

1 M. ELIZABETH DAY (SBN 177125)

2 eday@feinday.com

3 DAVID ALBERTI (SBN 220265)

4 dalberti@feinday.com

5 SAL LIM (SBN 211836)

6 slim@feinday.com

7 MARC BELLOLI (SBN 244290)

8 mbelloli@feinday.com

9 **FEINBERG DAY ALBERTI LIM &**
10 **BELLOLI LLP**

11 1600 El Camino Real, Suite 280

12 Menlo Park, CA 94025

13 Tel: 650.618.4360

14 Fax: 650.618.4368

15 Attorneys for Uniloc 2017 LLC and Uniloc

16 Licensing USA LLC.

17 UNITED STATES DISTRICT COURT

18 CENTRAL DISTRICT OF CALIFORNIA

19 UNILOC 2017 LLC and UNILOC
20 LICENSING USA LLC,

21 Plaintiffs,

22 v.

23 NETFLIX, INC.,

24 Defendant.

CASE NO. 8:18-cv-01899

**COMPLAINT FOR PATENT
INFRINGEMENT**

DEMAND FOR JURY TRIAL

1 Plaintiffs Uniloc 2017 LLC and Uniloc Licensing USA LLC (collectively
2 “Uniloc”), by and through the undersigned counsel, hereby file this Complaint and
3 make the following allegations of patent infringement relating to U.S. Patent Nos.
4 8,407,609, 6,584,229 and 6,519,005 against Defendant Netflix, Inc., and allege as
5 follows upon actual knowledge with respect to themselves and their own acts and
6 upon information and belief as to all other matters:

7 **NATURE OF THE ACTION**

8 1. This is an action for patent infringement. Uniloc alleges that Netflix
9 infringes U.S. Patent Nos. 8,407,609 (the “’609 patent”), 6,584,229 (the “’229
10 patent) and 6,519,005 (the “’005 patent”), copies of which are attached hereto as
11 Exhibits A-C (collectively, “the Asserted Patents”).

12 2. Uniloc alleges that Netflix directly infringes the Asserted Patents by
13 making, using, offering for sale, selling, licensing and/or importing products and
14 services that: (1) track digital media presentations such as Netflix Service, (2)
15 perform coding of image signals using the VP9 codec and (3) perform a method for
16 motion coding an uncompressed (pixel level) digital video data stream. Uniloc
17 seeks damages and other relief for Netflix’s infringement of the Asserted Patents.

18 **THE PARTIES**

19 3. Uniloc 2017 LLC is a Delaware corporation having places of business
20 at 1209 Orange Street, Wilmington, Delaware 19801 and 620 Newport Center
21 Drive, Newport Beach, California 92660.

22 4. Uniloc Licensing USA LLC is a Delaware corporation having places
23 of business at 1209 Orange Street, Wilmington, Delaware 19801 and 620 Newport
24 Center Drive, Newport Beach, California 92660.

25 5. Uniloc holds all substantial rights, title and interest in and to the
26 Asserted Patents.

27 6. Upon information and belief, Defendant Netflix, Inc., is a corporation
28

1 organized and existing under the laws of the State of Delaware, with at least the
2 following places of business in this District: 5808 Sunset Blvd, Los Angeles, CA
3 90028 and 335 N. Maple Dr. Beverly Hills, CA 90210. Netflix can be served with
4 process by serving its registered agent for service of process in California: CT
5 Corporation System, 818 W. Seventh St, Suite 930, Los Angeles, CA. 90017.

6 **JURISDICTION AND VENUE**

7 7. This action for patent infringement arises under the Patent Laws of the
8 United States, 35 U.S.C. § 1 et. seq. This Court has original jurisdiction under 28
9 U.S.C. §§ 1331 and 1338.

10 8. This Court has both general and specific jurisdiction over Netflix
11 because Netflix has committed acts within the Central District of California giving
12 rise to this action and has established minimum contacts with this forum such that
13 the exercise of jurisdiction over Netflix would not offend traditional notions of fair
14 play and substantial justice. Defendant Netflix, directly and through subsidiaries,
15 intermediaries (including distributors, retailers, franchisees and others), has
16 committed and continues to commit acts of patent infringement in this District, by,
17 among other things, making, using, testing, selling, licensing, importing and/or
18 offering for sale/license products and services that infringe the Asserted Patents.

19 9. Venue is proper in this district and division under 28 U.S.C. §§
20 1391(b)-(d) and 1400(b) because Netflix has committed acts of infringement in the
21 Central District of California and has multiple regular and established places of
22 business in the Central District of California.

23 **COUNT I – INFRINGEMENT OF U.S. PATENT NO. 8,407,609**

24 10. The allegations of paragraphs 1-9 of this Complaint are incorporated
25 by reference as though fully set forth herein.

26 11. The '609 patent, titled "System and Method For Providing And
27 Tracking The Provision of Audio and Visual Presentations Via A Computer
28

1 Network,” issued on March 26, 2013. A copy of the ’609 patent is attached as
2 Exhibit A.

3 12. Pursuant to 35 U.S.C. § 282, the ’609 patent is presumed valid.

4 13. Upon information and belief, Netflix makes, uses, offers for sale, sells
5 in the United States, licenses in the United States and/or imports into the United
6 States Netflix Service which provides a method for tracking digital media
7 presentations (collectively the “Accused Infringing Devices”).

8 14. Upon information and belief, the Accused Infringing Devices infringe
9 at least claim 1 in the exemplary manner described below.

10 15. The Accused Infringing Devices track digital media presentations
11 delivered from a first computer system to a user’s computer via a network.

Viewing Data

12
13
14 Who, What, When, Where, How Long



10/13/14

"city": "PLEASANTON",
"region_code": "CA",



Duration

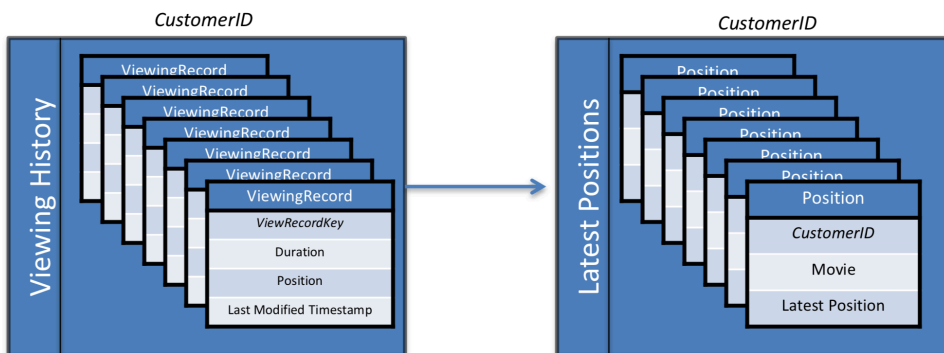
0:15:11

Latest
Position

14:41

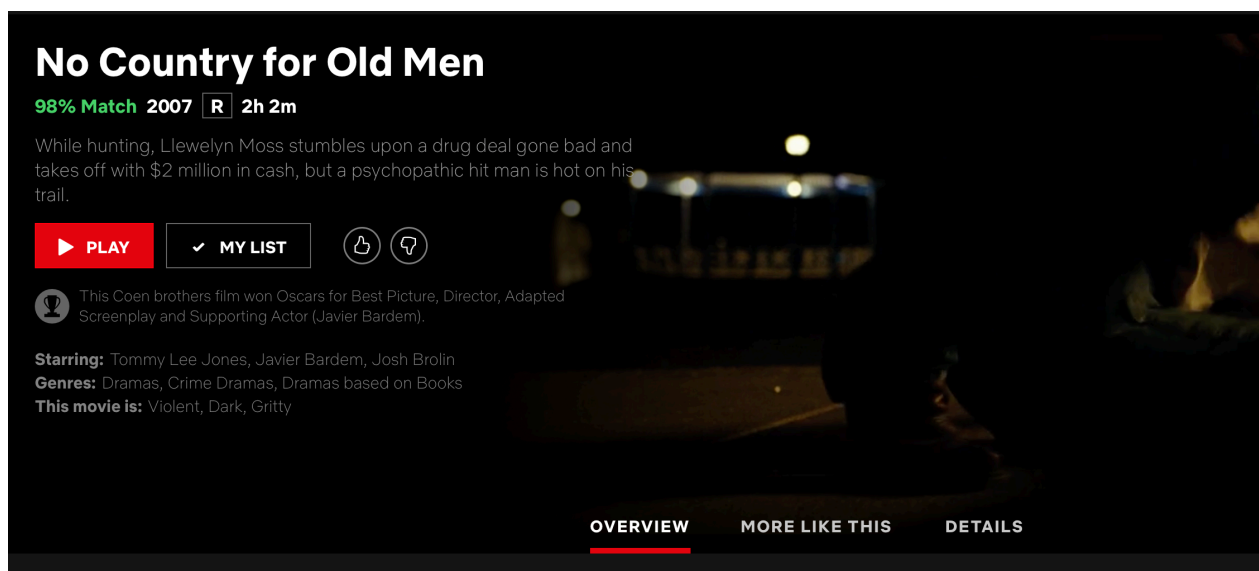
20 Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 9.

Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 30.

16. The Accused Infringing Devices provide a corresponding web page to the user's computer for each digital media presentation to be delivered using the first computer system.



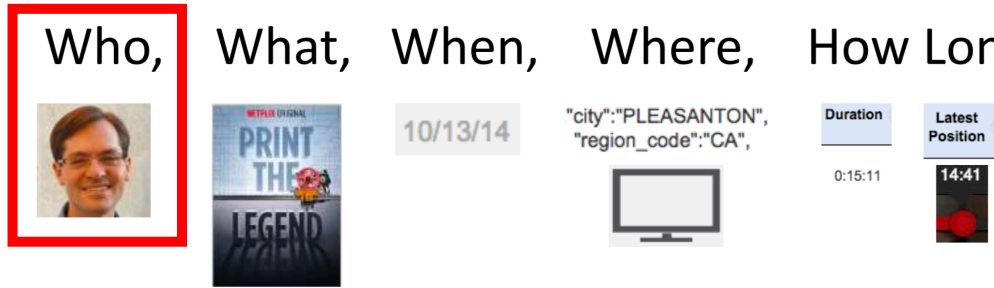
Source: Screenshot

17. The Accused Infringing Devices provide identifier data to the user's computer using the first computer system.

Viewing Data

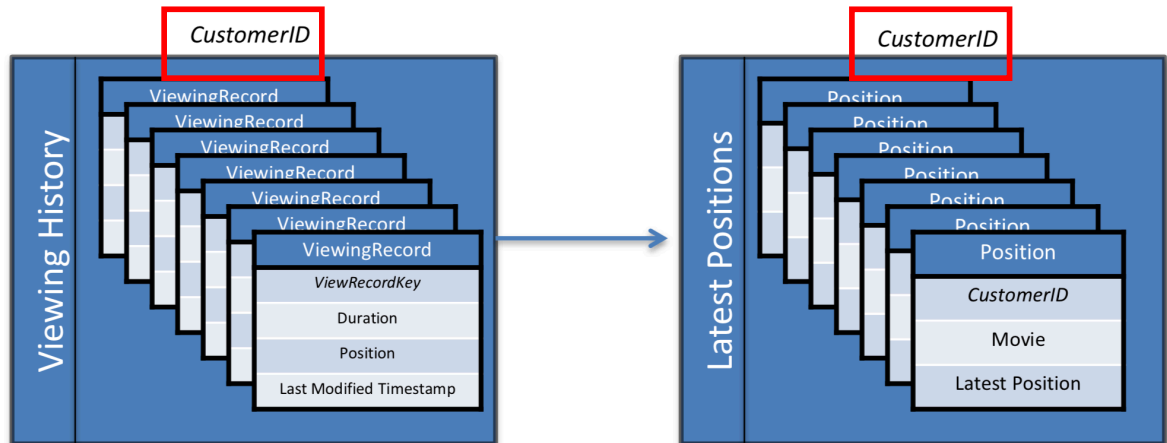
Who,

What, When, Where, How Long



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 9.

Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 30

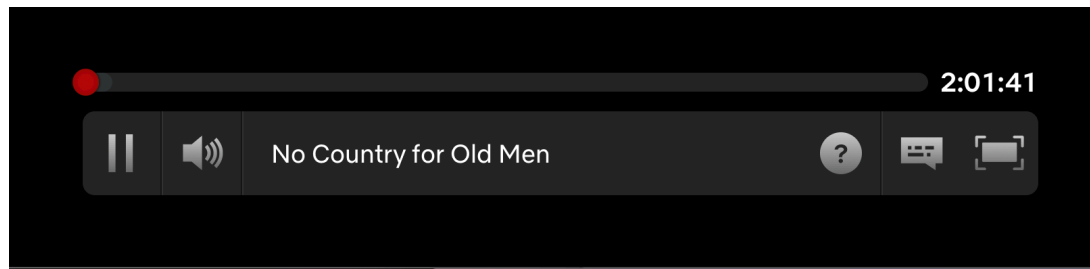
18. The Accused Infringing Devices provide an applet to the user's computer for each digital media presentation to be delivered using the first computer system. In particular, the Accused Infringing Devices provide a script that keeps track of how much of the presentation the user has watched, thus reflecting the operation of a time running in the background.

```

1      C.r("akira/components/controls/Progress/Progress.jsx", function(r, s, e) {
2          "use strict";
3          var t = r("prop-types"),
4              n = r("react"),
5              o = r("create-react-class"),
6              a = o({
7                  displayName: "Progress",
8                  contextTypes: {
9                      getI18nString: t.func.isRequired
10                 },
11                 render: function() {
12                     var r = this.props.variants,
13                         s = parseInt(this.props.bookmarkPosition / 60, 10),
14                         e = parseInt(this.props.runtime / 60, 10);
15                     if (!this.props.runtime)
16                         return null;
17                     var t = {
18                         width: parseInt(100 * this.props.bookmarkPosition / this.props.runtime, 10) + "%"
19                     },
20                     o = r ? r.map(function(r) {
21                         return "progress--" + r
22                     }).join(" ") : "";
23                     return n.createElement("div", {
24                         className: "progress " + o
25                     }, n.createElement("span", {
26                         className: "progress-bar"
27                     }, n.createElement("span", {
28                         role: "presentation",
29                         className: "progress-completed",
30                         style: t
31                     })), this.props.showSummary ? n.createElement("span", {
32                         className: "summary"
33                     }, this.context.getI18nString("discovery/akira/Common", "progress.summary", {
34                         BOOKMARK: s,
35                         RUNTIME: e
36                     }))) : null)
37                 }
38             });
39         s.exports = a
40     });

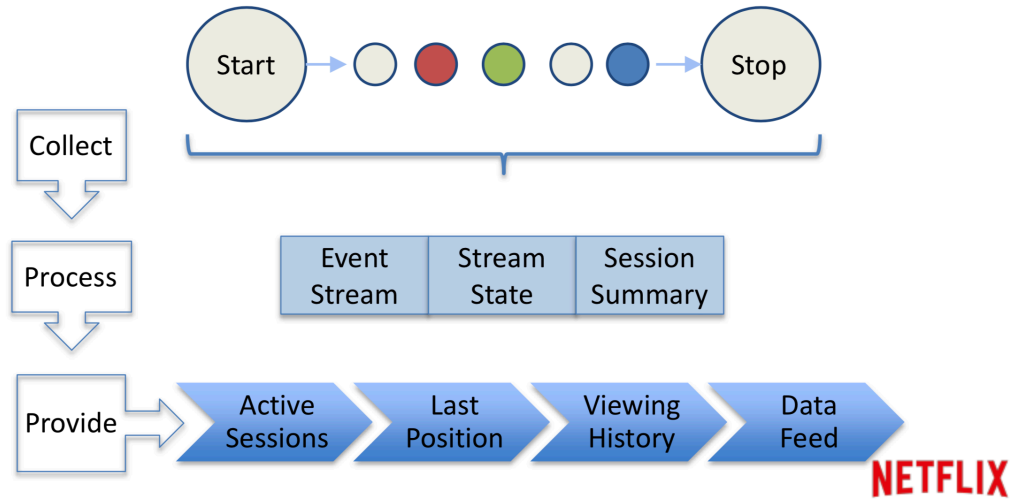
```

Source: Script from <http://codex.nflxext.com>.



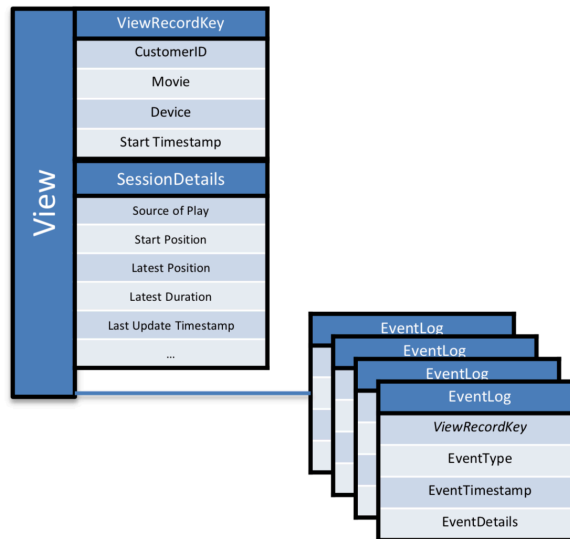
Source: Screenshot

Generic Architecture



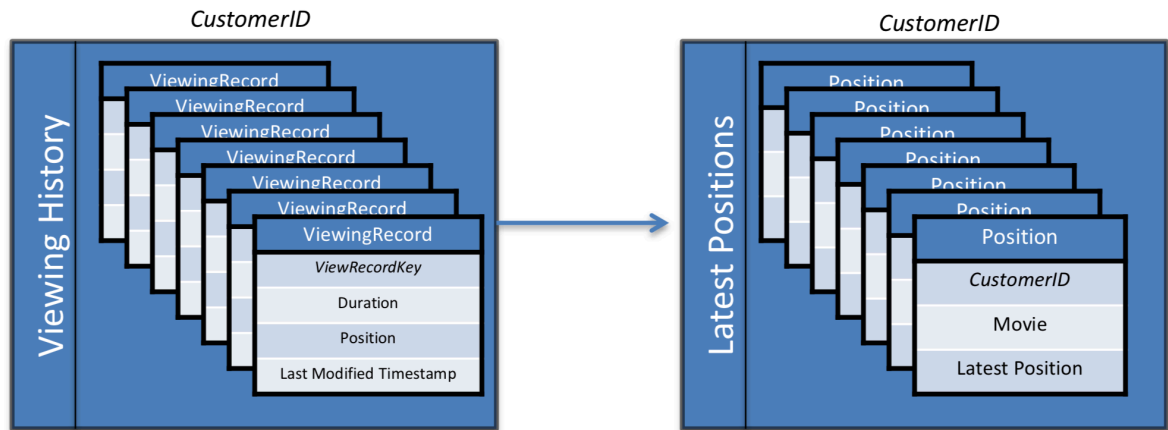
Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 24.

Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 29.

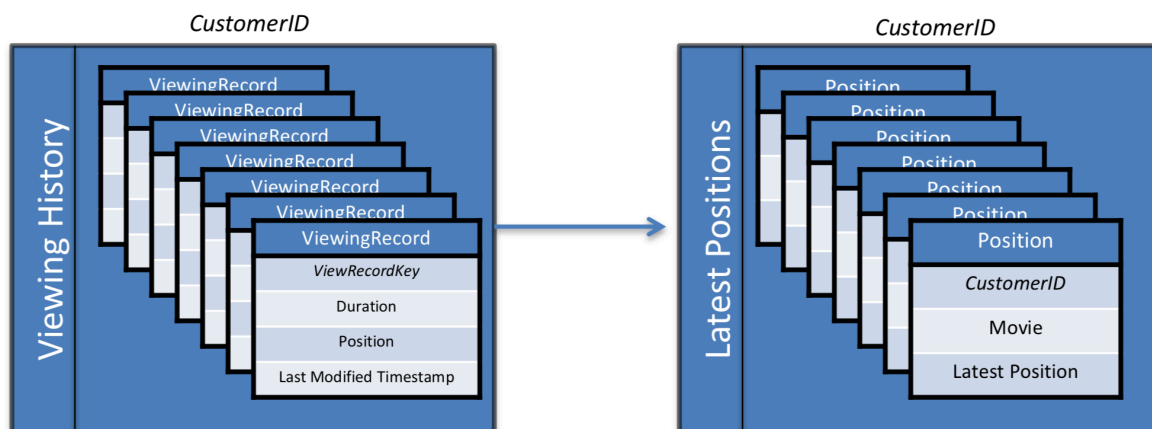
Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 30.

19. The Accused Infringing Devices receive at least a portion of the identifier data from the user's computer responsively to the timer applet each time a predetermined temporal period elapses using the first computer system. The Accused Infringing Devices maintain a viewing history for each user, shown below as a collection of consecutive viewing records. Each viewing record includes a ViewRecordKey to identify a program as well as the duration, position and last modified timestamp at the time the viewing record was updated. The viewing record is updated periodically, at least when a heartbeat is sent reflecting operation of a timer.

Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 30.

20. In addition, the Accused Infringing Devices' player periodically sends a heartbeat to the server to indicate the current playback position in the event the user needs to return to the current position in the event of an error or crash. The heartbeat is associated with the active session, which by definition, must identify the user. The heartbeat is sent at predetermined intervals. The Accused Infringing Devices also take snapshots of the viewing history, including positions, at intervals.

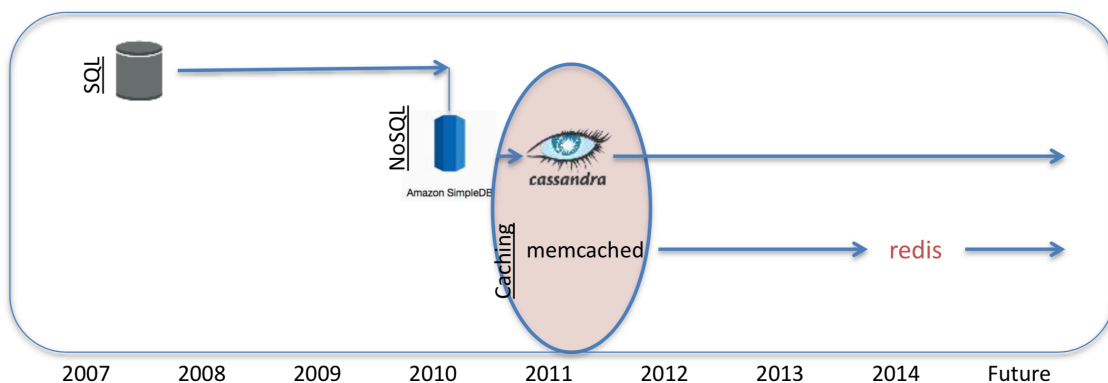
Command Use Cases

Action	Operation	Key	DataSet
Start	Insert	ViewRecordKey	ActiveSession ViewingRecord
Continue (heartbeat)	Update	ViewRecordKey	ActiveSession
Log	Insert	ViewRecordKey	EventLog
Stop	Update	ViewRecordKey	ActiveSession ViewingRecord
Snapshot	Insert/Update	CustomerID	ViewingHistory Positions

Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 31.

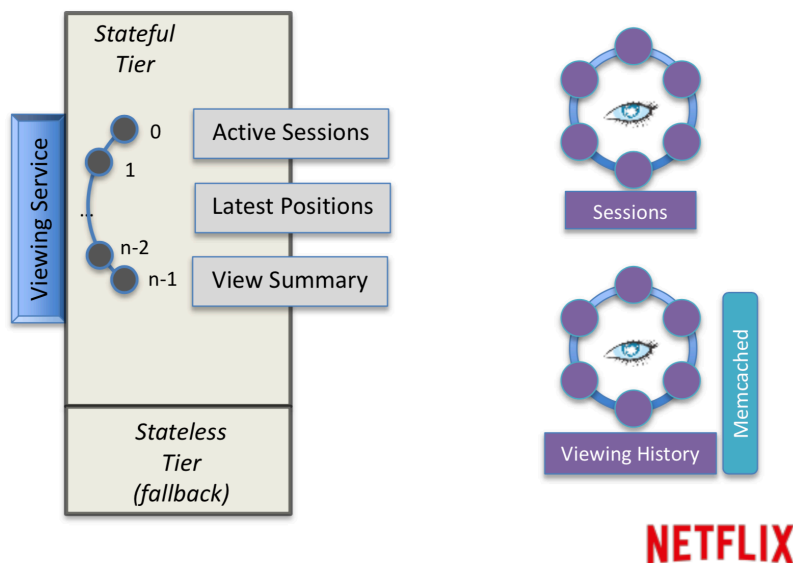
21. The Accused Infringing Devices store data indicative of the received at least portion of the identifier data using the first computer system. The user’s viewing history, updated every time a heartbeat is sent, is stored by the Accused Infringing Devices using one of Cassandra, the Viewing History Tier or memcached, as shown below.

Real Time Data – gen 3



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 42.

Real Time Data – gen 3



NETFLIX

Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 45.

gen 3 – Cluster Scale

Cluster	Scale
Cassandra Viewing History	~100 hi1.4xl nodes ~48 TB total space used
Viewing Service Stateful Tier	~1700 r3.2xl nodes 50GB heap memory per node
Memcached	~450 r3.2xl/xl nodes ~8TB memory used

Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 59.

22. Each provided webpage causes corresponding digital media presentation data to be streamed from a second computer system (e.g., the content delivery network, e.g., NOpen Connect), distinct from the user’s computer independent of the first computer system (e.g., the Netflix website).

1 Netflix Open Connect delivers 100% of our video traffic, currently *over 125 million hours*
2 *of viewing per day*. This amounts to tens of terabits per second of simultaneous peak
3 traffic, making Netflix Open Connect one of the highest-volume networks in the world.

4 Globally, close to 90% of our traffic is delivered via direct connections between Open
5 Connect and the residential Internet Service Providers (ISPs) our members use to access
6 the internet. Most of these connections are localized to the regional point of
7 interconnection that's geographically closest to the member who's watching. Because
8 connections to the Netflix Open Connect network are always free and our traffic delivery
9 is highly localized, thousands of ISPs around the world enthusiastically participate.

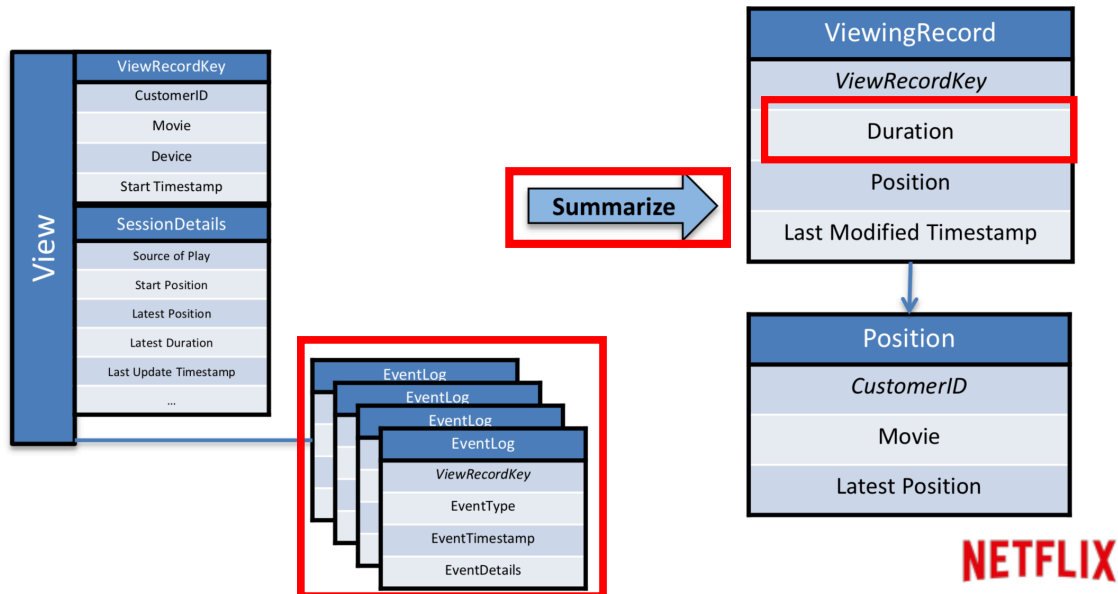
10 We also give qualifying ISPs the same Open Connect Appliances (OCAs) that we use in
11 our internet interconnection locations. After these appliances are installed in an ISP's
12 data center, almost all Netflix content is served from the local OCAs rather than
13 "upstream" from the internet. Many ISPs take advantage of this option, in addition to
14 local network interconnection, because it reduces the amount of capacity they need to
15 build to the rest of the internet since Netflix is no longer a significant factor in that
16 capacity. This has the dual benefit of reducing the ISP's cost of operation and ensuring
17 the best possible Netflix experience for their subscribers.

18 We now have Open Connect Appliances in close to 1,000 separate locations around the
19 world. In big cities like New York, Paris, London, Hong Kong, and Tokyo, as well as more
20 remote locations — as far north as Greenland and Tromsø, Norway and as far south as
21 Puerto Montt, Chile, and Hobart, Tasmania. ISPs have even placed OCAs in Macapá and
22 Manaus in the Amazon rainforest — on every continent, except Antarctica and on many
23 islands such as Jamaica, Malta, Guam, and Okinawa. This means that most of our
24 members are getting their Netflix audio and video bits from a server that's either inside
25 of, or directly connected to, their ISP's network within their local region.

26 **Source:** <https://media.netflix.com/en/company-blog/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience>.

27 23. The stored data is indicative of the amount of time the digital media
28 presentation is streamed from the second computer system to the user's computer.
The stored data indicates the duration and position of the user's current position,
which indicates the amount of time the presentation has been streamed to the user's
computer by the CDN.

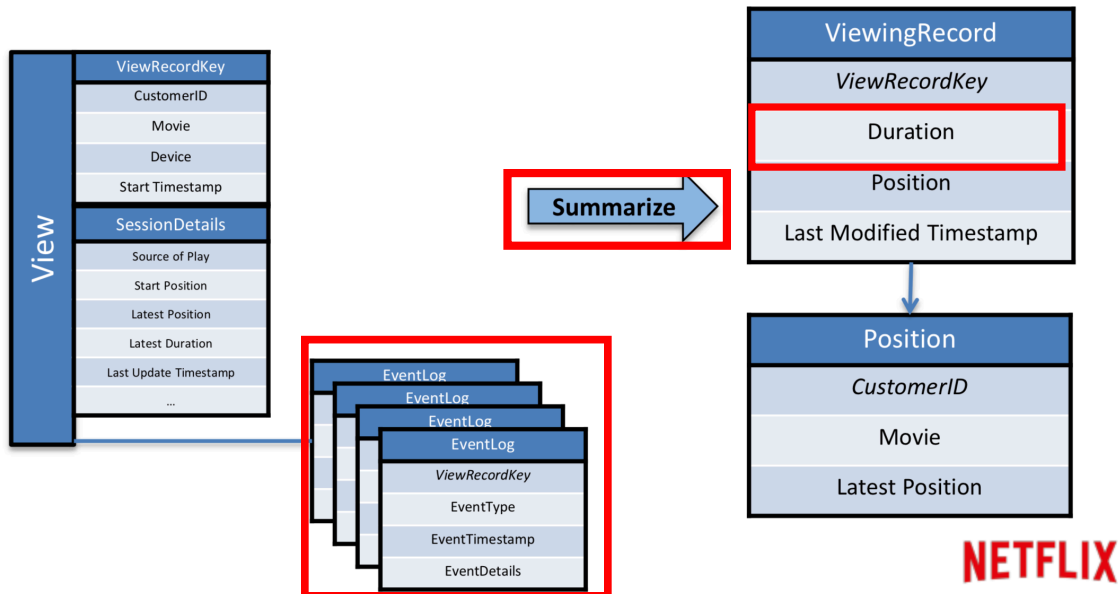
Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 29.

24. Each stored data is together indicative of a cumulative time the corresponding web page was displayed by the user's computer. The user visits netflix.com, which displays tiles and information screens for movies and TV shows. When the user elects to stream a movie or TV show, the player is loaded on the same page. The amount of time the user spends watching a movie or TV show is tracked by Netflix and also reflects the amount of time the preceding webpage was displayed by the user's computer.

Conceptual Data Model



Source: <https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014> at 29.

25. Netflix has infringed, and continues to infringe, at least claim 1 of the '609 patent in the United States, by making, using, offering for sale, selling, licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C. § 271(a).

26. Upon information and belief, Netflix may have infringed and continues to infringe the '609 patent through other software and devices utilizing the same or reasonably similar functionality, including other versions of the Accused Infringing Devices.

27. Netflix's acts of direct infringement have caused and continue to cause damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

COUNT II – INFRINGEMENT OF U.S. PATENT NO. 6,584,229

28. The allegations of paragraphs 1-9 of this Complaint are incorporated by reference as though fully set forth herein.

29. The '229 patent, titled "Macroblock-Based Object-Oriented Coding

1 Method Of Image Sequence Having A Stationary Background,” issued on June 24,
2 2003. A copy of the ’229 patent is attached as Exhibit B.

3 30. Pursuant to 35 U.S.C. § 282, the ’229 patent is presumed valid.

4 31. Upon information and belief, Netflix makes, uses, offers for sale, sells
5 in the United States, licenses in the United States and/or imports into the United
6 States products and services that use VP9 video encoding for coding image signals
7 (collectively the “Accused Infringing Devices”).

8 32. Upon information and belief, the Accused Infringing Devices infringe
9 at least claim 1 in the exemplary manner described below.

10 33. The Accused Infringing Devices practice a method for use in a macro-
11 block based object oriented coding of an image signal using the VP9 codec wherein
12 the image signal has a stationary background region and an object region and
13 contains a current frame and a previous frame, each frame including a plurality of
14 macroblocks.

15 16 **What’s new about our mobile encodes**

17 We are introducing two new types of mobile encodes—AVCHi-Mobile and
18 VP9-Mobile. The enhancements in the new bitstreams fall into three
19 categories: (1) new video compression formats, (2) more optimal encoder
20 settings, and (3) per-chunk bitrate optimization. All the changes combined
21 result in better video quality for the same bitrate compared to our current
streams (AVCMain).

22 **Source:** [https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-
23 downloads-625d7b082909](https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909)

1 VP9 has a number of tools which bring improvements in compression
2 efficiency over H.264/AVC, including:

- 3 • Motion-predicted blocks of sizes up to 64×64
- 4 • 1/8th pel motion vectors
- 5 • Three switchable 8-tap subpixel interpolation filters
- 6 • Better coding of motion vectors
- 7 • Larger discrete cosine transforms (DCT, 16×16, and 32×32)
- 8 • Asymmetric discrete sine transform (ADST)
- 9 • Loop filtering adapted to new block sizes
- 10 • Segmentation maps

11 **Source:** [https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-](https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909)
12 [downloads-625d7b082909](https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909)

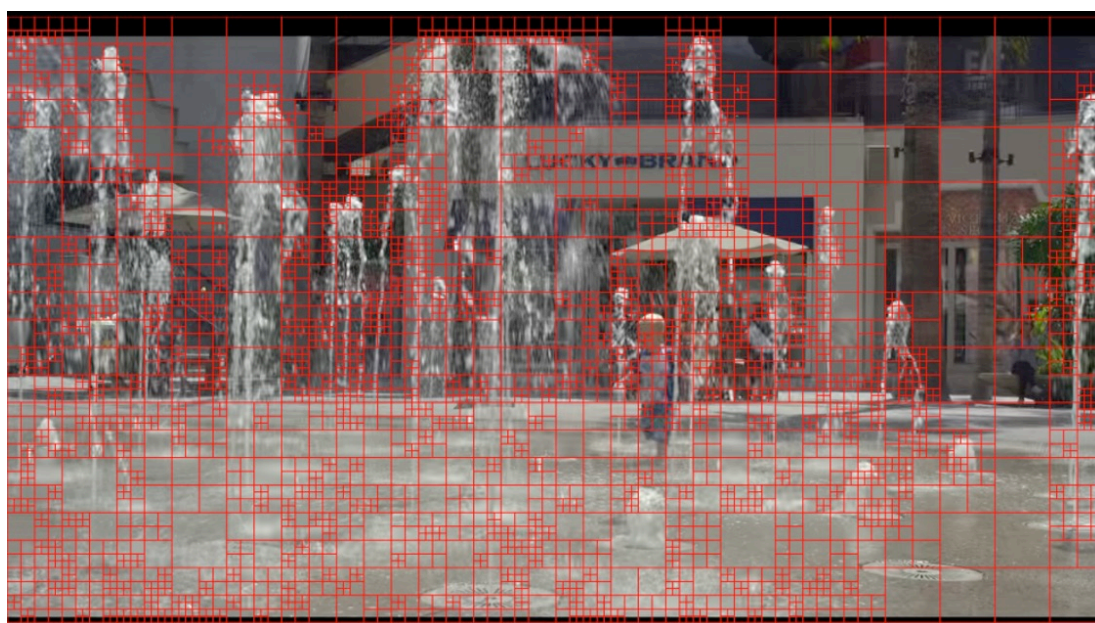
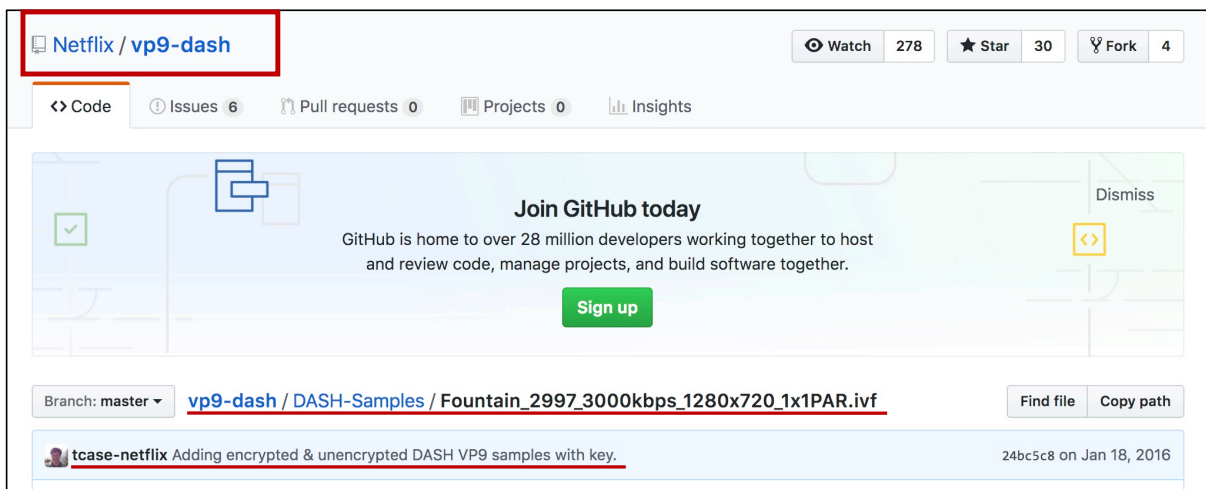
13 **2.4. Transforms**

14
15 VP9 supports the Discrete Cosine Transform (DCTs) at sizes 4x4, 8x8, 16x16 and
16 32x32 and removes the second-order transform that was employed in VP8. Only
17 transform sizes equal to, or smaller than, the prediction block size may be specified.
18 Modes B_PRED and 4x4 SPLITMV are thus restricted to using only the 4x4 transform;
19 modes I8X8_PRED and non-4x4 SPLITMV can use either the 4x4 or 8x8 transform; full-
20 size (16x16) macroblock predictors can be coupled with either the 4x4, 8x8 or 16x16
21 transforms, and superblocks can use any transform size up to 32x32. Further
22 restrictions on the available sub-set of transforms can be signaled at the frame-level,
23 by specifying a maximum allowable transform size, or at the macroblock level by
24 explicitly signaling which of the available transform sizes is used.

25 **Source:** <https://tools.ietf.org/id/draft-grange-vp9-bitstream-00.html#rfc.section.2.1>

26
27 34. Netflix supplies VP9 sample test files to developers. Those samples
28 demonstrate the block based coding of video images.

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

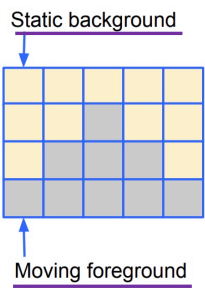


Source: https://github.com/NetfliX/vp9-dash/blob/master/DASH-Samples/Fountain_2997_3000kbps_1280x720_1x1PAR.ivf

35. VP9 coding uses segmentation to group together blocks with common characteristics, including static background and moving foreground (object) regions.

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

- Segmentation feature significantly enhanced in VP9
 - Groups together blocks that share common characteristics into segments.
 - Indicate segmentation id at block level
 - Differentially encode segmentation map temporally

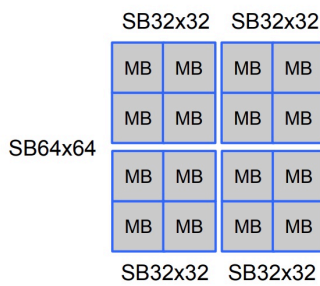


Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

36. The Accused Infringing Devices use inter mode prediction at a block level to predict a current block relative to a reference frame. The frame is coded using a variety of block sizes and a plurality of blocks to encode the image.

- Information conveyed at block end-points:
 - Prediction mode (can be INTER or INTRA)
 - If INTER mode:
 - Prediction reference frame(s)
 - Motion vector(s) if needed
 - Sub-pel filter choice
 - Skip flag indicating if there is any non-zero coefficient for prediction residual

- VP9 introduces Superblocks (SB mxn):
 - SB64x64: 64x64 block
 - SB64x32: 64x32 block
 - ...
 - MB16x16: 16x16 block
 - ...
 - SB8x8: 8x8 block



Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

37. The Accused Infringing Devices use VP9 segmentation to divide the inputted video into a stationary background and the object region on a block by block basis to create segmentation maps. For INTER blocks, VP9 uses the

1 calculation of motion vectors temporally between frames to find a difference
 2 (residual) between blocks in a given frame (next frame) and a reference frame.

3

4 ● Segmentation feature significantly enhanced in VP9

5 ○ Groups together blocks that share common characteristics into segments.

6 ○ Indicate segmentation id at block level

7 ■ Differentially encode segmentation map temporally

8

9 Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

10 How Does it Work?

11 While 4K video increases picture quality by making individual pixels smaller, VP9 codec and HEVC
 12 make them bigger to reduce the bitrate and file size. While this may seem conflicting, the encoding
 13 engine takes the larger pixels and turns them into a higher resolution output. Source video,
 14 consisting of video frames, is encoded or compressed to create a compressed video bitstream. Each
 15 individual frame is first broken up into blocks of pixels. The blocks are then analyzed for spatial
 16 redundancies and temporal linkages between frames are analyzed to take advantage of areas that
 17 do not change. These are encoded via motion vectors that predict qualities of the given block on
the next frame. The residual information is encoded using an efficient binary compression.

18 Source: <https://www.encoding.com/vp9/>

- 19
- 20 ● Information conveyed at block end-points:
- 21 ○ Prediction mode (can be INTER or INTRA)
- 22 ○ If INTER mode:
- 23 - Prediction reference frame(s)
- 24 - Motion vector(s) if needed
- 25 - Sub-pel filter choice
- 26 ○ Skip flag indicating if there is any non-zero
 27 coefficient for prediction residual

28 Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

38. The Accused Infringing Devices code macroblocks into segments

1 (regions) on a block by block basis to create segmentation maps that code the shape
 2 information.

- Segmentation feature significantly enhanced in VP9
 - Groups together blocks that share common characteristics into segments.
 - Indicate segmentation id at block level
 - Differentially encode segmentation map temporally

9 Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

	Segment ID by Blocksize				Hide
	64x64	32x32	16x16	8x8	4x4
0	13	37	49	16	0
1	6	7	52	52	0
2	12	42	219	384	0
3	26	124	461	720	0
4	5	35	325	916	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0

17 Source: Frame from the sample file showing VP9 image blocks using a VP9 analyzer showing
 18 the use of blocks and segments.

19 39. The Accused Infringing Devices use the well-known coding technique
 20 of motion vectors, prediction residuals and transform coding to code the pixel
 21 information of blocks in an object region by using, for example, INTER block
 22 coding.

28

- Information conveyed at block end-points:
 - Prediction mode (can be INTER or INTRA)
 - If INTER mode:
 - Prediction reference frame(s)
 - Motion vector(s) if needed
 - Sub-pel filter choice
 - Skip flag indicating if there is any non-zero coefficient for prediction residual

- VP9 uses different transforms for different modes:
 - 2D DCT for INTER modes and
 - Hybrid DCT/ADST transforms for INTRA modes
 - A lossless 4x4 transform for lossless encoding

Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

40. The Accused Infringing Devices use segments that are background (static) regions of inter blocks which contain blocks that are encoded from a fixed reference (previous) frame. The coded pixel information for a block contains the residual between the previous frame and the current frame, which can be zero (skip), using the coded pixel information of a previous frame block corresponding to a current frame block (only a motion vector pointing to the reference frame).

- the segment ID allows selecting a per-block quantizer and/or loop-filter strength level that is different from the frame-wide default, which allows for adaptive quantization/loop-filtering. The segment ID also allows encoding a fixed reference and/or marking a block as skipped, which is mainly useful for static content/background;

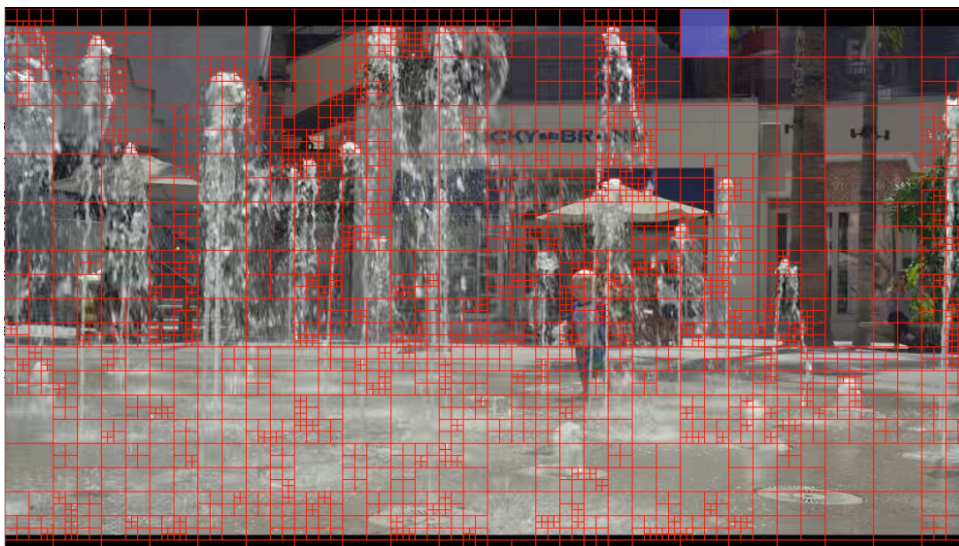
Source: <https://files.meetup.com/9842252/Overview-VP9.pdf>

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

	Txsize by Blocksize (inter)					Hide
	64x64	32x32	16x16	8x8	4x4	
Skip	40	112	610	1441	0	
32x32	3	77				
16x16	0	0	341			
8x8	0	0	0	192		
4x4	0	0	0	0	0	

Source: Frame from the sample file showing VP9 image blocks marked as skipped (Skip) using a VP9 analyzer.

41. Below shows a frame from the sample file showing a block marked as skipped in Segment 0, referencing a previous frame (Reference 0) and a motion vector using a VP9 analyzer.



Selected Block	Hide
X offset (col):	896
Y offset (row):	0
Block size:	64x64
Segment ID:	0
<u>Skip flag:</u>	<u>yes</u>
Intra flag:	no
Transform size:	32x32
Y intra mode:	
UV intra mode:	
<u>Reference(s):</u>	<u>0</u>
Filter:	smooth
MV prediction mode:	Nearestmv
Motion vector:	0,0

Source: Frame from the sample file showing a block marked as skipped in Segment 0, referencing a previous frame (Reference 0) and a motion vector using a VP9 analyzer.

42. The Accused Infringing Devices provide the ability to store the result of the video encoding to a file by creating downloadable coded data.

More Efficient Mobile Encodes for Netflix Downloads

Our first use case for these streams is the recently launched downloads feature on Android and iOS.

Source: <https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909>

43. The Accused Infringing Devices use INTER blocks marked as skipped (skip flag is true) to code a block without sending any coded pixel information for a macroblock in the current frame.

44. The decision to skip the coding of pixel information is based on a difference in pixel value for a macroblock between the current macroblock, and its corresponding macroblock in a previous frame. This difference is based on a test statistic between the macroblock in the current frame and a reference frame.

1 45. The Accused Infringing Devices perform this step by using test
2 statistics such as variance, or the magnitude of the residual frequency components
3 in macroblock to be coded or skipped, with comparisons to predetermined
4 thresholds of zero for the variance, **ac_thr** for the AC coefficients and **dc_thr** for
5 the DC coefficient.

```
6
7 616 // Check if all ac coefficients can be quantized to zero.
8 617 if (var_tx < ac_thr || var == 0) {
9 618     x->skip_txfm[0] = SKIP_TXFM_AC_ONLY;
10 619 // Check if dc coefficient can be quantized to zero.
11 620 if (sse_tx - var_tx < dc_thr || sse == var)
12 621     x->skip_txfm[0] = SKIP_TXFM_AC_DC;
13 622 } else {
14 623     if (sse_tx - var_tx < dc_thr || sse == var) skip_dc = 1;
15 624 }
```

16 **Source:**

17 https://chromium.googlesource.com/webm/libvpx/+/master/vp9/encoder/vp9_pickmode.c

18 46. Netflix has infringed, and continues to infringe, at least claim 1 of the
19 '229 patent in the United States, by making, using, offering for sale, selling,
20 licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C.
21 § 271(a).

22 47. Upon information and belief, Netflix may have infringed and
23 continues to infringe the '229 patent through other software and devices utilizing
24 the same or reasonably similar functionality, including other versions of the
25 Accused Infringing Devices.

26 48. Netflix's acts of direct infringement have caused and continue to cause
27 damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of
28 Netflix's wrongful acts in an amount subject to proof at trial.

COUNT III – INFRINGEMENT OF U.S. PATENT NO. 6,519,005

49. The allegations of paragraphs 1-9 of this Complaint are incorporated
by reference as though fully set forth herein.

1 50. The '005 patent, titled "Method of Concurrent Multiple-Mode Motion
2 Estimation For Digital Video," issued on February 11, 2003. A copy of the '005
3 patent is attached as Exhibit C.

4 51. Pursuant to 35 U.S.C. § 282, the '005 patent is presumed valid.

5 52. On information and belief, Netflix makes, uses, offers for sale, sells in
6 the United States, licenses in the United States and/or imports into the United States
7 products and services such as its H.264 encoders that practice a method for motion
8 coding an uncompressed digital video data stream (collectively the "Accused
9 Infringing Devices").

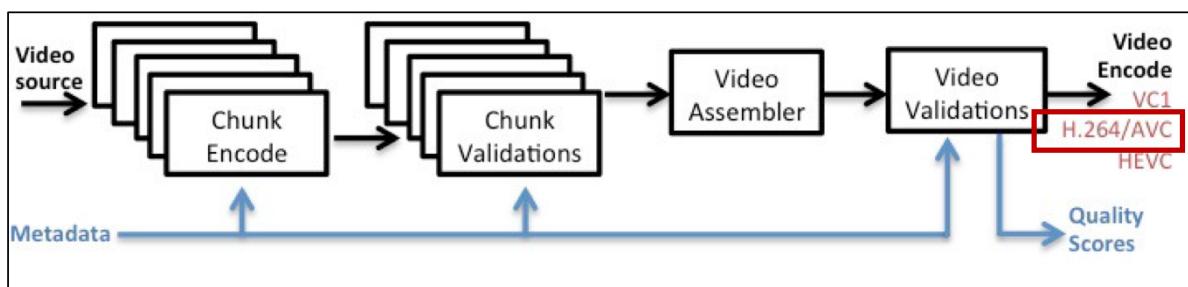
10 53. Upon information and belief, the Accused Infringing Devices infringe
11 at least claim 1 in the exemplary manner described below.

12 54. The Accused Infringing Devices provide a method for motion coding
13 an uncompressed (pixel level) digital video data stream. The Accused Infringing
14 Devices receive input video streams which are then encoded and/or transcoded
15 using at least the H.264 standard. The H.264 standard is a widely used video
16 compression format with decoder support on web browsers, TVs and other
17 consumer devices. Moreover, H.264 uses motion compressor and estimator for
18 motion coding video streams.

20 **Compression Performance**

21 In December 2016, we introduced AVCHi-Mobile and VP9-Mobile encodes for
22 downloads. For these mobile encodes, several changes led to improved
23 compression performance over per-title encodes, including longer GOPs,
24 flexible encoder settings and per-chunk optimization. These streams serve as
25 our high quality baseline for H.264/AVC and VP9 encoding with traditional
rate control settings.

26 **Source:** <https://medium.com/netflix-techblog/optimized-shot-based-encodes-now-streaming-4b9464204830>



6 parallel on different instances. For each chunk, bitstream-level and pixel-level
7 analysis is applied to detect errors and generate metadata such as temporal
8 and spatial fingerprints. After all the chunks are inspected, the results are

9 **Source:** <https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746>

10 11 12 **0.6 Overview of the design characteristics**

This subclause does not form an integral part of this Recommendation | International Standard.

13 The coded representation specified in the syntax is designed to enable a high compression capability for a desired image
14 quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 Intra,
15 CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles, and the I_PCM mode of operation in all profiles, the algorithm
16 is typically not lossless, as the exact source sample values are typically not preserved through the encoding and
17 decoding processes. A number of techniques may be used to achieve highly efficient compression. Encoding algorithms
18 (not specified in this Recommendation | International Standard) may select between inter and intra coding for block-
19 shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal
20 statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial
21 statistical dependencies in the source signal for a single picture. Motion vectors and intra prediction modes may be
22 specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform
23 to remove spatial correlation inside the transform block before it is quantised, producing an irreversible process that
24 typically discards less important visual information while forming a close approximation to the source samples. Finally,
25 the motion vectors or intra prediction modes are combined with the quantised transform coefficient information and
26 encoded using either variable length coding or arithmetic coding.

27 **0.6.1 Predictive coding**

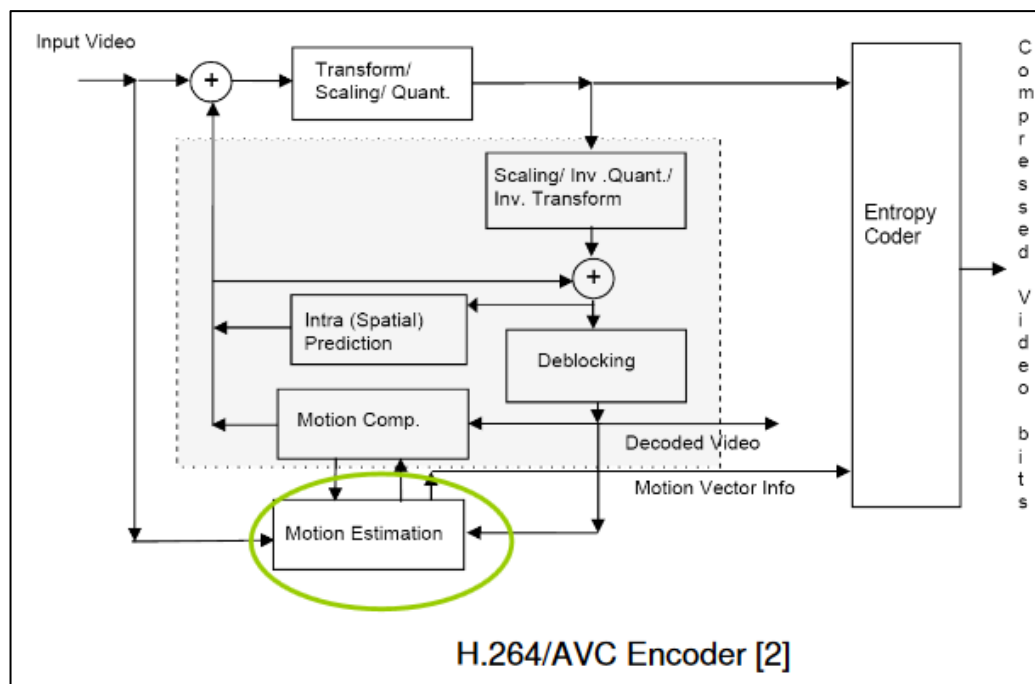
This subclause does not form an integral part of this Recommendation | International Standard.

28 Because of the conflicting requirements of random access and highly efficient compression, two main coding types are
specified. Intra coding is done without reference to other pictures. Intra coding may provide access points to the coded
sequence where decoding can begin and continue correctly, but typically also shows only moderate compression
efficiency. Inter coding (predictive or bi-predictive) is more efficient using inter prediction of each block of sample
values from some previously decoded picture selected by the encoder. In contrast to some other video coding standards,
pictures coded using bi-predictive inter prediction may also be used as references for inter coding of other pictures.

The application of the three coding types to pictures in a sequence is flexible, and the order of the decoding process is
generally not the same as the order of the source picture capture process in the encoder or the output order from the
decoder for display. The choice is left to the encoder and will depend on the requirements of the application. The

decoding order is specified such that the decoding of pictures that use inter-picture prediction follows later in decoding
order than other pictures that are referenced in the decoding process.

H.264 Encoder Block Diagram



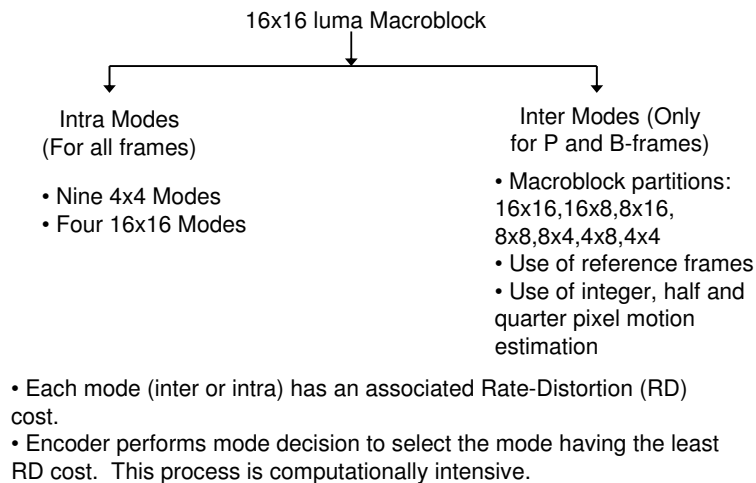
Source: <https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf>

55. The Accused Infringing Devices provide a method for comparing pixels of a first pixel array (e.g., a macroblock) in a picture currently being coded with pixels of a plurality of second pixel arrays in at least one reference picture and concurrently performing motion estimation for each of a plurality of different prediction modes in order to determine which of the prediction modes is an optimum prediction mode.

56. H.264 uses different motion estimation modes in inter-frame prediction. These modes are commonly referred to as inter-frame prediction modes, or inter modes. Each inter mode involves partitioning the current Macroblock into a different combination of sub blocks, and selecting the optimum motion vector for the current Macroblock based on the partition. The inter-frame prediction modes, or inter modes, can be further categorized by the number and position of the reference frames, as well as the choice of integer pixel, half pixel and quarter pixel values in motion estimation. The Netflix H.264 encoders concurrently perform motion estimation of a Macroblock for all inter-modes and select the most optimum prediction mode with least rate distortion cost.

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

Mode Decision

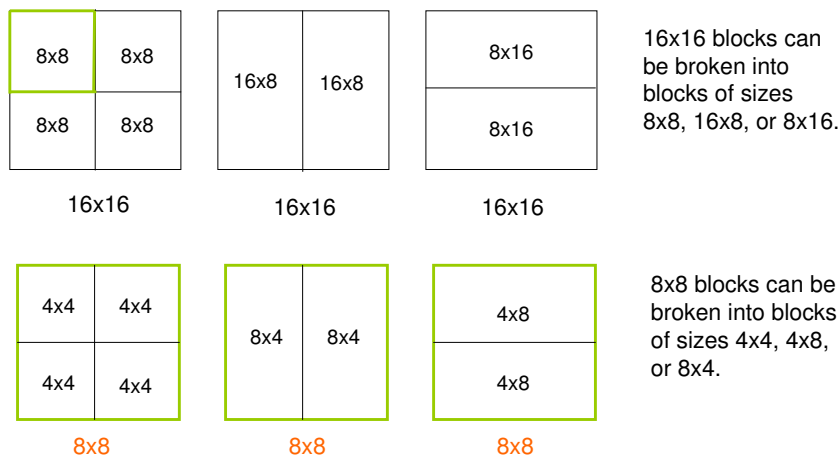


Source: <https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf>, p. 30

57. H.264 provides a hierarchical way to partition a Macroblock, with the available partitions shown in the following two figures. An exemplary inter-frame prediction mode, or inter mode, can be for a Macroblock to be partitioned to encompass a 16x8 sub block on the left, and two 8x8 sub blocks on the right.

Macroblock partitions for inter-frame prediction modes

Macroblock Partitions



Source: <https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf>, p. 4

H.264 provides macroblock partitions for inter-frame prediction modes

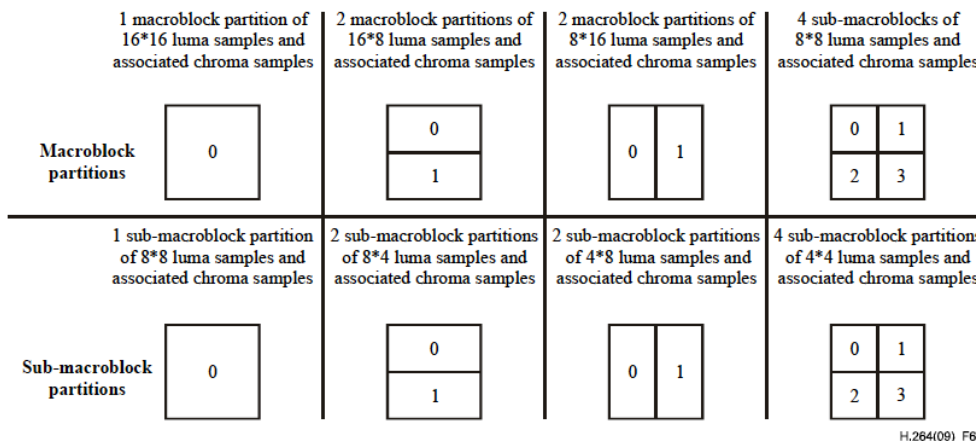


Figure 6-9 – Macroblock partitions, sub-macroblock partitions, macroblock partition scans, and sub-macroblock partition scans

Source: H.264 Standard (03-2010) at p. 26

58. The optimum prediction mode as chosen for the current Macroblock is embedded in the compressed bit stream of H.264, as shown in the following two syntaxes.

Macroblock prediction syntax in H.264

7.3.5.1 Macroblock prediction syntax

	C	Descriptor
mb_pred(mb_type) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx]	2	u(3) ae(v)
}		
intra_chroma_pred_mode	2	ue(v) ae(v)
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
ref_idx_l0[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
ref_idx_l1[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx][0][compIdx]	2	se(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx][0][compIdx]	2	se(v) ae(v)
}		
}		

Source: H.264 Standard (03-2010) at p. 57

Sub-macroblock prediction syntax in H.264

7.3.5.2 Sub-macroblock prediction syntax

	C	Descriptor
sub_mb_pred(mb_type) {		
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
sub_mb_type[mbPartIdx]	2	ue(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 mb_field_decoding_flag) && mb_type != P_8x8ref0 && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
ref_idx_l0[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 mb_field_decoding_flag) && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
ref_idx_l1[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
}		

Source: H.264 Standard (03-2010) at p. 58

59. The Accused Infringing Devices provide a method for determining which of the second pixel arrays (e.g., macroblock) constitutes a best match with respect to the first pixel array (e.g., macroblock) for the optimum prediction mode.

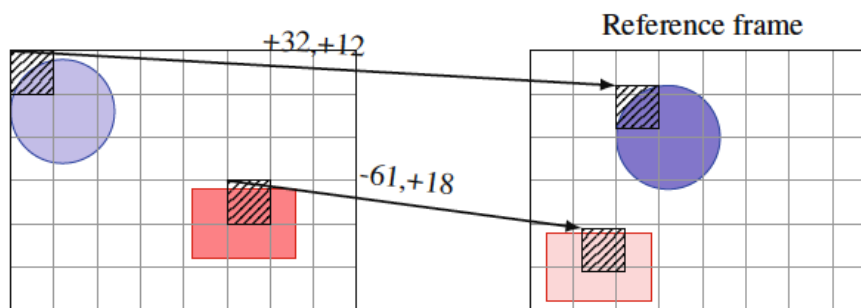


Fig. 2.4: Motion estimation. For each MB the best matching block in the reference frame is found. The encoder codes the differences (errors) between the MBs and their best matching blocks. Arrows indicate motion vectors and are labeled by the vector coordinates. In this example the shapes are identical but their colors are slightly larger/darker.

Source: B. Juurlink et al., Scalable Parallel Programming Applied to H.264, Chapter 2: Understanding the Application: An Overview of the H.264 Standard, p. 12

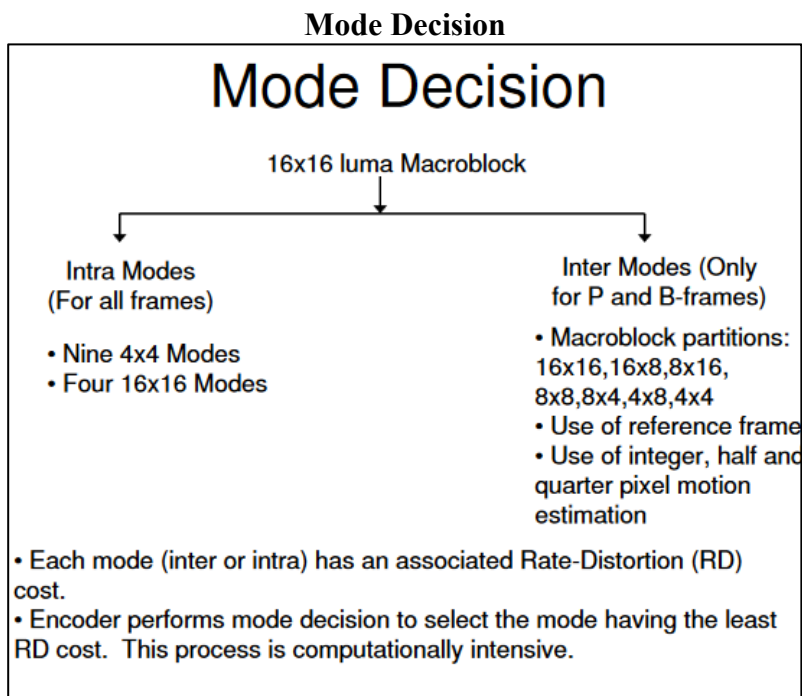
60. For example, the encoder performs mode decision to select the most optimum prediction mode with least rate distortion cost.

Macroblock layer semantics

The following semantics are assigned to the macroblock types in Table 7-13:

- P_L0_16x16: the samples of the macroblock are predicted with one luma macroblock partition of size 16x16 luma samples and associated chroma samples.
- P_L0_L0_MxN, with MxN being replaced by 16x8 or 8x16: the samples of the macroblock are predicted using two luma partitions of size MxN equal to 16x8, or two luma partitions of size MxN equal to 8x16, and associated chroma samples, respectively.
- P_8x8: for each sub-macroblock an additional syntax element (`sub_mb_type[mbPartIdx]` with `mbPartIdx` being the macroblock partition index for the corresponding sub-macroblock) is present in the bitstream that specifies the type of the corresponding sub-macroblock (see subclause 7.4.5.2).
- P_8x8ref0: has the same semantics as P_8x8 but no syntax element for the reference index (`ref_idx_10[mbPartIdx]` with `mbPartIdx = 0..3`) is present in the bitstream and `ref_idx_10[mbPartIdx]` shall be inferred to be equal to 0 for all sub-macroblocks of the macroblock (with indices `mbPartIdx = 0..3`).
- P_Skip: no further data is present for the macroblock in the bitstream.

Source: H.264 Standard (03-2010), p. 100

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

Source: <https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf>, p. 30

61. The Accused Infringing Devices provide a method for generating a motion vector for the first pixel array in response to the determining step. The encoder calculates the appropriate motion vectors and other data elements represented in the video data stream.

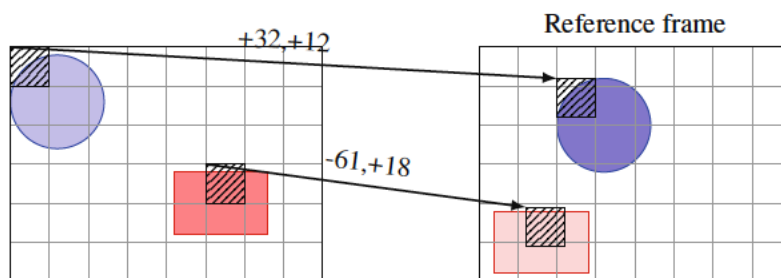


Fig. 2.4: Motion estimation. For each MB the best matching block in the reference frame is found. The encoder codes the differences (errors) between the MBs and their best matching blocks. Arrows indicate motion vectors and are labeled by the vector coordinates. In this example the shapes are identical but their colors are slightly larger/darker.

Source: B. Juurlink et al., Scalable Parallel Programming Applied to H.264, Chapter 2: Understanding the Application: An Overview of the H.264 Standard, p. 12

Motion Vector Derivation is described below

1. The derivation process for motion vector components and reference indices as specified in subclause 8.4.1 is invoked.

Inputs to this process are:

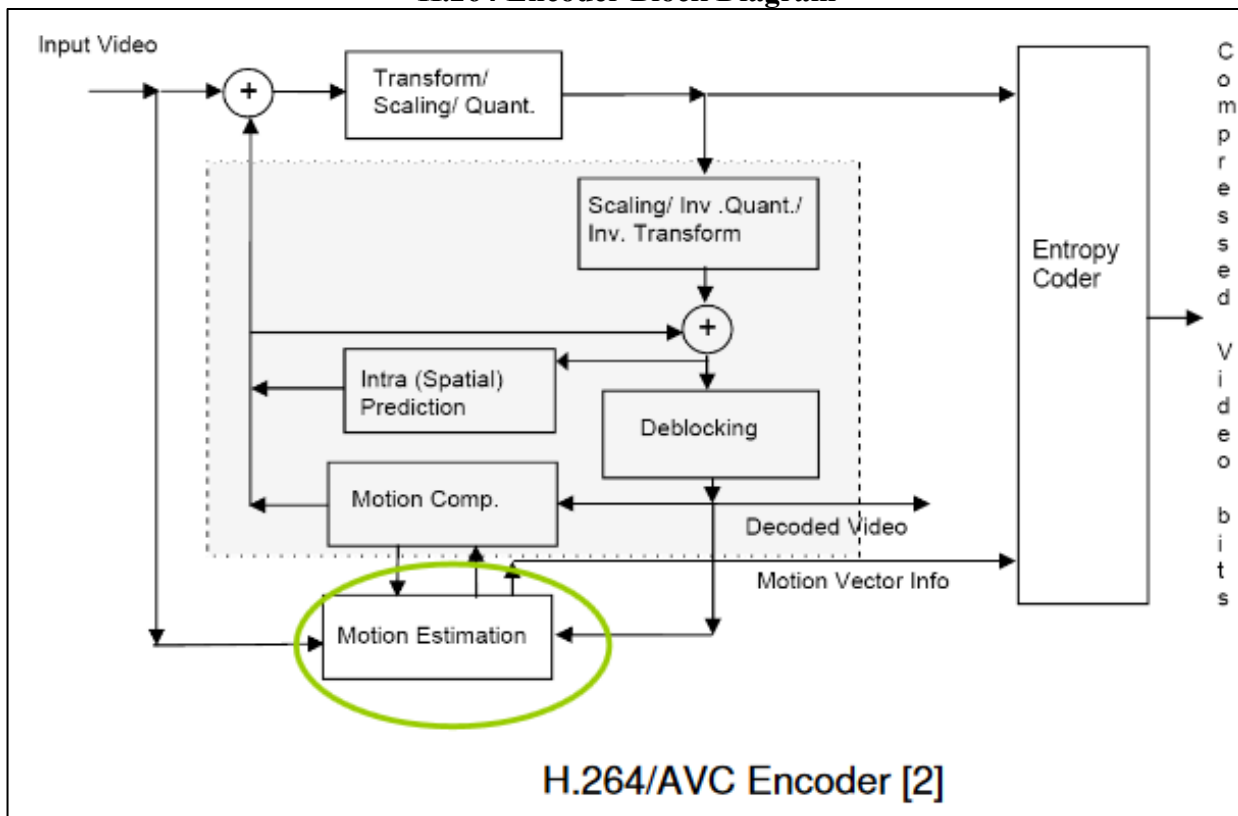
- a macroblock partition mbPartIdx,
- a sub-macroblock partition subMbPartIdx.

Outputs of this process are:

- luma motion vectors mvL0 and mvL1 and when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1
- reference indices refIdxL0 and refIdxL1
- prediction list utilization flags predFlagL0 and predFlagL1
- the sub-macroblock partition motion vector count subMvCnt.

Source: H.264 Standard (03-2010), p. 151

H.264 Encoder Block Diagram



H.264/AVC Encoder [2]

Source: <https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf>, p. 2

62. Netflix has infringed, and continues to infringe, at least claim 1 of the '005 patent in the United States, by making, using, offering for sale, selling, licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C.

1 § 271(a).

2 63. Upon information and belief, Netflix may have infringed and
3 continues to infringe the '005 patent through other software and devices utilizing
4 the same or reasonably similar functionality, including other versions of the
5 Accused Infringing Devices.

6 64. Netflix's acts of direct infringement have caused and continue to cause
7 damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of
8 Netflix's wrongful acts in an amount subject to proof at trial.

9 **PRAYER FOR RELIEF**

10 WHEREFORE, plaintiffs Uniloc 2017 LLC and Uniloc Licensing USA LLC
11 respectfully pray that the Court enter judgment in their favor and against Netflix as
12 follows:

13 a. A judgment that Netflix has infringed one or more claims of the
14 '609 Patent literally and/or under the doctrine of equivalents;

15 b. A judgment that Netflix has infringed one or more claims of the
16 '229 Patent literally and/or under the doctrine of equivalents;

17 c. A judgment that Netflix has infringed one or more claims of the
18 '005 Patent literally and/or under the doctrine of equivalents;

19 d. That for each Asserted Patent this Court judges infringed by
20 Netflix this Court award Uniloc its damages pursuant to 35 U.S.C. § 284 and any
21 royalties determined to be appropriate;

22 e. That this be determined to be an exceptional case under 35
23 U.S.C. § 285 and that Uniloc be awarded enhanced damages up to treble damages
24 for willful infringement as provided by 35 U.S.C. § 284;

25 f. That this Court award Uniloc prejudgment and post-judgment
26 interest on its damages;

27 g. That Uniloc be granted its reasonable attorneys' fees in this
28

1 action;

2 h. That this Court award Uniloc its costs; and

3 i. That this Court award Uniloc such other and further relief as the
4 Court deems proper.

5 **DEMAND FOR JURY TRIAL**

6 Uniloc hereby demands trial by jury on all issues so triable pursuant to Fed.
7 R. Civ. P. 38.

8
9 Dated: October 23, 2018

FEINBERG DAY ALBERTI LIM &
BELLOLI LLP

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

By: /s/ M. Elizabeth Day
M. Elizabeth Day

Attorneys for Plaintiffs
Uniloc 2017 LLC and Uniloc Licensing
USA LLC