

1 RUSS, AUGUST & KABAT  
 2 Marc A. Fenster, SBN 181067  
 Email: mfenster@raklaw.com  
 3 Reza Mirzaie (CA SBN 246953)  
 4 Email: rmirzaie@raklaw.com  
 Brian D. Ledahl (CA SBN 186579)  
 5 Email: bledahl@raklaw.com  
 Paul Kroeger (CA SBN 229074)  
 6 Email: pkroeger@raklaw.com  
 7 C. Jay Chung (CA SBN 252794)  
 Email: jchung@raklaw.com  
 8 Philip X. Wang (CA SBN 262239)  
 9 Email: pwang@raklaw.com  
 12424 Wilshire Boulevard, 12<sup>th</sup> Floor  
 10 Los Angeles, California 90025  
 11 Telephone: (310) 826-7474  
 12 Facsimile: (310) 826-6991

13 *Attorneys for Plaintiff*  
 14 *REALTIME ADAPTIVE STREAMING LLC*

15 **UNITED STATES DISTRICT COURT**  
 16 **CENTRAL DISTRICT OF CALIFORNIA**  
 17 **WESTERN DIVISION**

18 REALTIME ADAPTIVE STREAMING  
 19 LLC,  
 Plaintiff,  
 20 vs.  
 21 ADOBE SYSTEMS INC.,  
 22 Defendant.  
 23

Case No. 2:18-cv-09344-GW-JC  
**JURY TRIAL DEMANDED**

24 **AMENDED COMPLAINT FOR PATENT INFRINGEMENT**

25  
 26 This is an action for patent infringement arising under the Patent Laws of the  
 27 United States of America, 35 U.S.C. § 1 *et seq.* in which Plaintiff Realtime Adaptive  
 28

1 Streaming LLC (“Plaintiff” or “Realtime”) makes the following allegations against  
2 Defendant Adobe Systems Inc. (“Defendant” or “Adobe”).

3 **PARTIES**

4 1. Realtime is a Texas limited liability company. Realtime has a place of  
5 business at 1828 E.S.E. Loop 323, Tyler, Texas 75701.

6 2. On information and belief, Defendant Adobe Systems Inc. is a Delaware  
7 corporation with its principal place of business at 345 Park Ave, San Jose, California  
8 95110.

9 3. Defendant Adobe Systems Inc. has regular and established places of  
10 business in this District, specifically at least at One Newton Place, Newton, MA  
11 02458 and One Broadway, Cambridge, MA 02142. Defendant Adobe Systems Inc.  
12 offers their products and/or services, including those accused herein of infringement,  
13 to customers and potential customers located in Massachusetts and in this District.  
14 15 16

17 4. Defendant Adobe Systems Inc. may be served with process through its  
18 registered agent for service c/o Corporation Service Company at 84 State Street,  
19 Boston, MA 02109.

20 **JURISDICTION AND VENUE**

21 22 5. This action arises under the patent laws of the United States, Title 35 of  
23 the United States Code. This Court has original subject matter jurisdiction pursuant to  
24 28 U.S.C. §§ 1331 and 1338(a).

25 26 6. This Court has personal jurisdiction over Defendant in this action  
27 because Defendant has committed acts within the District of Massachusetts giving rise  
28

1 to this action and has established minimum contacts with this forum such that the  
2 exercise of jurisdiction over Defendant would not offend traditional notions of fair  
3 play and substantial justice. The Defendant has also committed and continues to  
4 commit acts of infringement in this District by, among other things, offering to sell  
5 and selling products and/or services that infringe the asserted patents.  
6

7 7. Venue is proper in this district, *e.g.*, under 28 U.S.C. § 1400(b).  
8 Defendant has a regular and established place of business in this District, for example  
9 at One Newton Place, Newton, MA 02458 and One Broadway, Cambridge, MA  
10 02142. Furthermore, upon information and belief, Defendant has transacted business  
11 in the District of Massachusetts and has committed acts of direct and indirect  
12 infringement in the District of Massachusetts.  
13  
14

#### 15 **THE PATENTS-IN-SUIT**

16 8. This action arises under 35 U.S.C. § 271 for Adobe's infringement of  
17 Realtime's United States Patent Nos. 7,386,046 (the "'046 patent"), RE46,777 (the  
18 "'777 patent"), 8,929,442 (the "'442 patent"), 8,934,535 (the "'535 patent"),  
19 9,578,298 (the "'298 patent"), 9,762,907 (the "'907 patent"), and 9,769,477 (the  
20 "'477 patent") (collectively, the "Patents-In-Suit").  
21  
22

23 9. The '046 patent, titled "Bandwidth Sensitive Data Compression and  
24 Decompression," was duly and properly issued by the United States Patent and  
25 Trademark Office ("USPTO") on June 10, 2008. A copy of the '046 patent is  
26 attached hereto as Exhibit A. Realtime is the owner and assignee of the '046 patent  
27 and holds the right to sue for and recover all damages for infringement thereof,  
28

1 including past infringement.

2 10. The '777 patent, titled "Quantization for Hybrid Video Coding," was  
3 duly and properly issued by the USPTO on April 3, 2018. The '777 patent is a reissue  
4 of U.S. Pat. No. 8,634,462, which was issued on January 21, 2014. A copy of the  
5 '777 patent is attached hereto as Exhibit B. Realtime is the owner and assignee of the  
6 '777 patent and holds the right to sue for and recover all damages for infringement  
7 thereof, including past infringement.  
8

9 11. The '442 patent, titled "System and method for video and audio data  
10 distribution," was duly and legally issued by the USPTO on January 6, 2015. A true  
11 and correct copy of the '442 patent is included as Exhibit C. Realtime is the owner  
12 and assignee of the '442 patent and holds the right to sue for and recover all damages  
13 for infringement thereof, including past infringement.  
14

15 12. The '535 patent, titled "Systems and methods for video and audio data  
16 storage and distribution," was duly and properly issued by the USPTO on January 13,  
17 2015. A copy of the '535 patent is attached hereto as Exhibit D. Realtime is the  
18 owner and assignee of the '535 patent and holds the right to sue for and recover all  
19 damages for infringement thereof, including past infringement.  
20

21 13. The '298 patent, titled "Method for Decoding 2D-Compatible  
22 Stereoscopic Video Flows," was duly and properly issued by the USPTO on February  
23 21, 2017. A copy of the '298 patent is attached hereto as Exhibit E. Realtime is the  
24 owner and assignee of the '298 patent and holds the right to sue for and recover all  
25 damages for infringement thereof, including past infringement.  
26  
27  
28



1 After Effects, Adobe After Effects CC, Adobe After Effects CS (e.g. CS4, CS5,  
2 CS5.5, CS6), Adobe HTTP Dynamic Streaming (HDS), and all versions and  
3 variations thereof since the issuance of the '046 patent ("Accused Instrumentalities").  
4

5 9. On information and belief, Adobe has directly infringed and continues to  
6 infringe the '046 patent, for example, through its sale, offer for sale, importation, use  
7 and testing of the Accused Instrumentalities, which practices the system claimed by,  
8 for example, Claim 40 of the '046 patent, namely, a system, comprising: a data  
9 compression system for compressing and decompressing data input; a plurality of  
10 compression routines selectively utilized by the data compression system, wherein a  
11 first one of the plurality of compression routines includes a first compression  
12 algorithm and a second one of the plurality of compression routines includes a second  
13 compression algorithm; and a controller for tracking throughput and generating a  
14 control signal to select a compression routine based on the throughput, wherein said  
15 tracking throughput comprises tracking a number of pending access requests to a  
16 storage device; and wherein when the controller determines that the throughput falls  
17 below a predetermined throughput threshold, the controller commands the data  
18 compression engine to use one of the plurality of compression routines to provide a  
19 faster rate of compression so as to increase the throughput. Upon information and  
20 belief, Adobe uses the Accused Instrumentalities to practice infringing methods for its  
21 own internal non-testing business purposes, while testing the Accused  
22 Instrumentalities, and while providing technical support and repair services for the  
23 Accused Instrumentalities to Adobe's customers.  
24  
25  
26  
27  
28

1           10. The Accused Instrumentalities include, or practice a system, comprising:  
2 a data compression system for compressing and decompressing data input. For  
3 example, the Accused Instrumentalities utilize H.264 and/or H.264 with Scalable  
4 Video Coding (SVC) and/or an equivalent codec. H.264 is “the most widely used  
5 codec on the planet.” See [http://www.streamingmedia.com/Articles/Editorial/What-Is-  
6 .../What-is-H.264-74735.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx). A “codec” is also a compression technology that has  
7 “two components, an encoder to compress the files, and a decoder to decompress.  
8 There are codecs for...video (Cinepark, MPEG-2, **H.264**, VP8).” See  
9 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-  
10 74487.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx); <https://forums.adobe.com/thread/729526> (“Compression is essential for  
11 reducing the size of movies so that they can be stored, transmitted, and played back  
12 effectively. Compression is achieved by an encoder; decompression is achieved by a  
13 decoder. Encoders and decoders are known by the common term codec.”). According  
14 to a website maintained by Adobe, “you **can produce H.264 video** with Adobe Media  
15 Encoder CS4 and Adobe Flash Media Encoding Server 3.5,” “Adobe added **H.264**  
16 **playback support** to Adobe Flash Player 9 Update 3 back in 2007,” Adobe “Flash  
17 Media Server and Flash Player can **both stream and play back any H.264 file** in  
18 virtually any format,” the encoding capabilities of the Adobe Flash Media Encoding  
19 Server using “**H.264-related parameters**” are described (e.g. using CAVLC, CABAC)  
20 and under a headline stating “Adobe Media Encoder,” the website states: “Adobe  
21 significantly enhanced the Flash Video Encoder in Creative Suite 4. There is now both  
22 stand-alone operation and batch encoding capabilities. As before, you can **access**  
23  
24  
25  
26  
27  
28

1 **H.264 encoding** by choosing different formats in the Format pop-up menu. When  
2 producing for Flash Player, you should always use the FLV|F4V option, which lets  
3 you produce both VP6- and **H.264-encoded files** for Flash Player distribution.” See  
4 [http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

5  
6 Another website maintained by Adobe mentions that “You have two primary options  
7 for using Adobe Media Encoder with After Effects to create videos in **H.264**.” The  
8 website also mentions that “Since Adobe Media Encoder already has superior  
9 H.264...exporting capabilities, it was more prudent to rely on Adobe Media Encoder  
10 for export of these formats.” See [https://helpx.adobe.com/after-effects/kb/export-](https://helpx.adobe.com/after-effects/kb/export-h264.html)

11 [h264.html](https://helpx.adobe.com/after-effects/kb/export-h264.html). Another website maintained by Adobe mentions that the “Adobe Flash  
12 Player 9 Update 3 is taking a step into the high-definition (HD) video realm in a major  
13 way by adding MPEG-4 video...[which] utilizes crisp, powerful **H.264 encoding**”  
14 and “The new Adobe Media Player...will **also support H.264**” and “The addition of  
15 **H.264...support** in Flash Player 9 Update 3 allows you to easily use high-definition,  
16 industry standard video and audio.” See

17 [http://www.adobe.com/devnet/flashplayer/articles/hd\\_video\\_flash\\_player.html](http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html). Adobe  
18 HTTP Dynamic Streaming (HDS) also supports additional data compression systems  
19 or codecs other than H.264 because it “enables high-quality (**H.264 or VP6**), network-  
20 efficient HTTP streaming for media delivery that is tightly integrated with Adobe®  
21 Access software for robust content protection in the Adobe Flash® Player 10.1 or later  
22 and Adobe AIR®2 or later runtimes,” is also “built on standards and deployed using  
23 standard HTTP servers (Apache), standard media format (MP4 fragment) using  
24  
25  
26  
27  
28



1 standard codecs (**H.264/AAC**), open APIs (Flash Player), and an open source  
2 framework for building media players (OSMF)” and “like other Flash Player  
3 supported delivery methods,” Adobe HDS “supports **H.264 video and VP6 codecs**  
4 required for the highest quality video on the platform.” *See*  
5 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS also  
6 allows users to encode content “using high-quality Flash Player compatible **codecs**  
7 (**VP6/MP3, H264/AAC**),” deliver “high-definition video up to 1080p, with bitrates  
8 from 700kbps up to and beyond 6Mbps, using either **H.264 or VP6 video codecs**,”  
9 and archive “live, high-definition streams on the server and enable HD DVR  
10 functionality (for example, instant replay and time shifting) with support for **H.264**  
11 stream recording for RTMP and HTTP Dynamic Streaming.” *See*  
12 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Adobe Media  
13 Encoder also “lets you **produce both VP6- and H.264-encoded files** for Flash Player  
14 distribution.” *See* [http://www.adobe.com/devnet/adobe-media-](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
15 [server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

20 11. The Accused Instrumentalities also include, or practice a system  
21 comprising: a plurality of compression routines selectively utilized by the data  
22 compression system, wherein a first one of the plurality of compression routines  
23 includes a first compression algorithm and a second one of the plurality of  
24 compression routines includes a second compression algorithm. First, based on  
25 various parameters (e.g. throughput, bitrate, max video bitrate, resolution), any H.264-  
26 compliant system such as the Accused Instrumentalities would determine which  
27  
28

1 profile (e.g., “baseline,” “extended,” “main”, or “high”) corresponds with that  
2 parameter, whether if that parameter (e.g. throughput) meets some criteria, then select  
3 between at least two asymmetric compressors or plurality of compression routines,  
4 where one asymmetric compressor may serve as the first compression algorithm and  
5 another asymmetric compressor may serve as the second compression algorithm. If  
6 baseline or extended is the corresponding profile, then the system will select a  
7 Context-Adaptive Variable Length Coding (“CAVLC”) entropy encoder, which may  
8 serve as the first or second compression algorithm or asymmetric compressor. If main  
9 or high is the corresponding profile, then the system will select a Context-Adaptive  
10 Binary Arithmetic Coding (“CABAC”) entropy encoder, which may serve as the first  
11 or second compression algorithm or asymmetric compressor. Both encoders are  
12 asymmetric compressors because it takes a longer period of time for them to compress  
13 data than to decompress data. *See* [https://sonnati.wordpress.com/2007/10/29/how-h-  
14 264-works-part-ii/](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/).

15  
16  
17  
18  
19 12. The Accused Instrumentalities also include, or practice a system  
20 comprising: a controller for tracking throughput and generating a control signal to  
21 select a compression routine based on the throughput, wherein said tracking  
22 throughput comprises tracking a number of pending access requests to a storage  
23 device. For example, Adobe HTTP Dynamic Streaming (HDS) contains the “ability to  
24 **shift quality depending on bandwidth and computer power**. HTTP progressive  
25 delivery consumes more bandwidth because it's not intelligent enough to throttle the  
26 delivery (for example, a 30-minute video will be fully downloaded whether the user  
27  
28

1 watches it or not).” See [https://www.adobe.com/products/hds-dynamic-](https://www.adobe.com/products/hds-dynamic-streaming/faq.html)  
2 [streaming/faq.html](https://www.adobe.com/products/hds-dynamic-streaming/faq.html). Adobe HDS can also “[d]etect the client’s bandwidth and  
3 **computer resources** and serve them content fragments encoded at the most  
4 appropriate bitrate.” See [https://www.adobe.com/products/hds-dynamic-](https://www.adobe.com/products/hds-dynamic-streaming/features.html)  
5 [streaming/features.html](https://www.adobe.com/products/hds-dynamic-streaming/features.html). On information and belief, the Accused Instrumentalities also  
6 include a storage device e.g., hard disk, disks, buffers, servers or other forms of  
7 memory/storage, that would receive pending access requests so the controller could  
8 track throughput by tracking a number of pending access requests to that storage  
9 device. See [https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html)  
10 [streaming-live-streaming.html](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html) (describing “Streams [from Adobe HDS] in a live event  
11 are packaged as fragments and written to **disk**,” “The **IO buffer** loads the **disk file**  
12 into an **in-memory buffer**,” and “To limit the amount of **storage the disk cache**  
13 uses,” and a “Content storage (HDS and HLS)” header describing a section where a  
14 “media player **requests content from the server**.” As shown below, when the  
15 controller generates a control signal to select a compression routine based on the  
16 throughput, it is clear that the compression routines and the first and second  
17 compression algorithms all utilize various parameters to compress or decompress data  
18 input including, of course, throughput and bandwidth, but also bitrate (or max video  
19 bitrate), and resolution. Different parameters also correspond with different end  
20 applications. H.264, a data compression system, and compression routine provides for  
21 multiple different ranges of such parameters, each included in the “profiles” and  
22 “levels” as defined by the H.264 standard, from the below shown paragraphs from a  
23  
24  
25  
26  
27  
28

1 white paper and Wikipedia. See

2 [http://www.axis.com/files/whitepaper/wp\\_h264\\_31669\\_en\\_0803\\_lo.pdf](http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf) at 5:

3  
4 **4. H.264 profiles and levels**

5 The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting  
6 options and features to a minimum. An important aspect of the standard, as with other video standards,  
7 is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes)  
8 that optimally support popular productions and common formats.

9 H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what  
10 feature set the encoder may use and limits the decoder implementation complexity.

11 Network cameras and video encoders will most likely use a profile called the baseline profile, which is  
12 intended primarily for applications with limited computing resources. The baseline profile is the most  
13 suitable given the available performance in a real-time encoder that is embedded in a network video  
14 product. The profile also enables low latency, which is an important requirement of surveillance video and  
15 also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

16 H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements.  
17 Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging  
18 from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

19 See [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC):

Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) <a href="#">Toggle additional details</a>
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	176x144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	176x144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	352x288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	352x288@15.2 (6)
1.3	3,041,280	11,880	101,376	396	768	960	2,304	352x288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	352x288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352x576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	720x576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	720x576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	1,280x720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280x1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048x1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048x1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048x1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	3,672x1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@56.3 (5)

1           13. A video data block is organized by the group of pictures (GOP) structure,  
2 which is a “collection of successive pictures within a coded video stream.” *See*  
3 [https://en.wikipedia.org/wiki/Group\\_of\\_pictures](https://en.wikipedia.org/wiki/Group_of_pictures). A GOP structure can contain intra  
4 coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame),  
5 bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture  
6 or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See*  
7 [https://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](https://en.wikipedia.org/wiki/Video_compression_picture_types) (for descriptions of I  
8 frames, P frames and B frames); <https://en.wikipedia.org/wiki/MPEG-1#D-frames> (for  
9 descriptions of D frames). Thus, at least a portion of a video data block would also  
10 make up a GOP structure and could also contain I frames, P frames, B frames and/or  
11 D frames. The GOP structure also reflects the size of a video data block, and the GOP  
12 structure can be controlled and used to fine-tune other parameters (e.g.  
13 throughput/bandwidth, bitrate, max video bitrate and resolution parameters) or even  
14 be considered as a parameter by itself.

15           14. The Accused Instrumentalities also practice wherein when the controller  
16 determines that the throughput falls below a predetermined throughput threshold, the  
17 controller commands the data compression engine to use one of the plurality of  
18 compression routines to provide a faster rate of compression so as to increase the  
19 throughput. Based on the throughput, bitrate, and/or resolution parameter identified  
20 (e.g. throughput, bandwidth, bitrate, max video bitrate, resolution, GOP structure or  
21 frame type within a GOP structure), any H.264-compliant system such as the Accused  
22 Instrumentalities would determine which profile (e.g., “baseline,” “extended,” “main”,  
23  
24  
25  
26  
27  
28

1 or “high”) corresponds with that parameter, whether if that parameter (e.g.  
 2 throughput) falls below a predetermined throughput threshold, then select between at  
 3 least two asymmetric compressors to, for example, provide a faster rate of  
 4 compression so as to increase the throughput. If baseline or extended is the  
 5 corresponding profile, then the system will select a Context-Adaptive Variable Length  
 6 Coding (“CAVLC”) entropy encoder. If main or high is the corresponding profile,  
 7 then the system will select a Context-Adaptive Binary Arithmetic Coding (“CABAC”)  
 8 entropy encoder. Both encoders are asymmetric compressors because it takes a longer  
 9 period of time for them to compress data than to decompress data. *See*  
 10 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
 11  
 12  
 13

	Baseline	Extended	Main	High	High 10
<b>I and P Slices</b>	Yes	Yes	Yes	Yes	Yes
<b>B Slices</b>	No	Yes	Yes	Yes	Yes
<b>SI and SP Slices</b>	No	Yes	No	No	No
<b>Multiple Reference Frames</b>	Yes	Yes	Yes	Yes	Yes
<b>In-Loop Deblocking Filter</b>	Yes	Yes	Yes	Yes	Yes
<b>CAVLC Entropy Coding</b>	Yes	Yes	Yes	Yes	Yes
<b>CABAC Entropy Coding</b>	No	No	Yes	Yes	Yes
<b>Flexible Macroblock Ordering (FMO)</b>	Yes	Yes	No	No	No
<b>Arbitrary Slice Ordering (ASO)</b>	Yes	Yes	No	No	No
<b>Redundant Slices (RS)</b>	Yes	Yes	No	No	No
<b>Data Partitioning</b>	No	Yes	No	No	No
<b>Interlaced Coding (PicAFF, MBAFF)</b>	No	Yes	Yes	Yes	Yes
<b>4:2:0 Chroma Format</b>	Yes	Yes	Yes	Yes	Yes
<b>Monochrome Video Format (4:0:0)</b>	No	No	No	Yes	Yes
<b>4:2:2 Chroma Format</b>	No	No	No	No	No
<b>4:4:4 Chroma Format</b>	No	No	No	No	No
<b>8 Bit Sample Depth</b>	Yes	Yes	Yes	Yes	Yes
<b>9 and 10 Bit Sample Depth</b>	No	No	No	No	Yes
<b>11 to 14 Bit Sample Depth</b>	No	No	No	No	No
<b>8x8 vs. 4x4 Transform Adaptivity</b>	No	No	No	Yes	Yes
<b>Quantization Scaling Matrices</b>	No	No	No	Yes	Yes
<b>Separate Cb and Cr QP control</b>	No	No	No	Yes	Yes
<b>Separate Color Plane Coding</b>	No	No	No	No	No
<b>Predictive Lossless Coding</b>	No	No	No	No	No

1 See [http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf) at  
2 7:

3 The following table summarizes the two major types of entropy coding: Variable Length  
4 Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers  
5 superior coding efficiency over VLC by adapting to the changing probability distribution  
6 of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit  
7 correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length  
8 Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of  
9 CABAC.

#### 10 H.264 Entropy Coding – Comparison of Approaches

11 Characteristics	12 Variable Length Coding (VLC)	13 Context Adaptive Binary Arithmetic Coding(CABAC)
14 • Where it is used	MPEG-2, MPEG-4 ASP	H.264/MPEG-4 AVC (high efficiency option)
15 • Probability distribution	<b>Static</b> - Probabilities never change	<b>Adaptive</b> - Adjusts probabilities based on actual data
16 • Leverages correlation between symbols	<b>No</b> - Conditional probabilities ignored	<b>Yes</b> - Exploits symbol correlations by using "contexts"
17 • Non-integer code words	<b>No</b> - Low coding efficiency for high probability symbols	<b>Yes</b> - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency

18 Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to  
19 determine the correct decoder for the corresponding encoder. As shown below, if the  
20 flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then  
21 CABAC must have been selected as the encoder. See

22 [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items)  
23 [E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items) (Rec. ITU-T H.264 (04/2013)) at 80:

24 **entropy\_coding\_mode\_flag** selects the entropy decoding method to be applied for the syntax elements for which two  
25 descriptors appear in the syntax tables as follows:

- 26 – If **entropy\_coding\_mode\_flag** is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).
- 27 – Otherwise (**entropy\_coding\_mode\_flag** is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

1           15. The Accused Instrumentalities also practice wherein when the controller  
2 determines that the throughput falls below a predetermined throughput threshold, the  
3 controller commands the data compression engine to use one of the plurality of  
4 compression routines to provide a faster rate of compression so as to increase the  
5 throughput. For example, after its selection, the asymmetric compressor (CAVLC or  
6 CABAC) then compresses the data input to provide compressed data blocks (which  
7 can also be organized in a GOP structure) so as to increase the throughput, as  
8 discussed previously above. See [https://sonnati.wordpress.com/2007/10/29/how-h-](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/)  
9 [264-works-part-ii/](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/):  
10  
11

### 12 **Entropy Coding**

13 For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive  
14 variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic  
15 coding (CABAC) which are complex techniques to losslessly compress syntax elements in the  
16 video stream knowing the probabilities of syntax elements in a given context. The use of  
CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total  
processing power to be accomplished.

17 *See*

18 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf)  
19 [p1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf) at 13:  
20

21 Typical compression ratios to maintain excellent quality are:

- 22 • 10:1 for general images using JPEG
  - 23 • 30:1 for general video using H.263 and MPEG-2
  - 24 • 60:1 for general video using H.264 and WMV9
- 25  
26  
27  
28



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

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles include the functionality of the Baseline Profile and add improvements to the prediction algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

See [http://www.ijera.com/papers/Vol3\\_issue4/BM34399403.pdf](http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf) at 2:

16. Therefore, from at least the above, Adobe has directly infringed and continues to infringe the '046 patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by, for example, Claim 40 of the '046 patent, namely, a system, comprising: a data compression system for compressing and decompressing data input; a plurality of compression routines selectively utilized by the data compression system, wherein a first one of the plurality of compression routines includes a first compression algorithm and a second one of the plurality of compression routines includes a second

1 compression algorithm; and a controller for tracking throughput and generating a  
2 control signal to select a compression routine based on the throughput, wherein said  
3 tracking throughput comprises tracking a number of pending access requests to a  
4 storage device; and wherein when the controller determines that the throughput falls  
5 below a predetermined throughput threshold, the controller commands the data  
6 compression engine to use one of the plurality of compression routines to provide a  
7 faster rate of compression so as to increase the throughput. Upon information and  
8 belief, Adobe uses the Accused Instrumentalities to practice infringing methods for its  
9 own internal non-testing business purposes, while testing the Accused  
10 Instrumentalities, and while providing technical support and repair services for the  
11 Accused Instrumentalities to their customers.

12  
13  
14  
15 17. On information and belief, Adobe also directly infringes and continues to  
16 infringe other claims of the '046 patent.

17  
18 18. On information and belief, all of the Accused Instrumentalities perform  
19 the claimed methods in substantially the same way, e.g., in the manner specified in the  
20 H.264 standard.

21  
22 19. On information and belief, use of the Accused Instrumentalities in their  
23 ordinary and customary fashion results in infringement of the systems and/or methods  
24 claimed by the '046 patent.

25  
26 20. On information and belief, Adobe has had knowledge of the '046 patent  
27 since at least the filing of this Complaint or shortly thereafter, and on information and  
28 belief, Adobe knew of the '046 patent and knew of its infringement, including by way

1 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
2 receiving such notice) that its continued actions would actively induce and contribute  
3 to the infringement of the claims of the '046 patent.

4  
5 21. Upon information and belief, Adobe's affirmative acts of making, using,  
6 and selling the Accused Instrumentalities, and providing implementation services and  
7 technical support to users of the Accused Instrumentalities, including, e.g., through  
8 training, demonstrations, brochures, installation and user guides, have induced and  
9 continue to induce users of the Accused Instrumentalities to use them in their normal  
10 and customary way to infringe the '046 patent by practicing a system, comprising: a  
11 data compression system for compressing and decompressing data input; a plurality of  
12 compression routines selectively utilized by the data compression system, wherein a  
13 first one of the plurality of compression routines includes a first compression  
14 algorithm and a second one of the plurality of compression routines includes a second  
15 compression algorithm; and a controller for tracking throughput and generating a  
16 control signal to select a compression routine based on the throughput, wherein said  
17 tracking throughput comprises tracking a number of pending access requests to a  
18 storage device; and wherein when the controller determines that the throughput falls  
19 below a predetermined throughput threshold, the controller commands the data  
20 compression engine to use one of the plurality of compression routines to provide a  
21 faster rate of compression so as to increase the throughput. For example, Adobe  
22 adopted H.264 and/or H.264 with Scalable Video Coding (SVC) and/or an equivalent  
23 codec as its video codec in its products/services, such as, e.g., Adobe's media encoder,  
24  
25  
26  
27  
28

1 Flash player, Flash media server, Flash media server encoder, special effects and  
2 video editing software. For similar reasons, Adobe also induces its customers to use  
3 the Accused Instrumentalities to infringe other claims of the '046 patent. Adobe  
4 specifically intended and was aware that these normal and customary activities would  
5 infringe the '046 patent. Adobe performed the acts that constitute induced  
6 infringement, and would induce actual infringement, with the knowledge of the '046  
7 patent and with the knowledge, or willful blindness to the probability, that the induced  
8 acts would constitute infringement. On information and belief, Adobe engaged in  
9 such inducement to promote the sales of the Accused Instrumentalities. Accordingly,  
10 Adobe has induced and continues to induce users of the Accused Instrumentalities to  
11 use the Accused Instrumentalities in their ordinary and customary way to infringe the  
12 '046 patent, knowing that such use constitutes infringement of the '046 patent.  
13 Accordingly, Adobe has been, and currently is, inducing infringement of the '046  
14 patent, in violation of 35 U.S.C. § 271(b).

15  
16  
17  
18  
19 22. Adobe has also infringed, and continues to infringe, claims of the '046  
20 patent by offering to commercially distribute, commercially distributing, making,  
21 and/or importing the Accused Instrumentalities, which are used in practicing the  
22 process, or using the systems, of the '046 patent, and constitute a material part of the  
23 invention. Adobe knows the components in the Accused Instrumentalities to be  
24 especially made or especially adapted for use in infringement of the '046 patent, not a  
25 staple article, and not a commodity of commerce suitable for substantial noninfringing  
26 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '046  
27  
28

1 patent, in violation of 35 U.S.C. § 271(c).

2 23. By making, using, offering for sale, selling and/or importing into the  
3 United States the Accused Instrumentalities, and touting the benefits of using the  
4 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
5 liable to Realtime for infringement of the '046 patent pursuant to 35 U.S.C. § 271.  
6

7 24. As a result of Adobe's infringement of the '046 patent, Plaintiff Realtime  
8 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
9 infringement, but in no event less than a reasonable royalty for the use made of the  
10 invention by Adobe, together with interest and costs as fixed by the Court.  
11

12 **COUNT II**

13 **INFRINGEMENT OF U.S. PATENT NO. RE46,777**

14  
15 25. Plaintiff re-alleges and incorporates by reference the foregoing  
16 paragraphs, as if fully set forth herein.

17  
18 26. On information and belief, Adobe has made, used, offered for sale, sold  
19 and/or imported into the United States products that infringe the '777 patent, and  
20 continues to do so. By way of illustrative example, these infringing products include,  
21 without limitation, Adobe's Adobe Media Encoder, Adobe Media Encoder Creative  
22 Cloud or CC (e.g. CC 2015.1), Adobe Media Encoder Creative Suite or CS (e.g. CS4,  
23 CS5, CS5.5, CS6), Adobe Premiere Pro, Adobe Premiere Pro CC (e.g. CC 2015.1),  
24 Adobe Premiere Pro CS (e.g. CS4, CS5, CS5.5, CS6), Adobe After Effects, Adobe  
25 After Effects CC, Adobe After Effects CS (e.g. CS4, CS5, CS5.5, CS6), and all  
26 versions and variations thereof since the issuance of the '777 patent ("Accused  
27  
28

1 Instrumentalities”).

2           27. On information and belief, Adobe has directly infringed and continues to  
3 infringe the ‘777 patent, for example, through its sale, offer for sale, importation, use  
4 and testing of the Accused Instrumentalities, which practices the method claimed by,  
5 for example, Claim 1 of the ‘777 patent, namely, a method for coding a video signal  
6 using hybrid coding, comprising: reducing temporal redundancy by block based  
7 motion compensated prediction in order to establish a prediction error signal;  
8 performing quantization on samples of the prediction error signal or on coefficients  
9 resulting from a transformation of the prediction error signal into the frequency  
10 domain to obtain quantized values, representing quantized samples or quantized  
11 coefficients respectively, wherein the prediction error signal includes a plurality of  
12 subblocks each including a plurality of quantized values; calculating a first  
13 quantization efficiency for the quantized values of at least one subblock of the  
14 plurality of subblocks; setting the quantized values of the at least one subblock to all  
15 zeroes; calculating a second quantization efficiency for the at least one subblock while  
16 all of the quantized values are zeroes; selecting which of the first and second  
17 quantization efficiencies is a higher efficiency; and selecting, for further proceeding,  
18 the at least one subblock with the quantized values prior to setting the quantized  
19 values of the at least one subblock to all zeroes if the first quantization efficiency is  
20 higher and selecting the at least one subblock with the quantized values set to zero, for  
21 further proceeding, if the second quantization efficiency is higher. Upon information  
22 and belief, Adobe uses the Accused Instrumentalities to practice infringing methods  
23  
24  
25  
26  
27  
28

1 for its own internal non-testing business purposes, while testing the Accused  
2 Instrumentalities, and while providing technical support and repair services for the  
3 Accused Instrumentalities to Adobe’s customers.

4           28. The Accused Instrumentalities utilize H.265 (HEVC) and/or its  
5 equivalents by practicing a method for coding a video signal using hybrid coding.  
6 “The video coding layer of **HEVC** employs **the same hybrid approach** (inter-  
7 /intrapicture prediction and 2-D transform coding) used in all video compression  
8 standards since H.261.” Gary J Sullivan et al., Overview of the High Efficiency Video  
9 Coding (HEVC) Standard, 22 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR  
10 VIDEO TECHNOLOGY 1650 (December 2012) (“IEEE HEVC”),  
11 [http://iphome.hhi.de/wiegand/assets/pdfs/2012\\_12\\_IEEE-HEVC-Overview.pdf](http://iphome.hhi.de/wiegand/assets/pdfs/2012_12_IEEE-HEVC-Overview.pdf); *see*  
12 *also id.* at 1654 (“As in all prior ITU-T and ISO/IEC JTC 1 video coding standards  
13 since H.261, **the HEVC design follows the classic block-based hybrid video coding**  
14 **approach** (as depicted in Fig. 1).”) (citations omitted). Furthermore, the aim of the  
15 hybrid coding process is the production of a bitstream, as defined in definition 3.12 of  
16 the ITU-T H.265 Series H: Audiovisual and Multimedia Systems, “Infrastructure of  
17 audiovisual services – Coding of moving video” High efficiency video coding  
18 (“HEVC Spec”): “bitstream: A sequence of bits, in the form of a NAL unit stream or  
19 a byte stream, that forms the representation of coded pictures and associated data  
20 forming one or more coded video sequences (CVSs).” *See also, e.g.*, HEVC Spec at  
21 0.7 “Overview of the design characteristics.” Moreover, according to a website  
22 maintained by Adobe, describing an “Overview of What’s New in Adobe Media  
23  
24  
25  
26  
27  
28

1 Encoder CC 2015.1,” the Adobe “Media Encoder will offer Expanded UHD  
2 capabilities with the addition of **support for...the new HEVC (H.265) codec**, which,  
3 at 60% the size of comparable H.264 files, is ideal for delivering pristine 4K content  
4 online.” See <https://blogs.adobe.com/creativecloud/adobe-media-encoder-cc-2015-1/>.  
5 Also, according to another website maintained by Adobe discussing “Supported file  
6 formats” for Adobe Premiere Pro or specifically “Supported native video and audio  
7 formats for import” for Adobe Premiere Pro, the format of “**HEVC (H.265)**” is listed  
8 a long with the text “H.265 media with resolutions up to 8192x4320.” See  
9 <https://helpx.adobe.com/premiere-pro/using/supported-file-formats.html>. According to  
10 another website, “One of the most significant new features in Adobe Premiere Pro CC  
11 2015.1 is the expanded native support for 4K Ultra HD & HDR formats to  
12 include...**HEVC (H.265)**,” Adobe “Media Encoder will offer Expanded UHD  
13 capabilities with the addition of support for...the new **HEVC/H.265 codec**, which, at  
14 60% the size of comparable H.264 files, is ideal for delivering pristine 4K content  
15 online.” See [http://www.4kshooters.net/2015/12/04/adobe-premiere-pro-cc-2015-1-  
16 update-brings-h-265-more-4k-native-support/](http://www.4kshooters.net/2015/12/04/adobe-premiere-pro-cc-2015-1-update-brings-h-265-more-4k-native-support/). The fact that both Adobe Media  
17 Encoder and Adobe Premiere Pro CC supported HEVC/H.265 as early as 2015 has  
18 also been confirmed by other websites. See e.g.,  
19 [https://larryjordan.com/articles/adobe-media-encoder-and-h-265-not-ready-for-prime-  
20 time/](https://larryjordan.com/articles/adobe-media-encoder-and-h-265-not-ready-for-prime-time/) (“One of the exciting features of the Nov. 2015 update to Adobe Media Encoder  
21 is its support for **H.265, also known as HEVC**, short for High Efficiency Video  
22 Coding, compression.”).



1           29. The Accused Instrumentalities also practice reducing temporal  
2 redundancy by block based motion compensated prediction in order to establish a  
3 prediction error signal. For example, clause 8.5.3 Decoding process for prediction  
4 units in inter prediction mode and the subclauses thereof of the HEVC Spec describe  
5 the block based motion compensation techniques used in the decoding process. *See*  
6 *also, e.g., IEEE HEVC at 1651-1652 6) Motion compensation: Quarter-sample*  
7 *precision is used for the MVs, and 7-tap or 8-tap filters are used for interpolation of*  
8 *fractional-sample positions (compared to six-tap filtering of half-sample positions*  
9 *followed by linear interpolation for quarter-sample positions in H.264/MPEG-4*  
10 *AVC). Similar to H.264/MPEG-4 AVC, multiple reference pictures are used. For each*  
11 *PB, either one or two motion vectors can be transmitted, resulting either in*  
12 *unipredictive or bipredictive coding, respectively. As in H.264/MPEG-4 AVC, a*  
13 *scaling and offset operation may be applied to the prediction signal(s) in a manner*  
14 *known as weighted prediction.”).*  
15  
16  
17  
18

19           30. The Accused Instrumentalities also practice performing quantization on  
20 samples of the prediction error signal or on coefficients resulting from a  
21 transformation of the prediction error signal into the frequency domain to obtain  
22 quantized values, representing quantized samples or quantized coefficients  
23 respectively. For example, the quantization parameter and the scaling (inverse  
24 quantization) are defined in definitions 3.112 (page 10) and 3.131 (page 11),  
25 respectively, the usage of the scaling process in the decoding being described in clause  
26 and 8.6 Scaling, transformation and array construction process prior to deblocking  
27  
28

1 filter process of the HEVC Spec. *See also, e.g.,* IEEE HEVC at 1652 (“8)  
2 Quantization control: As in H.264/MPEG-4 AVC, uniform reconstruction  
3 quantization (URQ) is used in HEVC, with quantization scaling matrices supported  
4 for the various transform block sizes.”).

6 31. The Accused Instrumentalities also practice performing a method  
7 wherein the prediction error signal includes a plurality of subblocks each including a  
8 plurality of quantized values. For example, the quantized samples or transform  
9 coefficients from the subblock are scaled and transformed as described in above  
10 mentioned clause 8.6 of the HEVC Spec. *See also, e.g.,* IEEE HEVC at 1652  
11 (“Prediction units and prediction blocks (PBs): The decision whether to code a picture  
12 area using interpicture or intrapicture prediction is made at the CU level. A PU  
13 partitioning structure has its root at the CU level. Depending on the basic prediction-  
14 type decision, the luma and chroma CBs can then be further split in size and predicted  
15 from luma and chroma prediction blocks (PBs). HEVC supports variable PB sizes  
16 from  $64 \times 64$  down to  $4 \times 4$  samples.”).

20 32. The Accused Instrumentalities also practice performing a method of  
21 calculating a first quantization efficiency for the quantized values of at least one  
22 subblock of the plurality of subblocks; setting the quantized values of the at least one  
23 subblock to all zeroes; calculating a second quantization efficiency for the at least one  
24 subblock while all of the quantized values are zeroes; selecting which of the first and  
25 second quantization efficiencies is a higher efficiency; and selecting, for further  
26 proceeding, the at least one subblock with the quantized values prior to setting the  
27  
28

1 quantized values of the at least one subblock to all zeroes if the first quantization  
2 efficiency is higher and selecting the at least one subblock with the quantized values  
3 set to zero, for further proceeding, if the second quantization efficiency is higher. For  
4 example, the bitstream resulting from the encoding as described in this last item of the  
5 claim contains all the relevant information as needed by the decoder for proper  
6 decoding. If the coefficients of the subblock are set to zero as a consequence of the  
7 efficiency calculation, the coded\_sub\_block\_flag, as described in clause 7.4.9.11  
8 Residual coding semantics, HEVC Spec, is set to 0, indicating that all the 16  
9 coefficients of the coded sub block have been set to 0:  
10  
11 “coded\_sub\_block\_flag[ xS ][ yS ] specifies the following for the sub-block at  
12 location ( xS, yS ) within the current transform block, where a sub-block is a (4x4)  
13 array of 16 transform coefficient levels: – If coded\_sub\_block\_flag[ xS ][ yS ] is equal  
14 to 0, the 16 transform coefficient levels of the sub-block at location ( xS, yS ) are  
15 inferred to be equal to 0.”  
16  
17

18  
19 33. When coded\_sub\_block\_flag[ xS ][ yS ] has not been set equal to 0, the  
20 position in the array of non 0 coefficients can be determined as follows:

21 – Otherwise (coded\_sub\_block\_flag[ xS ][ yS ] is equal to 1), the following  
22 applies:  
23

24 – If ( xS, yS ) is equal to ( 0, 0 ) and ( LastSignificantCoeffX,  
25 LastSignificantCoeffY ) is not equal to ( 0, 0 ), at least one of the 16 sig\_coeff\_flag  
26 syntax elements is present for the sub-block at location ( xS, yS ) .  
27

28 – Otherwise, at least one of the 16 transform coefficient levels of the sub-

1 block at location (  $x_S$ ,  $y_S$  ) has a non zero value.

2 When `coded_sub_block_flag[  $x_S$  ][  $y_S$  ]` is not present, it is inferred as follows:

3 – If one or more of the following conditions are true,

4 `coded_sub_block_flag[  $x_S$  ][  $y_S$  ]` is inferred to be equal to 1:

5 – (  $x_S$ ,  $y_S$  ) is equal to ( 0, 0 )

6 – (  $x_S$ ,  $y_S$  ) is equal to (  $\text{LastSignificantCoeffX} \gg 2$  ,

7  $\text{LastSignificantCoeffY} \gg 2$  )

8 – Otherwise, `coded_sub_block_flag[  $x_S$  ][  $y_S$  ]` is inferred to be equal to 0.

9  
10  
11 HEVC Spec at 7.4.9.11 Residual coding semantics. Therefore, even though the  
12 coding algorithms than can be used for reaching specific efficiency targets are not  
13 specified by the HEVC Spec (as stated in clause 0.7), this particular combination of  
14 choices produces a valid bitstream that has to be decoded by a conformant decoder.  
15

16 34. The infringement of the Accused Instrumentalities is also shown by way  
17 of considering the reference software (*see, e.g.*, <https://hevc.hhi.fraunhofer.de/>).

18  
19 Setting the flag `RDOQ=true` in the encoder configuration file enables rate-distortion-  
20 optimized quantization for transformed TUs. This feature is implemented in the HM  
21 reference software as function `xRateDistOptQuant` in file `TComTrQuant.cpp`. In the  
22 function `xRateDistOptQuant`, the efficiency for setting all quantized values to zero is  
23 calculated and stored in the variable `d64BestCost`. In the variable `iBestLastIdxP1`, a 0  
24 is stored indicating that all values starting from the 0th position are set to zero.  
25

26 Afterwards, the efficiency for keeping quantized values unequal to zero is calculated  
27 and stored in the variable `totalCost`. The variable `iBestLastIdxP1` is adjusted  
28

1 correspondingly to values unequal to 0. The two efficiencies d64BestCost and  
 2 totalCost are compared, and selecting for further proceeding either quantized values,  
 3 which are all set to zero or quantized values, which are not all set to zero. All values  
 4 starting from the position defined by the variable iBestLastIdxP1 are set to zero.  
 5

6 35. Calculation of the efficiency for setting all quantized values to zero and  
 7 storing the result in the variable d64BestCost:

```

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
Double d64BestCost = 0;
Int ui16CtxCbf = 0;
Int iBestLastIdxP1 = 0;
if( !pcCU->isIntra( uiAbsPartIdx ) && isLuma(compID) && pcCU->getTransformIdx( uiAbsPartIdx ) == 0 )
{
    ui16CtxCbf = 0;
    d64BestCost = d64BlockUncodedCost + xGetICost( m_pcEstBitsSbac->blockRootCbpBits[ ui16CtxCbf ][ 0 ] );
    d64BaseCost += xGetICost( m_pcEstBitsSbac->blockRootCbpBits[ ui16CtxCbf ][ 1 ] );
}
else
{
    ui16CtxCbf = pcCU->getCtxQtCbf( rTu, channelType );
    ui16CtxCbf += getCBFContextOffset(compID);
    d64BestCost = d64BlockUncodedCost + xGetICost( m_pcEstBitsSbac->blockCbpBits[ ui16CtxCbf ][ 0 ] );
    d64BaseCost += xGetICost( m_pcEstBitsSbac->blockCbpBits[ ui16CtxCbf ][ 1 ] );
}

```

HEVC Reference Software (<https://hevc.hhi.fraunhofer.de/>).

14 36. Calculating the efficiency for keeping quantized values unequal to zero  
 15 and storing the result in the variable totalCost:

```

17
18
19
20
21
22
23
24
25
26
27
28
Bool bFoundLast = false;
for (Int iCGScanPos = iCGLastScanPos; iCGScanPos >= 0; iCGScanPos--)
{
    UInt uiCGBlkPos = codingParameters.scanCG[ iCGScanPos ];

    d64BaseCost -= pdCostCoeffGroupSig[ iCGScanPos ];
    if (uiSigCoeffGroupFlag[ uiCGBlkPos ])
    {
        for (Int iScanPosinCG = uiCGSize-1; iScanPosinCG >= 0; iScanPosinCG--)
        {
            iScanPos = iCGScanPos*uiCGSize + iScanPosinCG;

            if (iScanPos > iLastScanPos) continue;
            UInt uiBlkPos = codingParameters.scan[iScanPos];

            if( piDstCoeff[ uiBlkPos ] )
            {
                UInt uiPosY = uiBlkPos >> uiLog2BlockWidth;
                UInt uiPosX = uiBlkPos - ( uiPosY << uiLog2BlockWidth );

                Double d64CostLast= codingParameters.scanType == SCAN_VER ? xGetRateLast( uiPosY, uiPosX, compID ) :
                xGetRateLast( uiPosX, uiPosY, compID );
                Double totalCost = d64BaseCost + d64CostLast - pdCostSig[ iScanPos ];
            }
        }
    }
}

```

HEVC Reference Software (<https://hevc.hhi.fraunhofer.de/>).

37. Comparing the two efficiencies d64BestCost and totalCost:

```

if( totalCost < d64BestCost )
{
    iBestLastIdxP1 = iScanPos + 1;
    d64BestCost = totalCost;
}

```

1 HEVC Reference Software (<https://hevc.hhi.fraunhofer.de/>).

2 38. Selecting for further proceeding either quantized values, which are all set  
3 to zero or quantized values, which are not all set to zero:  
4

```
5 //===== clean uncoded coefficients =====  
6 for ( Int scanPos = iBestLastIdxP1; scanPos <= iLastScanPos; scanPos++ )  
7 {  
8     piDstCoeff[ codingParameters.scan[ scanPos ] ] = 0;  
9 }
```

8 HEVC Reference Software (<https://hevc.hhi.fraunhofer.de/>).

9  
10 39. Therefore, from at least the above, Adobe has directly infringed and  
11 continues to infringe the '777 patent, for example, through its own use and testing of  
12 the Accused Instrumentalities, which when used, practices the method claimed by, for  
13 example, Claim 1 of the '777 patent, namely, a method for coding a video signal using  
14 hybrid coding, comprising: reducing temporal redundancy by block based motion  
15 compensated prediction in order to establish a prediction error signal; performing  
16 quantization on samples of the prediction error signal or on coefficients resulting from  
17 a transformation of the prediction error signal into the frequency domain to obtain  
18 quantized values, representing quantized samples or quantized coefficients  
19 respectively, wherein the prediction error signal includes a plurality of subblocks each  
20 including a plurality of quantized values; calculating a first quantization efficiency for  
21 the quantized values of at least one subblock of the plurality of subblocks; setting the  
22 quantized values of the at least one subblock to all zeroes; calculating a second  
23 quantization efficiency for the at least one subblock while all of the quantized values  
24 are zeroes; selecting which of the first and second quantization efficiencies is a higher  
25  
26  
27  
28

1 efficiency; and selecting, for further proceeding, the at least one subblock with the  
2 quantized values prior to setting the quantized values of the at least one subblock to all  
3 zeroes if the first quantization efficiency is higher and selecting the at least one  
4 subblock with the quantized values set to zero, for further proceeding, if the second  
5 quantization efficiency is higher. Upon information and belief, Adobe uses the  
6 Accused Instrumentalities to practice infringing methods for its own internal non-  
7 testing business purposes, while testing the Accused Instrumentalities, and while  
8 providing technical support and repair services for the Accused Instrumentalities to  
9 their customers.  
10  
11

12 40. On information and belief, Adobe also directly infringes and continues to  
13 infringe other claims of the '777 patent.  
14

15 41. On information and belief, all of the Accused Instrumentalities perform  
16 the claimed methods in substantially the same way, e.g., in the manner specified in the  
17 HEVC (or H.265) standard.  
18

19 42. On information and belief, use of the Accused Instrumentalities in their  
20 ordinary and customary fashion results in infringement of the methods and/or systems  
21 claimed by the '777 patent.  
22

23 43. On information and belief, Adobe has had knowledge of the '777 patent  
24 since at least the filing of this Complaint or shortly thereafter, and on information and  
25 belief, Adobe knew of the '777 patent and knew of its infringement, including by way  
26 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
27 receiving such notice) that its continued actions would actively induce and contribute  
28

1 to the infringement of the claims of the '777 patent.

2 44. Upon information and belief, Adobe's affirmative acts of making, using,  
3 and selling the Accused Instrumentalities, and providing implementation services and  
4 technical support to users of the Accused Instrumentalities, including, e.g., through  
5 training, demonstrations, brochures, installation and user guides, have induced and  
6 continue to induce users of the Accused Instrumentalities to use them in their normal  
7 and customary way to infringe the '777 patent by practicing a method for coding a  
8 video signal using hybrid coding, comprising: reducing temporal redundancy by block  
9 based motion compensated prediction in order to establish a prediction error signal;  
10 performing quantization on samples of the prediction error signal or on coefficients  
11 resulting from a transformation of the prediction error signal into the frequency  
12 domain to obtain quantized values, representing quantized samples or quantized  
13 coefficients respectively, wherein the prediction error signal includes a plurality of  
14 subblocks each including a plurality of quantized values; calculating a first  
15 quantization efficiency for the quantized values of at least one subblock of the  
16 plurality of subblocks; setting the quantized values of the at least one subblock to all  
17 zeroes; calculating a second quantization efficiency for the at least one subblock while  
18 all of the quantized values are zeroes; selecting which of the first and second  
19 quantization efficiencies is a higher efficiency; and selecting, for further proceeding,  
20 the at least one subblock with the quantized values prior to setting the quantized  
21 values of the at least one subblock to all zeroes if the first quantization efficiency is  
22 higher and selecting the at least one subblock with the quantized values set to zero, for  
23  
24  
25  
26  
27  
28



1 further proceeding, if the second quantization efficiency is higher. For example,  
2 Adobe adopted HEVC (or H.265) and/or its equivalents as its video codec in its  
3 products/services, such as in Adobe's media encoder, Flash player, Flash media  
4 server, Flash media server encoder, special effects and video editing software. For  
5 similar reasons, Adobe also induces its customers to use the Accused Instrumentalities  
6 to infringe other claims of the '777 patent. Adobe specifically intended and was  
7 aware that these normal and customary activities would infringe the '777 patent.  
8 Adobe performed the acts that constitute induced infringement, and would induce  
9 actual infringement, with the knowledge of the '777 patent and with the knowledge, or  
10 willful blindness to the probability, that the induced acts would constitute  
11 infringement. On information and belief, Adobe engaged in such inducement to  
12 promote the sales of the Accused Instrumentalities. Accordingly, Adobe has induced  
13 and continues to induce users of the Accused Instrumentalities to use the Accused  
14 Instrumentalities in their ordinary and customary way to infringe the '777 patent,  
15 knowing that such use constitutes infringement of the '777 patent. Accordingly,  
16 Adobe has been, and currently is, inducing infringement of the '777 patent, in  
17 violation of 35 U.S.C. § 271(b).  
18  
19  
20  
21  
22

23 45. Adobe has also infringed, and continues to infringe, claims of the '777  
24 patent by offering to commercially distribute, commercially distributing, making,  
25 and/or importing the Accused Instrumentalities, which are used in practicing the  
26 process, or using the systems, of the '777 patent, and constitute a material part of the  
27 invention. Adobe knows the components in the Accused Instrumentalities to be  
28

1 especially made or especially adapted for use in infringement of the '777 patent, not a  
2 staple article, and not a commodity of commerce suitable for substantial noninfringing  
3 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '777  
4 patent, in violation of 35 U.S.C. § 271(c).

5  
6 46. By making, using, offering for sale, selling and/or importing into the  
7 United States the Accused Instrumentalities, and touting the benefits of using the  
8 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
9 liable to Realtime for infringement of the '777 patent pursuant to 35 U.S.C. § 271.  
10

11 47. As a result of Adobe's infringement of the '777 patent, Plaintiff Realtime  
12 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
13 infringement, but in no event less than a reasonable royalty for the use made of the  
14 invention by Adobe, together with interest and costs as fixed by the Court.  
15

16 **COUNT III**

17 **INFRINGEMENT OF U.S. PATENT NO. 8,929,442**

18  
19 48. Plaintiff re-alleges and incorporates by reference the foregoing  
20 paragraphs, as if fully set forth herein.

21 49. On information and belief, Adobe has made, used, offered for sale, sold  
22 and/or imported into the United States Adobe products that infringe the '442 patent,  
23 and continues to do so. By way of illustrative example, these infringing products  
24 include, without limitation, Adobe's Adobe Premiere Pro, Adobe Premiere Pro CC,  
25 Adobe Premiere Pro CS (e.g. CS4, CS5, CS5.5, CS6), Adobe Flash, Adobe Flash  
26 Player (e.g. Adobe Flash Player 9), Adobe Flash Media Server, Adobe After Effects,  
27  
28

1 Adobe After Effects CC, Adobe After Effects CS (e.g. CS4, CS5, CS5.5, CS6), Adobe  
2 HTTP Dynamic Streaming (HDS), and all versions and variations thereof since the  
3 issuance of the '442 patent ("Accused Instrumentalities").

4  
5 50. On information and belief, Adobe has directly infringed and continues to  
6 infringe the '442 patent, for example, through its sale, offer for sale, importation, use  
7 and testing of the Accused Instrumentalities, which practices the apparatus claimed  
8 by, for example, Claim 8 of the '442 patent, namely, an apparatus, comprising: a data  
9 decompression system configured to decompress a compressed data block; and a  
10 storage medium configured to store at least a portion of the decompressed data block,  
11 wherein at least a portion of a data block having video or audio data was compressed  
12 with one or more compression algorithms selected from among a plurality of  
13 compression algorithms based upon a throughput of a communication channel and a  
14 parameter or an attribute of the at least the portion of the data block to create at least  
15 the compressed data block, and wherein at least one of the plurality of compression  
16 algorithms is asymmetric. Upon information and belief, Adobe uses the Accused  
17 Instrumentalities to practice infringing methods for its own internal non-testing  
18 business purposes, while testing the Accused Instrumentalities, and while providing  
19 technical support and repair services for the Accused Instrumentalities to Adobe's  
20 customers.

21  
22 51. The Accused Instrumentalities include, or practice an apparatus,  
23 comprising: a data decompression system configured to decompress a compressed  
24 data block. For example, the Accused Instrumentalities utilize H.264 and/or H.264  
25  
26  
27  
28

1 with Scalable Video Coding (SVC) and/or an equivalent codec. H.264 is “the most  
2 widely used codec on the planet.” *See*  
3 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx)  
4 [74735.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx). A “codec” is also a compression technology that has “two components,  
5 an encoder to compress the files, and a **decoder to decompress**. There are codecs  
6 for...video (Cinepark, MPEG-2, **H.264**, VP8).” *See*  
7 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx)  
8 [74487.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx); <https://forums.adobe.com/thread/729526> (“Compression is essential for  
9 reducing the size of movies so that they can be stored, transmitted, and played back  
10 effectively. Compression is achieved by an encoder; decompression is achieved by a  
11 decoder. Encoders and decoders are known by the common term codec.”). According  
12 to a website maintained by Adobe, “you **can produce H.264 video** with Adobe Media  
13 Encoder CS4 and Adobe Flash Media Encoding Server 3.5,” “Adobe added **H.264**  
14 **playback support** to Adobe Flash Player 9 Update 3 back in 2007,” Adobe “Flash  
15 Media Server and Flash Player can **both stream and play back any H.264 file** in  
16 virtually any format,” the encoding capabilities of the Adobe Flash Media Encoding  
17 Server using “**H.264-related parameters**” are described (e.g. using CAVLC, CABAC)  
18 and under a headline stating “Adobe Media Encoder,” the website states: “Adobe  
19 significantly enhanced the Flash Video Encoder in Creative Suite 4. There is now both  
20 stand-alone operation and batch encoding capabilities. As before, you can **access**  
21 **H.264 encoding** by choosing different formats in the Format pop-up menu. When  
22 producing for Flash Player, you should always use the FLV|F4V option, which lets  
23  
24  
25  
26  
27  
28

1 you produce both **VP6- and H.264-encoded files** for Flash Player distribution.” See

2 [http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

3 Another website maintained by Adobe mentions that “You have two primary options

4 for using Adobe Media Encoder with After Effects to create videos in **H.264**.” The

5 website also mentions that “Since Adobe Media Encoder already has superior

6 H.264...exporting capabilities, it was more prudent to rely on Adobe Media Encoder

7 for export of these formats.” See [https://helpx.adobe.com/after-effects/kb/export-](https://helpx.adobe.com/after-effects/kb/export-h264.html)

8 [h264.html](https://helpx.adobe.com/after-effects/kb/export-h264.html). Another website maintained by Adobe mentions that the “Adobe Flash

9 Player 9 Update 3 is taking a step into the high-definition (HD) video realm in a major

10 way by adding MPEG-4 video...[which] utilizes crisp, powerful **H.264 encoding**”

11 and “The new Adobe Media Player...will **also support H.264**” and “The addition of

12 **H.264...support** in Flash Player 9 Update 3 allows you to easily use high-definition,

13 industry standard video and audio.” See

14 [http://www.adobe.com/devnet/flashplayer/articles/hd\\_video\\_flash\\_player.html](http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html). Adobe

15 HTTP Dynamic Streaming (HDS) also supports additional data compression systems

16 or codecs other than H.264 because it “enables high-quality (**H.264 or VP6**), network-

17 efficient HTTP streaming for media delivery that is tightly integrated with Adobe®

18 Access software for robust content protection in the Adobe Flash® Player 10.1 or later

19 and Adobe AIR®2 or later runtimes,” is also “built on standards and deployed using

20 standard HTTP servers (Apache), standard media format (MP4 fragment) using

21 standard codecs (**H.264/AAC**), open APIs (Flash Player), and an open source

22 framework for building media players (OSMF)” and “like other Flash Player

1 supported delivery methods,” Adobe HDS “supports **H.264 video and VP6 codecs**  
2 required for the highest quality video on the platform.” *See*  
3 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS also  
4 allows users to encode content “using high-quality Flash Player compatible **codecs**  
5 (**VP6/MP3, H264/AAC**),” deliver “high-definition video up to 1080p, with bitrates  
6 from 700kbps up to and beyond 6Mbps, using either **H.264 or VP6 video codecs**,”  
7 and archive “live, high-definition streams on the server and enable HD DVR  
8 functionality (for example, instant replay and time shifting) with support for **H.264**  
9 stream recording for RTMP and HTTP Dynamic Streaming.” *See*  
10 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Adobe Media  
11 Encoder also “lets you produce both **VP6- and H.264-encoded files** for Flash Player  
12 distribution.” *See* [http://www.adobe.com/devnet/adobe-media-](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
13 [server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

17 52. The Accused Instrumentalities also include or practice an apparatus  
18 comprising: a storage medium configured to store at least a portion of the  
19 decompressed data block. For example, the Accused Instrumentalities include volatile  
20 and non-volatile memory (e.g., RAM, flash, etc.) configured to store at least a portion  
21 of the decompressed data block. *See* [https://streaminglearningcenter.com/articles/ram-](https://streaminglearningcenter.com/articles/ram-requirements-for-adobe-cs5-5.html)  
22 [requirements-for-adobe-cs5-5.html](https://streaminglearningcenter.com/articles/ram-requirements-for-adobe-cs5-5.html) (“RAM Requirements for Adobe CS5.5”);  
23 [https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html)  
24 [streaming.html](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html) (“Adobe HTTP Dynamic Streaming combines these approaches to  
25 introduce HTTP streaming to the Flash Platform. HTTP Dynamic Streaming packages  
26  
27  
28

1 media files into fragments that Flash Player clients can access instantly without  
2 downloading the entire file. Adobe HTTP Dynamic Streaming contains several  
3 components that work together to package media and stream it over HTTP to Flash  
4 Player and AIR. HTTP Dynamic Streaming supports multi-bitrate streaming, DVR,  
5 and Adobe® Flash® Access™ protection.”). On information and belief, the Accused  
6 Instrumentalities also include a storage medium configured to store at least a portion  
7 of the decompressed data block e.g., hard disk, disks, buffers, servers or other forms  
8 of memory/storage. *See* [https://helpx.adobe.com/adobe-media-server/dev/configure-](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html)  
9 [dynamic-streaming-live-streaming.html](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html) (describing “Streams [from Adobe HDS] in a  
10 live event are packaged as fragments and written to **disk**,” “The **IO buffer** loads the  
11 **disk file** into an **in-memory buffer**,” and “To limit the amount of **storage the disk**  
12 **cache** uses,” and a “Content storage (HDS and HLS)” header describing a section  
13 where a “media player **requests content from the server.**”  
14  
15  
16  
17

18 53. The Accused Instrumentalities also practice wherein at least a portion of  
19 a data block having video or audio data was compressed with one or more  
20 compression algorithms selected from among a plurality of compression algorithms  
21 based upon a throughput of a communication channel and a parameter or an attribute  
22 of the at least the portion of the data block to create at least the compressed data block.  
23 For throughput of a communication channel or bandwidth, Adobe HTTP Dynamic  
24 Streaming (HDS) contains the “**ability to shift quality depending on bandwidth and**  
25 **computer power**. HTTP progressive delivery consumes more bandwidth because it's  
26 not intelligent enough to throttle the delivery (for example, a 30-minute video will be  
27  
28

1 fully downloaded whether the user watches it or not).” See  
2 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS can  
3 also “[d]etect the client’s bandwidth and computer resources and serve them  
4 content fragments encoded at the most appropriate bitrate.” See  
5 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Examples of  
6 the aforementioned parameters or attributes of portions of data blocks include  
7 throughput or bandwidth, bitrate (or max video bitrate) and resolution parameters.  
8 Different parameters correspond with different end applications. H.264 provides for  
9 multiple different ranges of such parameters, each included in the “profiles” and  
10 “levels” as defined by the H.264 standard, from the below shown paragraphs from a  
11 white paper and Wikipedia. See  
12 [http://www.axis.com/files/whitepaper/wp\\_h264\\_31669\\_en\\_0803\\_lo.pdf](http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf) at 5:  
13  
14  
15  
16  
17

#### 18 4. H.264 profiles and levels

19 The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting  
20 options and features to a minimum. An important aspect of the standard, as with other video standards,  
21 is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes)  
22 that optimally support popular productions and common formats.

23 H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what  
24 feature set the encoder may use and limits the decoder implementation complexity.

25 Network cameras and video encoders will most likely use a profile called the baseline profile, which is  
26 intended primarily for applications with limited computing resources. The baseline profile is the most  
27 suitable given the available performance in a real-time encoder that is embedded in a network video  
28 product. The profile also enables low latency, which is an important requirement of surveillance video and  
also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements.  
Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging  
from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

See [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC):



Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) <a href="#">Toggle additional details</a>
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	176x144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	176x144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	352x288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	352x288@15.2 (6)
1.3	3,041,280	11,880	101,376	396	768	960	2,304	352x288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	352x288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352x576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	720x576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	720x576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	1,280x720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280x1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048x1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048x1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048x1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	3,672x1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@56.3 (5)

54. A video data block is organized by the group of pictures (GOP) structure, which is a “collection of successive pictures within a coded video stream.” *See* [https://en.wikipedia.org/wiki/Group\\_of\\_pictures](https://en.wikipedia.org/wiki/Group_of_pictures). A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See* [https://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](https://en.wikipedia.org/wiki/Video_compression_picture_types) (for descriptions of I frames, P frames and B frames); <https://en.wikipedia.org/wiki/MPEG-1#D-frames> (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP

1 structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max  
2 video bitrate and resolution parameters) or even be considered as a parameter by itself.

3         55. The Accused Instrumentalities also practice wherein at least one of the  
4 plurality of compression algorithms is asymmetric. Based on the bitrate and/or  
5 resolution parameter identified (e.g. bitrate, max video bitrate, resolution, GOP  
6 structure or frame type within a GOP structure), any H.264-compliant system such as  
7 the Accused Instrumentalities would determine which profile (e.g., “baseline,”  
8 “extended,” “main”, or “high”) corresponds with that parameter, then select between  
9 at least two asymmetric compressors. If baseline or extended is the corresponding  
10 profile, then the system will select a Context-Adaptive Variable Length Coding  
11 (“CAVLC”) entropy encoder. If main or high is the corresponding profile, then the  
12 system will select a Context-Adaptive Binary Arithmetic Coding (“CABAC”) entropy  
13 encoder. Both encoders are asymmetric compressors because it takes a longer period  
14 of time for them to compress data than to decompress data. *See*

15 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

	<b>Baseline</b>	<b>Extended</b>	<b>Main</b>	<b>High</b>	<b>High 10</b>
<b>I and P Slices</b>	Yes	Yes	Yes	Yes	Yes
<b>B Slices</b>	No	Yes	Yes	Yes	Yes
<b>SI and SP Slices</b>	No	Yes	No	No	No
<b>Multiple Reference Frames</b>	Yes	Yes	Yes	Yes	Yes
<b>In-Loop Deblocking Filter</b>	Yes	Yes	Yes	Yes	Yes
<b>CAVLC Entropy Coding</b>	Yes	Yes	Yes	Yes	Yes
<b>CABAC Entropy Coding</b>	No	No	Yes	Yes	Yes
<b>Flexible Macroblock Ordering (FMO)</b>	Yes	Yes	No	No	No
<b>Arbitrary Slice Ordering (ASO)</b>	Yes	Yes	No	No	No
<b>Redundant Slices (RS)</b>	Yes	Yes	No	No	No
<b>Data Partitioning</b>	No	Yes	No	No	No
<b>Interlaced Coding (PicAFF, MBAFF)</b>	No	Yes	Yes	Yes	Yes
<b>4:2:0 Chroma Format</b>	Yes	Yes	Yes	Yes	Yes
<b>Monochrome Video Format (4:0:0)</b>	No	No	No	Yes	Yes
<b>4:2:2 Chroma Format</b>	No	No	No	No	No
<b>4:4:4 Chroma Format</b>	No	No	No	No	No
<b>8 Bit Sample Depth</b>	Yes	Yes	Yes	Yes	Yes
<b>9 and 10 Bit Sample Depth</b>	No	No	No	No	Yes
<b>11 to 14 Bit Sample Depth</b>	No	No	No	No	No
<b>8x8 vs. 4x4 Transform Adaptivity</b>	No	No	No	Yes	Yes
<b>Quantization Scaling Matrices</b>	No	No	No	Yes	Yes
<b>Separate Cb and Cr QP control</b>	No	No	No	Yes	Yes
<b>Separate Color Plane Coding</b>	No	No	No	No	No
<b>Predictive Lossless Coding</b>	No	No	No	No	No

See

[http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf) at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

#### H.264 Entropy Coding – Comparison of Approaches

Characteristics	Variable Length Coding (VLC)	Context Adaptive Binary Arithmetic Coding(CABAC)
• Where it is used	MPEG-2, MPEG-4 ASP	H.264/MPEG-4 AVC (high efficiency option)
• Probability distribution	<b>Static</b> - Probabilities never change	<b>Adaptive</b> - Adjusts probabilities based on actual data
• Leverages correlation between symbols	<b>No</b> - Conditional probabilities ignored	<b>Yes</b> - Exploits symbol correlations by using "contexts"
• Non-integer code words	<b>No</b> - Low coding efficiency for high probability symbols	<b>Yes</b> - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items) (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy\_coding\_mode\_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

- If **entropy\_coding\_mode\_flag** is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).
- Otherwise (**entropy\_coding\_mode\_flag** is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

56. The Accused Instrumentalities practice at least a portion of a data block having video or audio data that was compressed with one or more compression

1 algorithms selected from among a plurality of compression algorithms...and wherein  
2 at least one of the plurality of compression algorithms is asymmetric. The compressed  
3 data blocks can also be organized in a GOP structure (see above). After its selection,  
4 the **asymmetric compressor** (CAVLC or CABAC) will compress the video or audio  
5 data to provide various compressed data blocks, which can also be organized in a  
6 GOP structure, as discussed previously above. *See*

8 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
9

10 **Entropy Coding**

11 For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive  
12 variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic  
13 coding (CABAC) which are complex techniques to losslessly compress syntax elements in the  
14 video stream knowing the probabilities of syntax elements in a given context. The use of  
15 CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total  
16 processing power to be accomplished.

17 *See*

18 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=  
19 pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf) at 13:  
20

21 Typical compression ratios to maintain excellent quality are:

- 22 • 10:1 for general images using JPEG
  - 23 • 30:1 for general video using H.263 and MPEG-2
  - 24 • 60:1 for general video using H.264 and WMV9
- 25  
26  
27  
28

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

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles include the functionality of the Baseline Profile and add improvements to the prediction algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

See [http://www.ijera.com/papers/Vol3\\_issue4/BM34399403.pdf](http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf) at 2:

57. Therefore, from at least the above, Adobe has directly infringed and continues to infringe the '442 patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the apparatus claimed by, for example, Claim 8 of the '442 patent, namely, an apparatus, comprising: a data decompression system configured to decompress a compressed data block; and a storage medium configured to store at least a portion of the decompressed data block, wherein at least a portion of a data block having video or audio data was compressed with one or more compression algorithms selected from among a plurality of

1 compression algorithms based upon a throughput of a communication channel and a  
2 parameter or an attribute of the at least the portion of the data block to create at least  
3 the compressed data block, and wherein at least one of the plurality of compression  
4 algorithms is asymmetric. Upon information and belief, Adobe uses the Accused  
5 Instrumentalities to practice infringing methods for its own internal non-testing  
6 business purposes, while testing the Accused Instrumentalities, and while providing  
7 technical support and repair services for the Accused Instrumentalities to their  
8 customers.  
9  
10

11 58. On information and belief, Adobe also directly infringes and continues to  
12 infringe other claims of the '442 patent.  
13

14 59. On information and belief, all of the Accused Instrumentalities perform  
15 the claimed methods in substantially the same way, e.g., in the manner specified in the  
16 H.264 standard.  
17

18 60. On information and belief, use of the Accused Instrumentalities in their  
19 ordinary and customary fashion results in infringement of the systems and/or methods  
20 claimed by the '442 patent.  
21

22 61. On information and belief, Adobe has had knowledge of the '442 patent  
23 since at least the filing of this Complaint or shortly thereafter, and on information and  
24 belief, Adobe knew of the '442 patent and knew of its infringement, including by way  
25 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
26 receiving such notice) that its continued actions would actively induce and contribute  
27 to the infringement of the claims of the '442 patent.  
28

1           62. Upon information and belief, Adobe's affirmative acts of making, using,  
2 and selling the Accused Instrumentalities, and providing implementation services and  
3 technical support to users of the Accused Instrumentalities, including, e.g., through  
4 training, demonstrations, brochures, installation and user guides, have induced and  
5 continue to induce users of the Accused Instrumentalities to use them in their normal  
6 and customary way to infringe the '442 patent by practicing, for example, an  
7 apparatus, comprising: a data decompression system configured to decompress a  
8 compressed data block; and a storage medium configured to store at least a portion of  
9 the decompressed data block, wherein at least a portion of a data block having video  
10 or audio data was compressed with one or more compression algorithms selected from  
11 among a plurality of compression algorithms based upon a throughput of a  
12 communication channel and a parameter or an attribute of the at least the portion of  
13 the data block to create at least the compressed data block, and wherein at least one of  
14 the plurality of compression algorithms is asymmetric. For example, Adobe adopted  
15 H.264 and/or H.264 with Scalable Video Coding (SVC) and/or an equivalent codec as  
16 its video codec in its products/services, such as, e.g., Adobe's media encoder, Flash  
17 player, Flash media server, Flash media server encoder, special effects and video  
18 editing software. For similar reasons, Adobe also induces its customers to use the  
19 Accused Instrumentalities to infringe other claims of the '442 patent. Adobe  
20 specifically intended and was aware that these normal and customary activities would  
21 infringe the '442 patent. Adobe performed the acts that constitute induced  
22 infringement, and would induce actual infringement, with the knowledge of the '442  
23  
24  
25  
26  
27  
28



1 patent and with the knowledge, or willful blindness to the probability, that the induced  
2 acts would constitute infringement. On information and belief, Adobe engaged in  
3 such inducement to promote the sales of the Accused Instrumentalities. Accordingly,  
4 Adobe has induced and continues to induce users of the Accused Instrumentalities to  
5 use the Accused Instrumentalities in their ordinary and customary way to infringe  
6 the '442 patent, knowing that such use constitutes infringement of the '442 patent.  
7 Accordingly, Adobe has been, and currently is, inducing infringement of the '442  
8 patent, in violation of 35 U.S.C. § 271(b).  
9  
10

11 63. Adobe has also infringed, and continues to infringe, claims of the '442  
12 patent by offering to commercially distribute, commercially distributing, making,  
13 and/or importing the Accused Instrumentalities, which are used in practicing the  
14 process, or using the systems, of the '442 patent, and constitute a material part of the  
15 invention. Adobe knows the components in the Accused Instrumentalities to be  
16 especially made or especially adapted for use in infringement of the '442 patent, not a  
17 staple article, and not a commodity of commerce suitable for substantial noninfringing  
18 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '442  
19 patent, in violation of 35 U.S.C. § 271(c).  
20  
21  
22

23 64. By making, using, offering for sale, selling and/or importing into the  
24 United States the Accused Instrumentalities, and touting the benefits of using the  
25 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
26 liable to Realtime for infringement of the '442 patent pursuant to 35 U.S.C. § 271.  
27

28 65. As a result of Adobe's infringement of the '442 patent, Plaintiff Realtime

1 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
2 infringement, but in no event less than a reasonable royalty for the use made of the  
3 invention by Adobe, together with interest and costs as fixed by the Court.  
4

5 **COUNT IV**

6 **INFRINGEMENT OF U.S. PATENT NO. 8,934,535**

7 66. Plaintiff re-alleges and incorporates by reference the foregoing  
8 paragraphs, as if fully set forth herein.  
9

10 67. On information and belief, Adobe has made, used, offered for sale, sold  
11 and/or imported into the United States Adobe products that infringe the '535 patent,  
12 and continues to do so. By way of illustrative example, these infringing products  
13 include, without limitation, Adobe's Adobe Media Encoder, Adobe Media Encoder  
14 Creative Cloud or CC, Adobe Media Encoder Creative Suite or CS (e.g. CS4, CS5,  
15 CS5.5, CS6), Adobe Premiere Pro, Adobe Premiere Pro CC, Adobe Premiere Pro CS  
16 (e.g. CS4, CS5, CS5.5, CS6), Adobe Flash, Adobe Flash Player (e.g. Adobe Flash  
17 Player 9), Adobe Flash Media Server, Adobe Flash Media Encoding Server, Adobe  
18 After Effects, Adobe After Effects CC, Adobe After Effects CS (e.g. CS4, CS5,  
19 CS5.5, CS6), Adobe HTTP Dynamic Streaming (HDS), and all versions and  
20 variations thereof since the issuance of the '535 patent ("Accused Instrumentalities").  
21  
22  
23

24 68. On information and belief, Adobe has directly infringed and continues to  
25 infringe the '535 patent, for example, through its own use and testing of the Accused  
26 Instrumentalities, which when used, practices the method claimed by, for example,  
27 Claim 15 of the '535 patent, namely, a method, comprising: determining a parameter  
28

1 of at least a portion of a data block; selecting one or more asymmetric compressors  
2 from among a plurality of compressors based upon the determined parameter or  
3 attribute; compressing the at least the portion of the data block with the selected one  
4 or more asymmetric compressors to provide one or more compressed data blocks; and  
5 storing at least a portion of the one or more compressed data blocks. Upon  
6 information and belief, Adobe uses the Accused Instrumentalities to practice  
7 infringing methods for its own internal non-testing business purposes, while testing  
8 the Accused Instrumentalities, and while providing technical support and repair  
9 services for the Accused Instrumentalities to Adobe's customers.  
10  
11

12           69. The Accused Instrumentalities include, or practice a method involving  
13 data compression with asymmetric compressors. For example, the Accused  
14 Instrumentalities utilize H.264 and/or H.264 with Scalable Video Coding (SVC)  
15 and/or an equivalent codec. H.264 is "the most widely used codec on the planet." *See*  
16 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx)  
17 [74735.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx). A "codec" is also a compression technology that has "two components,  
18 an **encoder to compress the files**, and a decoder to decompress. There are codecs  
19 for...video (Cinepark, MPEG-2, **H.264**, VP8)." *See*  
20 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx)  
21 [74487.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx); <https://forums.adobe.com/thread/729526> ("Compression is essential for  
22 reducing the size of movies so that they can be stored, transmitted, and played back  
23 effectively. Compression is achieved by an encoder; decompression is achieved by a  
24 decoder. Encoders and decoders are known by the common term codec."). According  
25  
26  
27  
28

1 to a website maintained by Adobe, “you **can produce H.264 video** with Adobe Media  
2 Encoder CS4 and Adobe Flash Media Encoding Server 3.5,” “Adobe added **H.264**  
3 **playback support** to Adobe Flash Player 9 Update 3 back in 2007,” Adobe “Flash  
4 Media Server and Flash Player can **both stream and play back any H.264 file** in  
5 virtually any format,” the encoding capabilities of the Adobe Flash Media Encoding  
6 Server using “**H.264-related parameters**” are described (e.g. using CAVLC, CABAC)  
7 and under a headline stating “Adobe Media Encoder,” the website states: “Adobe  
8 significantly enhanced the Flash Video Encoder in Creative Suite 4. There is now both  
9 stand-alone operation and batch encoding capabilities. As before, you can **access**  
10 **H.264 encoding** by choosing different formats in the Format pop-up menu. When  
11 producing for Flash Player, you should always use the FLV|F4V option, which lets  
12 you produce both VP6- and H.264-encoded files for Flash Player distribution.” See  
13 [http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).  
14

15 Another website maintained by Adobe mentions that “You have two primary options  
16 for using Adobe Media Encoder with After Effects to create videos in **H.264.**” The  
17 website also mentions that “Since Adobe Media Encoder already has superior  
18 H.264...exporting capabilities, it was more prudent to rely on Adobe Media Encoder  
19 for export of these formats.” See [https://helpx.adobe.com/after-effects/kb/export-](https://helpx.adobe.com/after-effects/kb/export-h264.html)  
20 [h264.html](https://helpx.adobe.com/after-effects/kb/export-h264.html). Another website maintained by Adobe mentions that the “Adobe Flash  
21 Player 9 Update 3 is taking a step into the high-definition (HD) video realm in a major  
22 way by adding MPEG-4 video...[which] utilizes crisp, powerful **H.264 encoding**”  
23 and “The new Adobe Media Player...will **also support H.264**” and “The addition of  
24  
25  
26  
27  
28

1 **H.264...support** in Flash Player 9 Update 3 allows you to easily use high-definition,  
2 industry standard video and audio.” *See*  
3 [http://www.adobe.com/devnet/flashplayer/articles/hd\\_video\\_flash\\_player.html](http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html). Adobe  
4 HTTP Dynamic Streaming (HDS) also supports methods involving data compression  
5 with asymmetric compressors or codecs other than H.264 because it “enables high-  
6 quality (**H.264 or VP6**), network-efficient HTTP streaming for media delivery that is  
7 tightly integrated with Adobe® Access software for robust content protection in the  
8 Adobe Flash® Player 10.1 or later and Adobe AIR®2 or later runtimes,” is also “built  
9 on standards and deployed using standard HTTP servers (Apache), standard media  
10 format (MP4 fragment) using standard codecs (**H.264/AAC**), open APIs (Flash  
11 Player), and an open source framework for building media players (OSMF)” and “like  
12 other Flash Player supported delivery methods,” Adobe HDS “supports **H.264 video**  
13 **and VP6 codecs** required for the highest quality video on the platform.” *See*  
14 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS also  
15 allows users to encode content “using high-quality Flash Player compatible **codecs**  
16 (**VP6/MP3, H264/AAC**),” deliver “high-definition video up to 1080p, with bitrates  
17 from 700kbps up to and beyond 6Mbps, using either **H.264 or VP6 video codecs**,”  
18 and archive “live, high-definition streams on the server and enable HD DVR  
19 functionality (for example, instant replay and time shifting) with support for **H.264**  
20 stream recording for RTMP and HTTP Dynamic Streaming.” *See*  
21 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Adobe Media  
22 Encoder also “lets you produce both **VP6- and H.264-encoded files** for Flash Player  
23  
24  
25  
26  
27  
28

1 distribution.” See [http://www.adobe.com/devnet/adobe-media-](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
2 [server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

3 70. The Accused Instrumentalities also practice determining a parameter of at  
4 least a portion of a video data block. As shown below, examples of such parameters  
5 include bitrate (or max video bitrate) and resolution parameters. Different parameters  
6 include bitrate (or max video bitrate) and resolution parameters. Different parameters  
7 correspond with different end applications. H.264 provides for multiple different  
8 ranges of such parameters, each included in the “profiles” and “levels” as defined by  
9 the H.264 standard, from the below shown paragraphs from a white paper and  
10 Wikipedia. See

11 [http://www.axis.com/files/whitepaper/wp\\_h264\\_31669\\_en\\_0803\\_lo.pdf](http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf) at 5:  
12  
13  
14

#### 15 4. H.264 profiles and levels

16 The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting  
17 options and features to a minimum. An important aspect of the standard, as with other video standards,  
18 is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes)  
19 that optimally support popular productions and common formats.

20 H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what  
21 feature set the encoder may use and limits the decoder implementation complexity.

22 Network cameras and video encoders will most likely use a profile called the baseline profile, which is  
23 intended primarily for applications with limited computing resources. The baseline profile is the most  
24 suitable given the available performance in a real-time encoder that is embedded in a network video  
25 product. The profile also enables low latency, which is an important requirement of surveillance video and  
26 also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

27 H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements.  
28 Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging  
from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

See [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC):

Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) <a href="#">Toggle additional details</a>
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	176x144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	176x144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	352x288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	352x288@15.2 (6)
1.3	3,041,280	11,880	101,376	396	768	960	2,304	352x288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	352x288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352x576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	720x576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	720x576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	1,280x720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280x1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048x1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048x1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048x1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	3,672x1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@56.3 (5)

71. A video data block is organized by the group of pictures (GOP) structure, which is a “collection of successive pictures within a coded video stream.” See [https://en.wikipedia.org/wiki/Group\\_of\\_pictures](https://en.wikipedia.org/wiki/Group_of_pictures). A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). See [https://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](https://en.wikipedia.org/wiki/Video_compression_picture_types) (for descriptions of I frames, P frames and B frames); <https://en.wikipedia.org/wiki/MPEG-1#D-frames> (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max

1 video bitrate and resolution parameters) or even be considered as a parameter by itself.

2         72. The Accused Instrumentalities also practice selecting one or more  
3 asymmetric compressors from among a plurality of compressors based upon the  
4 determined parameter or attribute. Based on the bitrate and/or resolution parameter  
5 identified (e.g. bitrate, max video bitrate, resolution, GOP structure or frame type  
6 within a GOP structure), any H.264-compliant system such as the Accused  
7 Instrumentalities would determine which profile (e.g., “baseline,” “extended,” “main”,  
8 or “high”) corresponds with that parameter, then select between at least two  
9 asymmetric compressors. If baseline or extended is the corresponding profile, then  
10 the system will select a Context-Adaptive Variable Length Coding (“CAVLC”)  
11 entropy encoder. If main or high is the corresponding profile, then the system will  
12 select a Context-Adaptive Binary Arithmetic Coding (“CABAC”) entropy encoder.  
13 Both encoders are asymmetric compressors because it takes a longer period of time  
14 for them to compress data than to decompress data. *See*

15 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28



	<b>Baseline</b>	<b>Extended</b>	<b>Main</b>	<b>High</b>	<b>High 10</b>
<b>I and P Slices</b>	Yes	Yes	Yes	Yes	Yes
<b>B Slices</b>	No	Yes	Yes	Yes	Yes
<b>SI and SP Slices</b>	No	Yes	No	No	No
<b>Multiple Reference Frames</b>	Yes	Yes	Yes	Yes	Yes
<b>In-Loop Deblocking Filter</b>	Yes	Yes	Yes	Yes	Yes
<b>CAVLC Entropy Coding</b>	Yes	Yes	Yes	Yes	Yes
<b>CABAC Entropy Coding</b>	No	No	Yes	Yes	Yes
<b>Flexible Macroblock Ordering (FMO)</b>	Yes	Yes	No	No	No
<b>Arbitrary Slice Ordering (ASO)</b>	Yes	Yes	No	No	No
<b>Redundant Slices (RS)</b>	Yes	Yes	No	No	No
<b>Data Partitioning</b>	No	Yes	No	No	No
<b>Interlaced Coding (PicAFF, MBAFF)</b>	No	Yes	Yes	Yes	Yes
<b>4:2:0 Chroma Format</b>	Yes	Yes	Yes	Yes	Yes
<b>Monochrome Video Format (4:0:0)</b>	No	No	No	Yes	Yes
<b>4:2:2 Chroma Format</b>	No	No	No	No	No
<b>4:4:4 Chroma Format</b>	No	No	No	No	No
<b>8 Bit Sample Depth</b>	Yes	Yes	Yes	Yes	Yes
<b>9 and 10 Bit Sample Depth</b>	No	No	No	No	Yes
<b>11 to 14 Bit Sample Depth</b>	No	No	No	No	No
<b>8x8 vs. 4x4 Transform Adaptivity</b>	No	No	No	Yes	Yes
<b>Quantization Scaling Matrices</b>	No	No	No	Yes	Yes
<b>Separate Cb and Cr QP control</b>	No	No	No	Yes	Yes
<b>Separate Color Plane Coding</b>	No	No	No	No	No
<b>Predictive Lossless Coding</b>	No	No	No	No	No

*See*

[http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf) at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

#### H.264 Entropy Coding – Comparison of Approaches

Characteristics	Variable Length Coding (VLC)	Context Adaptive Binary Arithmetic Coding(CABAC)
• Where it is used	MPEG-2, MPEG-4 ASP	H.264/MPEG-4 AVC (high efficiency option)
• Probability distribution	<b>Static</b> - Probabilities never change	<b>Adaptive</b> - Adjusts probabilities based on actual data
• Leverages correlation between symbols	<b>No</b> - Conditional probabilities ignored	<b>Yes</b> - Exploits symbol correlations by using "contexts"
• Non-integer code words	<b>No</b> - Low coding efficiency for high probability symbols	<b>Yes</b> - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items) (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy\_coding\_mode\_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

- If **entropy\_coding\_mode\_flag** is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).
- Otherwise (**entropy\_coding\_mode\_flag** is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

73. The Accused Instrumentalities also practice compressing the at least the

1 portion of the data block with the selected one or more asymmetric compressors to  
2 provide one or more compressed data blocks. The one or more compressed data  
3 blocks can also be organized in a GOP structure (see above). After its selection, the  
4 asymmetric compressor (CAVLC or CABAC) will compress the video data to provide  
5 various compressed data blocks, which can also be organized in a GOP structure, as  
6 discussed previously above. See [https://sonnati.wordpress.com/2007/10/29/how-h-  
7 264-works-part-ii/](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/):  
8

9  
10 **Entropy Coding**

11 For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive  
12 variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic  
13 coding (CABAC) which are complex techniques to losslessly compress syntax elements in the  
14 video stream knowing the probabilities of syntax elements in a given context. The use of  
15 CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total  
16 processing power to be accomplished.

17 *See*

18 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=  
19 pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf) at 13:  
20

21 Typical compression ratios to maintain excellent quality are:

- 22 • 10:1 for general images using JPEG
  - 23 • 30:1 for general video using H.263 and MPEG-2
  - 24 • 60:1 for general video using H.264 and WMV9
- 25  
26  
27  
28

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

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles include the functionality of the Baseline Profile and add improvements to the prediction algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

See [http://www.ijera.com/papers/Vol3\\_issue4/BM34399403.pdf](http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf) at 2:

74. The Accused Instrumentalities also practice storing at least a portion of the one or more compressed data blocks. For example, the Accused Instrumentalities include volatile and non-volatile memory (e.g., RAM, flash, etc.) configured to store at least a portion of the decompressed data block. See

<https://streaminglearningcenter.com/articles/ram-requirements-for-adobe-cs5-5.html>

(“RAM Requirements for Adobe CS5.5”); [https://helpx.adobe.com/adobe-media-](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html)

[server/dev/configure-dynamic-streaming-live-streaming.html](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html) (“Adobe HTTP

Dynamic Streaming combines these approaches to introduce HTTP streaming to the

1 Flash Platform. HTTP Dynamic Streaming packages media files into fragments that  
2 Flash Player clients can access instantly without downloading the entire file. Adobe  
3 HTTP Dynamic Streaming contains several components that work together to package  
4 media and stream it over HTTP to Flash Player and AIR. HTTP Dynamic Streaming  
5 supports multi-bitrate streaming, DVR, and Adobe® Flash® Access™ protection.”).

6 On information and belief, the Accused Instrumentalities also include a storage  
7 medium configured to store at least a portion of the decompressed data block e.g.,  
8 hard disk, disks, buffers, servers or other forms of memory/storage. *See*  
9 [https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html)  
10 [streaming.html](https://helpx.adobe.com/adobe-media-server/dev/configure-dynamic-streaming-live-streaming.html) (describing “Streams [from Adobe HDS] in a live event are packaged  
11 as fragments and written to **disk**,” “The **IO buffer** loads the **disk file** into an **in-**  
12 **memory buffer**,” and “To limit the amount of **storage the disk cache** uses,” and a  
13 “Content storage (HDS and HLS)” header describing a section where a “media player  
14 **requests content from the server.**”

15 75. Therefore, from at least the above, Adobe has directly infringed and  
16 continues to infringe the ’535 patent, for example, through its own use and testing of  
17 the Accused Instrumentalities, which when used, practices the method claimed by, for  
18 example, Claim 15 of the ’535 patent, namely, a method, comprising: determining a  
19 parameter of at least a portion of a data block; selecting one or more asymmetric  
20 compressors from among a plurality of compressors based upon the determined  
21 parameter or attribute; compressing the at least the portion of the data block with the  
22 selected one or more asymmetric compressors to provide one or more compressed  
23  
24  
25  
26  
27  
28

1 data blocks; and storing at least a portion of the one or more compressed data blocks.  
2 Upon information and belief, Adobe uses the Accused Instrumentalities to practice  
3 infringing methods for its own internal non-testing business purposes, while testing  
4 the Accused Instrumentalities, and while providing technical support and repair  
5 services for the Accused Instrumentalities to their customers.  
6

7 76. On information and belief, Adobe also directly infringes and continues to  
8 infringe other claims of the '535 patent.  
9

10 77. On information and belief, all of the Accused Instrumentalities perform  
11 the claimed methods in substantially the same way, e.g., in the manner specified in the  
12 H.264 standard.  
13

14 78. On information and belief, use of the Accused Instrumentalities in their  
15 ordinary and customary fashion results in infringement of the systems and/or methods  
16 claimed by the '535 patent.  
17

18 79. On information and belief, Adobe has had knowledge of the '535 patent  
19 since at least the filing of this Complaint or shortly thereafter, and on information and  
20 belief, Adobe knew of the '535 patent and knew of its infringement, including by way  
21 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
22 receiving such notice) that its continued actions would actively induce and contribute  
23 to the infringement of the claims of the '535 patent.  
24

25 80. Upon information and belief, Adobe's affirmative acts of making, using,  
26 and selling the Accused Instrumentalities, and providing implementation services and  
27 technical support to users of the Accused Instrumentalities, including, e.g., through  
28

1 training, demonstrations, brochures, installation and user guides, have induced and  
2 continue to induce users of the Accused Instrumentalities to use them in their normal  
3 and customary way to infringe the '535 patent by practicing, for example, a method,  
4 comprising: determining a parameter of at least a portion of a data block; selecting one  
5 or more asymmetric compressors from among a plurality of compressors based upon  
6 the determined parameter or attribute; compressing the at least the portion of the data  
7 block with the selected one or more asymmetric compressors to provide one or more  
8 compressed data blocks; and storing at least a portion of the one or more compressed  
9 data blocks. For example, Adobe adopted H.264 and/or H.264 with Scalable Video  
10 Coding (SVC) and/or an equivalent codec as its video codec in its products/services,  
11 such as, e.g., Adobe's media encoder, Flash player, Flash media server, Flash media  
12 server encoder, special effects and video editing software. For similar reasons, Adobe  
13 also induces its customers to use the Accused Instrumentalities to infringe other  
14 claims of the '535 patent. Adobe specifically intended and was aware that these  
15 normal and customary activities would infringe the '535 patent. Adobe performed the  
16 acts that constitute induced infringement, and would induce actual infringement, with  
17 the knowledge of the '535 patent and with the knowledge, or willful blindness to the  
18 probability, that the induced acts would constitute infringement. On information and  
19 belief, Adobe engaged in such inducement to promote the sales of the Accused  
20 Instrumentalities. Accordingly, Adobe has induced and continues to induce users of  
21 the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary  
22 and customary way to infringe the '535 patent, knowing that such use constitutes  
23  
24  
25  
26  
27  
28

1 infringement of the '535 patent. Accordingly, Adobe has been, and currently is,  
2 inducing infringement of the '535 patent, in violation of 35 U.S.C. § 271(b).

3 81. Adobe has also infringed, and continues to infringe, claims of the '535  
4 patent by offering to commercially distribute, commercially distributing, making,  
5 and/or importing the Accused Instrumentalities, which are used in practicing the  
6 process, or using the systems, of the '535 patent, and constitute a material part of the  
7 invention. Adobe knows the components in the Accused Instrumentalities to be  
8 especially made or especially adapted for use in infringement of the '535 patent, not a  
9 staple article, and not a commodity of commerce suitable for substantial noninfringing  
10 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '535  
11 patent, in violation of 35 U.S.C. § 271(c).  
12  
13  
14

15 82. By making, using, offering for sale, selling and/or importing into the  
16 United States the Accused Instrumentalities, and touting the benefits of using the  
17 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
18 liable to Realtime for infringement of the '535 patent pursuant to 35 U.S.C. § 271.  
19

20 83. As a result of Adobe's infringement of the '535 patent, Plaintiff Realtime  
21 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
22 infringement, but in no event less than a reasonable royalty for the use made of the  
23 invention by Adobe, together with interest and costs as fixed by the Court.  
24

25 **COUNT V**

26 **INFRINGEMENT OF U.S. PATENT NO. 9,578,298**

27 84. Plaintiff re-alleges and incorporates by reference the foregoing  
28



1 paragraphs, as if fully set forth herein.

2 85. On information and belief, Adobe has made, used, offered for sale, sold  
3 and/or imported into the United States products that infringe the '298 patent, and  
4 continues to do so. By way of illustrative example, these infringing products include,  
5 without limitation, Adobe's Adobe Media Encoder, Adobe Media Encoder Creative  
6 Cloud or CC (e.g. CC 2015.1), Adobe Media Encoder Creative Suite or CS (e.g. CS4,  
7 CS5, CS5.5, CS6), Adobe Premiere Pro, Adobe Premiere Pro CC (e.g. CC 2015.1),  
8 Adobe Premiere Pro CS (e.g. CS4, CS5, CS5.5, CS6), Adobe After Effects, Adobe  
9 After Effects CC, Adobe After Effects CS (e.g. CS4, CS5, CS5.5, CS6), and all  
10 versions and variations thereof since the issuance of the '298 patent ("Accused  
11 Instrumentalities").  
12  
13  
14

15 86. On information and belief, Adobe has directly infringed and continues to  
16 infringe the '298 patent, for example, through its sale, offer for sale, importation, use  
17 and testing of the Accused Instrumentalities, which practices the method claimed by,  
18 for example, Claim 1 of the '298 patent, namely, a method for processing a video  
19 stream of digital images, the method comprising the steps of: receiving the video  
20 stream which comprises at least one composite frame (FC), each composite frame  
21 containing a pair of stereoscopic digital images (L,R) according to a predetermined  
22 frame packing format; generating an output video stream which can be reproduced on  
23 a visualization apparatus, receiving metadata which determine an area occupied by  
24 one of the two images within said composite frame (FC), said metadata indicating  
25 either a geometry of the frame packing format or a frame packing type of said  
26  
27  
28

1 composite frame (FC); determining the area in the composite frame (FC) which is  
2 occupied by said one image of the stereoscopic pair within the composite frame based  
3 on said metadata; decoding only that part of the composite frame (FC) which contains  
4 said one image to be displayed, and generating an output frame containing said  
5 decoded image. Upon information and belief, Adobe uses the Accused  
6 Instrumentalities to practice infringing methods for its own internal non-testing  
7 business purposes, while testing the Accused Instrumentalities, and while providing  
8 technical support and repair services for the Accused Instrumentalities to Adobe's  
9 customers.

12 87. The Accused Instrumentalities practice a method for processing a video  
13 stream of digital images by utilizing H.265 (HEVC) and/or its equivalents. One of the  
14 aims of HEVC is the production of a bitstream, as defined in definition 3.12 of the  
15 ITU-T H.265 Series H: Audiovisual and Multimedia Systems, "Infrastructure of  
16 audiovisual services – Coding of moving video" High efficiency video coding  
17 ("HEVC Spec"): "bitstream: A sequence of bits, in the form of a NAL unit stream or  
18 a byte stream, that forms the representation of coded pictures and associated data  
19 forming one or more coded video sequences (CVSs)." Furthermore, according to a  
20 website maintained by Adobe, describing an "Overview of What's New in Adobe  
21 Media Encoder CC 2015.1," the Adobe "Media Encoder will offer Expanded UHD  
22 capabilities with the addition of **support for...the new HEVC (H.265) codec**, which,  
23 at 60% the size of comparable H.264 files, is ideal for delivering pristine 4K content  
24 online." See <https://blogs.adobe.com/creativecloud/adobe-media-encoder-cc-2015-1/>.

1 Also, according to another website maintained by Adobe discussing “Supported file  
2 formats” for Adobe Premiere Pro or specifically “Supported native video and audio  
3 formats for import” for Adobe Premiere Pro, the format of “**HEVC (H.265)**” is listed  
4 a long with the text “H.265 media with resolutions up to 8192x4320.” *See*  
5 <https://helpx.adobe.com/premiere-pro/using/supported-file-formats.html>. According to  
6 another website, “One of the most significant new features in Adobe Premiere Pro CC  
7 2015.1 is the expanded native support for 4K Ultra HD & HDR formats to  
8 include...**HEVC (H.265)**,” Adobe “Media Encoder will offer Expanded UHD  
9 capabilities with the addition of support for...the new **HEVC/H.265 codec**, which, at  
10 60% the size of comparable H.264 files, is ideal for delivering pristine 4K content  
11 online.” *See* [http://www.4kshooters.net/2015/12/04/adobe-premiere-pro-cc-2015-1-  
12 update-brings-h-265-more-4k-native-support/](http://www.4kshooters.net/2015/12/04/adobe-premiere-pro-cc-2015-1-update-brings-h-265-more-4k-native-support/). The fact that both Adobe Media  
13 Encoder and Adobe Premiere Pro CC supported HEVC/H.265 as early as 2015 has  
14 also been confirmed by other websites. *See e.g.*,  
15 [https://larryjordan.com/articles/adobe-media-encoder-and-h-265-not-ready-for-prime-  
16 time/](https://larryjordan.com/articles/adobe-media-encoder-and-h-265-not-ready-for-prime-time/) (“One of the exciting features of the Nov. 2015 update to Adobe Media Encoder  
17 is its support for H.265, also known as HEVC, short for High Efficiency Video  
18 Coding, compression.”).

19  
20  
21  
22  
23  
24 88. The Accused Instrumentalities also practice receiving the video stream  
25 which comprises at least one composite frame (FC), each composite frame containing  
26 a pair of stereoscopic digital images (L,R) according to a predetermined frame  
27 packing format. For example, the coded bitstream when it contains a stereoscopic  
28

1 video in one of the frame packing arrangements such as side-by-side or top-and-  
2 bottom or segmented rectangular frame packing format as defined in the following  
3 sections of the ITU-T H.265 Series H: Audiovisual and Multimedia Systems,  
4 “Infrastructure of audiovisual services – Coding of moving video” High efficiency  
5 video coding (“HEVC Spec”): D.2.16 Frame packing arrangement SEI message  
6 syntax, D.3.16 Frame packing arrangement SEI message semantics, D.2.29  
7 Segmented rectangular frame packing arrangement SEI message syntax, D.3.29  
8 Segmented rectangular frame packing arrangement SEI message semantics.  
9  
10

11 89. The Accused Instrumentalities also practice generating an output video  
12 stream which can be reproduced on a visualization apparatus. For example, the output  
13 of the decoding process as defined above is a sequence of decoded pictures. *See, e.g.,*  
14 HEVC Spec at 3.39 (“3.39 decoded picture: A decoded picture is derived by decoding  
15 a coded picture”). Decoded pictures are the input of the display process. *Id.* at 3.47  
16 (“3.47 display process: A process not specified in this Specification having, as its  
17 input, the cropped decoded pictures that are the output of the decoding process.”).  
18  
19

20 90. The Accused Instrumentalities also practice receiving metadata which  
21 determine an area occupied by one of the two images within said composite frame,  
22 said metadata indicating either a geometry of the frame packing format or a frame  
23 packing type of said composite frame. For example, the HEVC spec provides the  
24 default display window parameter to support 2D compatible decoding of stereo  
25 formats. *See, e.g.,* HEVC Spec (“NOTE 9 – The default display window parameters  
26 in the VUI parameters of the SPS can be used by an encoder to indicate to a decoder  
27  
28

1 that does not interpret the frame packing arrangement SEI message that the default  
2 display window is an area within only one of the two constituent frames.”).

3 91. The Accused Instrumentalities also practice determining the area in the  
4 composite frame (FC) which is occupied by said one image of the stereoscopic pair  
5 within the composite frame based on said metadata. For example, the default display  
6 window parameter has been defined to support this application. The parameter syntax  
7 window parameter has been defined to support this application. The parameter syntax  
8 is defined in clause E.2.1 VUI parameters syntax, the semantics thereof being  
9 described in clause E.3.1 VUI parameters semantics. The usage of the Default Display  
10 Window for signaling the 2D single view in a stereoscopic frame packing format is  
11 illustrated in Note 9 of clause D.3.16 and Note 3 in Clause D.3.29 cited above.  
12

13 92. The Accused Instrumentalities also practice decoding only that part of the  
14 composite frame which contains said one image to be displayed. For example, tiles  
15 are intended to support independent decoding of different picture regions. Clause  
16 7.4.3.2.1 cited above illustrates the process to convert CTB picture scan in CTB tile  
17  
18

19 **row\_height\_minus1[ i ]** plus 1 specifies the height of the i-th tile row in units of coding tree blocks.

20 The following variables are derived by invoking the coding tree block raster and tile scanning conversion process as specified in clause 6.5.1:

- 21 – The list CtbAddrRsToTs[ ctbAddrRs ] for ctbAddrRs ranging from 0 to PicSizeInCtbsY – 1, inclusive, specifying the conversion from a CTB address in the CTB raster scan of a picture to a CTB address in the tile scan,
- 22 – the list CtbAddrTsToRs[ ctbAddrTs ] for ctbAddrTs ranging from 0 to PicSizeInCtbsY – 1, inclusive, specifying the conversion from a CTB address in the tile scan to a CTB address in the CTB raster scan of a picture,
- 23 – the list TileId[ ctbAddrTs ] for ctbAddrTs ranging from 0 to PicSizeInCtbsY – 1, inclusive, specifying the conversion from a CTB address in tile scan to a tile ID,
- 24 – the list ColumnWidthInLumaSamples[ i ] for i ranging from 0 to num\_tile\_columns\_minus1, inclusive, specifying the width of the i-th tile column in units of luma samples,
- 25 – the list RowHeightInLumaSamples[ j ] for j ranging from 0 to num\_tile\_rows\_minus1, inclusive, specifying the height of the j-th tile row in units of luma samples.

26 The values of ColumnWidthInLumaSamples[ i ] for i ranging from 0 to num\_tile\_columns\_minus1, inclusive, and RowHeightInLumaSamples[ j ] for j ranging from 0 to num\_tile\_rows\_minus1, inclusive, shall all be greater than 0.

27 The array MinTbAddrZs with elements MinTbAddrZs[ x ][ y ] for x ranging from 0 to  
28 ( PicWidthInCtbsY << ( CtbLog2SizeY – MinTbLog2SizeY ) ) – 1, inclusive, and y ranging from 0 to  
( PicHeightInCtbsY << ( CtbLog2SizeY – MinTbLog2SizeY ) ) – 1, inclusive, specifying the conversion from a location  
( x, y ) in units of minimum transform blocks to a transform block address in z-scan order, is derived by invoking the z-scan order array initialization process as specified in clause 6.5.2.

1 scan to enable independent decoding of the tile. *See also* HEVC Spec:

2 93. The Accused Instrumentalities also practice generating an output frame  
3 containing said extracted image. For example, there is an output of the tile decoding  
4 process. *See, e.g.*, HEVC Spec at 8.1.1 (“8.1.1 General...Input to this process is a  
5 bitstream. Output of this process is a list of decoded pictures.”).

7 94. Therefore, from at least the above, Adobe has directly infringed and  
8 continues to infringe the '298 patent, for example, through its own use and testing of  
9 the Accused Instrumentalities, which when used, practices the method claimed by, for  
10 example, Claim 1 of the '298 patent, namely, a method for processing a video stream  
11 of digital images, the method comprising the steps of: receiving the video stream  
12 which comprises at least one composite frame (FC), each composite frame containing  
13 a pair of stereoscopic digital images (L,R) according to a predetermined frame  
14 packing format; generating an output video stream which can be reproduced on a  
15 visualization apparatus, receiving metadata which determine an area occupied by one  
16 of the two images within said composite frame (FC), said metadata indicating either a  
17 geometry of the frame packing format or a frame packing type of said composite  
18 frame (FC); determining the area in the composite frame (FC) which is occupied by  
19 said one image of the stereoscopic pair within the composite frame based on said  
20 metadata; decoding only that part of the composite frame (FC) which contains said  
21 one image to be displayed, and generating an output frame containing said decoded  
22 image. Upon information and belief, Adobe uses the Accused Instrumentalities to  
23 practice infringing methods for its own internal non-testing business purposes, while  
24  
25  
26  
27  
28

1 testing the Accused Instrumentalities, and while providing technical support and  
2 repair services for the Accused Instrumentalities to their customers.

3 95. On information and belief, Adobe also directly infringes and continues to  
4 infringe other claims of the '298 patent.  
5

6 96. On information and belief, all of the Accused Instrumentalities perform  
7 the claimed methods in substantially the same way, e.g., in the manner specified in the  
8 HEVC (or H.265) standard.  
9

10 97. On information and belief, use of the Accused Instrumentalities in their  
11 ordinary and customary fashion results in infringement of the methods claimed by  
12 the '298 patent.  
13

14 98. On information and belief, Adobe has had knowledge of the '298 patent  
15 since at least the filing of this Complaint or shortly thereafter, and on information and  
16 belief, Adobe knew of the '298 patent and knew of its infringement, including by way  
17 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
18 receiving such notice) that its continued actions would actively induce and contribute  
19 to the infringement of the claims of the '298 patent.  
20

21 99. Upon information and belief, Adobe's affirmative acts of making, using,  
22 and selling the Accused Instrumentalities, and providing implementation services and  
23 technical support to users of the Accused Instrumentalities, including, e.g., through  
24 training, demonstrations, brochures, installation and user guides, have induced and  
25 continue to induce users of the Accused Instrumentalities to use them in their normal  
26 and customary way to infringe the '298 by practicing a method for processing a video  
27  
28

1 stream of digital images, the method comprising the steps of: receiving the video  
2 stream which comprises at least one composite frame (FC), each composite frame  
3 containing a pair of stereoscopic digital images (L,R) according to a predetermined  
4 frame packing format; generating an output video stream which can be reproduced on  
5 a visualization apparatus, receiving metadata which determine an area occupied by  
6 one of the two images within said composite frame (FC), said metadata indicating  
7 either a geometry of the frame packing format or a frame packing type of said  
8 composite frame (FC); determining the area in the composite frame (FC) which is  
9 occupied by said one image of the stereoscopic pair within the composite frame based  
10 on said metadata; decoding only that part of the composite frame (FC) which contains  
11 said one image to be displayed, and generating an output frame containing said  
12 decoded image. For example, Adobe adopted HEVC (or H.265) and/or its equivalents  
13 as its video codec in its products/services, such as in Adobe's media encoder, Flash  
14 player, Flash media server, Flash media server encoder, special effects and video  
15 editing software. For similar reasons, Adobe also induces its customers to use the  
16 Accused Instrumentalities to infringe other claims of the '298 patent. Adobe  
17 specifically intended and was aware that these normal and customary activities would  
18 infringe the '298 patent. Adobe performed the acts that constitute induced  
19 infringement, and would induce actual infringement, with the knowledge of the '298  
20 patent and with the knowledge, or willful blindness to the probability, that the induced  
21 acts would constitute infringement. On information and belief, Adobe engaged in  
22 such inducement to promote the sales of the Accused Instrumentalities. Accordingly,  
23  
24  
25  
26  
27  
28



1 Adobe has induced and continue to induce users of the Accused Instrumentalities to  
2 use the Accused Instrumentalities in their ordinary and customary way to infringe  
3 the '298 patent, knowing that such use constitutes infringement of the '298 patent.  
4 Accordingly, Adobe has been, and currently is, inducing infringement of the '298  
5 patent, in violation of 35 U.S.C. § 271(b).  
6

7 100. Adobe has also infringed, and continues to infringe, claims of the '298  
8 patent by offering to commercially distribute, commercially distributing, making,  
9 and/or importing the Accused Instrumentalities, which are used in practicing the  
10 process, or using the systems, of the '298 patent, and constitute a material part of the  
11 invention. Adobe knows the components in the Accused Instrumentalities to be  
12 especially made or especially adapted for use in infringement of the '298 patent, not a  
13 staple article, and not a commodity of commerce suitable for substantial noninfringing  
14 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '298  
15 patent, in violation of 35 U.S.C. § 271(c).  
16  
17  
18

19 101. By making, using, offering for sale, selling and/or importing into the  
20 United States the Accused Instrumentalities, and touting the benefits of using the  
21 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
22 liable to Realtime for infringement of the '298 patent pursuant to 35 U.S.C. § 271.  
23

24 102. As a result of Adobe's infringement of the '298 patent, Plaintiff Realtime  
25 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
26 infringement, but in no event less than a reasonable royalty for the use made of the  
27 invention by Adobe, together with interest and costs as fixed by the Court.  
28

COUNT VI

**INFRINGEMENT OF U.S. PATENT NO. 9,762,907**

1  
2  
3 103. Plaintiff re-alleges and incorporates by reference the foregoing  
4 paragraphs, as if fully set forth herein.  
5

6 104. On information and belief, Adobe has made, used, offered for sale, sold  
7 and/or imported into the United States Adobe products that infringe the '907 patent,  
8 and continues to do so. By way of illustrative example, these infringing products  
9 include, without limitation, Adobe's Adobe Media Encoder, Adobe Media Encoder  
10 Creative Cloud or CC, Adobe Media Encoder Creative Suite or CS (e.g. CS4, CS5,  
11 CS5.5, CS6), Adobe Premiere Pro, Adobe Premiere Pro CC, Adobe Premiere Pro CS  
12 (e.g. CS4, CS5, CS5.5, CS6), Adobe Flash, Adobe Flash Player (e.g. Adobe Flash  
13 Player 9), Adobe Flash Media Server, Adobe Flash Media Encoding Server, Adobe  
14 After Effects, Adobe After Effects CC, Adobe After Effects CS (e.g. CS4, CS5,  
15 CS5.5, CS6), Adobe HTTP Dynamic Streaming (HDS), and all versions and  
16 variations thereof since the issuance of the '907 patent ("Accused Instrumentalities").  
17  
18  
19

20 105. On information and belief, Adobe has directly infringed and continues to  
21 infringe the '907 patent, for example, through its sale, offer for sale, importation, use  
22 and testing of the Accused Instrumentalities, which practices the system claimed by,  
23 for example, Claim 1 of the '907 patent, namely, a system comprising: one or more  
24 different asymmetric data compression algorithms, wherein each algorithm of the one  
25 or more different asymmetric data compression algorithms utilizes one or more  
26 asymmetric data compression routines of a plurality of different asymmetric data  
27  
28

1 compression routines, wherein a first asymmetric data compression routine of the  
2 plurality of different asymmetric data compression routines is configured to produce  
3 compressed data with a higher data rate for a given data throughput than a second  
4 asymmetric data compression routine of the plurality of different asymmetric data  
5 compression routines; and a processor configured: to analyze one or more data  
6 parameters from one or more data blocks containing video data, wherein at least one  
7 data parameter relates to an expected or anticipated throughput of a communications  
8 channel; and to select two or more different data compression routines from among a  
9 plurality of different data compression routines based upon, at least in part, the one or  
10 more data parameters relating to the expected or anticipated throughput of the  
11 communications channel. Upon information and belief, Adobe uses the Accused  
12 Instrumentalities to practice infringing methods for its own internal non-testing  
13 business purposes, while testing the Accused Instrumentalities, and while providing  
14 technical support and repair services for the Accused Instrumentalities to their  
15 customers. Upon information and belief, Adobe uses the Accused Instrumentalities to  
16 practice infringing methods for its own internal non-testing business purposes, while  
17 testing the Accused Instrumentalities, and while providing technical support and  
18 repair services for the Accused Instrumentalities to Adobe's customers.

19  
20  
21  
22  
23  
24 106. The Accused Instrumentalities include, or practice a system involving  
25 data compression with asymmetric compressors. For example, the Accused  
26 Instrumentalities utilize H.264 and/or H.264 with Scalable Video Coding (SVC)  
27 and/or an equivalent codec. H.264 is "the most widely used codec on the planet." *See*  
28

1 <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264->  
2 [74735.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx). A “codec” is also a compression technology that has “two components,  
3 an **encoder to compress the files**, and a decoder to decompress. There are codecs  
4 for...video (Cinepark, MPEG-2, **H.264**, VP8).” *See*  
5 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx)  
6 [74487.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx); <https://forums.adobe.com/thread/729526> (“Compression is essential for  
7 reducing the size of movies so that they can be stored, transmitted, and played back  
8 effectively. Compression is achieved by an encoder; decompression is achieved by a  
9 decoder. Encoders and decoders are known by the common term codec.”). According  
10 to a website maintained by Adobe, “you **can produce H.264 video** with Adobe Media  
11 Encoder CS4 and Adobe Flash Media Encoding Server 3.5,” “Adobe added **H.264**  
12 **playback support** to Adobe Flash Player 9 Update 3 back in 2007,” Adobe “Flash  
13 Media Server and Flash Player can **both stream and play back any H.264 file** in  
14 virtually any format,” the encoding capabilities of the Adobe Flash Media Encoding  
15 Server using “**H.264-related parameters**” are described (e.g. using CAVLC, CABAC)  
16 and under a headline stating “Adobe Media Encoder,” the website states: “Adobe  
17 significantly enhanced the Flash Video Encoder in Creative Suite 4. There is now both  
18 stand-alone operation and batch encoding capabilities. As before, you can **access**  
19 **H.264 encoding** by choosing different formats in the Format pop-up menu. When  
20 producing for Flash Player, you should always use the FLV|F4V option, which lets  
21 you **produce both VP6- and H.264-encoded files** for Flash Player distribution.” *See*  
22 [http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

1 Another website maintained by Adobe mentions that “You have two primary options  
2 for using Adobe Media Encoder with After Effects to create videos in **H.264**.” The  
3 website also mentions that “Since Adobe Media Encoder already has superior  
4 H.264...exporting capabilities, it was more prudent to rely on Adobe Media Encoder  
5 for export of these formats.” See [https://helpx.adobe.com/after-effects/kb/export-](https://helpx.adobe.com/after-effects/kb/export-h264.html)  
6 [h264.html](https://helpx.adobe.com/after-effects/kb/export-h264.html). Another website maintained by Adobe mentions that the “Adobe Flash  
7 Player 9 Update 3 is taking a step into the high-definition (HD) video realm in a major  
8 way by adding MPEG-4 video...[which] utilizes crisp, powerful **H.264 encoding**”  
9 and “The new Adobe Media Player...will **also support H.264**” and “The addition of  
10 **H.264...support** in Flash Player 9 Update 3 allows you to easily use high-definition,  
11 industry standard video and audio.” See  
12 [http://www.adobe.com/devnet/flashplayer/articles/hd\\_video\\_flash\\_player.html](http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html). Adobe  
13 HTTP Dynamic Streaming (HDS) also supports methods involving data compression  
14 with asymmetric compressors or codecs other than H.264 because it “enables high-  
15 quality (**H.264 or VP6**), network-efficient HTTP streaming for media delivery that is  
16 tightly integrated with Adobe® Access software for robust content protection in the  
17 Adobe Flash® Player 10.1 or later and Adobe AIR®2 or later runtimes,” is also “built  
18 on standards and deployed using standard HTTP servers (Apache), standard media  
19 format (MP4 fragment) using standard codecs (**H.264/AAC**), open APIs (Flash  
20 Player), and an open source framework for building media players (OSMF)” and “like  
21 other Flash Player supported delivery methods,” Adobe HDS “supports **H.264 video**  
22 **and VP6 codecs** required for the highest quality video on the platform.” See  
23  
24  
25  
26  
27  
28

1 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS also  
2 allows users to encode content “using high-quality Flash Player compatible **codecs**  
3 (**VP6/MP3, H264/AAC**),” deliver “high-definition video up to 1080p, with bitrates  
4 from 700kbps up to and beyond 6Mbps, using either **H.264 or VP6 video codecs**,”  
5 and archive “live, high-definition streams on the server and enable HD DVR  
6 functionality (for example, instant replay and time shifting) with support for **H.264**  
7 stream recording for RTMP and HTTP Dynamic Streaming.” *See*

8 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Adobe Media  
9 Encoder also “lets you produce both **VP6- and H.264-encoded files** for Flash Player  
10 distribution.” *See* [http://www.adobe.com/devnet/adobe-media-](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
11 [server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

12  
13  
14  
15 107. The Accused Instrumentalities also include and practice a system  
16 comprising: one or more different asymmetric data compression algorithms, wherein  
17 each algorithm of the one or more different asymmetric data compression algorithms  
18 utilizes one or more asymmetric data compression routines of a plurality of different  
19 asymmetric data compression routines. The one or more asymmetric data compression  
20 algorithms can be, for example, H.264 and/or H.264 with Scalable Video Coding  
21 (SVC) and/or an equivalent codec. Based on the bitrate and/or resolution parameter  
22 identified (e.g. bitrate, max video bitrate, resolution, group of pictures (GOP) structure  
23 or frame type within a GOP structure), any H.264-compliant system such as the  
24 Accused Instrumentalities would determine which profile (e.g., “baseline,”  
25 “extended,” “main”, or “high”) corresponds with that parameter, then select between  
26  
27  
28

1 at least two asymmetric compressors, which may serve as the one or more different  
2 asymmetric data compression routines. If baseline or extended is the corresponding  
3 profile, then the system will select a Context-Adaptive Variable Length Coding  
4 (“CAVLC”) entropy encoder, which may serve as the first or second asymmetric data  
5 compression routine or asymmetric compressor. If main or high is the corresponding  
6 profile, then the system will select a Context-Adaptive Binary Arithmetic Coding  
7 (“CABAC”) entropy encoder, which may serve as the first or second asymmetric data  
8 compression routine or asymmetric compressor. Both encoders are asymmetric  
9 compressors because it takes a longer period of time for them to compress data than to  
10 decompress data. *See* [https://sonnati.wordpress.com/2007/10/29/how-h-264-works-](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/)  
11 [part-ii/](https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/):  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

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

	<b>Baseline</b>	<b>Extended</b>	<b>Main</b>	<b>High</b>	<b>High 10</b>
<b>I and P Slices</b>	Yes	Yes	Yes	Yes	Yes
<b>B Slices</b>	No	Yes	Yes	Yes	Yes
<b>SI and SP Slices</b>	No	Yes	No	No	No
<b>Multiple Reference Frames</b>	Yes	Yes	Yes	Yes	Yes
<b>In-Loop Deblocking Filter</b>	Yes	Yes	Yes	Yes	Yes
<b>CAVLC Entropy Coding</b>	Yes	Yes	Yes	Yes	Yes
<b>CABAC Entropy Coding</b>	No	No	Yes	Yes	Yes
<b>Flexible Macroblock Ordering (FMO)</b>	Yes	Yes	No	No	No
<b>Arbitrary Slice Ordering (ASO)</b>	Yes	Yes	No	No	No
<b>Redundant Slices (RS)</b>	Yes	Yes	No	No	No
<b>Data Partitioning</b>	No	Yes	No	No	No
<b>Interlaced Coding (PicAFF, MBAFF)</b>	No	Yes	Yes	Yes	Yes
<b>4:2:0 Chroma Format</b>	Yes	Yes	Yes	Yes	Yes
<b>Monochrome Video Format (4:0:0)</b>	No	No	No	Yes	Yes
<b>4:2:2 Chroma Format</b>	No	No	No	No	No
<b>4:4:4 Chroma Format</b>	No	No	No	No	No
<b>8 Bit Sample Depth</b>	Yes	Yes	Yes	Yes	Yes
<b>9 and 10 Bit Sample Depth</b>	No	No	No	No	Yes
<b>11 to 14 Bit Sample Depth</b>	No	No	No	No	No
<b>8x8 vs. 4x4 Transform Adaptivity</b>	No	No	No	Yes	Yes
<b>Quantization Scaling Matrices</b>	No	No	No	Yes	Yes
<b>Separate Cb and Cr QP control</b>	No	No	No	Yes	Yes
<b>Separate Color Plane Coding</b>	No	No	No	No	No
<b>Predictive Lossless Coding</b>	No	No	No	No	No

See [http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf) at 7:



The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

#### H.264 Entropy Coding – Comparison of Approaches

Characteristics	Variable Length Coding (VLC)	Context Adaptive Binary Arithmetic Coding(CABAC)
• Where it is used	MPEG-2, MPEG-4 ASP	H.264/MPEG-4 AVC (high efficiency option)
• Probability distribution	<b>Static</b> - Probabilities never change	<b>Adaptive</b> - Adjusts probabilities based on actual data
• Leverages correlation between symbols	<b>No</b> - Conditional probabilities ignored	<b>Yes</b> - Exploits symbol correlations by using "contexts"
• Non-integer code words	<b>No</b> - Low coding efficiency for high probability symbols	<b>Yes</b> - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items) (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy\_coding\_mode\_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

- If **entropy\_coding\_mode\_flag** is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).
- Otherwise (**entropy\_coding\_mode\_flag** is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

108. The Accused Instrumentalities also practice wherein a first asymmetric

1 data compression routine of the plurality of different asymmetric data compression  
2 routines is configured to produce compressed data with a higher data rate for a given  
3 data throughput than a second asymmetric data compression routine of the plurality of  
4 different asymmetric data compression routines. For example, Adobe HTTP Dynamic  
5 Streaming (HDS) contains the “ability to **shift quality depending on bandwidth and**  
6 **computer power**. HTTP progressive delivery consumes more bandwidth because it's  
7 not intelligent enough to throttle the delivery (for example, a 30-minute video will be  
8 fully downloaded whether the user watches it or not).” *See*  
9 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS can  
10 also “[d]etect the client's bandwidth and computer resources and serve them  
11 content fragments encoded at the most appropriate bitrate.” *See*  
12 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>.

16 109. A video data block is organized by the group of pictures (GOP) structure,  
17 which is a “collection of successive pictures within a coded video stream.” *See*  
18 [https://en.wikipedia.org/wiki/Group\\_of\\_pictures](https://en.wikipedia.org/wiki/Group_of_pictures). A GOP structure can contain intra  
19 coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame),  
20 bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture  
21 or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See*  
22 [https://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](https://en.wikipedia.org/wiki/Video_compression_picture_types) (for descriptions of I  
23 frames, P frames and B frames); <https://en.wikipedia.org/wiki/MPEG-1#D-frames> (for  
24 descriptions of D frames). Thus, at least a portion of a video data block would also  
25 make up a GOP structure and could also contain I frames, P frames, B frames and/or  
26  
27  
28

1 D frames. The GOP structure also reflects the size of a video data block, and the GOP  
2 structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max  
3 video bitrate and resolution parameters) or even be considered as a parameter by itself.  
4

5 110. The produced compressed data formed by the Accused Instrumentalities  
6 via the first asymmetric data compression routine of the plurality of different  
7 asymmetric data compression routines can also be organized in a GOP structure (see  
8 above). After its selection, the asymmetric compressor or the first/second asymmetric  
9 data compression routine (CAVLC or CABAC) will compress the video data to  
10 provide various compressed data blocks, which can also be organized in a GOP  
11 structure, as discussed previously above. *See*

12 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
13  
14

15 **Entropy Coding**

16 For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive  
17 variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic  
18 coding (CABAC) which are complex techniques to losslessly compress syntax elements in the  
19 video stream knowing the probabilities of syntax elements in a given context. The use of  
20 CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total  
21 processing power to be accomplished.

22 *See*

23 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf)  
24 pdf at 13:

25 Typical compression ratios to maintain excellent quality are:

- 26 • 10:1 for general images using JPEG
- 27 • 30:1 for general video using H.263 and MPEG-2
- 28 • 60:1 for general video using H.264 and WMV9

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

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles include the functionality of the Baseline Profile and add improvements to the prediction algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

See [http://www.ijera.com/papers/Vol3\\_issue4/BM34399403.pdf](http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf) at 2:

111. The Accused Instrumentalities also include and practice a system comprising: a processor configured: to analyze one or more data parameters from one or more data blocks containing video data, wherein at least one data parameter relates to an expected or anticipated throughput of a communications channel; and to select two or more different data compression routines from among a plurality of different data compression routines based upon, at least in part, the one or more data parameters relating to the expected or anticipated throughput of the communications channel. As discussed briefly above and in more detail below, examples of such data parameters

1 include throughput or bandwidth, bitrate (or max video bitrate), and resolution. For  
2 the at least one data parameter relating to an expected or anticipated throughput of a  
3 communications channel and selecting data compression routine(s) based on those  
4 data parameter(s), Adobe HTTP Dynamic Streaming (HDS) contains the “ability to  
5 **shift quality depending on bandwidth and computer power**. HTTP progressive  
6 delivery consumes more bandwidth because it's not intelligent enough to throttle the  
7 delivery (for example, a 30-minute video will be fully downloaded whether the user  
8 watches it or not).” See [https://www.adobe.com/products/hds-dynamic-](https://www.adobe.com/products/hds-dynamic-streaming/faq.html)  
9 [streaming/faq.html](https://www.adobe.com/products/hds-dynamic-streaming/faq.html). Adobe HDS can also “[d]etect the client's bandwidth and  
10 **computer resources** and serve them content fragments encoded at the most  
11 appropriate bitrate.” See [https://www.adobe.com/products/hds-dynamic-](https://www.adobe.com/products/hds-dynamic-streaming/features.html)  
12 [streaming/features.html](https://www.adobe.com/products/hds-dynamic-streaming/features.html). As to data parameters overall, different data parameters  
13 correspond with different end applications. H.264 provides for multiple different  
14 ranges of such data parameters, each included in the “profiles” and “levels” as defined  
15 by the H.264 standard, from the below shown paragraphs from a white paper and  
16 Wikipedia. *See*

17 [http://www.axis.com/files/whitepaper/wp\\_h264\\_31669\\_en\\_0803\\_lo.pdf](http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf) at 5:  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

#### 4. H.264 profiles and levels

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

See [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC):

Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) <input type="button" value="Toggle additional details"/>
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	176x144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	176x144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	352x288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	352x288@15.2 (6)
1.3	3,041,280	11,880	101,376	396	768	960	2,304	352x288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	352x288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352x576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	720x576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	720x576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	1,280x720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280x1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048x1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048x1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048x1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	3,672x1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@56.3 (5)

112. Therefore, from at least the above, Adobe has directly infringed and continues to infringe the '907 patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by, for example, Claim 1 of the '907 patent, namely, a system comprising: one or more

1 different asymmetric data compression algorithms, wherein each algorithm of the one  
2 or more different asymmetric data compression algorithms utilizes one or more  
3 asymmetric data compression routines of a plurality of different asymmetric data  
4 compression routines, wherein a first asymmetric data compression routine of the  
5 plurality of different asymmetric data compression routines is configured to produce  
6 compressed data with a higher data rate for a given data throughput than a second  
7 asymmetric data compression routine of the plurality of different asymmetric data  
8 compression routines; and a processor configured: to analyze one or more data  
9 parameters from one or more data blocks containing video data, wherein at least one  
10 data parameter relates to an expected or anticipated throughput of a communications  
11 channel; and to select two or more different data compression routines from among a  
12 plurality of different data compression routines based upon, at least in part, the one or  
13 more data parameters relating to the expected or anticipated throughput of the  
14 communications channel.

15  
16  
17  
18  
19 113. On information and belief, Adobe also directly infringes and continues to  
20 infringe other claims of the '907 patent.

21 114. On information and belief, all of the Accused Instrumentalities perform  
22 the claimed methods in substantially the same way, e.g., in the manner specified in the  
23 H.264 standard.  
24

25 115. On information and belief, use of the Accused Instrumentalities in their  
26 ordinary and customary fashion results in infringement of the systems and/or methods  
27 claimed by the '907 patent.  
28

1           116. On information and belief, Adobe has had knowledge of the '907 patent  
2 since at least the filing of this Complaint or shortly thereafter, and on information and  
3 belief, Adobe knew of the '907 patent and knew of its infringement, including by way  
4 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
5 receiving such notice) that its continued actions would actively induce and contribute  
6 to the infringement of the claims of the '907 patent.  
7

8           117. Upon information and belief, Adobe's affirmative acts of making, using,  
9 and selling the Accused Instrumentalities, and providing implementation services and  
10 technical support to users of the Accused Instrumentalities, including, e.g., through  
11 training, demonstrations, brochures, installation and user guides, have induced and  
12 continue to induce users of the Accused Instrumentalities to use them in their normal  
13 and customary way to infringe the '907 patent by practicing a system comprising: one  
14 or more different asymmetric data compression algorithms, wherein each algorithm of  
15 the one or more different asymmetric data compression algorithms utilizes one or  
16 more asymmetric data compression routines of a plurality of different asymmetric data  
17 compression routines, wherein a first asymmetric data compression routine of the  
18 plurality of different asymmetric data compression routines is configured to produce  
19 compressed data with a higher data rate for a given data throughput than a second  
20 asymmetric data compression routine of the plurality of different asymmetric data  
21 compression routines; and a processor configured: to analyze one or more data  
22 parameters from one or more data blocks containing video data, wherein at least one  
23 data parameter relates to an expected or anticipated throughput of a communications  
24  
25  
26  
27  
28



1 channel; and to select two or more different data compression routines from among a  
2 plurality of different data compression routines based upon, at least in part, the one or  
3 more data parameters relating to the expected or anticipated throughput of the  
4 communications channel. For example, Adobe adopted H.264 and/or H.264 with  
5 Scalable Video Coding (SVC) and/or an equivalent codec as its video codec in its  
6 products/services, such as, e.g., Adobe's media encoder, Flash player, Flash media  
7 server, Flash media server encoder, special effects and video editing software. For  
8 similar reasons, Adobe also induces its customers to use the Accused Instrumentalities  
9 to infringe other claims of the '907 patent. Adobe specifically intended and was  
10 aware that these normal and customary activities would infringe the '907 patent.  
11 Adobe performed the acts that constitute induced infringement, and would induce  
12 actual infringement, with the knowledge of the '907 patent and with the knowledge, or  
13 willful blindness to the probability, that the induced acts would constitute  
14 infringement. On information and belief, Adobe engaged in such inducement to  
15 promote the sales of the Accused Instrumentalities. Accordingly, Adobe has induced  
16 and continues to induce users of the Accused Instrumentalities to use the Accused  
17 Instrumentalities in their ordinary and customary way to infringe the '907 patent,  
18 knowing that such use constitutes infringement of the '907 patent. Accordingly,  
19 Adobe has been, and currently is, inducing infringement of the '907 patent, in  
20 violation of 35 U.S.C. § 271(b).  
21  
22  
23  
24  
25

26 118. Adobe has also infringed, and continues to infringe, claims of the '907  
27 patent by offering to commercially distribute, commercially distributing, making,  
28

1 and/or importing the Accused Instrumentalities, which are used in practicing the  
2 process, or using the systems, of the '907 patent, and constitute a material part of the  
3 invention. Adobe knows the components in the Accused Instrumentalities to be  
4 especially made or especially adapted for use in infringement of the '907 patent, not a  
5 staple article, and not a commodity of commerce suitable for substantial noninfringing  
6 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '907  
7 patent, in violation of 35 U.S.C. § 271(c).  
8  
9

10 119. By making, using, offering for sale, selling and/or importing into the  
11 United States the Accused Instrumentalities, and touting the benefits of using the  
12 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
13 liable to Realtime for infringement of the '907 patent pursuant to 35 U.S.C. § 271.  
14

15 120. As a result of Adobe's infringement of the '907 patent, Plaintiff Realtime  
16 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
17 infringement, but in no event less than a reasonable royalty for the use made of the  
18 invention by Adobe, together with interest and costs as fixed by the Court.  
19

## 20 **COUNT VII**

### 21 **INFRINGEMENT OF U.S. PATENT NO. 9,769,477**

22 23 121. Plaintiff re-alleges and incorporates by reference the foregoing  
24 paragraphs, as if fully set forth herein.

25 26 122. On information and belief, Adobe has made, used, offered for sale, sold  
27 and/or imported into the United States Adobe products that infringe the '477 patent,  
28 and continues to do so. By way of illustrative example, these infringing products

1 include, without limitation, Adobe's Adobe Media Encoder, Adobe Media Encoder  
2 Creative Cloud or CC, Adobe Media Encoder Creative Suite or CS (e.g. CS4, CS5,  
3 CS5.5, CS6), Adobe Premiere Pro, Adobe Premiere Pro CC, Adobe Premiere Pro CS  
4 (e.g. CS4, CS5, CS5.5, CS6), Adobe Flash, Adobe Flash Player (e.g. Adobe Flash  
5 Player 9), Adobe Flash Media Server, Adobe Flash Media Encoding Server, Adobe  
6 After Effects, Adobe After Effects CC, Adobe After Effects CS (e.g. CS4, CS5,  
7 CS5.5, CS6), Adobe HTTP Dynamic Streaming (HDS), and all versions and  
8 variations thereof since the issuance of the '477 patent ("Accused Instrumentalities").  
9  
10

11 123. On information and belief, Adobe has directly infringed and continues to  
12 infringe the '477 patent, for example, through its sale, offer for sale, importation, use  
13 and testing of the Accused Instrumentalities that practice, for example, Claim 1 of  
14 the '477 patent, namely, a system, comprising: a plurality of different asymmetric data  
15 compression encoders, wherein each asymmetric data compression encoder of the  
16 plurality of different asymmetric data compression encoders is configured to utilize  
17 one or more data compression algorithms, and wherein a first asymmetric data  
18 compression encoder of the plurality of different asymmetric data compression  
19 encoders is configured to compress data blocks containing video or image data at a  
20 higher data compression rate than a second asymmetric data compression encoder of  
21 the plurality of different asymmetric data compression encoders; and one or more  
22 processors configured to: determine one or more data parameters, at least one of the  
23 determined one or more data parameters relating to a throughput of a communications  
24 channel measured in bits per second; and select one or more asymmetric data  
25  
26  
27  
28

1 compression encoders from among the plurality of different asymmetric data  
2 compression encoders based upon, at least in part, the determined one or more data  
3 parameters. Upon information and belief, Adobe uses the Accused Instrumentalities to  
4 practice infringing methods for its own internal non-testing business purposes, while  
5 testing the Accused Instrumentalities, and while providing technical support and  
6 repair services for the Accused Instrumentalities to Adobe's customers.  
7

8           124. The Accused Instrumentalities include, or practice a system involving  
9 data compression with asymmetric compressors. For example, the Accused  
10 Instrumentalities utilize H.264 and/or H.264 with Scalable Video Coding (SVC)  
11 and/or an equivalent codec. H.264 is "the most widely used codec on the planet." *See*  
12 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx)  
13 [74735.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-H.264-74735.aspx). A "codec" is also a compression technology that has "two components,  
14 an **encoder to compress the files**, and a decoder to decompress. There are codecs  
15 for...video (Cinepark, MPEG-2, **H.264**, VP8)." *See*  
16 [http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx)  
17 [74487.aspx](http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx); <https://forums.adobe.com/thread/729526> ("Compression is essential for  
18 reducing the size of movies so that they can be stored, transmitted, and played back  
19 effectively. Compression is achieved by an encoder; decompression is achieved by a  
20 decoder. Encoders and decoders are known by the common term codec."). According  
21 to a website maintained by Adobe, "you **can produce H.264 video** with Adobe Media  
22 Encoder CS4 and Adobe Flash Media Encoding Server 3.5," "Adobe added **H.264**  
23 **playback support** to Adobe Flash Player 9 Update 3 back in 2007," Adobe "Flash  
24  
25  
26  
27  
28

1 Media Server and Flash Player can **both stream and play back any H.264 file** in  
2 virtually any format,” the encoding capabilities of the Adobe Flash Media Encoding  
3 Server using “**H.264-related parameters**” are described (e.g. using CAVLC, CABAC)  
4 and under a headline stating “Adobe Media Encoder,” the website states: “Adobe  
5 significantly enhanced the Flash Video Encoder in Creative Suite 4. There is now both  
6 stand-alone operation and batch encoding capabilities. As before, you can **access**  
7 **H.264 encoding** by choosing different formats in the Format pop-up menu. When  
8 producing for Flash Player, you should always use the FLV|F4V option, which lets  
9 you **produce both VP6- and H.264-encoded files** for Flash Player distribution.” *See*  
10 [http://www.adobe.com/devnet/adobe-media-server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).  
11

12  
13 Another website maintained by Adobe mentions that “You have two primary options  
14 for using Adobe Media Encoder with After Effects to create videos in **H.264.**” The  
15 website also mentions that “Since Adobe Media Encoder already has superior  
16 H.264...exporting capabilities, it was more prudent to rely on Adobe Media Encoder  
17 for export of these formats.” *See* [https://helpx.adobe.com/after-effects/kb/export-](https://helpx.adobe.com/after-effects/kb/export-h264.html)  
18 [h264.html](https://helpx.adobe.com/after-effects/kb/export-h264.html). Another website maintained by Adobe mentions that the “Adobe Flash  
19 Player 9 Update 3 is taking a step into the high-definition (HD) video realm in a major  
20 way by adding MPEG-4 video...[which] utilizes crisp, powerful **H.264 encoding**”  
21 and “The new Adobe Media Player...will **also support H.264**” and “The addition of  
22 **H.264...support** in Flash Player 9 Update 3 allows you to easily use high-definition,  
23 industry standard video and audio.” *See*  
24 [http://www.adobe.com/devnet/flashplayer/articles/hd\\_video\\_flash\\_player.html](http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html). Adobe  
25  
26  
27  
28

1 HTTP Dynamic Streaming (HDS) also supports methods involving data compression  
2 with asymmetric compressors or codecs other than H.264 because it “enables high-  
3 quality (**H.264 or VP6**), network-efficient HTTP streaming for media delivery that is  
4 tightly integrated with Adobe® Access software for robust content protection in the  
5 Adobe Flash® Player 10.1 or later and Adobe AIR®2 or later runtimes,” is also “built  
6 on standards and deployed using standard HTTP servers (Apache), standard media  
7 format (MP4 fragment) using standard codecs (**H.264/AAC**), open APIs (Flash  
8 Player), and an open source framework for building media players (OSMF)” and “like  
9 other Flash Player supported delivery methods,” Adobe HDS “supports **H.264 video**  
10 **and VP6 codecs** required for the highest quality video on the platform.” *See*  
11 <https://www.adobe.com/products/hds-dynamic-streaming/faq.html>. Adobe HDS also  
12 allows users to encode content “using high-quality Flash Player compatible **codecs**  
13 (**VP6/MP3, H264/AAC**),” deliver “high-definition video up to 1080p, with bitrates  
14 from 700kbps up to and beyond 6Mbps, using either **H.264 or VP6 video codecs**,”  
15 and archive “live, high-definition streams on the server and enable HD DVR  
16 functionality (for example, instant replay and time shifting) with support for **H.264**  
17 stream recording for RTMP and HTTP Dynamic Streaming.” *See*  
18 <https://www.adobe.com/products/hds-dynamic-streaming/features.html>. Adobe Media  
19 Encoder also “lets you **produce both VP6- and H.264-encoded files** for Flash Player  
20 distribution.” *See* [http://www.adobe.com/devnet/adobe-media-](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
21 [server/articles/h264\\_encoding.html](http://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html).

22  
23  
24  
25  
26  
27  
28 125. The Accused Instrumentalities also include or practice a system,

1 comprising: a plurality of different asymmetric data compression encoders, wherein  
2 each asymmetric data compression encoder of the plurality of different asymmetric  
3 data compression encoders is configured to utilize one or more data compression  
4 algorithms. The one or more asymmetric data compression encoders can be, for  
5 example, H.264 and/or H.264 with Scalable Video Coding (SVC) and/or an  
6 equivalent codec. Based on a bitrate, throughput and/or resolution parameter identified  
7 (e.g. bitrate, max video bitrate, resolution, group of pictures (GOP) structure or frame  
8 type within a GOP structure), any H.264-compliant system such as the Accused  
9 Instrumentalities would determine which profile (e.g., “baseline,” “extended,” “main”,  
10 or “high”) corresponds with that parameter, then select between at least two  
11 asymmetric compressors, which may serve as an asymmetric data compression  
12 encoder or the one or more data compression algorithms. If baseline or extended is  
13 the corresponding profile, then the system will select a Context-Adaptive Variable  
14 Length Coding (“CAVLC”) entropy encoder, which may serve as an asymmetric data  
15 compression encoder, data compression algorithm or asymmetric compressor. If main  
16 or high is the corresponding profile, then the system will select a Context-Adaptive  
17 Binary Arithmetic Coding (“CABAC”) entropy encoder, which may serve as another  
18 asymmetric data compression encoder, data compression algorithm or asymmetric  
19 compressor. Both encoders are asymmetric compressors because it takes a longer  
20 period of time for them to compress data than to decompress data. *See*  
21 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
22  
23  
24  
25  
26  
27  
28

	<b>Baseline</b>	<b>Extended</b>	<b>Main</b>	<b>High</b>	<b>High 10</b>
<b>I and P Slices</b>	Yes	Yes	Yes	Yes	Yes
<b>B Slices</b>	No	Yes	Yes	Yes	Yes
<b>SI and SP Slices</b>	No	Yes	No	No	No
<b>Multiple Reference Frames</b>	Yes	Yes	Yes	Yes	Yes
<b>In-Loop Deblocking Filter</b>	Yes	Yes	Yes	Yes	Yes
<b>CAVLC Entropy Coding</b>	Yes	Yes	Yes	Yes	Yes
<b>CABAC Entropy Coding</b>	No	No	Yes	Yes	Yes
<b>Flexible Macroblock Ordering (FMO)</b>	Yes	Yes	No	No	No
<b>Arbitrary Slice Ordering (ASO)</b>	Yes	Yes	No	No	No
<b>Redundant Slices (RS)</b>	Yes	Yes	No	No	No
<b>Data Partitioning</b>	No	Yes	No	No	No
<b>Interlaced Coding (PicAFF, MBAFF)</b>	No	Yes	Yes	Yes	Yes
<b>4:2:0 Chroma Format</b>	Yes	Yes	Yes	Yes	Yes
<b>Monochrome Video Format (4:0:0)</b>	No	No	No	Yes	Yes
<b>4:2:2 Chroma Format</b>	No	No	No	No	No
<b>4:4:4 Chroma Format</b>	No	No	No	No	No
<b>8 Bit Sample Depth</b>	Yes	Yes	Yes	Yes	Yes
<b>9 and 10 Bit Sample Depth</b>	No	No	No	No	Yes
<b>11 to 14 Bit Sample Depth</b>	No	No	No	No	No
<b>8x8 vs. 4x4 Transform Adaptivity</b>	No	No	No	Yes	Yes
<b>Quantization Scaling Matrices</b>	No	No	No	Yes	Yes
<b>Separate Cb and Cr QP control</b>	No	No	No	Yes	Yes
<b>Separate Color Plane Coding</b>	No	No	No	No	No
<b>Predictive Lossless Coding</b>	No	No	No	No	No

See

[http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf) at 7:



The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

#### H.264 Entropy Coding – Comparison of Approaches

Characteristics	Variable Length Coding (VLC)	Context Adaptive Binary Arithmetic Coding(CABAC)
• Where it is used	MPEG-2, MPEG-4 ASP	H.264/MPEG-4 AVC (high efficiency option)
• Probability distribution	<b>Static</b> - Probabilities never change	<b>Adaptive</b> - Adjusts probabilities based on actual data
• Leverages correlation between symbols	<b>No</b> - Conditional probabilities ignored	<b>Yes</b> - Exploits symbol correlations by using "contexts"
• Non-integer code words	<b>No</b> - Low coding efficiency for high probability symbols	<b>Yes</b> - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items) (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy\_coding\_mode\_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

- If **entropy\_coding\_mode\_flag** is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).
- Otherwise (**entropy\_coding\_mode\_flag** is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

126. The Accused Instrumentalities also practice wherein a first asymmetric data compression encoder of the plurality of different asymmetric data compression encoders is configured to compress data blocks containing video or image data at a

1 higher data compression rate than a second asymmetric data compression encoder of  
2 the plurality of different asymmetric data compression encoders. For example, Adobe  
3 HTTP Dynamic Streaming (HDS) contains the “ability to shift quality [and  
4 **compression rate] depending on bandwidth and computer power.** HTTP  
5 progressive delivery consumes more bandwidth because it's not intelligent enough to  
6 throttle the delivery (for example, a 30-minute video will be fully downloaded  
7 whether the user watches it or not).” See [https://www.adobe.com/products/hds-](https://www.adobe.com/products/hds-dynamic-streaming/faq.html)  
8 [dynamic-streaming/faq.html](https://www.adobe.com/products/hds-dynamic-streaming/faq.html). Adobe HDS can also “[d]etect the client's bandwidth  
9 **and computer resources** and serve them content fragments encoded at the most  
10 appropriate bitrate.” See [https://www.adobe.com/products/hds-dynamic-](https://www.adobe.com/products/hds-dynamic-streaming/features.html)  
11 [streaming/features.html](https://www.adobe.com/products/hds-dynamic-streaming/features.html); [https://www.adobe.com/devnet/adobe-media-](https://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html)  
12 [server/articles/h264\\_encoding.html](https://www.adobe.com/devnet/adobe-media-server/articles/h264_encoding.html) (“Like MPEG-2, H.264 uses three types of frames,  
13 meaning that each group of pictures (GOP) is comprised of I-, B-, and P-frames, with  
14 I-frames like the DCT-based compression used in DV and B- and P-frames  
15 referencing redundancies in other frames **to increase compression**...Finally, though  
16 it's not technically related to B-frames, consider the number of Slices per picture,  
17 which can be 1, 2, or 4. At a value of 4, the encoder divides each frame into four  
18 regions and searches for redundancies in other frames only within the respective  
19 region. This can **accelerate encoding** on multicore computers because the encoder  
20 can assign the regions to different cores.”).

21  
22  
23  
24  
25  
26  
27  
28  
127. A video data block is organized by the group of pictures (GOP) structure,  
which is a “collection of successive pictures within a coded video stream.” See

1 [https://en.wikipedia.org/wiki/Group\\_of\\_pictures](https://en.wikipedia.org/wiki/Group_of_pictures). A GOP structure can contain intra  
2 coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame),  
3 bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture  
4 or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See*  
5 [https://en.wikipedia.org/wiki/Video\\_compression\\_picture\\_types](https://en.wikipedia.org/wiki/Video_compression_picture_types) (for descriptions of I  
6 frames, P frames and B frames); <https://en.wikipedia.org/wiki/MPEG-1#D-frames> (for  
7 descriptions of D frames). Thus, at least a portion of a video data block would also  
8 make up a GOP structure and could also contain I frames, P frames, B frames and/or  
9 D frames. The GOP structure also reflects the size of a video data block, and the GOP  
10 structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max  
11 video bitrate and resolution parameters) or even be considered as a parameter by itself.

12  
13  
14  
15 128. The compressed data blocks that the different asymmetric data  
16 compression encoders are configured to compress can also be organized in a GOP  
17 structure (see above). After its selection, the asymmetric compressor (CAVLC or  
18 CABAC) will compress the video data to provide various compressed data blocks,  
19 which can also be organized in a GOP structure, as discussed previously above. *See*  
20 <https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/>:  
21

### 22 **Entropy Coding**

23 For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive  
24 variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic  
25 coding (CABAC) which are complex techniques to losslessly compress syntax elements in the  
26 video stream knowing the probabilities of syntax elements in a given context. The use of  
27 CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total  
28 processing power to be accomplished.

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

*See*

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf> at 13:

Typical compression ratios to maintain excellent quality are:

- 10:1 for general images using JPEG
- 30:1 for general video using H.263 and MPEG-2
- 60:1 for general video using H.264 and WMV9

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles includes the functionality of the Baseline Profile and add improvements to the predictions algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

*See* [http://www.ijera.com/papers/Vol3\\_issue4/BM34399403.pdf](http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf) at 2:

129. The Accused Instrumentalities also include or practice a system, comprising: one or more processors configured to: determine one or more data

1 parameters, at least one of the determined one or more data parameters relating to a  
2 throughput of a communications channel measured in bits per second; and select one  
3 or more asymmetric data compression encoders from among the plurality of different  
4 asymmetric data compression encoders based upon, at least in part, the determined  
5 one or more data parameters. As shown below, examples of such data parameters  
6 include throughput or bandwidth, bitrate (or max video bitrate), and resolution.  
7 Different parameters correspond with different end applications. The asymmetric data  
8 compression encoder will then be selected based on the determined one or more data  
9 parameters, as discussed above. For example, based on the determined one or more  
10 data parameters, such as a bitrate, throughput and/or resolution parameter that is  
11 determined (e.g. bitrate, max video bitrate, resolution, group of pictures (GOP)  
12 structure or frame type within a GOP structure), any H.264-compliant system such as  
13 the Accused Instrumentalities would determine which profile (e.g., “baseline,”  
14 “extended,” “main”, or “high”) corresponds with that data parameter, then accordingly  
15 select an asymmetric data compression encoder based on that data parameter. H.264  
16 provides for multiple different ranges of such data parameters, each included in the  
17 “profiles” and “levels” as defined by the H.264 standard, from the below shown  
18 paragraphs from a white paper and Wikipedia. See  
19 [http://www.axis.com/files/whitepaper/wp\\_h264\\_31669\\_en\\_0803\\_lo.pdf](http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf) at 5:  
20  
21  
22  
23  
24  
25  
26  
27  
28

4. H.264 profiles and levels

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

See [https://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC](https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC):

Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) <a href="#">Toggle additional details</a>
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	176x144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	176x144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	352x288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	352x288@15.2 (6)
1.3	3,041,280	11,880	101,376	396	768	960	2,304	352x288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	352x288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352x576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	720x576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	720x576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	1,280x720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280x1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048x1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048x1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048x1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	3,672x1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	4,096x2,304@56.3 (5)

130. Therefore, from at least the above, Adobe has directly infringed and continues to infringe the '477 patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by, for

1 example, Claim 1 of the '477 patent, namely, a system, comprising: a plurality of  
2 different asymmetric data compression encoders, wherein each asymmetric data  
3 compression encoder of the plurality of different asymmetric data compression  
4 encoders is configured to utilize one or more data compression algorithms, and  
5 wherein a first asymmetric data compression encoder of the plurality of different  
6 asymmetric data compression encoders is configured to compress data blocks  
7 containing video or image data at a higher data compression rate than a second  
8 asymmetric data compression encoder of the plurality of different asymmetric data  
9 compression encoders; and one or more processors configured to: determine one or  
10 more data parameters, at least one of the determined one or more data parameters  
11 relating to a throughput of a communications channel measured in bits per second;  
12 and select one or more asymmetric data compression encoders from among the  
13 plurality of different asymmetric data compression encoders based upon, at least in  
14 part, the determined one or more data parameters. Upon information and belief,  
15 Adobe uses the Accused Instrumentalities to practice infringing methods for its own  
16 internal non-testing business purposes, while testing the Accused Instrumentalities,  
17 and while providing technical support and repair services for the Accused  
18 Instrumentalities to their customers.

19  
20  
21  
22  
23  
24 131. On information and belief, Adobe also directly infringes and continues to  
25 infringe other claims of the '477 patent.

26  
27 132. On information and belief, all of the Accused Instrumentalities perform  
28 the claimed methods in substantially the same way, e.g., in the manner specified in the

1 H.264 standard.

2 133. On information and belief, use of the Accused Instrumentalities in their  
3 ordinary and customary fashion results in infringement of the systems and/or methods  
4 claimed by the '477 patent.  
5

6 134. On information and belief, Adobe has had knowledge of the '477 patent  
7 since at least the filing of this Complaint or shortly thereafter, and on information and  
8 belief, Adobe knew of the '477 patent and knew of its infringement, including by way  
9 of this lawsuit. By the time of trial, Adobe will have known and intended (since  
10 receiving such notice) that its continued actions would actively induce and contribute  
11 to the infringement of the claims of the '477 patent.  
12

13 135. Upon information and belief, Adobe's affirmative acts of making, using,  
14 and selling the Accused Instrumentalities, and providing implementation services and  
15 technical support to users of the Accused Instrumentalities, including, e.g., through  
16 training, demonstrations, brochures, installation and user guides, have induced and  
17 continue to induce users of the Accused Instrumentalities to use them in their normal  
18 and customary way to infringe the '477 patent by practicing a system, comprising: a  
19 plurality of different asymmetric data compression encoders, wherein each  
20 asymmetric data compression encoder of the plurality of different asymmetric data  
21 compression encoders is configured to utilize one or more data compression  
22 algorithms, and wherein a first asymmetric data compression encoder of the plurality  
23 of different asymmetric data compression encoders is configured to compress data  
24 blocks containing video or image data at a higher data compression rate than a second  
25  
26  
27  
28



1 asymmetric data compression encoder of the plurality of different asymmetric data  
2 compression encoders; and one or more processors configured to: determine one or  
3 more data parameters, at least one of the determined one or more data parameters  
4 relating to a throughput of a communications channel measured in bits per second;  
5 and select one or more asymmetric data compression encoders from among the  
6 plurality of different asymmetric data compression encoders based upon, at least in  
7 part, the determined one or more data parameters. For example, Adobe adopted  
8 H.264 and/or H.264 with Scalable Video Coding (SVC) and/or an equivalent codec as  
9 its video codec in its products/services, such as, e.g., Adobe's media encoder, Flash  
10 player, Flash media server, Flash media server encoder, special effects and video  
11 editing software. For similar reasons, Adobe also induces its customers to use the  
12 Accused Instrumentalities to infringe other claims of the '477 patent. Adobe  
13 specifically intended and was aware that these normal and customary activities would  
14 infringe the '477 patent. Adobe performed the acts that constitute induced  
15 infringement, and would induce actual infringement, with the knowledge of the '477  
16 patent and with the knowledge, or willful blindness to the probability, that the induced  
17 acts would constitute infringement. On information and belief, Adobe engaged in  
18 such inducement to promote the sales of the Accused Instrumentalities. Accordingly,  
19 Adobe has induced and continues to induce users of the Accused Instrumentalities to  
20 use the Accused Instrumentalities in their ordinary and customary way to infringe  
21 the '477 patent, knowing that such use constitutes infringement of the '477 patent.  
22 Accordingly, Adobe has been, and currently is, inducing infringement of the '477  
23  
24  
25  
26  
27  
28

1 patent, in violation of 35 U.S.C. § 271(b).

2 136. Adobe has also infringed, and continues to infringe, claims of the '477  
3 patent by offering to commercially distribute, commercially distributing, making,  
4 and/or importing the Accused Instrumentalities, which are used in practicing the  
5 process, or using the systems, of the '477 patent, and constitute a material part of the  
6 invention. Adobe knows the components in the Accused Instrumentalities to be  
7 especially made or especially adapted for use in infringement of the '477 patent, not a  
8 staple article, and not a commodity of commerce suitable for substantial noninfringing  
9 use. Accordingly, Adobe has been, and currently is, contributorily infringing the '477  
10 patent, in violation of 35 U.S.C. § 271(c).

11 137. By making, using, offering for sale, selling and/or importing into the  
12 United States the Accused Instrumentalities, and touting the benefits of using the  
13 Accused Instrumentalities' compression features, Adobe has injured Realtime and is  
14 liable to Realtime for infringement of the '477 patent pursuant to 35 U.S.C. § 271.

15 138. As a result of Adobe's infringement of the '477 patent, Plaintiff Realtime  
16 is entitled to monetary damages in an amount adequate to compensate for Adobe's  
17 infringement, but in no event less than a reasonable royalty for the use made of the  
18 invention by Adobe, together with interest and costs as fixed by the Court.

19  
20  
21  
22  
23  
24 **PRAYER FOR RELIEF**

25 WHEREFORE, Plaintiff Realtime respectfully requests that this Court enter:  
26  
27  
28

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

- a. A judgment in favor of Plaintiff that Defendant has infringed, literally and/or under the doctrine of equivalents, the '046, '777, '442, '535, '298, '907, and '477 patents (the "Asserted Patents");
- b. A judgment and order requiring Defendant to pay Plaintiff its damages, costs, expenses, and prejudgment and post-judgment interest for its infringement of the Asserted Patents, as provided under 35 U.S.C. § 284;
- c. A judgment and order requiring Defendant to provide an accounting and to pay supplemental damages to Realtime, including without limitation, prejudgment and post-judgment interest;
- d. A permanent injunction prohibiting Defendant from further acts of infringement of the Asserted Patents;
- e. A judgment and order finding that this is an exceptional case within the meaning of 35 U.S.C. § 285 and awarding to Plaintiff its reasonable attorneys' fees against Defendant; and
- f. Any and all other relief as the Court may deem appropriate and just under the circumstances.

**DEMAND FOR JURY TRIAL**

Plaintiff, under Rule 38 of the Federal Rules of Civil Procedure, requests a trial by jury of any issues so triable by right.

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

Dated: December 21, 2018

Respectfully Submitted,

/s/ Marc A. Fenster

RUSS AUGUST & KABAT  
Marc A. Fenster, SBN 181067  
Email: mfenster@raklaw.com  
Reza Mirzaie (CA SBN 246953)  
Email: rmirzaie@raklaw.com  
Brian D. Ledahl (CA SBN 186579)  
Email: bledahl@raklaw.com  
Paul Kroeger (CA SBN 229074)  
Email: pkroeger@raklaw.com  
C. Jay Chung (CA SBN 252794)  
Email: jchung@raklaw.com  
Philip X. Wang (CA SBN 262239)  
Email: pwang@raklaw.com

*Attorneys for Plaintiff*  
*REALTIME ADAPTIVE STREAMING LLC*

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

**CERTIFICATE OF SERVICE**

I hereby certify that the foregoing document was served on all counsel of record via electronic service on December 21, 2018.

/s/ Marc A. Fenster