

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION

Petitioner,

v.

TELEFONAKTIEBOLAGET LM ERICSSON,

Patent Owner.

Case No. IPR2024-01009
U.S. Patent 10,142,659

DECLARATION OF IAIN RICHARDSON

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	2
III.	MATERIALS CONSIDERED	4
IV.	SCOPE OF OPINIONS	5
V.	LEGAL STANDARDS	5
A.	Presumption of Validity	5
B.	Obviousness.....	7
C.	Motivations to Combine	9
D.	Objective Indicia of Nonobviousness	10
VI.	LEVEL OF ORDINARY SKILL IN THE ART	10
VII.	CLAIM CONSTRUCTION	11
VIII.	TECHNOLOGY BACKGROUND.....	12
A.	Introduction to Video Coding	12
B.	Introduction to Deblocking Filters	17
IX.	U.S. PATENT NO. 10,142,659 (“’659 Patent”) (Ex-1001).....	25
A.	Priority Date	25
B.	’659 Patent Overview	26
C.	The ’659 Patent’s Prosecution History	42
X.	SIMULATION OF DEBLOCKING FILTER DESIGNS.....	43
A.	Testing Methodology	43
B.	Matlab Scripts.....	46
C.	Analysis of Test Simulation Results	55
1.	Using Thr1/Thr2 from the ’659 patent and using the Q15-J-28 disclosed method for calculating Diff values:	58
2.	Using Thr1/Thr2 from the ’659 patent and the ‘image.c’ method for calculating Diff values:	60
3.	Using S01/S02 from Q15-J-28 as the ‘threshold’ comparisons and the Q15-J-28 disclosed method for calculating Diff values:	61

4.	Using S01/S02 from Q15-J-28 as the ‘threshold’ comparisons and the ‘image.c’ method for calculating Diff values:.....	61
XI.	ASSERTED REFERENCES	62
A.	List.....	62
B.	Bjontegaard	64
C.	Bjontegaard’s Code	69
D.	B118	77
E.	VCEG-M21	81
F.	Bjontegaard ’048	81
G.	Aksu.....	82
A.	Fu.....	84
B.	B205	87
XII.	THE ASSERTED CLAIMS OF THE ’659 PATENT ARE NOT OBVIOUS.....	90
A.	No Motivation to Combine Fu with “the knowledge of a POSITA as evidenced by List, Bjontegaard, B205 and B118”	90
B.	Petitioner’s Ground 1 Fails Because The Challenged Claims Are Not Obvious In View Of Fu And A POSITA’s Knowledge As Evidenced By List, Bjontegaard, B205, And B118.....	96
1.	Independent Claims 1 and 10.....	97
2.	Dependent Claim 4	106
C.	Petitioner’s Ground 2 Fails Because The Challenged Claims Are Not Obvious In View Of Fu And A POSITA’s Knowledge As Evidenced By List, Bjontegaard, B205, And B118.....	117
D.	No Motivation to Combine Bjontegaard with “the knowledge of a POSITA as evidenced by List, Fu, B205, and B118”	117
1.	Independent Claims 1 and 10.....	123
2.	Dependent Claim 4	128
XIII.	SUMMARY AND OTHER REMARKS	138

TABLE OF CLAIMS

Reference	Claim Limitation
Claim 1	
1[pre]	A method of reducing blocking artifacts at a block boundary between a block of multiple pixels and a neighboring block of multiple pixels in a video frame, said method comprises:
1[a]	calculating a first offset to be equal to $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4,$ wherein $\gg$ denotes a right shift operation, p_0 denotes a pixel value of a pixel closest to said block boundary in a line of pixels in said block, wherein said line of pixels is perpendicular to said block boundary, p_1 denotes a pixel value of a pixel next closest to said block boundary in said line of pixels, q_0 denotes a pixel value of a pixel closest to said block boundary in a corresponding line of pixels in said neighboring block, wherein said corresponding line of pixels is perpendicular to said block boundary and q_1 denotes a pixel value of a pixel next closest to said block boundary in said corresponding line of pixels;
1[b]	modifying said pixel value p_0 of said pixel closest to said block boundary in said line of pixels by adding said first offset to said pixel value p_0 of said pixel closest to said block boundary in said line of pixels to form a modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels; and
1[c]	modifying said pixel value q_0 of said pixel closest to said block boundary in said corresponding line of pixels by subtracting said first offset from said pixel value q_0 of said pixel closest to said block boundary in said corresponding line of pixels to form a modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels.
Claim 2	
2[a]	The method according to claim 1, further comprising clipping said first offset to be within an interval of $-t_c$ and t_c by setting said first offset to be equal to $-t_c$ if said first offset is smaller than $-t_c$ and by setting said first offset to be equal to t_c if said first offset is larger than t_c .

Reference	Claim Limitation
2[b]	wherein t_c is a threshold value that is dependent on a quantization parameter value assigned to said block.
Claim 3	
3[a]	The method according to claim 1, further comprising: clipping said modified pixel value $p0'$ of said pixel closest to said block boundary in said line of pixels to be within an interval of zero and a defined maximum value by setting said modified pixel value $p0'$ of said pixel closest to said block boundary in said line of pixels to be equal to zero if said modified pixel value $p0'$ of said pixel closest to said block boundary in said line of pixels is smaller than zero and by setting said modified pixel value $p0'$ of said pixel closest to said block boundary in said line of pixels to be equal to said defined maximum value if said modified pixel value $p0'$ of said pixel closest to said block boundary in said line of pixels is larger than said defined maximum value; and
3[b]	clipping said modified pixel value $q0'$ of said pixel closest to said block boundary in said corresponding line of pixels to be within an interval of zero and said defined maximum value by setting said modified pixel value $q0'$ of said pixel closest to said block boundary in said corresponding line of pixels to be equal to zero if said modified pixel value $q0'$ of said pixel closest to said block boundary in said corresponding line of pixels is smaller than zero and by setting said modified pixel value $q0'$ of said pixel closest to said block boundary in said corresponding line of pixels to be equal to said defined maximum if said modified pixel value $q0'$ of said pixel closest to said block boundary in said corresponding line of pixels is larger than said defined maximum value.
Claim 4	
4[a]	The method according to claim 1, further comprising: calculating a second offset to be based on $(p_2 + p_0 - 2 \times p_1 + 2 \times \Delta)/4$, wherein p_2 denotes a pixel value of a pixel second next closest to said block boundary in said line of pixels and Δ denotes said first offset;
4[b]	calculating a third offset to be based on $(q_2 + q_0 - 2 \times q_1 - 2 \times \Delta)/4$, wherein q_2 denotes a pixel value of a pixel second next closest to said block boundary in said corresponding line of pixels;

Reference	Claim Limitation
4[c]	modifying said pixel value p1 of said pixel next closest to said block boundary in said line of pixels by adding said second offset to said pixel value p1 of said pixel next closest to said block boundary in said line of pixels to form a modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels; and
4[d]	modifying said pixel value q1 of said pixel next closest to said block boundary in said corresponding line of pixels by adding said third offset to said pixel value q1 of said pixel next closest to said block boundary in said corresponding line of pixels to form a modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels.
Claim 5	
5[a]	The method according to claim 4, wherein calculating said second offset comprises calculating said second offset to be equal to $((p2+p0+1)>>1)-p1+\Delta>>1$, wherein $>>$ denotes a right shift operation; and
5[b]	calculating said third offset comprises calculating said third offset to be equal to $((q2+q0+1)>>1)-q1-\Delta>>1$
Claim 6	
6[a]	The method according to claim 4, further comprising: clipping said second offset to be within an interval of $-t_{c2}$ and t_{c2} by setting said second offset to be equal to $-t_{c2}$ if said second offset is smaller than $-t_{c2}$ and by setting said second offset to be equal to t_{c2} if said second offset is larger than t_{c2} ,
6[b]	wherein $-t_{c2}$ is a threshold value that is dependent on a quantization parameter value assigned to said block; and
6[c]	clipping said third offset within an interval of $-t_{c2}$ and t_{c2} by setting said third offset to be equal to $-t_{c2}$ if said third offset is smaller than $-t_{c2}$ and by setting said third offset to be equal to t_{c2} if said third offset is larger than t_{c2} .
Claim 7	
7[a]	The method according to claim 4, further comprising: clipping said modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels to be within an interval of zero and a defined maximum value by setting said modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels to be equal to zero if said

Reference	Claim Limitation
	modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels is smaller than zero and by setting said modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels to be equal to said defined maximum value if said modified pixel value p1' of said pixel next closest to said block boundary in said line of pixels is larger than said defined maximum value; and
7[b]	clipping said modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels to be within an interval of zero and said defined maximum value by setting said modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels to be equal to zero if said modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels is smaller than zero and by setting said modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels to be equal to said defined maximum if said modified pixel value q1' of said pixel next closest to said block boundary in said corresponding line of pixels is larger than said defined maximum value.
Claim 10	
10[pre]	A user equipment comprising a deblocking filter unit for reducing blocking artifacts at a block boundary between a block of multiple pixels and a neighboring block of multiple pixels in a video frame, wherein the deblocking filter unit comprises:
10[a]	a first offset calculator configured to calculate a first offset to be equal to $(9 \times (q_0 - p_0) - 3 \times q_1 - p_1) + 8 \gg 4$, wherein $\gg$ denotes a right shift operation, p0 denotes a pixel value of a pixel closest to said block boundary in a line of pixels in said block, wherein said line of pixels is perpendicular to said block boundary, p1 denotes a pixel value of a pixel next closest to said block boundary in said line of pixels, q0 denotes a pixel value of a pixel closest to said block boundary in a corresponding line of pixels in said neighboring block, wherein said corresponding line of pixels is perpendicular to said block boundary, and q1

Reference	Claim Limitation
	denotes a pixel value of a pixel next closest to said block boundary in said corresponding line of pixels; and
10[b]	a pixel value modifier configured to modify said pixel value p_0 of said pixel closest to said block boundary in said line of pixels by adding said first offset to said pixel value p_0 of said pixel closest to said block boundary in said line of pixels to form a modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels
10[c]	and to modify said pixel value q_0 of said pixel value q_0 of said pixel closest to said block boundary in said corresponding line of pixels to form a modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels.
Claim 11	
11[a]	The user equipment according to claim 10, wherein the deblocking filter unit further comprises a first clipping unit configured to clip said first offset within an interval of $-t_c$ and t_c by setting a first offset to be equal to $-t_c$ if said first offset is smaller than $-t_c$ and by setting a first offset to be equal to t_c if said first offset is larger than t_c .
11[b]	wherein t_c is a threshold value that is dependent on a quantization parameter value assigned to said block.
Claim 12	
12[a]	The user equipment according to claim 10, wherein the deblocking filter unit further comprises a second clipping unit configured to clip said modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels to be within an interval of zero and a defined maximum value by setting said modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels to be equal to zero if said modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels is smaller than zero and by setting said modified pixel value p_0' of said pixel closest to said block boundary in said line of pixels to be equal to said defined maximum value if said modified pixel value of p_0' of said pixel closest to said block boundary in said line of pixels is larger than said defined maximum value and

Reference	Claim Limitation
12[b]	to clip said modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels to be within an interval of zero and said defined maximum value by setting said modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels to be equal to zero if said modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels is smaller than zero and by setting said modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels to be equal to said defined maximum if said modified pixel value q_0' of said pixel closest to said block boundary in said corresponding line of pixels is larger than said defined maximum value.

Table of Exhibits

Exhibit No.	Description
Ex-2002	CV of Dr. Iain Richardson
Ex-2003	ITU-T Rec. H.261, first published in 1988, https://www.itu.int/rec/T-REC-H.261-198811-S/en .)H.261
Ex-2004	FiltScript2.m
Ex-2005	TestFilt2.m
Ex-2006	FiltBlocksAltThr 2a.txt
Ex-2007	FiltBlocksAltThr 2b.txt
Ex-2008	FiltBlocksThr1Thr2_2a.txt
Ex-2009	FiltBlocksThr1Thr2_2a.txt
Ex-2010	FiltStatsAltThr 2a.txt
Ex-2011	FiltStatsAltThr 2b.txt
Ex-2012	FiltStatsThr1Thr2_2a.txt
Ex-2013	FiltStatsThr1Thr2_2b.txt
Ex-2014	TestUint8.c
Ex-2015	PCT/FI01/00049 (“Aksu”)
Ex-2016	Sullivan et al., <i>HEVC: Algorithms and Architectures</i> (2014)

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. 10,142,659 (“the ’659 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 Petitioner and/or opinions rendered by Petitioner’s experts, or in light of any relevant orders from the Board.

3. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials.

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 Ex-2002 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. I have reviewed the '659 patent, including the challenged claims, prosecution history, and relevant patent family. The '659 Patent was marked as Ex-1001 and its prosecution history was marked as Ex-1002.

17. I also reviewed the Petition, and each of Exhibits 1001-1133 attached to the Petition.

18. In forming my opinions, I have considered the materials listed above and any other documents cited in this declaration. I have also relied on my own education, knowledge, and experience in the relevant art.

19. I have also considered the understanding of a person of ordinary skill around the time of the invention of the '659 Patent.

IV. SCOPE OF OPINIONS

20. I set forth my opinions throughout this declaration.

21. It is my opinion that the challenged claims of the '659 Patent are not obvious in view of the asserted references, alone or in combination, discussed herein.

22. I understand that while Petitioner does not necessarily combine each of the asserted references, Petitioner argues that the challenged claims are obvious over the primary references Fu (Ground 1) and Bjontegaard (Ground 2) "in view of a POSITA'S understanding of the art," as evidenced by List (Ex-1024), B205 (Ex-1018), B118 (Ex-1023), and, for Ground 2 only, Bjontegaard's Code (Exs-1010-1015).

V. LEGAL STANDARDS

23. 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

24. 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).

25. 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, 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. Obviousness

26. I 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.

27. 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.

28. I understand that if a reference or proposed combination of references does not disclose 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.

29. 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. I understand, however, that routine experimentation does not necessarily preclude patentability.

30. 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.

C. Motivations to Combine

31. 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.

D. Objective Indicia of Nonobviousness

32. 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

33. 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.

34. 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 dates of the Asserted Patents under this definition.

35. 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 ’659 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 or image coding/decoding and processing could substitute for working experience in the field.” Ex-1003, ¶ 227.

36. I disagree with Dr. Schonfeld’s opinion because in my opinion, two years of experience in video decoding or encoding technologies is an appropriate level of experience in the context of the Asserted Patents. 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.

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

VII. CLAIM CONSTRUCTION

38. I understand that unless claim terms are provided an express construction, the terms must be given their plain and ordinary meaning.

39. I also understand that Petitioner does not propose any constructions for any claim term. Ex-1003, ¶ 230.

40. I also understand that in separate ITC proceedings, Lenovo and Patent Owner dispute whether the preamble of claim 10 is limiting in its entirety. I understand that Patent Owner proposes that the entire preamble of claim 10—“[a] user equipment comprising a deblocking filter unit for reducing blocking artifacts at a block boundary

between a block of multiple pixels and a neighboring block of multiple pixels in a video frame...” is limiting. Ex-1003, ¶¶ 229-234. Petitioner contends that only the phrase “user equipment comprising” is limiting. Petition, 12.

41. I do not offer an opinion on the correct construction of any disputed terms, because, in my opinion, the Challenged Claims are not taught or suggested by the prior art. I reserve the right to respond to any future claim construction argument asserted by Petitioner or Dr. Schonfeld.

42. I also understand that Patent Owner and Lenovo agreed that the term “configured to” as used in Challenged Claims 10 through 12 should be afforded its plain and ordinary meaning—i.e., “programmed to, as opposed to being merely capable of, performing a task.” Ex-1026, 4.

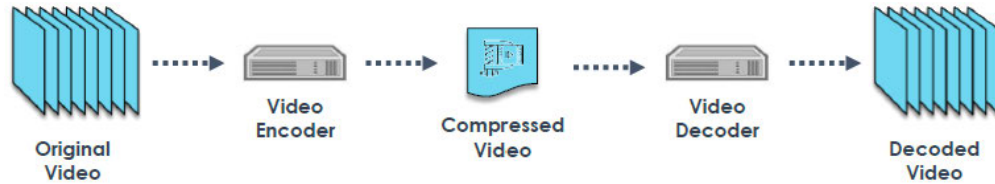
VIII. TECHNOLOGY BACKGROUND

43. 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

44. 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:



45. 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.

46. 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, a 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.

47. 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.

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

49. 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.

50. 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).

51. 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.

52. 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.

53. 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.

54. 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.

55. 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.

56. 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.

57. 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.

58. 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

59. Current video codecs, including H.265, utilize hybrid coding schemes. Ex-2016 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 based on pictures 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:

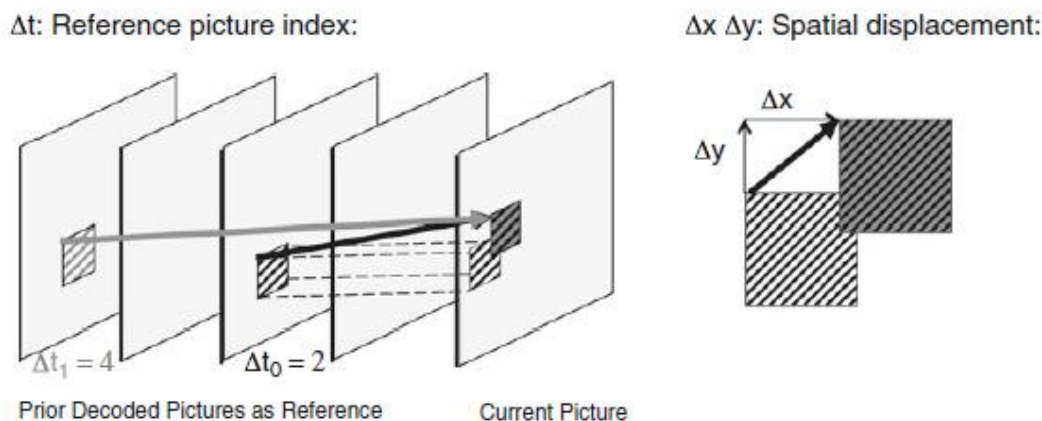


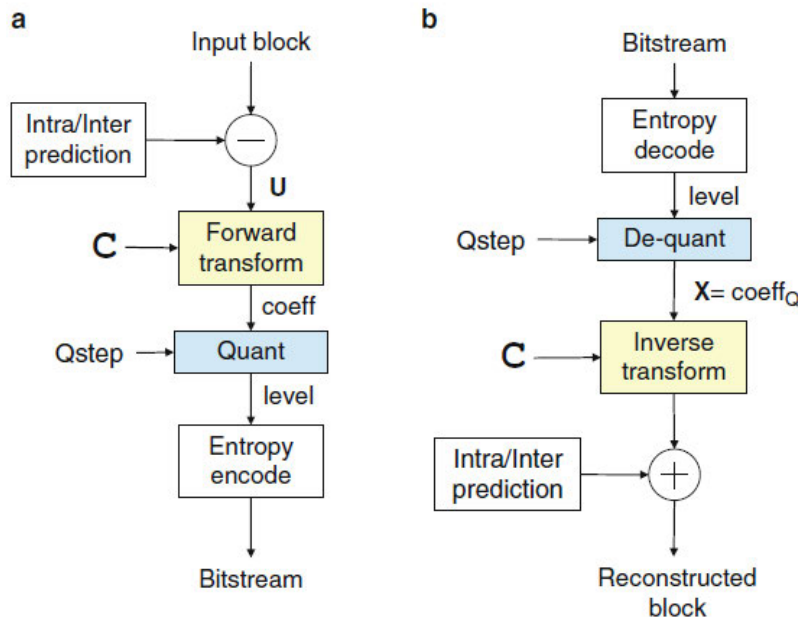
Fig. 5.1 Inter-picture prediction concept and parameters using a translational motion model

Id.

60. 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.

61. 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:



Id. at 142.

62. 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.

63. In H.265/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. Ex-2016.

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

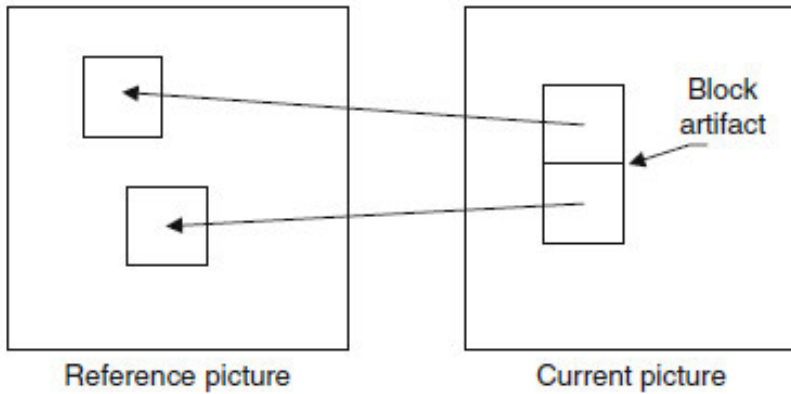
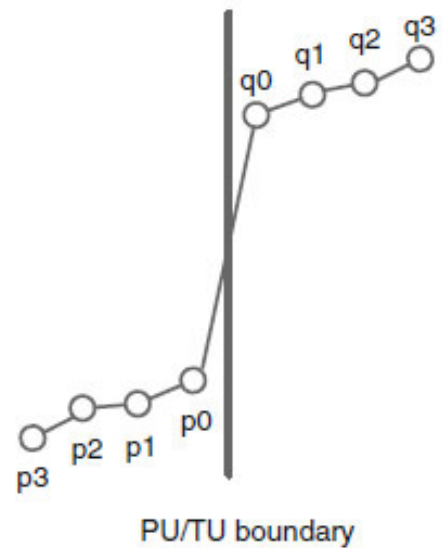


Fig. 7.3 Example of block artifact in one dimension [37]



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



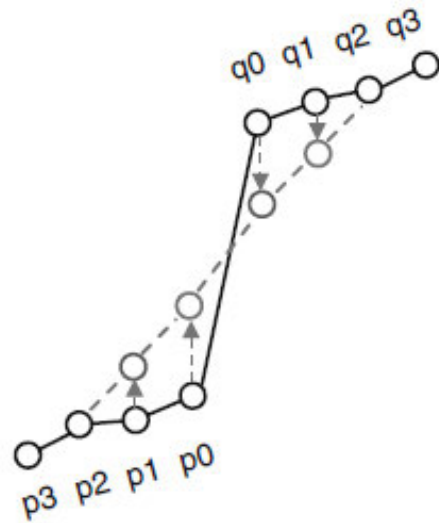
Ex-1003, ¶ 44.

65. 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-200N 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 analyses the values of reconstructed samples on the sides of the block boundary.” *Id.*

66. 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.

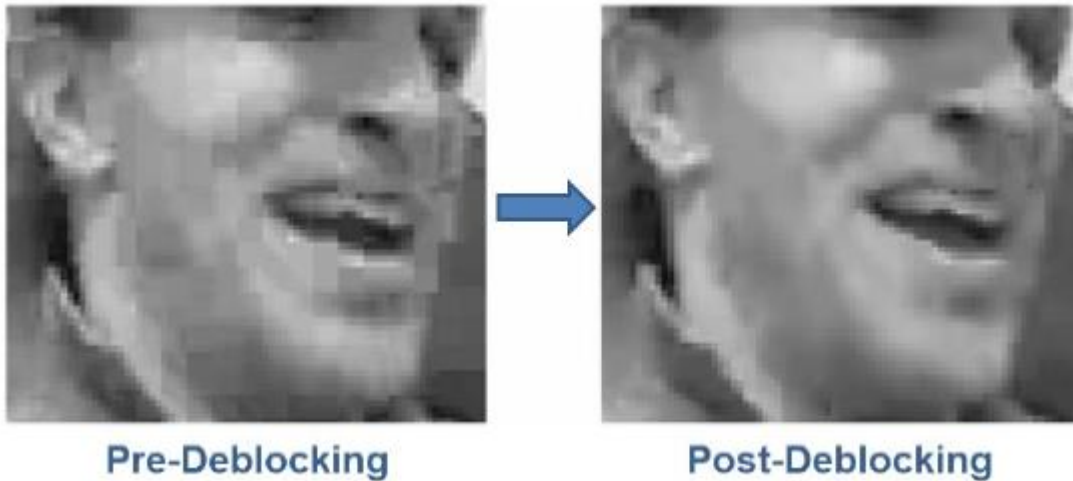
Fig. 7.9 Illustration of normal filtering mode operations: original block boundary (solid black line) and modified block boundary (dashed gray line)



Ex-2016 at 181.

67. 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. 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.



Petition, 5.

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 of 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 '659 Patent. Ex-2003 (ITU-T Rec. H.261, first published in 1988, <https://www.itu.int/rec/T-REC-H.261-198811-S/en>).

IX. U.S. PATENT NO. 10,142,659 (“’659 Patent”) (Ex-1001)

A. Priority Date

70. The '659 Patent claims priority to U.S. provisional application No. 61/432,751, which has a filing date of January 14, 2011.

71. Petitioner does not challenge this priority date for purposes of the Petition, and I have assumed that each of the challenged claims is entitled to a priority date of January 14, 2011. Petition, 8.

B. '659 Patent Overview

72. United States Patent No. 10,142,659 discloses a “deblocking filter [which] has good low-pass characteristics and is efficient for reducing blocking artifact[s].” Ex-1001, Abstract. In particular, the 659 Patent “defines a method of reducing blocking artifacts at a block boundary between a block of pixels and a neighboring block of pixels in a video frame.” Ex-1001, 3:44-48. Illustrative examples include Figures 2A and 2B, which show a vertical block boundary (Fig. 2A) and a horizontal block boundary (Fig. 2B). Figures 2A and 2B further show the related naming convention for pixels within each block on a given side of the block boundary, with one side references as the “p” block and the other side of the boundary referenced as the “q” block. Ex-1001, Figs. 2A, 2B. In particular, each pixel is given a letter (i.e. “p” or “q”) based on which side of the boundary the pixel is on, a number referencing how many pixels are between a given pixel and the block boundary, and another subscript number referencing which row or column the pixel is in (based on whether the boundary is vertical or horizontal). For example, a pixel denoted by p_{02} in Figure 2A for a vertical boundary would reference the pixel within the left (“p” block), in the column 0 closest to the block boundary, and in the second row from the top.

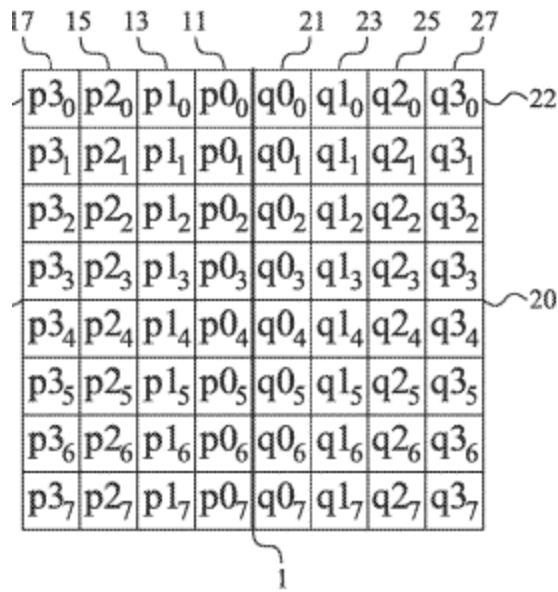


Fig. 2A

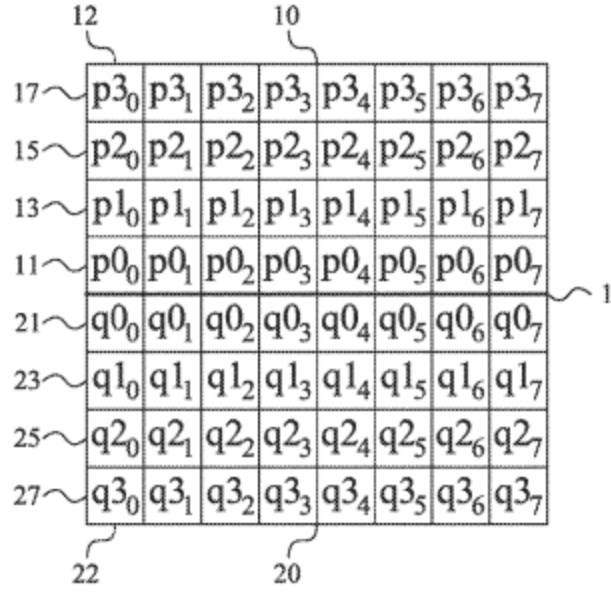


Fig. 2B

73. In draft versions of the HEVC standard which had been developed by the Joint Collaborative Team on Video coding (JCT-VC) before U.S. Application No. 15/639,229 (which became the 659 Patent) was filed, deblocking filtering involved modification of pixels near the block boundary, where a first offset equation (“ Δ ”) would be calculated to determine how much each pixel in a given row should be filtered. Ex-1001, 2:57-63. In particular, the proposed offset was calculated using the equation $\Delta = \text{Clip}(-t_c, t_c, (13 \times (q0_i - p0_i) + 4 \times (q1_i - p1_i) - 5 \times (q2_i - p2_i) + 16) \gg 5)$. *Id.* This calculated offset value, Δ , was then used to modify the “p” pixel and “q” pixel closest to the block boundary for that row, by adding the offset to the “p” pixel, and subtracting the offset from the “q” pixel. ’659 Patent at 2:63-67 ($p0_i = \text{Clip0-255}(p0_i + \Delta)$ and $q0_i = \text{Clip0-255}(q0_i - \Delta)$). Each modified pixel would then be clipped to make sure it was between the values of 0 and 255, which was the allowable range for an 8-bit pixel value. Ex-1001, 12:8-18.

74. The method of the then-draft version of the HEVC standard would next modify the pixels in the row of the “p” and “q” block next closest to the block boundary, i.e. p_1 and q_1 . The difference was, for these pixels, they would be modified using the calculated first offset divided by two, which was then either added or subtracted from the given pixel value. Ex-1001, 3:1-3 ($p_{1i} = \text{Clip}_{0-255}(p_{1i} + \Delta/2)$ and $q_{1i} = \text{Clip}_{0-255}(q_{1i} - \Delta/2)$). In this way, the pixel values p_1 and q_1 would always be modified by $\frac{1}{2}$ the amount of the calculated first offset, and the p_1 pixel would be modified by the exact same amount as the q_1 pixel. Ex-1001, 2:56-3:33 (“the offset for pixel p_{1i} is equal in magnitude to the offset of pixel q_{1i} , although the structure of the signal can be different on different sides of the block boundary.”). However, the inventors of the ’659 Patent realized that it was not necessarily optimal to modify pixel values by the same amount on both sides of the block boundary, as this filter would “not adapt well to the local structure of the side of the block boundary” and “cannot remove higher frequencies that can appear near the block boundary and cannot remove ringing.” Ex-1001, 3:18-24, 3:30-33.

75. The inventors of the ’659 Patent were involved in the development of H.265 Standard and recognized that they could solve these problems of the draft HEVC deblocking filter through a number of innovations. First, the ’659 Patent discloses a more efficient first offset calculation that is based on different pixels, and uses different weights associated with those pixel values. Ex-1001, 3:47-48 (“The method calculates a first

offset to be based on $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1)) / 16$). This improved first offset equation is then used to modify the pixel values closest to the block boundary by adding the calculated first offset to the nearest pixel value on one side, and subtracting the calculated first offset from the nearest pixel value on the other side of the block boundary. *Id.*, 3:55-63.

76. The inventors of the '659 Patent also disclosed several additional exemplary embodiments, for instance defining the first offset "so that the calculation of the first offset is efficiently performed in hardware. In such a case, it is generally preferred not to have any divisions and/or define the function so that the first offset will be an integer value. In an embodiment, $(X+8) \gg 4$ is used as an integer-expression of $X/16$, where $\gg$ denotes a right shift operation. Thus, in a particular embodiment step S1 calculates the first offset to be based on and preferably equal to $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$." Ex-1001, 12:50-60.

77. As noted above, hardware implementation of the '659 Patent deblocking filter could also include clipping the calculated first offset (Δ) where "Clip3(A, B, x) is defined as Clip3(A, B, x)=A if $x < A$, Clip3(A, B, x)=B if $x > B$ and Clip3(A, B, x)=x if $A \leq x \leq B$ and Clip(x) is defined as Clip(0, M, x)." *Id.*, 13:7-10. In this way, an exemplary embodiment of the 659 Patent deblocking filter discloses "the method further comprises clipping the first offset to be within an interval of $-t_C$ and t_C by setting the first offset to be equal to $-t_C$ if the first offset is smaller than $-t_C$ and by setting the first offset to be

equal to t_C if the first offset is larger than t_C , wherein t_C is a threshold value that is dependent on a quantization parameter value assigned to the block.” Ex-1001 4:1-7. Where the first offset is clipped, an exemplary embodiment shows how the clipped first offset is used to modify pixels p_0 and q_0 to create modified pixel values p_0' and q_0' :

$$\Delta = \text{Clip}3(-t_C, t_C, (9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) > 4)$$

$$p_0' = \text{Clip}(p_0 + \Delta)$$

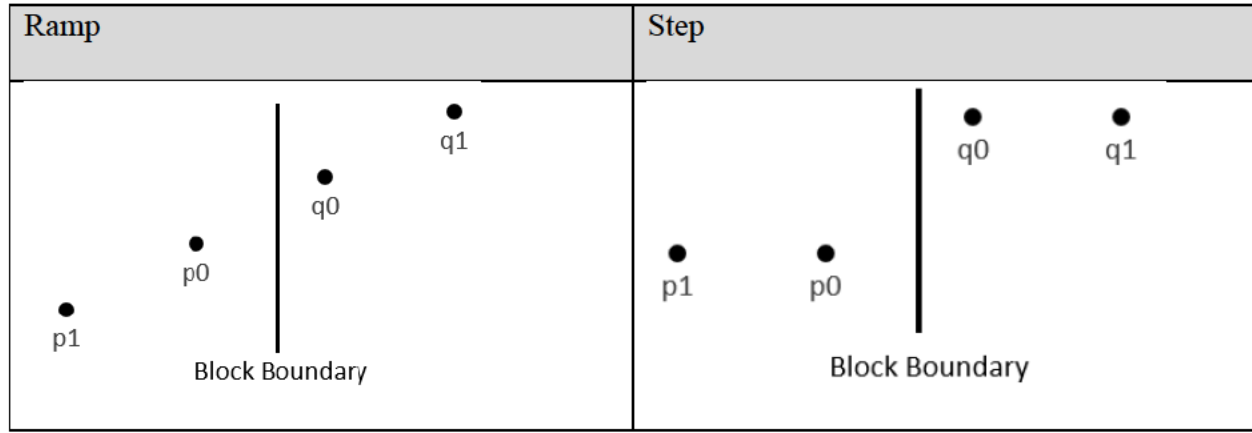
$$q_0' = \text{Clip}(q_0 - \Delta)$$

Ex-1001, 13:2-7.

78. The deblocking filter of the '659 Patent further discloses that another exemplary embodiment “generates a first offset for the pixels closest to the block boundary with a formula that produces approximately zero, preferably exactly zero, when applied to a ramp of pixel values and which produces an offset value that smooths a step in the pixel values when applied to a step of pixel values.” '659 Patent at 13:15-21. The specification of the '659 Patent further discloses specific evaluations of given pixel values, and how they are interpreted within the deblocking filter of the '659 Patent. Ex-1001, 13:15-34.

79. A “ramp” as referenced in the '659 Patent is a series of constantly increasing or decreasing pixel values. Ex-1001, 13:21-24. In contrast a “step” refers to a structure across a block boundary that occurs when there is a sudden or instantaneous change in the values of adjacent pixels from one pixel value that is constant to another. Id at 13:26-

27. A step can correspond to an increase or decrease in pixel values. *Id.* I illustrate these two structures relevant to the deblocking design of the '659 Patent below.



80. The '659 Patent specification also discloses a benefit that “[w]hen applied to a ramp signal, the deblocking filter will not modify it,” but “when applied to a step signal, i.e. a deblocking artifact, the deblocking filter will smooth it, i.e. reduce the blocking artifact. . . . [T]he proposed deblocking filter will smooth and thereby attenuate small ripples on the sides of the block boundary.” Ex-1001 at 13:61-14:3; *see also* 17:15-45. Thus, the deblocking design of the '659 Patent does not “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. This implies that the deblocking filter will have good lowpass characteristics and can remove high frequencies that can appear near the block boundary.” *Id.*, 8:57-63. The '659 Patent achieves this design goal by the particular expressions claimed as the first, second, and third offsets. *See, e.g.*, Ex-1001, cl. 1, 10, 4.

81. Another benefit of the '659 Patent is asymmetric filtering decisions for each side of the block boundary, such that the magnitude of an offset on one side of the block boundary is allowed to be different than the magnitude of the offset on the other side of the block boundary. Ex-1001, 4:20-26 (“the method further comprises calculating a second offset to be based on $(p_2+p_0-2\times p_1+2\times\Delta)/4$. . . The method also comprises, in this optional embodiment, calculating a third offset to be based on $(q_2+q_0-2\times q_1-2\times\Delta)/4$ ”). These particular improvements “enable[e] efficient reduction in blocking artifacts and provid[e] good low-pass characteristics.” Ex-1001, 8:5-7. The '659 Patent further discloses that “the proposed deblocking filter is, in this embodiment, a low-pass filter on all the filter positions p_0 , p_1 , q_0 , q_1 . When applied to a ramp signal, the deblocking filter will not modify it since Δ , Δp , Δq will all be equal to zero, wherein Δp denotes the second offset and Δq denotes the third offset. When applied to a step signal, i.e. a deblocking artifact, the deblocking filter will smooth it, i.e. reduce the blocking artifact. In clear contrast to the current HEVC filter, the proposed deblocking filter will smooth and thereby attenuate small ripples on the sides of the block boundary.” Ex-1001, 13:61-14:3.

82. The '659 Patent specification also discusses that “an advantage of this embodiment is that the offsets for the pixels second next closest to the block boundary can have different values that allows for a better local adaptation.” Ex-1001, 14:7-10. In this embodiment, “the second and third offsets can be clipped to be within an interval of

$-t_{C2}$ and t_{C2} , where the threshold value t_{C2} is determined based on the QP value assigned to the block. In a particular embodiment the threshold value t_{C2} is determined based on the threshold value t_C used for clipping the first offset in FIG. 3.” Ex-1001, 14:11-16. In this embodiment, the threshold value t_{c2} can be equal to $t_c/2$, or $t_{c2}=t_c \gg 1$ with a hardware adapted implementation. Ex-1001, 14:16-22.

83. As an alternative embodiment of the second and third offsets disclosed above, the '659 Patent specification discusses another embodiment where in which the second and third offsets can be “adapted for a hardware implementation and/or be an integer representation of $(p_2+p_0-2 \times p_1+2 \times \Delta)/4$ and $(q_2+q_0-2 \times q_1-2 \times \Delta)/4$. A particular example of such a hardware implementation that produces integer values is to calculate the second offset to be based on, preferably equal to, $((p_2+p_0+1) \gg 1) - p_1 + \Delta \gg 1$. Correspondingly, the third offset is preferably calculated to be based on, preferably equal to, $((q_2+q_0+1) \gg 1) - q_1 - \Delta \gg 1$.” *Id.*, 15:3-11. The '659 Patent specification goes on to disclose several embodiments for hardware implemented second and third offsets, as shown below:

In such a case, the modified pixel values are calculated as:

$$\Delta_p = (((p_2 + p_0 + 1) \gg 1) - p_1 + \Delta) \gg 1$$

$$p_1' = p_1 + \Delta_p$$

$$\Delta_q = (((q_2 + q_0 + 1) \gg 1) - q_1 + \Delta) \gg 1$$

$$q_1' = q_1 + \Delta_q$$

or if clipping is used:

$$\Delta_p = \text{Clip3}(-t_{C2}, t_{C2}, (((p_2 + p_0 + 1) \gg 1) - p_1 + \Delta) \gg 1)$$

$$p_1' = \text{Clip}(p_1 + \Delta_p)$$

$$\Delta_q = \text{Clip3}(-t_{C2}, t_{C2}, (((q_2 + q_0 + 1) \gg 1) - q_1 + \Delta) \gg 1)$$

$$q_1' = \text{Clip}(q_1 + \Delta_q)$$

Id., 15:12-27.

84. “In another [hardware implemented] embodiment, one can use a stronger low-pass filter in combination with the first offset. In this case the second offset is calculated based on, such as equal to, $(p_2 + p_0 - 2 \times p_1 + \Delta) / 2$ or, in a hardware-adapted implementation, based on, preferably equal to, $((p_2 + p_0 + 1 + \Delta) \gg 1) - p_1$ or alternatively $((q_2 + p_0 + \Delta - (p_1 \ll 1) + 1) \gg 1)$. The third offset could then be calculated based on, such as equal to, $(q_2 + q_0 - 2 \times q_1 - \Delta) / 2$ or, in a hardware-adapted implementation, based on, preferably equal to, $((q_2 + q_0 + 1 - \Delta) \gg 1) - q_1$ or alternatively $((q_2 + q_0 - \Delta - (q_1 \ll 1) + 1) \gg 1)$.” Ex-1001, 15:42-51.

85. The '659 Patent discusses another alternative embodiment in which the embodiment “calculates the first, second and third offsets for a line of pixels in the block to thereby modify the pixel values of both the pixels closest to the block boundary and the pixels next closest to the block boundary.” Ex-1001, 15:51-56.

86. It is also important to note that the '659 Patent was created to be implemented for HEVC/H.265, such that it would be capable of being implemented not just with HEVC/H.265, but the predecessor standard H.264 as well. '659 Patent at 17:1-15. Another important component of the deblocking filter of the 659 Patent is its ability to determine when a line of pixels should or should not be filtered. In particular, one embodiment discloses “[t]he first offset is calculated based on these pixel values and a formula that produces an offset value that approximates zero, preferably equals zero, when the pixel values are linearly increasing or decreasing or are the same when traveling along the line of pixels and the corresponding line of pixels and produces an offset value that smooths a step in the pixel values when the pixel values increase or decrease in a step when traveling along the line of pixels and the corresponding line of pixels.” Ex-1001, 17:15-45.

87. The '659 Patent discloses that the invention of the '659 Patent can be implemented in a physical deblocking filter unit, where sub-units can be “physically separate units 110, 120 and 210-290 in the device 100, 200, and all may be special purpose circuits, such as ASICs (Application Specific Integrated Circuits).” Ex-1001, 17:49-19:37. The '659 Patent also discloses implementing the invention of the '659 Patent within a computer, a user equipment, decoder, or computer program product. Ex-1001, 20:60-23:54. In particular, the '659 Patent can be implanted within a deblocking filter module of a computer. *Id.*, Fig. 10, 20:60-21:42. The deblocking filter module can

“essentially perform the steps of the flow diagram in FIG. 1 when run on the processing unit 72. Thus, when the different modules 310, 320 are run on the processing unit 72 they correspond to the corresponding units 110, 120 of FIG. 8 and units 210, 220 of FIG. 9.” ’659 Patent at 21:20-25. With the difference being that the deblocking filter module is implemented on a computer, and requires a processing unit 72 and code means 70. Ex-1001, Fig. 10, 20:60-21:42. The ’659 Patent also discloses that some or all of the sub-units of the physical deblocking filter unit 100 or 200 can be implemented in through one or more sub-modules within the deblocking filter modules of a computer, where “alternative embodiments of the device 100, 200 are possible where some or all of the units 110, 120 and 210-290 are implemented as computer program modules running on a general-purpose processor. Such an embodiment is disclosed in FIG. 10.” Ex-1001, 20:50-59.

88. The prior art asserted by Petitioner demonstrates that there are fundamentally different approaches to how to modify similarly positioned pixels. For example, prior art deblocking filters do not modify 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. The prior art shows 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, each of the cited references herein disclose deblocking filters that use different offsets for pixels p0 and q0 and thus remove and/or smooth visual artifacts differently to the invention of the '659 Patent.

89. The following provides examples of the different approaches to modification of pixels p0 and q0 disclosed in the art. 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	$(p2 + 2p1 + 2p0 + 2q0 + q1 + 4) \gg 3$	$(4(q0 - p0) + (p1 - q1) + 4) \gg 3$
q0	$(p1 + 2p0 + 2q0 + 2q1 + q2 + 4) \gg 3$	

Ex-1024 at X; Ex-1018 (B205) at X.

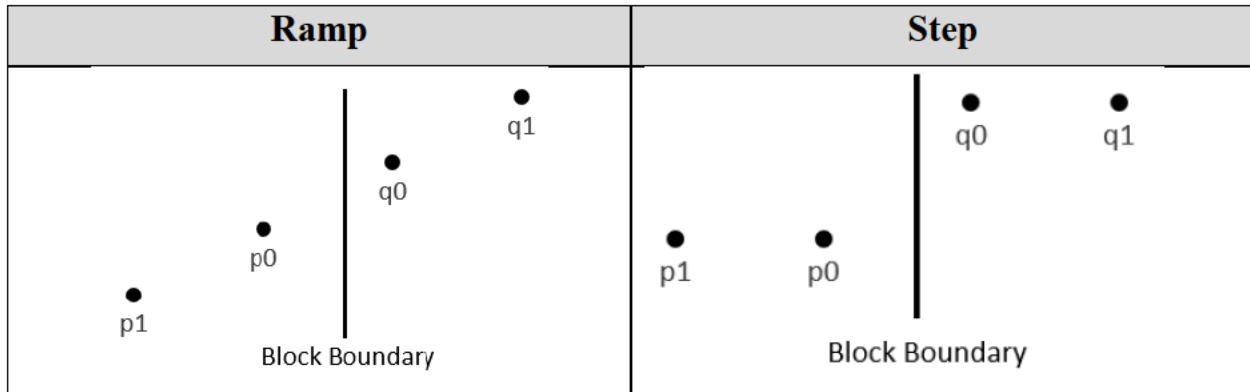
90. As shown here, one difference between the H.264 strong and weak filter is that the weak filter applies the same offset to p0 and q0, whereas the strong filter uses a different offset for p0 and q0. Further, the pixels values used in the weak filter offset are different than the pixels values used in the strong filter offset. For example, the weak filter does not use pixels p2 or q2, but the strong filter does. 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.

91. As another example, the HEVC Draft reflected in B205 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	$(4(q_0 - p_0) + (p_1 - q_1) + 4) \gg 3$	$(13(q_0 - p_0) + 4(q_1 - p_1) - 5(q_2 - p_2) + 16) \gg 5$

See Ex-1024; Ex-1018.

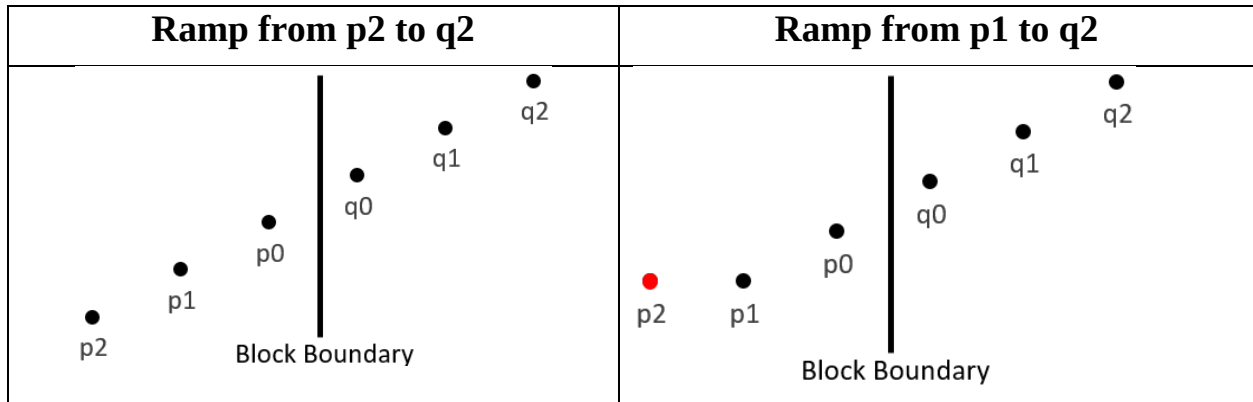
92. Specifically, the use of the offsets described in each prior art reference do not “modify the pixel values over a block boundary if these pixel values form a ramp but smooth the pixel values if they instead are in the form of a step. This implies that the deblocking filter will have good lowpass characteristics and can remove high frequencies that can appear near the block boundary.” Ex-1001, 8:57-63. As illustrated below, a ramp is a series of linearly increasing or decreasing pixels. Ex-1001, 13:21-24. A step occurs when there is a sudden or instantaneous change from one constant value to another in the values of adjacent pixels (i.e. a step increase or decrease). Ex-1001, 13:26-27.



93. The deblocking filter of the 659 Patent is designed such that “[w]hen applied to a ramp signal, the deblocking filter will not modify it,” but “when applied to a step signal, i.e. a deblocking artifact, the deblocking filter will smooth it, i.e. reduce the blocking artifact. . . . [T]he proposed deblocking filter will smooth and thereby attenuate small ripples on the sides of the block boundary.” Ex-1001, 13:61-14:3;17:15-45.

94. To achieve this design, the 659 Patent describes utilizing a first offset based on $9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) / 16$. Ex-1001, 3:47-48, 12:18-28. The ’659 Patent explains that, when this offset is calculated in hardware, the offset is preferably equal to $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$.¹ *Id.*, 3:65-67, 12:51-69. 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.*, 3:55-63. The pixel values are modified by physically altering the pixels of a video frame, thereby attenuating blocking artifacts. *Id.* An important aspect of this design is that 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:

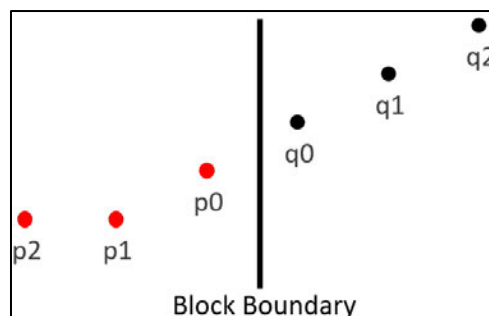
¹ The 659 Patent describes that $(X+8) \gg 4$ is used as an integer-expression of $X/16$, i.e., a way of implementing $X/16$ using integer arithmetic. Ex-1001, 12:55-60.



95. The figure on the left includes a ramp that extends from pixel p2 through to q2, while the ramp in the figure on the right only extends from pixels p1 through to q2. The extent of the ramp in the figure on the right is more limited because the transition from pixel p2 to p1 does not form a ramp structure. Even though the figures are different (and the structure that the pixel values form are different), the offset for p0 and q0 will be 0 in both examples because the first offset of the '659 Patent ignores the pixel p2.

96. The '659 Patent's second and third Offset provide localized modification of pixels p1 and q1 of a video frame. The '659 Patent also provides an alternative embodiment where pixels p1 and q1 can be modified by the deblocking filter. The prior HEVC draft described in B205 "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." *Id.* at 3:24-33. The '659 Patent also criticizes the HEVC Draft for not allowing offsets of different magnitudes for different sides of the block boundary. *Id.*

97. The '659 Patent discloses that the filters for the coefficients at the second position from the block boundary (p1, q1) are produced by using a combination of the first offset and a symmetrical low-pass filter. An advantage of this embodiment is that the offsets for the pixels second next closest to the block boundary can have different values allowing for a better local adaptation. Ex-1001, 14:4-10. Specifically, the 659 Patent discloses that the pixel p1 should be modified by a second offset calculated based on $(p_2 + p_0 - 2 \times p_1 + 2 \times \Delta) / 4$, and the pixel q1 should be modified by a third offset calculated based on $(q_2 + q_0 - 2 \times q_1 - 2 \times \Delta) / 4$. See, e.g., Ex-1001, 14:38-47. 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. *Id.*, 8:5-7, 14:4-10 (“An advantage of this embodiment is that the offsets for the pixels second next closest to the block boundary can have different values that allows for a better local adaptation.”). When considering the illustration 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.



98. Specifically, the second offset, 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 a ramp. Thus, the '659 Patent provides for a specific deblocking filter that accounts for localized structures on each side of the block, does not filter ramps within the localized structure, but smooths steps.

C. The '659 Patent's Prosecution History

99. I have reviewed the prosecution history of the '659 Patent. Because the rejections and responses to rejections made during the prosecution history were unrelated to the prior art asserted here, in my view the prosecution history does not affect my opinions.

100. The Examiner filed a non-final rejection on January 26, 2018. This rejection noted that the Examiner was rejecting the '659 Patent based on nonstatutory double-patenting in view of U.S. Patent No. 8,526,509, which is in the same patent family as the '659 Patent. Ex-1002 at 93-94. The Examiner also rejected claims 1-18 under 35 U.S.C. § 101, alleging the claims were directed to non-statutory subject matter, but also noted that the claims would otherwise be allowed. Ex-1002 at 94-96. In particular, the examiner noted that the closest identified prior art (Hsu (Ex-1039) and Ugur (Ex-1059)) failed to anticipate or render obvious the claimed limitations. *Id.*

101. In addition to certain minimal claim amendments to fix typographical errors, the Applicant filed a terminal disclaimer to overcome the non-statutory double patenting rejection and a notice of allowance subsequently followed. Ex-1002 at 121-123.

X. SIMULATION OF DEBLOCKING FILTER DESIGNS

A. Testing Methodology

102. As I explain above, in his opinions with respect to the '659 Patent, Dr. Schonfeld relies on "Fu" (Ex-1022) as the primary reference for Ground 1, and "Bjontegaard" (Ex-1009) and "Bjontegaard's Code" (Exs-1010-1015) as primary references for Ground 2. Petition, 4.

103. To illustrate, analyze, and compare certain aspects of Bjontegaard/Bjontegaard's Code and the deblocking filter design claimed in the '659 Patent, I created a test simulation based on pseudo-code and description provided in the specification and claims of the '659 Patent and a test simulation that is based on the deblocking filtering design disclosed in Bjontegaard (Q15-J-28, Ex-1009). 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 '659 Patent and the references that Dr. Schonfeld relies on for his opinions, including Bjontegaard, Bjontegaard's Code, H.264, Fu, B118, VCEG-M21, H.241, 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 '659 Patent

compared with (for example) Bjontegaard, that in my opinion would not have been obvious to a POSITA as a part of routine experimentation.

104. 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 (e.g., Q15-J-28, Ex-1009 and Bjontegaard's Code, Ex-1010-1015) for Ground 2. TestFilt2.m implements deblocking filter processes disclosed in the '659 Patent. Ex-2004 – Ex-2014 attached to the Patent Owner's Preliminary Response includes FiltScript2.m and TestFilt2.m and corresponding results files described below.

105. I implemented portions of Bjontegaard, Bjontegaard's Code and the claims of the '659 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.

106. 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-1011-Ex-

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 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 ("Bjontegaard") and Bjontegaard's Code). I therefore determined it would be more efficient and effective to implement certain portions of Bjontegaard (Q15-J-28), Bjontegaard's Code (Ex-1011-1015) and the '659 Patent 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.

107. 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 the lcodec4.0 software, which I discuss below.

B. Matlab Scripts

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

109. 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 '659 Patent.

110. 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 '659 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 '659 Patent, without the use of certain

clipping operations, and (d) a third filtered version of each pair of blocks, based on the disclosure of '659 Patent and including certain clipping operations, as explained in the tables below.

111. 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 '659 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)
'659 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]

112. 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 '659 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 ’659 Patent (the “’659 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 ’659 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 (841/659). 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 '659 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 '659 Patent, the "'659 method". Set up (a) the Beta table (Table 1, '659 Column 16) and (b) the Tc and Tc2 arrays ('659 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 '659 patent and claim 4 of the '659 patent.
292-296	Filter pixels p0 and q0 as per '659 patent claim 4 and '659 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 '659 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 '659 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.

113. 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 '659 patents as set forth in detail below.

114. 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.

115. 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
...		...	

116. 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.

117. 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.

118. 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 $QP+6$.

119. Next, I derived the QP_{mod} that provided an equivalent Quantizer Step Size for the methods of the '659 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)

120. 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 '659 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.

121. I compare the behaviour of the Q15-J-28 filtering method and the '659 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 '659 Patent method is calculated by calculating thresholds Thr1 and Thr2 based on Beta, '659 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

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

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

124. Block 1:

125. Input blocks:

126. 110 102 94 86 | 79 72 66 60

127. 111 103 99 90 | 85 79 73 65

128. 109 105 102 95 | 90 86 82 74

129. 106 98 94 86 | 80 77 71 66

130. 105 100 96 87 | 83 80 75 68

131. 107 99 93 89 | 82 75 67 58

132. 108 100 93 87 | 78 72 64 59

133. 109 101 94 89 | 82 73 67 62

134. 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.

135. The difference between Block 1 and the version filtered using the simulated method of the Patents (without the Clip functionality of the '659 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

136. 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

137. 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.

138. 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 '659 method, without the clipping functionality of '659 claims 2 and 6.

1. *Using Thr1/Thr2 from the '659 patent and using the Q15-J-28 disclosed method for calculating Diff values:*

139. 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.

140. 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.

141. We can compare how often the '659 Patent's method and the Q15-J-28 method result in the **same** or **different** filtering choices. Consider for example the filtering of pixel $p1$ ('659 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 '659 method also adjusts $p1$ by -1), or by a different amount (e.g. Q15-J-28 adjusts c by -1 and the '659 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

142. 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 '659 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 '659 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

143. 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

144. 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

2. *Using Thr1/Thr2 from the '659 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

3. *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

4. *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

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

146. For the pixels which were modified by one or both of the simulated Q15-J-28 and '659 Patent's methods during filtering:

- Pixels p0 ('659 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 ('659 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.

147. 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 increase 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.

XI. ASSERTED REFERENCES

A. List

148. List is an article titled *Adaptive Deblocking Filter*, published by the Institute of Electrical and Electronics Engineers ("IEEE") in July 2003. List is cited on the face of the '659 Patent as describing the context of H.264 deblocking, which the 659 Patent explicitly improves upon. Ex-1001 at 2:14-16. List describes the same spatial activity condition on either side of the block boundary for filtering decisions that are disclosed in the prior generation H.264 standard. List specifically describes that the boundary strength

parameter, Bs, which determines the strength of the filtering applied on the boundary and can take on an integer value 1 to 4. Ex. 1024 at 615. List defines two filtering modes that apply based on the Bs parameter for a set of pixels. Further, List discloses a basic filtering operation for luma pixels where p_0' and q_0' are calculated using the Δ value provided below.

$$\Delta_{0i} = (4(q_0 - p_0) + (p_1 - q_1) + 4) \gg 3.$$

$$\begin{aligned} p_0' &= p_0 + \Delta_0 \text{ and} \\ q_0' &= q_0 - \Delta_0 \end{aligned}$$

149. List discloses that the two equations below are used to determine the extent of the filtering for luma pixel samples. In particular, pixel p_1 is modified if the corresponding (8) condition is true and pixel q_1 is modified if the corresponding condition (9) is true:

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

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

Ex-1024 at 617. List discloses the filtered value of p_1' and q_1' calculated as set forth below using Δ_{p1} and Δ_{q1} , respectively. The value of Δ_{q1} is obtained by substituting q_2 and q_1 for p_2 and p_1 in the equation below:

$$\Delta_{p1i} = (p_2 + ((p_0 + q_0 + 1) \gg 1) - 2p_1) \gg 1.$$

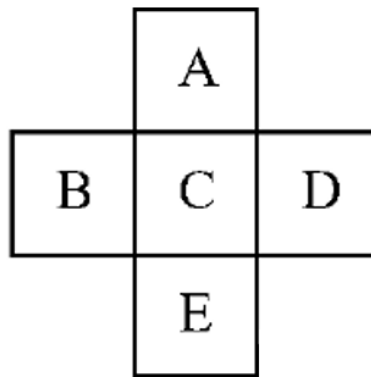
$$p_1' = p_1 + \Delta_{p1}.$$

$$q'_1 = q_1 + \Delta_{q1}.$$

Id.

B. Bjontegaard

150. 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 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.

151. The Strength value is determined for each of the above exemplary blocks and begins with an initially assigned 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 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.*

152. In the scenario where Strength1 and Strength2 are both non-zero, the filtering takes place and the following parameters are calculated:

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

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

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

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

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

Ex-1009 at 19. These values may then be used to calculate the value of parameters Act₁, Act₁₁, Act₂, Act₂₂, 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)x2ⁱ. Next, the activity on each side is calculated as Act₁ and Act₂ 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.

153. 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.

154. 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:

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

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

155. 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 and Dr. Schonfeld’s analysis of Bjontegaard suffers the same shortcoming of selecting one offset (*e.g.*, $Diff = 3(c - f) - 8(d - e)/16$) 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 could be calculated three different ways based on the value of Act_{edge} as shown below, and Petitioner fails to address why or how often $Diff = (3(c - f) - 8(d - e))/16$ 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)

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

157. 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: $\text{Diff} = 3(a + b - 4c + d + e)/16$

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

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

Ex-1009 at 20.

158. 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 $\text{lim}(qp)$, 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 $\lim(qp)$, and $Act_{overall}$, 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 $Act_{overall}$ is less than Act_1 and/or Act_2 .

159. 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 $c_{modified}$ as shown below:

Initially set Diff = 0

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

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

$Diff = \max(-Clip(qp, Strength_2), \min(Clip(qp, Strength_2), Diff))$

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

160. 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 $f_{modified}$. 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.

C. Bjontegaard's Code

161. In his opinions with respect to the '659 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-2015)

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

162. 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 algorithm 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. .

163. 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, $i0=4*(img->li[2]-img->li[5])-12*(img->li[3]-img->li[4])$, $i1=9*(img->li[2]+img->li[5]-img->li[3]-img->li[4])$. 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 $iaa=abs(i0)+abs(i1)$. 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     }

```

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++;

```

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.

164. 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.

165. 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      ill=img->li[3]+id;
484      irl=img->li[4]-id;
485      img->li[3]=ill; /* set back left edge */
486      img->li[4]=irl; /* set back right edge */

```

Ex-1012, lines 475-486. 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.

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

C

```

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;

```

167. 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.

168. 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 .

169. 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 equations for $S0_1$ in Section 4.5.6.1 of Bjontegaard is different to 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.

170. 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+Diff in Bjontegaard) and “redefine[d]” prior to reusing li[3] in the calculation of c+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+Diff. *Id.*

171. 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).

172. 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.

173. 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)

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

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

D. B118

176. The B118 reference is the Working Draft Number 2 (WD-2) released by the Joint Video (JVT) that described the draft standard developed for H.26L, which

eventually became H.264 Ex-1023, 1. B118 was generated on March 15, 2002. *Id.* However, the deblocking design described in B118 was eventually rejected in favor of the deblocking design that was included in the published version of H.264, which is described in List. Ex-1024. 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, 146.

177. The B118 deblocking filter first determines the boundary strength Bs for the block boundary between two neighboring blocks. Ex-1023, 147.

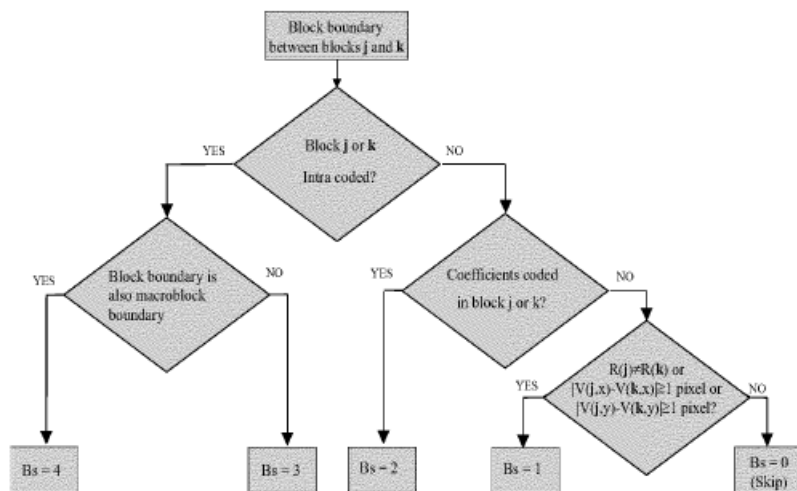


FIGURE 14

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

Ex-1023, 147.

178. 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 different naming convention

than the '659 Patent, with the pixels closest to the block boundary denoted p_1 and q_1 , rather than p_0 and q_0 .

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

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

Ex-1023, 147.

179. Based on these boundary strength Bs calculations, one of two types of filtering is performed – filtering for the default case, or stronger filtering for intra coded blocks.

Ex-1023, 148. 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 p_1 and q_1

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

180. 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 p_2 will be filtered and again by one if q_2 will be filtered. *Id.* C0 is defined using the equation $C0 = \text{ClipTable}[QP, Bs]$ (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	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	5	5	6	6	6	8	8	9	10

181. Once the Δ is clipped within a range of -C to C, it is either added or subtracted to the pixel values p1 and q1, to create modified pixel values P1 and Q1:

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

182. The B118 deblocking filter will then determine two intermediate values a_p and a_q , that are used to determine whether to filter p2 and q2. If it is determined to filter p2 and q2, these intermediate values 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 p2 and q2 are filtered. These pixels are only processed for luminance.

If (Luma && $a_p < \beta$) p2 is filtered by:

$$P_2 = p_2 + \text{Clip}(-C0, C0, (p_3 + P_1 - p_2 < 1) > 1)$$

If (Luma && $a_q < \beta$) q2 is filtered by:

$$Q_2 = q_2 + \text{Clip}(-C0, C0, (q_3 + Q_1 - q_2 < 1) > 1)$$

183. 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:

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

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

184.

E. VCEG-M21

185. 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.

F. Bjontegaard '048

186. 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 '659 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 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.

G. Aksu

187. PCT/FI01/00049 (“Aksu”) is an International Patent Application filed on January 22, and published on July 26, 2001. While Petitioner does not assert Aksu as a reference against the '659 Patent, Aksu is notably absent from Petitioner and Dr. Schonfeld’s description of prior art deblocking filters. In my view, Aksu is an example of an approach to deblocking filtering that undermine Petitioner and Dr. Schonfeld’s argument that the offsets in the prior art took on a similar form with the same essential components. This is because Aksu discloses an averaging filter. Ex-2016 at 15.

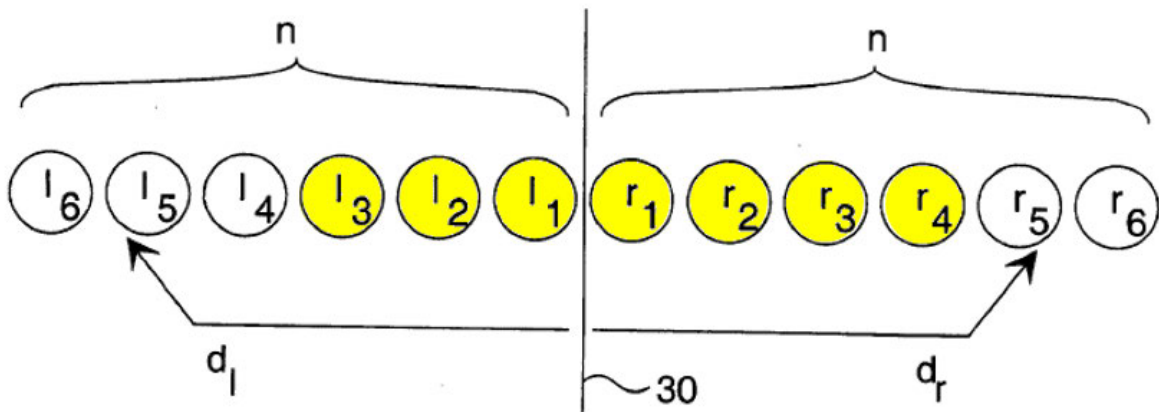
188. Aksu is directed to a “method and associated device for filtering digital video images, and discloses that it relates to a method for reducing visual artefacts in a frame of a digital video signal, which is coded by blocks and then decoded. Ex-1019 at 1:1-5. Aksu references prior art video transmission 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 the design included in the published H.264 Standard, described in List. Ex-1024.

189. Deblocking filtering is described using Table 2, wherein 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-2016 at 15:15-24.

$d_r (d_l > 1)$	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-2016, Table 2.

190. Based on the foregoing disclosure of the design in Aksu, I set forth the following example of how the deblocking filter is calculated in Aksu. Aksu discloses an averaging filter that is based on a sliding filtering window. The pixels in the filtering window of Aksu include zero to seven possible pixels, centering on the pixel to be filtered. Table 2 above specifies the pixels in the filtering window based on the value of d_r or d_l . For example, if r_1 is the pixel to be filtered and the filtering window includes three pixels on either side of r_1 , then filtering window used to calculate the offset is shown below.



191. The offset applied to r_1 is given by:

$$\frac{l_3 + l_2 + l_1 + r_1 + r_2 + r_3 + r_4}{7}$$

192. Ex-2001 at Fig. 2, 15:2-15. The possible filters that can be applied to r_1 (which corresponds to p_0) are described as follows:

	Offset to Modify r_1
3-pixel window	$\frac{l_1 + r_1 + r_2}{3}$
5-pixel window	$\frac{l_2 + l_1 + r_1 + r_2 + r_3}{5}$
7-pixel window	$\frac{l_3 + l_2 + l_1 + r_1 + r_2 + r_3 + r_4}{7}$

A. Fu

193. 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. 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” in “the field of image processing.” Ex-1022 at 1:12-13, 1:15-38; Ex-1003 ¶ 182. 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.

194. 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. 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:

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

195. Ex. 1022, 5:1-6. The filter applied can further vary such that either a strong filter or weak filter is used based on the value of a_p and a_q . *Id.* at 5:7-14. Process 2 of Fu describes filtering the pixels on both sides of the luma block boundary when boundary filtering strength is equal to 2. *Id.* at 7:20-21. In this process, only p_0 , p_1 , q_0 , and q_1 are filtered using the following offsets, wherein the filter applied can also vary based on the value of a_p and a_q as in Process 1:

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

Ex. 1022, 7:20-25.

196. Finally, Process 3 of Fu describes filtering pixels on both sides of the luma block boundary when BS is equal to 1. Ex-1022 at 8:3-4. In Fu's process 3, p_0 and q_0 are filtered and then whether there is a need to filter p_1 and q_1 is determined, which occurs when (i) the block boundary is a chroma block boundary (e.g., not a luma), (ii) p_1 is filtered because a_p is greater than 0, and (iii) q_1 is filtered because a_q is greater than 0. For cases (ii) and (iii), the filtered values of pixels p_1 and q_1 are given by:

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

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

Ex-1022 at 8:12-25. Notably, without explaining the basis for doing so, Dr. Schonfeld states that “[f]or purposes of his analysis, a BS of one is assumed,” and analyzes the Challenged Claims in view of Fu's Process 3. Ex-1003, ¶¶ 248-251. If other offsets were used as is possible in Fu, they would produce different filtering results.

B. B205

197. 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. B205 is mentioned in the Background of the '659 Patent as the prior HEVC draft standard that the inventions of the '659 Patent improve upon. Ex-1001, 2:19-22. The '659 Patent explicitly 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-1001 (659 Patent) at 3:18-33. The filter disclosed in the ’659 Patent explicitly improves upon B205 to address these shortcomings.

198. 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.

199. 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.

$$|p_{22} - 2*p_{12} + p_{02}| + |q_{22} - 2*q_{12} + q_{02}| + |p_{25} - 2*p_{15} + p_{05}| + |q_{25} - 2*q_{15} + q_{05}| < \beta$$

200. 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** $|p0_i - q0_i| < ((5 * t_c + 1) \gg 1),$

201. 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.

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

202. 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,

$\Delta = \text{Clip}(-t_c, t_c, (13 * (q0_i - p0_i) + 4 * (q1_i - p1_i) - 5 * (q2_i - p2_i) + 16) \gg 5);$	$i = 0, 7$
$p0_i = \text{Clip}_{0-255}(p0_i + \Delta)$	$i = 0, 7$
$q0_i = \text{Clip}_{0-255}(q0_i - \Delta)$	$i = 0, 7$
$p1_i = \text{Clip}_{0-255}(p1_i + \Delta/2)$	$i = 0, 7$
$q1_i = \text{Clip}_{0-255}(q1_i - \Delta/2)$	$i = 0, 7$

$$q2_0 = \text{Clip}_{0-255}((p0_i + q0_i + q1_i + 3*q2_i + 2*q3_i + 4) >> 3);$$

$$i = 0, 7$$

XII. THE ASSERTED CLAIMS OF THE '659 PATENT ARE NOT OBVIOUS

A. No Motivation to Combine Fu with “the knowledge of a POSITA as evidenced by List, Bjontegaard, B205 and B118”

203. I understand that Petitioner’s Ground 1 argument alleges that the Challenged Claims are Obvious in View of Fu and a POSITA’s knowledge as evidenced by List, Bjontegaard, B205, and B118. Petition, 12.

204. After reviewing the Petition and Dr. Schonfeld’s declaration, other than conclusory statements (*see, e.g.*, Petition, 38; Ex1003, ¶345), I do not see any articulated motivated to combine Fu with any or all of List, Bjontegaard, B205, and/or B118. *See* Petition, Ex-1003. Petitioner and Dr. Schonfeld instead state that a number of references could be considered together as “the knowledge of a POSITA.” Regardless of the Petition’s silence on a motivation to combine the prior art, it is my opinion that a POSITA would not be motivated to combine Fu with the other prior art.

205. Bjontegaard contemplate versions of a deblocking filter design proposed to H.26L in 2000, which was the evolution of the H.263 standardisation effort, and a subsequent version of which (excluding the Bjontegaard proposals) became the H.264 Standard. Fu is a Chinese patent application from several years later (2008) that discusses the then-current state of the published H.264 standard, which did not include the deblocking filter disclosed in Bjontegaard and/or Bjontegaard’s Code. The process of developing a standard such as H.264 is/was an in-depth, involved, iterative process,

whereby hundreds of engineers come together to evaluate ideas and proposals to create the best video coding standard possible at the time. As part of this process, certain proposals are adopted into the standard, and ultimately are integrated into the version of the standard published by the standardisation body. Other proposals, such as Bjontegaard and Bjontegaard's Code, are proposed, and may even be temporarily or partially adopted, before ultimately being rejected, with the standardisation body choosing to adopt a different approach. This happened with the deblocking filter design disclosed in Bjontegaard and Bjontegaard's Code. This is evidenced by the working draft JVT-B118 and the deblocking filter design that was ultimately adopted in the published H.264 Standard, which is very different from the deblocking design in Bjontegaard and Bjontegaard's Code, and does not include those approaches. Dr. Schonfeld has not provided an opinion as to why a POSITA would be motivated to combine a proposed deblocking filter from 2008, which takes as its starting point the then-current state of the H.264 Standard, with previous proposals to the H.264 development process that were rejected and not integrated into the final H.264 published standard. Further, Fu explicitly criticizes the deblocking design of H.264, and Bjontegaard and B118 reflect technical submissions to the H.264 Standardization that were rejected in favor of H.264's deblocking filter described in List. Moreover, the continual development of deblocking designs during standardisation where proposals are adopted and later rejected shows that deblocking designs are diverse and sensitive to changes, and persons of skill in the art

recognize that certain deblocking filter designs outperform other designs. Thus, in my view, Fu's criticism of H.264 would serve as teaching away from modifying Fu using references that were rejected by Fu.

206. I do not think a POSITA would have been motivated to combine these references in order to optimize or calibrate their design according to the '659 Patent. My Test Simulations, discussed herein, illustrate that differences in deblocking filter designs result in significant differences in filtering outcomes. Thus, I would expect the filtering outcome values to be very different if applying Fu's offset equation and Bjontegaard's, B118's, Lists, or B205's offset equation to the same set of pixels. Dr. Schonfeld provides no explanation, other than hindsight, as to why a POSITA would increase Bjontegaard's first coefficient from $8/16$ to $9/16$ but keep Bjontegaard's second coefficient at $3/16$ based on Fu's disclosed coefficients of $6/16$ and $2/16$, (Ex-1003, ¶¶ 488-490). As illustrated in my simulations, such an arbitrary change to filter coefficients would in fact have an impact on other factors such as the filter impulse response and the behavior of the filter, including the effect on visual quality and on the performance of inter prediction of further frames from the filtered pixel values. Moreover, Dr. Schonfeld states that such a change would likely have "expected slight differences in filtering performance" (Ex-1003, ¶ 282), but has provided no testing or analysis to support this statement. Ex-1003, ¶¶ 262-274, 281-292. In fact, Dr. Schonfeld's own examples illustrate that even an apparently slight change in filter coefficients can have a significant effect on filtering outcomes, for

example changing the outcome for the majority of Dr Schonfeld's example pixel rows (*Id.*, ¶ 489).

207. A POSITA would not have had a reasonable expectation of success in modifying Fu's or Bjontegaard's first coefficient of the first offset because in my view adjusting the coefficients in this way is certainly not "routine experimentation" as Dr. Schonfeld asserts. Ex-1003, ¶¶ 281-283. Changing the coefficients is not a mere edit of a number in a single line of code, but as explained above, changing the filter coefficients changes the frequency response of the filter and is a non-trivial change in operation that produces very different filtering outcomes, as illustrated by my simulation testing above and as illustrated by Dr. Schonfeld's own examples (*Id.*, ¶ 489). Petitioner has not provided any testing results to support such an argument.

208. Dr. Schonfeld's own summary of the normalized versions of the first offset equations from the different references show that the differences in the design of the first offset expression can vary in coefficients used as well as the pixels used. Ex-1003, ¶ 272. These slight differences are exacerbated by the additional differences in the filtering designs that are not limited to only the differences in offset equations. For example, as explained above, Fu relies on threshold values using absolute difference calculations in its "Process 3" to determine whether to filter the pixels next closest to the block boundary, whereas Bjontegaard uses three-pixel Laplacians in some instances, not to determine whether to filter the pixels next closest to the block boundary, but rather to calculate the

actual value of the offset applied to the pixels next closest to the block boundary. Ex-1009 at 19. These differences are exemplary, and as illustrated by my testing, the many factors considered as part of the overall filter design are sensitive to changes, and will result in significant difference in filtering outcome. *See* § X above.

209. In addition, in my opinion, a POSITA would not have been motivated to modify F_u in light of B205 for similar reasons—that is, the filtering approaches are fundamentally different. When comparing the offsets calculated by B205 versus F_u , it would have been apparent to a POSITA that the approaches are fundamentally different in choices of source pixel values, and the actual coefficients used, which would affect the frequency response and therefore the filtering results. Moreover, while F_u uses flatness to determine the boundary strength that is used to determine how to filter a pixel, B205 uses a different metric of slope. Thus a POSITA would, in my view, understand that these references teach away from each other because of their different approaches.

210. Further, all of the cited references asserted as the “basis” for a POSITA’s understanding of the art (e.g., List, Bjontegaard, B118, and JCTVC-B205) disclose different approaches to calculating the offsets applied to the pixels next closest to the block boundary (e.g., second and third offsets). For example, if each of the second offset expressions are normalized, as Dr. Schonfeld and Petitioner assert is necessary to compare the expressions, then the second offset expressions in the prior art compared to the ’659 Patent are given below:

	Second Offset
659 Patents	$\frac{64(p_2 - 2p_1 + p_0) + 72(q_0 - p_0) - 24(q_1 - p_1)}{256}$
Bjontegaard	$\frac{80(p_2 - 2p_1 + p_0) + 40(q_0 - p_0) - 15(q_1 - p_1)}{256}$
Fu	$\frac{32(p_2 - 3p_1 + 3p_0 - q_0) + 48(q_0 - p_0) - 16(q_1 - p_1)}{256}$
H.264/List	$\frac{128p_2 - 256p_1 + 64p_0 + 64q_0}{256}$
B205	$\frac{60(q_0 - p_0) + 16(q_1 - p_1) - 20(q_2 - p_2)}{256}$
B118	$\frac{128(p_2 - 2p_1 + p_0) + 64(q_0 - p_0) - 16(q_1 - p_1)}{256}$

211. I normalized these expressions by inserting the previously calculated Δ , or first offset expression where it appears in the second offset expression.

212. As shown by the table above for the second offset equations, not only are the coefficients (i.e., weights) assigned to each element substantively different among the different solutions provided by the prior art, but there is also no uniform, agreed-upon approach to which pixels are used, or what metric should be used to calculate the second offset. For example, certain expressions use equations that measure slope and absolute difference (e.g., B118, Fu, Bjontegaard), while others use equations that measure absolute difference only (e.g., B205, List). The differences between these offset expressions would have led to significantly different filtering results because the frequency response of the filter and the response to different types of structures in the blocks of pixels would necessarily be different.

213. Further, Bjontegaard proposed a deblocking filter design to H.26L in 2000, the development effort which subsequently led to the publication of the H.264 Standard. Ex-1009. Approximately two years later, the Joint Video Team (the standardization body that created the H.264 Standard) published JVTB118, which was a working draft of the then-current version of the H.264 Standard being developed. Ex-1023. Notably, Bjontegaard and B118 contain different offset equations for modifying the pixels closest and next closest to the block boundary. *Compare Ex-1023 with Ex-1009.* The 2003 version of the published H.264 Standard is a further evolution of the working draft published as B118, with identical first offset equations. However, the adopted H.264 Standard altered the second and third offset equations as compared to B118, removing the second and third offset equations' reliance on the previously calculated modified pixel values closest to the block boundary. *Compare Ex-1023 (B118) with Ex-1024 (List).*

214. Thus, Petitioner's arguments with respect to motivation to combine and obviousness are unsupported because they are based on disparate solutions in the prior art. It was not until the invention of the '659 Patent that anyone determined the specific, appropriate second and third offsets as claimed.

B. Petitioner's Ground 1 Fails Because The Challenged Claims Are Not Obvious In View Of Fu And A POSITA's Knowledge As Evidenced By List, Bjontegaard, B205, And B118

215. While I address certain limitations to demonstrate that each of the challenged claims are not unpatentable, I reserve the right to supplement this declaration to include

additional information and opinions related to any and all challenged claims and claim elements.

1. Independent Claims 1 and 10

- a. **1[a]/10[a]: calculating a first offset to be equal to $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$, wherein $\gg$ denotes a right shift operation, p_0 denotes a pixel value of a pixel closest to said block boundary in a line of pixels in said block, wherein said line of pixels is perpendicular to said block boundary, p_1 denotes a pixel value of a pixel next closest to said block boundary in said line of pixels, q_0 denotes a pixel value of a pixel closest to said block boundary in a corresponding line of pixels in said neighboring block, wherein said corresponding line of pixels is perpendicular to said block boundary and q_1 denotes a pixel value of a pixel next closest to said block boundary in said corresponding line of pixels;**

216. Dr. Schonfeld opines that Fu renders obvious this claim element, based in part on an equation for calculating a delta value, reproduced below. Ex-1003, ¶ 251.

$$\text{delta} = \text{Clip3} \left(-C, C, \left(((q_0 - p_0) \times 3 + (p_1 - q_1) + 4) \gg 3 \right) \right)$$

$$P_0 = \text{Clip1}(p_0 + \text{delta})$$

$$Q_0 = \text{Clip1}(q_0 - \text{delta})$$

217. Fu's first offset equation is demonstrably different than the '659 Patent's first offset equation. See, e.g., Ex-1003, ¶ 261.

Reference	First Offset Equation
'659 Patent	$(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$
Fu	$(6 \times (q_0 - p_0) - 2 \times (q_1 - p_1) + 8) \gg 4$

218. Specifically, Fu fails to disclose the particular invention of the '659 Patent filter design—that it does not “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 at 8:57-60. The '659 Patent explains that its deblocking filter will have good lowpass characteristics and can remove high frequencies that can appear near the block boundary.” 659 Patent, 8:57-63. Fu does not have any such disclosure about its particular response to certain structures in a block of pixels. Instead, Fu’s design is dependent on the region flatness generated by the underlying signal, which is used to determine how to filter the pixels.

219. Dr. Schonfeld states that a POSITA would have naturally been motivated to experiment with Fu’s coefficients to achieve a desired amount of filtering of blocking artifacts without also degrading the overall image quality. Ex-1003, ¶ 282; *see also id.*, ¶¶ 272-274 (comparing different coefficients in the prior art). I disagree with this proposition because, what Dr. Schonfeld fails to acknowledge is that the calculation of the first offset in Fu, and in many other references, would necessarily affect the overall filter design. For example, the first offset in Fu is used to calculate the modification of pixels p_1 and q_1 , and there is therefore, in my opinion no basis for an assertion that

modifying Fu's first offset as suggested would yield predictable results from the perspective of a POSITA.

220. I disagree. While many deblocking filters were known prior to the '659 Patent, none of those deblocking filters disclosed the '659 Patent's claimed first offset, despite over a decade of engineers working to adjust these deblocking filters, at least since Bjontegaard in 2000. Dr. Schonfeld has also provided no rationale why a POSITA would view the deblocking filter of Fu as incomplete, or be motivated to experiment with the disclosed deblocking filters. In fact, Fu and the other listed inventors did not identify or suggest anywhere on their patent application that the equations were incomplete, or required additional experimentation. Ex-1022. Dr. Schonfeld has also provided no explanation, other than the fact that other proposals were made in the prior art, as to why the '659 Patent's deblocking filter coefficients were known to be result-effective variables. Ex-1003, ¶ 290. Deblocking filters are extremely complicated, such that one variable cannot be predictably tweaked without affecting the rest of the filter response. I also disagree with Dr. Schonfeld that there was a known finite range of coefficients that could be used within the constraints required for a deblocking filter, that would result in discovering the coefficients of the '659 Patent through routine experimentation. Ex-1003, ¶¶ 274-281. In fact, none of the deblocking filters identified by Dr. Schonfeld use a coefficient value equal to or larger than the 9/16 value used with the '659 Patent's first offset equation:

Table 1: Condensed Offset Equation	
'659 Patent	$\frac{9(q_0 - p_0) - 3(q_1 - p_1)}{16}$
Fu	$\frac{6(q_0 - p_0) - 2(q_1 - p_1)}{16}$
H.264, List	$\frac{8(q_0 - p_0) - 2(q_1 - p_1)}{16}$
B118	$\frac{8(q_0 - p_0) - 2(q_1 - p_1)}{16}$
Bjontegaard	$\frac{8(q_0 - p_0) - 3(q_1 - p_1)}{16}$
VCEG-M21	$\frac{8(q_0 - p_0) - 2.5(q_1 - p_1)}{16}$
Bjontegaard '048	$\frac{8(q_0 - p_0) + 4(q_1 - p_1) - 4(q_2 - p_2)}{16}$
B205	$\frac{7.5(q_0 - p_0) + 2(q_1 - p_1) - 2.5(q_2 - p_2)}{16}$

Ex-1003, ¶¶ 272-273.

221. In my view, the prior art teaches away from modifying Fu to reach the exact expression of the '659 Patent because those skilled in the art came up with many different solutions over the course of the decades of development of deblocking filtering, but never created the specific equation and coefficients of the '659 Patent. Further, in my view, the table above does not necessarily reflect the full range of first offset equations in the prior art. For example, the averaging filter in Aksu described above, in my view, demonstrates that not all offset equations use difference calculations, coefficients applied to those difference calculations, or a limited range of pixels, because Aksu used the average of between three and 7 pixels. Ex-2016 at 15.

222. Dr. Schonfeld also opines that a POSITA would have arrived at the claimed equation from Fu because “the ratio of coefficients in Fu is the same as in the claimed

equation.” Ex-1003, ¶ 275. I disagree. Nothing in Fu (or any of the other identified references) would suggest that a POSITA would (1) want to experiment and change the coefficients associated with the deblocking filter, but (2) must keep the relative ratio between the p_0/q_0 values and the p_1/q_1 values at 3/1 when performing this experimentation. In fact, not a single reference identified by Dr. Schonfeld aside from Fu discloses using a ratio of 3/1 between p_0/q_0 values and the p_1/q_1 values. Ex-1003, ¶¶ 272-273. Dr. Schonfeld also suggests that a POSITA would have been motivated to adjust the magnitude of the offset, but does not provide any explanation for why a POSITA would require that the magnitude of individual components must increase, let alone increase or decrease by the same magnitude (especially in light of Dr. Schonfeld’s arguments throughout his declaration that it would have been obvious to modify all manner of equations, variables, and coefficients would be altered). Ex-1003, ¶ 275.

223. Dr. Schonfeld further declares that “POSITA would have found the claimed offset equation obvious because the coefficients (9 and 3) were nothing but a scaled version of Fu’s coefficients (6 and 2) by 50%.” Ex-1003, ¶ 282.

224. I disagree. Dr. Schonfeld appears to choose the value 50% based on the claimed first offset of the ’659 Patent, rather than providing any motivation in Fu for increasing the filter coefficient values, let alone by that particular factor. Further, even after this arbitrary change, the altered coefficient values of Fu *are still different than the coefficient values of Bjontegaard and List*. Compare altered Fu coefficients of (9/16 and

3/16) with Bjontegaard (8/16 and 3/16) and List (8/16 and 2/16). Ex-1003, ¶¶ 272, 282.

There is no rationale to alter Fu in this way. I have seen no motivation or justification as to why increasing Fu's coefficient of 6 to 9 and 2 to 3 would have been a simple step that would yield predictable and successful results. In my view, the evidence demonstrates the opposite. For example, in my test simulations, the change from "8" to "9" while the second coefficient remained constant between two designs (along with the other aspects of the deblocking design) lead to approximately 40% of pixels being filtered differently.

225. Dr. Schonfeld also opines that a POSITA may have known that the larger-valued coefficients in the art would produce "slightly stronger" deblocking than Fu. Ex-1003, ¶ 283. But Dr. Schonfeld does not point to any disclosure in Fu (or any of the other references for that matter) that Fu did not have aggressive enough deblocking, and would instead require larger coefficients. As noted above regarding the motivation to combine, Dr. Schonfeld has provided no rationale for why a POSITA would look to a prior proposal that was not incorporated into the adopted technical standard and a working draft of a standard that was itself ultimately not adopted, to determine how to best to alter or optimize the Fu deblocking filter.

226. Dr. Schonfeld also relies on Gonzales to opine that "it was known that the "DC" gain of the offset should equal 1." Ex-1003, ¶ 278. I disagree. A POSITA would not look to Gonzales (which is not even asserted as part of the obviousness combination) to determine whether or how Fu should be altered. Similarly, Dr. Schonfeld does not

point to any disclosure in any of the references of the obviousness combination to support this opinion. *Id.*

227. I disagree that a POSITA would have looked specifically to Fu coefficient ratio of 3, and its effect on deblocking of ramp conditions, in creating this newly modified offset equation. Ex-1003, ¶ 490. Dr. Schonfeld also acknowledges that even though Bjontegaard does not use this 3/1 ratio between p_0/q_0 values and the p_1/q_1 values, “it would produce an offset of approximately $\frac{3}{8}$, just slightly more than Bjontegaard’s original offset equation” *Id.* While I do not acknowledge that this difference would result in no visually noticeable improvement, a POSITA would (1) not be motivated to use the 3/1 ratio of Fu when there is no alleged improvement over the pre-existing H.264 standard (or Bjontegaard), and (2) a POSITA would not be motivated to modify Fu by increasing its offset coefficients to more closely match H.264 and/or Bjontegaard when Fu does not provide any visually noticeable improvement over those references, based on Dr. Schonfeld’s declaration. In my opinion, Dr. Schonfeld suggests that while a POSITA would look to the prior art equations and therefore determine to adjust Fu’s coefficient value of $\frac{3}{8}$, they would not look to the prior art to determine to adjust Fu’s coefficient ratio of 3. Ex-1003, ¶ 283. I cannot see a justification other than hindsight for these inconsistencies in Dr. Schonfeld’s arguments.

228. Dr. Schonfeld’s (Ex-1003, ¶¶ 275-280) and Petitioner’s (Petition, 27-30) citation to the several design constraints that would have allegedly limited the first-offset

filter design also undermine their obviousness argument in light of the prior art. For example, Dr. Schonfeld alleges that a POSITA would have experimented with six alleged design constraints to optimize Fu'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. (Ex-1003, ¶¶ 275-280). Under this 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 '659 Patent, with its specific combination of coefficients, window, DC gain, and right shift. Thus, Dr. Schonfeld's opinion that these design constraints would have been used by a POSITA to modify the first offset in Fu to arrive at the claimed offset indicates hindsight bias, especially in light of the fact that even with such alleged constraints, the 659 Patent's coefficients remained undiscovered. Further, in my opinion, the so-called design constraints fail if actually tested. For example, the claimed offsets of $9/16$ and $3/16$ sum to $12/16$ or $2/3$, which is outside the range that Petitioner states is workable range. Petition, 27.

229. Moreover, these so-called design constraints render the number of possibilities generated for calculating the appropriate first offset expression to be well

over 100, and Dr. Schonfeld's design constraints fail, in my view, to set forth any motivation as to why a POSITA would select one design constraint over another. Thus, in my opinion, determining the precise expressions disclosed in the '659 Patent in view of the wide range of possibilities would not have been a task that could be completed by a POSITA using routine experimentation. Instead, extensive subjective and objective testing would be required, without any guarantee that a successful deblocking filter would result.

230. The invention of the '659 Patent was carefully designed to have specific qualities, including not filtering ramps while smoothing other structures, such as steps, in a particular way. No prior art reference achieved these same qualities. For example, as demonstrated by Dr. Schonfeld's analysis, while Fu's first offset does not filter ramps, it does not smooth step, line or random structures in the same way that the '659 Patent first offset does. These differences in filter response further support the non-obviousness of the '659 Patent and the notion 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, and modifying aspects such as filter coefficient values has significant effects on the overall filtering response and filtering results.

p2	p1	p0	q0	q1	Type	$\frac{9(q0 - p0) - 3(q0 - p0)}{16}$	$\frac{6(q0 - p0) - 2(q1 - p1)}{16}$
200	150	150	150	150	Step (p1)	0	0
200	200	150	150	150	Step (p0)	9	6
200	200	200	150	150	Step (q0)	-19	-13
150	160	170	180	190	Ramp (10)	0	0
125	150	175	200	225	Ramp (25)	0	0
200	250	200	250	200	Alternating	38	25
255	50	50	255	255	Line	77	51
255	255	50	50	255	Line	0	0
175	200	150	150	175	Random	5	3
50	125	100	150	125	Random	28	19
75	100	150	75	125	Random	-47	-31

Ex-1003, ¶282.

2. Dependent Claim 4

- a. **4[a]: The method according to claim 1 calculating a second offset to be based on $(p2+p0 \ 2 \times p1 + 2 \times \Delta)/4$, wherein p2 denotes a pixel value of a pixel second next closest to said block boundary in said line of pixels and Δ denotes said first offset**

231. In my opinion, Fu does not disclose or suggest element 4[a] alone or in combination.

232. First, for the reasons set forth above in Section XII.A, it is my opinion that there is no motivation to combine Fu with any of the references that Petitioner and Dr. Schonfeld assert. My discussion in that section is incorporated by reference.

233. Dr. Schonfeld shows the difference between the Second Offset of the '659

Patent and that of Fu:

Reference	Second Offset Equation
'659 Patent	$\Delta = \frac{1(p_2 - 2p_1 + p_0)}{4} + \frac{2\Delta}{4}$
Fu	$\Delta = \frac{1(p_2 - 3p_1 + 3p_0 - q_0)}{8} + \frac{2\Delta}{4}$

Ex-1003, ¶ 343.

234. Dr. Schonfeld opines that the offsets are different in two ways: (1) the exact high-pass filter used and (2) the coefficients applied to that component. *Id.* at ¶344.

235. In my opinion, these differences are critical to the different frequency responses of the two filters and the resulting different filtering outcomes. Thus, in my view, these differences cannot be minimized as Dr. Schonfeld does in his analysis such that the equation of the '659 Patent could have been reached by a POSITA by “optimizing” the equation of Fu using routine experimentation. Offset expressions are carefully designed to yield a specific frequency response. In particular, the '659 second offset uses a Laplacian that only uses p pixels (pixels in the same block) for the “high-pass component.” This design choice enable the deblocking filter to (i) consider the local structure near the block boundary and (ii) avoid filtering ramp structures. Moreover, the frequency response is relevant to other types of structures, and the coefficients ensure

that other structures, such as lines and steps are smoothed in a particular way. In my opinion, Fu's use of q_0 would therefore affect the frequency response.

236. The differences between these two offsets is further illustrated when the Δ , or first offset is considered. The following table reflect the actual deblocking filters normalized based on the value of Δ .

	Second Offset
659 Patent	$\frac{8(p_2 - 2p_1 + p_0) + 9(q_0 - p_0) - 3(q_1 - p_1)}{32}$
Fu	$\frac{4(p_2 - 3p_1 + 3p_0 - q_0) + 6(q_0 - p_0) - 2(q_1 - p_1)}{32}$

237. Dr. Schonfeld provides no analysis to address these differences. 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 an analysis based on the careful selection of other aspects of offsets disclosed in other references to attempt to fashion the claimed offset from the prior art. For example, Dr. Schonfeld opines that the "three tap filter equation" in B118's second offset could be considered, but further states that B118's equation would further be scaled by dividing by 2 if more filtering was desired, or dividing by 4 if a POSITA wanted to preserve Fu's level of filtering. However, Dr. Schonfeld does not explain one way or the other why a POSITA would want to

preserve the amount of filtering in F_u , or increase the amount of filtering given by B118's second offset such that a POSITA would arrive at the claimed second offset expression. Dr. Schonfeld's table of examples illustrates my point that these differences cannot be minimized to a mere difference in more or less filtering because the difference offsets that use different metrics as "high-pass components" behave differently in response to different structures. For example, the table below shows that F_u 's "high-pass component" will smooth step structures in a certain way, whereas a scaled Laplacian will not filter certain step structures. The differences in behavior of these different offsets is evident from Dr. Schonfeld's own analysis. I therefore disagree with Dr. Schonfeld's conclusion that the offsets operate "extremely similarly."

p2	p1	p0	q0	q1	Type	F_u	Laplacian/4	Laplacian/2
200	150	150	150	150	Step (p1)	6	13	25
200	200	150	150	150	Step (p0)	-13	-13	-25
200	200	200	150	150	Step (q0)	6	0	0
150	160	170	180	190	Ramp (10)	0	0	0
125	150	175	200	225	Ramp (25)	0	0	0
200	250	200	250	200	Alternating	-25	-25	-50
255	50	50	255	255	Line	0	51	103
255	255	50	50	255	Line	-51	-51	-103
175	200	150	150	175	Random	-16	-19	-38
50	125	100	150	125	Random	-22	-25	-50
75	100	150	75	125	Random	19	6	13

Ex-1003, ¶352.

238. Dr. Schonfeld also relies on other offsets disclosed in Bjontegaard and B205, neither of which would have rendered obvious the claimed second offset. For example, the normalized offsets are given in the table below.

Bjontegaard	$\frac{5(p_2 - 2p_1 + p_0)}{16} + \frac{5\Delta_{p0}}{16}$
B205	$\frac{\Delta}{2}$

Ex-1003, ¶¶358-359. As an initial matter, I do not agree that a POSITA would have found it obvious to replace d with d_{modified} in Bjontegaard’s equation such that Δ is included in the equation. Nonetheless, in each of these equations, Dr. Schonfeld fails to account for the fact that the delta in each equation would necessarily impact the overall filter design, and has not performed any analysis that would explain the effect.

239. Further, in my view, Dr. Schonfeld has not provided any motivation whatsoever for a POSITA to 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 ’659 filter

design, the differences in the design led to the pixel next closest to the block boundary being filtered differently by the two filters most of the time.

240. 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 supported 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 fails in light of the many approaches that collectively teach away from the notion that one particular expression would have been achieved by a POSITA using routine experimentation to optimize an equation with many different possible components.

- a. **4[b] calculating a third offset to be based on $(q_2 + q_0 - 2 \times q_1 - 2 \times \Delta)/4$, wherein q_2 denotes a pixel value of a pixel second next closest to said block boundary in said corresponding line of pixels;**

241. In my opinion, Fu does not disclose or suggest element 4[b] alone or in combination.

242. First, for the reasons set forth above in Section XII.A, it is my opinion that there is no motivation to combine Fu with any of the references that Petitioner and Dr.

Schonfeld assert illustrates the knowledge of a POSITA. My discussion in that section is incorporated by reference.

243. Dr. Schonfeld shows the difference between the third offset of the '659 Patent and that of Fu:

Reference	Third Offset Equation
'659 Patent	$\Delta = \frac{1(q_0 - 2q_1 + q_2)}{4} - \frac{2\Delta}{4}$
Fu	$\Delta = \frac{1(p_0 - 3q_0 + 3q_1 - q_2)}{8} + \frac{2\Delta}{4}$

Ex-1003, ¶ 379.

244. Dr. Schonfeld opines that the offsets are different in two ways: the exact high-pass filter used; and (2) the coefficients applied to that component. *Id.* at ¶381

245. In my opinion, these differences are critical to the differing frequency responses of the two filters and the resulting filtering outcomes. Thus, in my view, these differences cannot be minimized as Dr. Schonfeld suggests in his analysis such that the equation of the '659 Patent could have been understood by a POSITA by “optimizing” the equation of Fu using routine experimentation. Instead, each offset expression reflects a very specific design with an intentionally designed frequency response. In particular, the '659 third offset uses a Laplacian that only uses q pixels for the “high-pass component.” This was a specific design choice of the overall '659 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. Fu's use of pixel p_0 would necessarily change the frequency response.

246. Further, as I explained above, the difference between the two offsets is illustrated when the value of delta is propagated in the equations. Specifically, while both offsets include a delta, the delta is a reference to the first offset. Thus, like the table above normalizing the second offset, the third offsets can be normalized in a similar way.

247. Dr. Schonfeld provides no analysis to address these differences. 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. Dr. Schonfeld then engages in an analysis based on the careful selection of other aspects of offsets disclosed in other references to attempt to fashion the claimed offset from the prior art. For example, Dr. Schonfeld opines that the "three tap filter equation" in B118's second offset could be considered, but further states that B118's equation would further be scaled by dividing by 2 if more filtering was desired, or dividing by 4 if a POSITA wanted to preserve Fu's level of filtering. However, Dr. Schonfeld does not explain one way or the other why a POSITA would want to preserve the amount of filtering in Fu, or increase the amount of filtering given by B118's third offset such that a POSITA would arrive at the claimed second offset expression. Dr.

Schonfeld's table illustrates my point that these differences cannot be minimized to a mere difference in more or less filtering because the difference offsets that use different metrics as "high-pass components" behave differently in response to different structures. For example, the table below shows that Fu's "high-pass component" will smooth step structures in a certain way, whereas a scaled Laplacian will not filter certain step functions. The differences in behavior of these different offsets is evident from Dr. Schonfeld's analysis. I therefore disagree with Dr. Schonfeld's conclusion that the offsets operate "extremely similarly."

p2	p1	p0	q0	q1	Type	Fu	Laplacian/4	Laplacian/2
200	150	150	150	150	Step (p1)	6	13	25
200	200	150	150	150	Step (p0)	-13	-13	-25
200	200	200	150	150	Step (q0)	6	0	0
150	160	170	180	190	Ramp (10)	0	0	0
125	150	175	200	225	Ramp (25)	0	0	0
200	250	200	250	200	Alternating	-25	-25	-50
255	50	50	255	255	Line	0	51	103
255	255	50	50	255	Line	-51	-51	-103
175	200	150	150	175	Random	-16	-19	-38
50	125	100	150	125	Random	-22	-25	-50
75	100	150	75	125	Random	19	6	13

Ex-1003, ¶352.²

248. Dr. Schonfeld also relies on other offsets disclosed in Bjontegaard and B205, neither of which would have rendered obvious the claimed second offset. For example, the normalized offsets are given in the table below.

Bjontegaard	$\frac{5(q_2 - 2q_1 + q_0)}{16} - \frac{5\Delta}{16}$
B205	$\frac{\Delta}{2}$

Ex-1003, ¶¶358-359. Nonetheless, in each of these equations, Dr. Schonfeld fails to account for the fact that the delta in each equation would necessarily impact the overall filter design, and has not performed any analysis that would explain the effect.

249. Further, in my view, Dr. Schonfeld has not provided any motivation whatsoever for a POSITA to 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, Dr. Schonfeld’s analysis that it would have been obvious to modify Fu’s third offset by (1) substituting a three-tap, high pass filter component $q_2 - 2q_1 + q_0$ like B118 and Bjontegaard and (2) applying a weight of $\frac{1}{4}$ to this component for a ”more efficient” third offset equation is driven only by hindsight based on the claimed third offset. In particular,

² While this table uses pixels for the second offset, Dr. Schonfeld refers to his analysis of the second offset when discussing the third offset, and therefore the same reasons apply as to why the third offset would not have been obvious.

Dr. Schonfeld has not opined that any of these references sought to achieve the specific goal of not filtering ramp structures and smoothing step structures with their disclosed offsets, which is the stated and achieved goal of the '659 Patent. Thus, Dr. Schonfeld's opinion that the filter would be "more efficient" is unsupported, in my opinion.

250. 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 '659 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.

251. 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 supported 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 third 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 third 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. Petitioner’s Ground 2 Fails Because The Challenged Claims Are Not Obvious In View Of Fu And A POSITA’s Knowledge As Evidenced By List, Bjontegaard, B205, And B118

D. No Motivation to Combine Bjontegaard with “the knowledge of a POSITA as evidenced by List, Fu, B205, and B118”

252. I understand that Petitioner’s Ground 2 argument alleges that the Challenged Claims are Obvious in View of Bjontegaard and a POSITA’s knowledge as evidenced by List, Fu, B205, and B118. Petition, 12.

253. After reviewing the Petition and Dr. Schonfeld’s declaration, other than conclusory statements (*see, e.g.*, Petition, 38; Ex1003, ¶345), I do not see any articulated motivated to combine Bjontegaard with any or all of List, Fu, B205, and/or B118. *See* Petition, Ex-1003. Petitioner and Dr. Schonfeld instead state that a number of references could be considered together as “the knowledge of a POSITA.” Regardless of the Petition’s silence on a motivation to combine the prior art, it is my opinion that a POSITA would not be motivated to combine Bjontegaard with the other prior art.

254. Bjontegaard contemplates versions of a deblocking filter design proposed in 2000 to H.26L, which was the evolution of the H.263 standardisation effort, and a subsequent version of which (without Bjontegaard) became the published H.264 Standard. Fu is a Chinese patent application from several years later (2008) that discusses the then-current state of the published H.264 standard, which did not include the

deblocking filter disclosed in Bjontegaard and/or Bjontegaard's Code. The process of developing a standard such as H.264 is/was an in-depth, involved, iterative process, whereby hundreds of engineers come together to evaluate ideas and proposals to create the best video coding standard possible at the time. As part of this process, certain proposals are adopted into the standard, and ultimately are integrated into the version of the standard published by the standardisation body. Other proposals, such as Bjontegaard and Bjontegaard's Code, are proposed, and may even be temporarily or partially adopted, before ultimately being rejected, with the standardisation body choosing to adopt a different approach instead. This happened with the deblocking filter design disclosed in Bjontegaard and Bjontegaard's Code. This is evidenced by the working draft JVT-B118 and the deblocking filter design that was ultimately adopted in the published H.264 Standard, which is very different from the deblocking design in Bjontegaard and Bjontegaard's Code, and does not include those approaches. Dr. Schonfeld has not provided an opinion as to why a POSITA would be motivated to combine a proposed deblocking filter from 2008, which takes as its starting point the then-current state of the H.264 Standard, with previous proposals to the H.264 development process that were rejected and not integrated into the final H.264 published standard. Further, Fu explicitly criticizes the deblocking design of H.264, and Bjontegaard and B118 reflect technical submissions to the H.264 Standardization that were rejected in favor of H.264's deblocking filter described in List. Thus, it is my opinion that a POSITA would not have

been motivated to reach back in the standards history for previously submitted and rejected proposals in the final published standard in combination with other references that explicitly criticize that same standard.

255. I do not think a POSITA would have been motivated to combine these references in order to optimize or calibrate their design according to the '659 Patent. My Test Simulations illustrate that differences in deblocking filter designs result in significant differences in filtering outcomes. Thus, I would expect the filtering outcome values to be very different if applying Bjontegaard's offset equation and Fu's, B118's, Lists, or B205's offset equation to the same set of pixels. Dr. Schonfeld provides no explanation as to why a POSITA would increase Bjontegaard's first coefficient from 8/16 to 9/16 but keep Bjontegaard's second coefficient at 3/16 based on Fu's disclosed coefficients of 6/16 and 2/16, (Ex-1003, ¶¶ 488-490) because, as illustrated in my simulations, such an arbitrary change would in fact have an impact on other factors such as the filter impulse response and the behavior of the filter. Moreover, Dr. Schonfeld states that such a change would likely have "expected slight differences in filtering performance" (Ex-1003, ¶ 282), but has provided no testing or analysis to support this statement. Ex-1003, ¶¶ 262-274, 281-292.

256. A POSITA would not have had a reasonable expectation of success in modifying Fu's or Bjontegaard's first coefficient for the first offset because in my view adjusting the coefficients in this way is certainly not "routine experimentation" as Dr.

Schonfeld asserts. Ex-1003, ¶¶ 281-283. Changing the coefficients is not a mere edit of a number in a single line of code, but as explained above, changing the filter coefficients changes the frequency response of the filter and is a non-trivial change in operation that produces very different filtering outcomes, as illustrated by my simulation testing above. Petitioner has not provided any testing results to support such an argument.

257. Dr. Schonfeld's own summary of the normalized versions of the first offset equations from the different references show that the differences in the design of the first offset expression can vary in coefficients used as well as the pixels used. Ex-1003, ¶ 272. These so-called "slight" differences are exacerbated by the additional differences in the filtering designs that are not limited to only the differences in offset equations. For example, as explained above, Fu relies on threshold values using absolute difference calculations in its "Process 3" to determine whether to filter the pixels next closest to the block boundary, whereas Bjontegaard uses three-pixel Laplacians in some instances, not to determine whether to filter the pixels next closest to the block boundary, but rather to calculate the actual value of the offset applied to the pixels next closest to the block boundary. See §XI.B. These differences are exemplary, and as illustrated by my testing, the many factors considered as part of the overall filter design are sensitive to changes, and will result in significant difference in filtering outcome. See § X above.

258. In addition, in my opinion, a POSITA would not have been motivated to modify Bjontegaard in light of B205 for similar reasons—that is, the filtering approach

is fundamentally different. When comparing the offsets calculated by B205 versus Bjontegaard it would have been apparent to a POSITA that the approaches are fundamentally different in their use of which and how many pixels, and the actual coefficients used, which would affect the frequency response. Moreover, in my view, a POSITA would not have been motivated to combine B205, the then-current draft of the H.265 Standard with a previously submitted and ultimately rejected proposal to the prior generation codec from a decade before. Such a motivation is contrary to the nature of standardization development.

259. Further, all of the cited references asserted as the “basis” for a POSITA’s understanding of the art (e.g., List, Fu, B118, and JCTVC-B205) disclose different approaches to what the offsets applied to the pixels next closest to the block boundary (e.g., second and third offsets). For example, if each of the second offset expressions is normalized, as Dr. Schonfeld and Petitioner assert is necessary to compare the expressions, then the second offset expressions in the prior art compared to the ’659 Patent are given below:

	Second Offset
659 Patents	$\frac{64(p_2 - 2p_1 + p_0) + 72(q_0 - p_0) - 24(q_1 - p_1)}{256}$
Bjontegaard	$\frac{80(p_2 - 2p_1 + p_0) + 40(q_0 - p_0) - 15(q_1 - p_1)}{256}$
Fu	$\frac{32(p_2 - 3p_1 + 3p_0 - q_0) + 48(q_0 - p_0) - 16(q_1 - p_1)}{256}$
H.264/List	$\frac{128p_2 - 256p_1 + 64p_0 + 64q_0}{256}$
B205	$\frac{60(q_0 - p_0) + 16(q_1 - p_1) - 20(q_2 - p_2)}{256}$
B118	$\frac{128(p_2 - 2p_1 + p_0) + 64(q_0 - p_0) - 16(q_1 - p_1)}{256}$

260. I normalized these expressions by inserting the previously calculated Δ , or first offset expression, where it appears in the second offset expression.

261. As shown by the table above for the second offset equations, not only are the coefficients (e.g. weights) assigned to each element substantively different among the different solutions provided by the prior art, but there is also no uniform, agreed-upon approach to which pixels are used, or what metric, should be used to calculate the second offset. For example, some expressions use equations that measure slope and absolute difference (e.g., B118, Fu, Bjontegaard), while others use equations that measure absolute difference only (e.g., B205, List). The differences between these offset expressions would result in significantly different filtering results because the frequency response of the filter and the behavior in response to different types of structures in the blocks of pixels would necessarily be different.

262. Thus, Petitioner's arguments with respect to motivation to combine and obviousness are unsupported because they are based on disparate solutions in the prior art. It was not until the invention of the '659 Patent that anyone determined the specific, appropriate second and third offsets as claimed.

263. Further, Bjontegaard proposed a deblocking filter design to H.26L in 2000, which subsequently became the H.264 Standard. Ex-1009. Approximately two years later, the Joint Video Team (the standardization body that created the H.264 Standard) published JVTB118, which was a working draft of the then current version of the H.264 Standard being developed. Ex-1023. Notably, Bjontegaard and B118 contain different offset equations for modifying the pixels closest and next closest to the block boundary. *Compare* Ex-1023 with Ex-1009. The 2003 version of the H.264 Standard is a further evolution of the working draft published as B118, with identical first offset equations. However, the adopted H.264 Standard altered the second and third offset equations as compared to B118, removing the second and third offset equations' reliance on the previously calculated modified pixel values closest to the block boundary. *Compare* Ex-1023 (B118) with Ex-1024 (List).

1. *Independent Claims 1 and 10*

- a. **1[a]/10[a]: calculating a first offset to be equal to $(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$, wherein $\gg$ denotes a right shift operation, p_0 denotes a pixel value of a pixel closest to said block boundary in a line of pixels in said block, wherein said line of pixels is perpendicular to said block boundary, p_1 denotes a**

pixel value of a pixel next closest to said block boundary in said line of pixels, q_0 denotes a pixel value of a pixel closest to said block boundary in a corresponding line of pixels in said neighboring block, wherein said corresponding line of pixels is perpendicular to said block boundary and q_1 denotes a pixel value of a pixel next closest to said block boundary in said corresponding line of pixels;

264. Dr. Schonfeld opines that “Bjontegaard teaches that a first offset equation” reproduced below:

$$Diff = (3(c - f) - 8(d - e))/16$$

Ex-1003, ¶ 479.

265. The equations are set forth below:

Reference	First Offset Equation
'659 Patent	$(9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$
Bjontegaard	$(8 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4$

Ex-1003, at ¶ 487.

266. It is undisputed that the Bjontegaard’s “First Offset” is different from the Claimed First Offset based at least on the use of the coefficient 8/16 versus 9/16. I disagree with Dr. Schonfeld that modifying Bjontegaard’s First Offset would have been obvious to a POSITA to arrive at the claimed First Offset for all of the reasons I explained with respect to Fu in Ground 1 above.

267. **First**, Dr. Schonfeld ignores that Bjontegaard's "First Offset" identified by Dr. Schonfeld is not always applied to the pixels d and e, as other values for Diff can be calculated and applied. Instead, it is a part of a different filtering design than the one disclosed in the '659 Patent. Specifically, Bjontegaard's First Offset is only applied if, in the algorithms for Q15-J-28, for example, Act_{edge} equals 1 or 2. Ex-1003, ¶ 475-476. Otherwise, e.g., Act_{edge} equals 3, a different Diff equation is used to modify the edge pixels closest to the block boundary. *Id.* Diff could also be equal to 0. This is an exemplary illustration of the significantly different behavior of the design disclosed in Bjontegaard compared with the Claims of the '659 Patent as illustrated by my simulation test results explained above. As I explained, when different filtering designs are implemented, the filter value is actually applied to the underlying signal being filtered, which is exacerbated by the fact that the filter must be applied to millions of pixels in the loop. Thus, what may superficially be a minor change in a coefficient between 8/16 and 9/16 is actually significant in operation due to the calibration and extensive objective and subjective testing needed to develop an elegant and effective deblocking filtering solution. In particular, the designs are fundamentally different because the offset that is claimed will not filter ramp structures and modifies other structures, such as step structures in a particular way. In contrast, the first offset of Bjontegaard will filter ramp structures and does not smooth other structures the same way that the '659 claimed offset does. This is demonstrated by Dr. Schonfeld's analysis.

						'659 Patent	Bjontegaard
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

268. Ex-1003, ¶489.

269. **Second**, I incorporate my analysis of the substantial differences between the design of the filter for Bjontegaard and the claimed filter. As shown in my simulations, the filtering decisions made by the simulated Q15-J-28 method are different to those of the simulated claimed method, for most of the filtered pixels. This is supported by Dr Schonfeld's own analysis, in which Bjontegaard's design and the claimed filter produce different results in most cases (Ex-1003 at 489). As discussed with respect to Bjontegaard, the evidence shows that the H.26L/H.264

industry standards effort rejected Bjontegaard (proposed in 2000) in favor of fundamentally different designs. Thus, there is no evidence to justify Dr. Schonfeld's opinion that use of different filter coefficients would be the result of routine experimentation. Instead, the evidence suggests that the deblocking filter is a complex design that is sensitive to even the smallest changes, and thus requires extensive testing and simulations to verify any potential design change.

270. In addition, a POSITA would not have found it obvious to substitute Bjontegaard's of $8/16$ with a coefficient of $9/16$ in view of the teaching of Fu, which teaches coefficients of $3/8$ and $1/8$. As explained above, a POSITA would expect the filtering outcome to be very different if applying Fu's offset equation and Bjontegaard's offset equation to the same set of pixels. I also disagree with Dr. Schonfeld opinion that a POSITA would seek to keep a 3:1 ratio between coefficients. Ex-1003, ¶¶ 262, 490. Despite identifying numerous filter designs, Dr. Schonfeld did not identify any other reference that actually maintained a 3:1 ratio. Ex-1003, ¶¶ 262-273. Thus, Dr. Schonfeld has not shown that POSITA would be motivated to incorporate a 3:1 ratio. I incorporate my discussion with regard to Ground 1 of Dr. Schonfeld's opinions that a POSITA would have reached the claimed coefficients because of certain alleged "design constraints" in the prior art.

271. Moreover, I disagree that a POSITA would have had a reasonable expectation of success in modifying Bjontegaard's first coefficient for the first offset because

adjusting the coefficients in this way is not “routine experimentation” as Dr. Schonfeld asserts. Ex-1003, ¶ 488. Redesigning the filter coefficients is not a mere edit of a number in a single line of code, but as I explained above, changing the filter coefficients changes the frequency response of the filter and is a non-trivial change in operation that produces very different filtering outcomes, as illustrated in my simulation testing above.

2. *Dependent Claim 4*

- a. **4[a]: The method according to claim 1 calculating a second offset to be based on $(p_2 + p_0 - 2 \times p_1 + 2 \times \Delta)/4$, wherein p_2 denotes a pixel value of a pixel second next closest to said block boundary in said line of pixels and Δ denotes said first offset**

272. In my opinion, Bjontegaard does not disclose or suggest element 4[a] alone or in combination.

273. First, for the reasons set forth above in Section XII.C, it is my opinion that there is no motivation to combine Bjontegaard with any of the references that Petitioner and Dr. Schonfeld suggests represents the knowledge of a POSITA. My discussion in that section is incorporated by reference.

274. Dr. Schonfeld shows the difference between the Second Offset of the '659 Patent and that of Bjontegaard (with his own modification to Bjontegaard to include the extra parameter $5\Delta/16$, Ex-1003 at 210):

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, ¶ 546.

275. 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, ¶ 547.

276. 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 ’659 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 ’659 second offset uses a Laplacian that only uses p pixels for the “high-pass component.” This was a specific design choice of the overall ’659 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.

277. The differences between these two offsets is further illustrated when the Δ is considered. Though I disagree that a POSITA would have read Bjontegaard’s “second offset to use d_{modified} instead of d as explained above, I will assume for purposes of demonstrating non-obviousness under Dr. Schonfeld’s own analysis that the Diff calculations use d_{modified} and e_{modified} instead of d and e as they are written.

278. Specifically, while both offsets include a delta (under Dr. Schonfeld’s interpretation of Bjontegaard), the Δ is a reference to the first offset. Thus, the following table reflect the actual deblocking filters normalized based on the value of delta.

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

279. 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 has not in my opinion, shown that the coefficients in the prior art offset equations were within a known workable range, because Dr. Schonfeld has not provided any opinions as to how modifying the offsets as suggested

would affect the overall filter design. 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 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 '659 Patent. In my opinion, Dr. Schonfeld cannot justify such an opinion. For example, Dr. Schonfeld's table of examples illustrates my point that these differences cannot be minimized to a mere difference in coefficients because the frequency response and filtering result is substantially different. Critically, the '659 Patent second offset does not filter ramp structures, while Bjontegaard's second offset does filter ramp structures. Moreover, Bjontegaard's second offset smooths other structures differently than the '659 Second offset.

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, ¶552.

280. 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 I compared a simulation of Bjontegaard's filter design with a simulation of the '659 filter design, the differences in the design led to the pixel next closest to the block boundary being filtered differently by the two filters around 76% of the time.

281. 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.

- b. 4[b] calculating a third offset to be based on $(q_2 + q_0 - 2 \times q_1 - 2 \times \Delta)/4$, wherein q_2 denotes a pixel value of a pixel second next closest to said block boundary in said corresponding line of pixels;**

282. In my opinion, Bjontegaard does not disclose or suggest element 4[b] alone or in combination.

283. First, for the reasons set forth above in Section XII.C, it is my opinion that there is no motivation to combine Bjontegaard with any of the references that Petitioner

and Dr. Schonfeld assert illustrates the knowledge of a POSITA. My discussion in that section is incorporated by reference.

284. Dr. Schonfeld shows the difference between the Third Offset of the '659 Patent and that of Bjontegaard (with his own modification to Bjontegaard to include the extra parameter Δ):

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

Ex-1003, ¶ 565.

285. 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 (e.g., Section X.B.4.a, Ex-1003, ¶568), 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 the same elements it is my opinion that the coefficients would not have been obvious. *See* Ex-1003, ¶ 547.

286. 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 '659 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 '659 second offset uses a Laplacian that only uses p pixels for the “high-pass component.” This was a specific design choice of the overall '659 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.

287. 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 delta equation, and demonstrates that substantially different filtering coefficients.

288. 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. 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 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, ¶554. 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 '659 Patent. In my opinion, Dr. Schonfeld cannot support such an opinion. 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 '659 Patent third offset does not filter ramp structures, while Bjontegaard's third offset does filter ramp structures. Moreover, Bjontegaard's third offset does not smooth the other structures, such as a step structures, in the same way that the '659 Patent's offset does.³

³ While the table below is a demonstrationg 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, ¶552.

289. 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 '659 filter design, the differences in the design led to the pixel next closest to the block boundary being filtered differently by the two filters around 76% of the time.

290. 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 entire equations with many different possible components.

XIII. SUMMARY AND OTHER REMARKS

291. It is my opinion that Petitioner has failed to show that any of the Challenged Claims set forth in Grounds 1 and 2 are unpatentable as obvious as explained above.

292. 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 hereby 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.