schonfeld acknowledges that Aksu discloses that “[t]he values of the pixels l_1 , l_2 , and l_3 are determined in the same manner [as their corresponding r block pixels] according to the value of the parameter d_l .” Ex-1019 at 15:19-21; *see also* Ex-1003 ¶415. Thus, as I explained above, Aksu fails to teach or suggest these limitations for the same reasons set forth in the immediately preceding limitation 1[c] as applied to the left block. *See supra* §XII.D.2.e.

351. Furthermore, it is my opinion that 1[d] is not rendered obvious for the same reasons described above for limitation 1[d] of Ground 1 and limitation 1[c] for Ground 2A. *See supra* §§XII.D.2.b, XII.B.1.e.

352. Therefore, in my opinion, Aksu does not teach or suggest these limitations.

H. Petitioner's Ground 2B Fails Because The Challenged Claims Are Not Obvious In View of the Combination of Aksu, Gonzalez, H.241, and B205/AAPA

353. I understand that Petitioner alleges that claims 2, 6, and 11 are obvious over Aksu, Gonzalez, H.241, and B205. In my opinion, Petitioner's Ground 2B fails because claims 2, 6, and 11 depend from claim 1 (claim 2) and claim 5 (claims 6 and 11). Thus, for the same reasons that Aksu in combination with Gonzalez fails to disclose or render obvious claims 1 and 5 discussed above, the dependent claims are also not obvious.

I. Petitioner's Ground 2C Fails Because The Challenged Claims Are Not Obvious in View of the Combination of Aksu, Gonzalez, Fu, and B118

354. Based on my opinions discussed in more detail below, it is my opinion that Aksu in combination with Gonzalez, Fu, and B118 does not teach or suggest Claim 10.

1. No Motivation to Combine Aksu with Gonzalez, Fu, and/or B118

355. I disagree with Dr. Schonfeld that a POSITA would have been motivated to combine Aksu with Fu and/or B118. A POSITA would not have been motivated to combine Aksu's deblocking design with Fu's offset equations. Aksu describes a filter that calculates the mean of pixel values that appear in a filtering

window. Ex-1019 at 15:4-8. In my opinion, a POSITA would not look to combine the method of calculating the mean of a range of number of pixels between 1 and 6 that is disclosed in Aksu with Fu and/or B118's offset equations, without the benefit of hindsight. Further, a POSITA would not have a reasonable expectation of success using Fu and/or B118's offset equations with Aksu because the varying discrepancies between the references' filtering designs, in my view, would result in an arbitrary combination of components or processes that I would not expect to work as a coherent design without extensive testing and further modification.

356. For example, while Aksu discloses calculating a series of absolute difference equations using up to 6 pixels for the calculation of d_l and d_r as parameters for measuring image activity on either side of a block boundary and comparing those calculations to a variable β divided by the number of pixels used for the calculation, Fu discloses calculating the absolute difference of just 2 pixels (p_2 and p_0 and q_2 and q_0) on each side of the block boundary and comparing that difference to a variable β . Fu, Ex-1022 at 5:13-15. Likewise, B118 uses the absolute difference of the same pixels disclosed in Fu and the same comparison to β . Ex-1023 at 48.

357. Thus, a POSITA would not expect to have success using the calculations of d_l and d_r in Aksu with the offsets disclosed in Fu and/or B118 because the deblocking designs of Fu and B118 use significantly different equations for the filtering decisions. In addition Aksu discloses an exemplary embodiment where the

filter is based on the average of a sliding filtering window that could be based on up to 7 pixels. This is fundamentally different filtering process from Fu and/or B118 which disclose filters based a fixed number of pixels in the same position relative to the block boundary rather than centered on the pixel to be modified. In my opinion, a POSITA would not expect to piece together disparate aspects of different deblocking designs and arrive at an effective solution without undue testing and experimentation, and without the benefit of hindsight, for example, using the claims of the '841 Patent as a guide. Thus, I disagree with Dr. Schonfeld's opinions and I disagree that a POSITA would not have been motivated to combine Aksu's deblocking calculations with the offset equations disclosed in Fu and/or B118 with a reasonable expectation of success.

J. Claim 10 (Ground 2C)

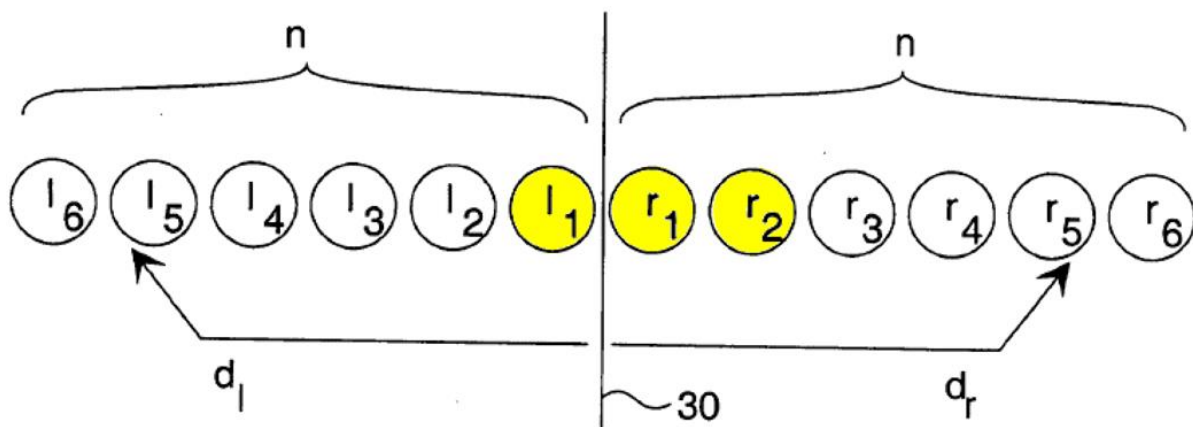
358. Claim 10 depends from Claim 5. Thus, Aksu in combination with Gonzalez, Fu, and B118 does not teach or suggest Claim 10.

- a. 10[a]: a first offset calculator configured to calculate a first offset Δ based on $\frac{9 \times (q_{0j} - p_{0j}) - 3 \times (q_{1j} - p_{1j})}{16}$, wherein p_{0j} denotes a pixel value of a pixel closest to, in said first line of pixels, said block boundary, p_{1j} denotes a pixel value of a pixel next closest to, in said first line of pixels, said block boundary, q_{0j} denotes a pixel value of a pixel in said neighboring block closest to, in said corresponding first line of pixels, said block boundary and q_{1j} denotes a pixel value of a pixel of said neighboring block next closest to, in said corresponding first line of pixels, said block boundary;"*

359. Claim 10 depends from claim 5. Thus, Aksu in combination with Gonzalez, Fu, and B118 does not teach or suggest Limitation 10[a] for the same reasons disclosed above in connection with claim elements 5[pre]-5[d]. *See supra* §XII.D.4.a-e.

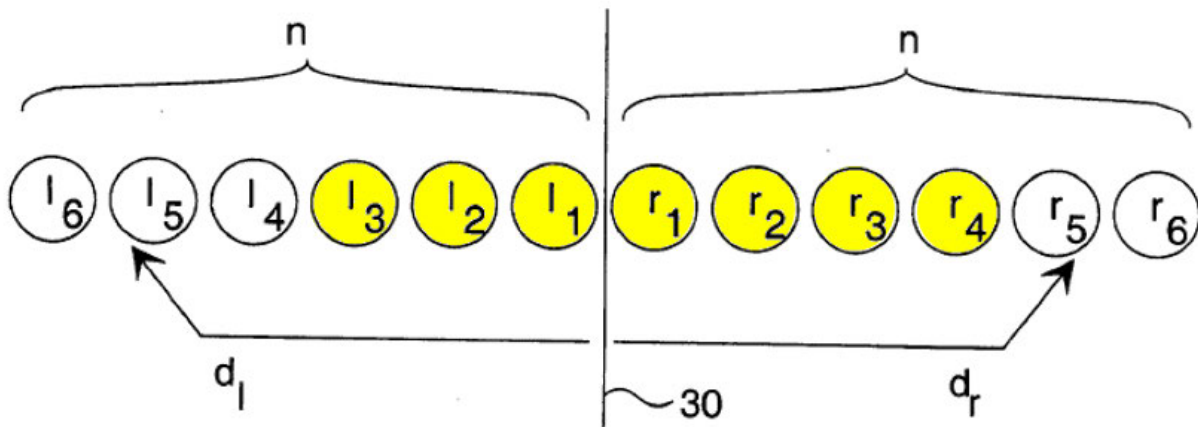
360. Aksu discloses a filter that calculates the average values of the pixels in a designated filtering window. Ex-1019 at 15:6-8. Thus, Aksu's filtering windows can vary between 3 and 7 pixels and are not limited to the two pixels closest to the block boundary as the claimed first offset is (e.g., calculating a first offset using pixels q0, p0, q1 and p1). Ex-1019 at 13:10-26; 15:16-26; Table 2; Fig. 2; Section X.C. As described above in Sections X.C and XI.D.1, in my view a POSITA would not have arrived at the claimed First Offset based on routine experimentation by modifying the "averaging filter" of Aksu where the average of the pixels in the filtering window is used to determine the new (e.g., filtered) pixel. Ex-1019 at 15:2-15. Petitioner asserts that weighted offset equations would have been obvious because they are "efficient" and provide more control over the filter's frequency response than Aksu's averaging filter. But Aksu's fundamentally different averaging filter is bound up with the filter decisions in Aksu that uses a sliding filter window, where the pixels considered for calculating the actual filter applied to a pixel are centered around the pixel to be filtered. Ex-1019 (Aksu) at 15. I provide some examples of the different possible averaging filters below.

361. Aksu explains that Table 2 above “shows the determination of the width of the filtering window for the pixels r_1 , r_2 , and r_3 according to the value of the parameters d_r in a preferred embodiment of the invention.” Ex-1019 at 15:17-19. Thus, if d_r is equal to 2, then 3 pixels are in the filtering window. Ex-1019 at 15:22-24. Thus, if r_1 is in the left block and closest to the block boundary as Dr. Schonfeld suggests, then the pixel to the left of r_1 (l_2) and the pixel to the right of r_1 (l_1), which is the pixel closest to the block boundary in the neighboring block will be included in the filtering window. I illustrate this example below using Figure 2 of Aksu, which can be expanded upon using the other values in Table 2. In the example below, the filtered pixels are highlighted in yellow. For example, I illustrate this example below using Figure 2 of Aksu, which can be developed using the other values in Table 2. In the example below, the filtered pixels are highlighted in yellow.



Ex-1019, Fig. 2 (annotated). In this case, the filter is calculated using Aksu's averaging filter as $\frac{l_1+r_1+r_2}{3}$.

362. Illustrating another example, if $d_r = 6$, then 7 pixels are in the filtering window as shown below because there are 3 pixels from each side of pixel r_1 . In this case, the filter for modifying r_1 is calculated using Aksu's averaging calculator as $\frac{l_3+l_2+l_1+r_1+r_2+r_3+r_4}{7}$.



Ex-1019, Fig. 2 (annotated).

363. Dr. Schonfeld has not provided any analysis as to why a POSITA would expect to obtain a successful deblocking filter design by substituting a “weighted coefficient” equation that could successfully work with Aksu's disclosure of using the sliding filtering window that centers on the pixel to be filtered. In my opinion, a POSITA would not have had a reasonable expectation of success in making this

substitution given the illustration of significant differences in filtering outcomes when comparing different deblocking filtering designs.

364. However, if Aksu's averaging filter was substituted, the claimed first offset is still not disclosed or obvious, in my opinion, in light of Fu and/or B118. As demonstrated by Dr. Schonfeld's own analysis of the prior art offsets, such as those in Fu and B-118, these variations in the prior art demonstrate that no uniform approach was known for calculating a first offset in the prior art. For the reasons set forth above in Ground 1A, I disagree that a would have experimented with "weighted-offset" equations like Fu's or B118's to optimize the coefficients used and the number of pixels used to come up with the solution in the '841 Patent that solved the same deblocking problem that had gone unsolved despite the decade-long development of various standardized video codecs between Aksu's priority date and

the '841 Patent. *Id.* The extensive possibilities available undermine this proposition.

'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	$\frac{7.5(q_0 - p_0) + 2(q_1 - p_1) - 2.5(q_2 - p_2)}{16}$

365. I also disagree that a POSITA would have recognized, without hindsight, that only using pixel differences ($q_0 - p_0$) and ($q_1 - p_1$) is a preferred starting point. Notably, Dr. Schonfeld's explanation for why a POSITA would prefer this approach is deficient, except for a conclusory statement asserting that the "equations are simpler and more computationally efficient." See Ex-1003 ¶504. Notably, Dr. Schonfeld's opinion, again, ignores the ultimate effect on the filtering that would result from using this design. These effects to the filtering are reflected in his own chart, which is reproduced below:

p2	p1	p0	q0	q1	Type	9 and 3 (claims)	6 and 2 (Fu)	8 and 3 (Bjont.)	8 and 2 (B118)
200	150	150	150	150	Step (p1)	0	0	0	0
200	200	150	150	150	Step (p0)	9	6	9	6
200	200	200	150	150	Step (q0)	-19	-13	-16	-19
150	160	170	180	190	Ramp (10)	0	0	-1	1
125	150	175	200	225	Ramp (25)	0	0	-2	3
200	250	200	250	200	Alternating	38	25	34	31
255	50	50	255	255	Line	77	51	64	77
255	255	50	50	255	Line	0	0	0	0
175	200	150	150	175	Random	5	3	5	3
50	125	100	150	125	Random	28	19	25	25
75	100	150	75	125	Random	-47	-31	-42	-41

Ex-1003 ¶505. Notably, only the '841 Patent does not filter ramp structures while smoothing step structures in a particular way that is not achieved by any of the other filter designs. Moreover, my simulations demonstrate the typically significantly different effect on filtering outcome when different filter designs are applied to the same set of pixels.

366. Thus, the combination of Aksu, Gonzalez, Fu, and B118 does not disclose 10[a].

- b. 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 $\frac{p0j+p2j-2p1j+2\Delta}{4}$, wherein $p2j$ denotes a pixel value

*of a pixel second next closest to, in said first line of pixels,
said block boundary;*

367. It is my opinion that Aksu in combination with Gonzalez, Fu, and B118 does not teach or suggest Limitation 10[d] for at least the same reasons discussed above for claims 2, 5 and limitation 10[a] of Ground 2C and limitation 10[d] for Ground 1A. §§ XII.D.2, XII.D.4, XII.F.2.a, XII.B.6.d.

368. First, a POSITA would not have been motivated to combine Aksu with Gonzalez, Fu, and/or B118 for all the reasons described above in Section XII.F.1. Dr. Schonfeld does not offer a motivation to combine Aksu with Gonzalez, Fu, and/or B118.

369. Petitioner alleges that d_r in Aksu is the first filter decision value, but Aksu does not disclose or suggest calculating its averaging filter if d_r is less than a threshold, which is required by the claims. Ex-2001 ¶590. Indeed, Petitioner fails to make any argument that Aksu in combination with any other reference contains any such disclosure. Petition at 80-82.

370. Nor does Aksu in combination with Gonzalez, Fu, B118 and a POSITA's knowledge render obvious the claimed second offset. Indeed, Petitioner's analysis would require a POSITA to "experiment with different high-pass filter equations" and "the coefficients assigned to each of the two components." However, Aksu's second offset is an averaging filter, which is fundamentally different from

any of the “high-pass filter” equations that Dr. Schonfeld allege render the second offset. Dr. Schonfeld has not provided any analysis as to why a POSITA would expect to obtain a successful deblocking filter design by substituting a weighted high-pass filter equation within Aksu’s disclosure of using a sliding filtering window that centers on the pixel to be filtered. In my opinion, a POSITA would not have had a reasonable expectation of success in making this substitution given the significant differences in filtering outcomes when comparing different deblocking filtering designs, which I have explained at length herein.

371. Further, as explained above, the fact that no asserted prior art reference working to solve the same problem of calculating a second offset in a deblocking filter reached the specific components and coefficients of the claimed second offset shows that the claimed expression is non-obvious nor could it be obtained through routine experimentation. For example, throughout a decade of video codec development between Aksu and the ’841 Patent, there was no agreement on a single preferred design for a second offset, including factors such as coefficient values, how many pixels used, the metric used, and which pixels are used. Moreover, no reference disclosed the offset of the ’841 Patent despite the access that POSITAs had to the high-pass components and coefficients that Petitioner and Dr. Schonfeld assert were well-known in the art. Petition at 80-81. Further, Dr. Schonfeld’s analysis fails to explain that a POSITA would understand the importance and effect of changing

these weights and scaling the result because the offset is the actual filter value that is applied to the underlying signal. Thus, the adjustments that Dr. Schonfeld makes to reach the claimed second offset would result in significant changes to the frequency response of the filter, and would have a significant effect on the ultimate filtering outcome and therefore on subjective quality and inter prediction performance. Thus, as explained above in element 10[a], I disagree that reaching the claimed offset would have been the result of routine experimentation because no prior reference discloses the claimed second offset when addressing the same problem of deblocking filtering over the course of at least a decade, during the development of H.264 and H.265, between the time of Aksu and the '841 Patent.

372. Thus, 10[d] is not suggested or taught by Petitioner's combination.

c. 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 $\frac{q_{0j}+q_{2j}-2q_{1j}-2\Delta}{4}$, wherein q_{2j} 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;

373. It is my opinion that Aksu in combination with Gonzalez, Fu and B118 does not teach or suggest Limitation 10[f] for at least the same reasons discussed above for claims 2, 5 and limitation 10[a] of Ground 2C and limitation 10[d] for Ground 1A. §§ XII.D.2, XII.D.4, XII.F.2.a, XII.B.6.d.

374. First, a POSITA would not have been motivated to combine Aksu with Gonzalez, Fu, and/or B118 for all the reasons described above in Section XII.F.1. Dr. Schonfeld does not offer a motivation to combine Aksu with Gonzalez, Fu, and/or B118.

375. Petitioner alleges that d_l in Aksu is the first filter decision value, but Aksu does not disclose or suggest calculating its averaging filter if d_l is less than a threshold, which is required by the claims. Ex-2001 ¶590. Indeed, Petitioner fails to make any argument that Aksu in combination with any other reference contains any such disclosure.

376. Nor does Aksu in combination with Gonzalez, Fu, B118, and a POSITA's knowledge render obvious the claimed third offset. Indeed, Petitioner's analysis would require a POSITA to "experiment with different high-pass filter equations" and "the coefficients assigned to each of the two components." However, Aksu's third offset is an averaging filter, which is fundamentally different from any of the "high-pass filter" equations that Dr. Schonfeld allege render the third offset obvious. Dr. Schofeld has not provided any analysis as to why a POSITA would expect to obtain a successful deblocking filter design by substituting a weighted high-pass filter equation that within Aksu's disclosure of using a sliding filtering window that centers on the pixel to be filtered. In my opinion, a POSITA would not have had a reasonable expectation of success in making this substitution given the

significant differences in filtering outcomes when comparing different deblocking filtering designs, as I have explained at length herein.

377. Further, as explained above, the fact that no asserted prior art reference working to solve the same problem of calculating a second offset in a deblocking filter reached the specific components and coefficients of the claimed third offset shows that the claimed expression is non-obvious nor could it be obtained through routine experimentation. For example, throughout a decade of video codec development between Aksu and the '841 Patent, there was no agreement on a single preferred design for a third offset, including factors such as coefficient values, how many pixels used, the metric used, and which pixels are used. Moreover, no reference disclosed the offset of the '841 Patent despite the access that POSITAs had to the high-pass components and coefficients that Petitioner and Dr. Schonfeld assert were well-known in the art. Further, Dr. Schonfeld's analysis fails to explain that a POSITA would understand the importance and effect of changing these weights and scaling the result because the offset is the actual filter value that is applied to the underlying signal. Thus, the adjustments that Dr. Schonfeld makes to reach the claimed third offset would result in significant changes to the frequency response of the filter, and would have a significant effect on the ultimate filtering outcome and therefore on subjective quality and inter prediction performance. Thus, as explained above in element 10[a], I disagree that reaching the claimed offset would have been

the result of routine experimentation because no prior reference discloses the claimed second offset when addressing the same problem of deblocking filtering over the course of at least a decade, during the development of H.264 and H.265, between the time of Aksu and the '841 Patent

378. Thus, 10[f] is not suggested or taught by Petitioner's combination.

XIII. SUMMARY AND OTHER REMARKS

379. It is my opinion that Petitioner has failed to show that any of the Challenged Claims set forth in Grounds 1A, 1B, 2A, 2B, and/or 2C are unpatentable as obvious as explained above.

380. My opinions expressed in this Declaration are based on the information available to me at this time. To the extent any additional information becomes available, I reserve the right to supplement my opinions contained herein.

I declare under penalty of perjury under the laws of the United States of
America that the foregoing is true and correct.

Date: 9/20/2024

Iain E. Richardson
Iain Richardson, Ph.D.