# Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders

Dan Grois, *Senior Member, IEEE,*
Detlev Marpe, *Senior Member*, *IEEE*

Image Processing Department,
Fraunhofer Heinrich-Hertz-Institute (HHI),
Berlin, Germany
E-mail: grois@ieee.org

Amit Mulayoff, Benaya Itzhaky,
and Ofer Hadar, *Senior Member, IEEE*

Communication Systems Engineering Department
Ben-Gurion University of the Negev
Beer-Sheva, Israel

***Abstract* — This work presents a performance comparison of the two latest video coding standards H.264/MPEG-AVC and H.265/MPEG-HEVC (High-Efficiency Video Coding) as well as the recently published proprietary video coding scheme VP9. According to the experimental results, which were obtained for a whole test set of video sequences by using similar encoding configurations for all three examined representative encoders, H.265/MPEG-HEVC provides significant average bit-rate savings of 43.3% and 39.3% relative to VP9 and H.264/MPEG-AVC, respectively. As a particular aspect of the conducted experiments, it turned out that the VP9 encoder produces an average bit-rate overhead of 8.4% at the same objective quality, when compared to an open H.264/MPEG-AVC encoder implementation – the x264 encoder. On the other hand, the typical encoding times of the VP9 encoder are more than 100 times higher than those measured for the x264 encoder. When compared to the full-fledged H.265/MPEG-HEVC reference software encoder implementation, the VP9 encoding times are lower by a factor of 7.35, on average.**

***Index Terms* — *H.265, High Efficiency Video Coding (HEVC), VP9, H.264, AVC, x264, video coding, coding efficiency.***

## I. INTRODUCTION

Major milestones in the evolution of video coding standards are the well-known H.262/MPEG-2 Video [1] and H.264/MPEG-4 Advanced Video Coding (AVC) [2] standards, the development of which was coordinated by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Pictures Expert Group (MPEG). The first version of the H.264/MPEG-AVC standard (and its reference software JM [3]) was developed in the period between 1999 and 2003 to satisfy the growing need for higher coding efficiency, especially with regard to standard-definition TV and video transmission over low data rate channels.

As a result, the H.264/MPEG-AVC standard successfully achieved an increase of about 50% in coding efficiency compared to its predecessor H.262/MPEG-2 Video. H.264/MPEG-AVC was designed for both low- and high bit-rate video coding in order to accommodate the increasing diversification of transport layers and storage media. In turn, this gave rise to a wide variety of H.264/MPEG-AVC-based products and services [2], [4]. Throughout subsequent stages of development, additional efforts were made (mainly from 2003 to 2009) for further improving the coding efficiency as well as for integrating additional functionalities and features into the design of H.264/MPEG-AVC by means of the so-called Fidelity Range Extensions (FRExt) with its prominent High profile, the Scalable Video Coding (SVC) extension and finally, the Multiview Video Coding (MVC) extension.

This work was carried out during Dr. Dan Grois tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant n° 246016.

As already noted above, H.264/MPEG-AVC provided significant bit-rate savings compared to H.262/MPEG-2 Video. However, both video coding standards, at least their first editions, were not initially designed for High Definition (HD) and Ultra High-Definition (UHD) video content, the demand for which is expected to dramatically increase in the near future (Note that the term UHD often refers to both 3840x2160 (4K) or 7680x4320 (8K) resolutions in terms of luma samples).

As a consequence, ITU-T VCEG and ISO/IEC MPEG established a Joint Collaborative Team on Video Coding (JCT-VC) and issued a joint call for proposals (CfP) on video coding technology in 2010. In response to this CfP, a lot of proposals were submitted both from representatives of industry and academia, which in turn led to an intensive development of the so-called High-Efficiency Video Coding (HEVC) standard during the next two and the half years. The first edition of HEVC was officially finalized in January 2013, and after that, the final aligned specification was approved by ITU-T as Recommendation H.265 and by ISO/IEC as MPEG-H, Part 2 [5].

The H.265/MPEG-HEVC standard was designed to be applicable for almost all existing H.264/MPEG-AVC applications, while putting emphasis on high-resolution video coding. Since the development process of H.265/MPEG-HEVC was also driven by the most recent scientific and technological achievements in the field of video coding, dramatic bit-rate savings were achieved for substantially the same visual quality, when compared to its predecessor like H.264/MPEG-AVC [6]-[8].

In parallel with the open video coding standardization processes of ITU-T and ISO/IEC, a few companies individually developed their own video codecs, which often were based partly on their own secretly kept technologies and partly on variants of the state-of-the-art technologies used in their standardized counterparts, available at that time. One of these kind of proprietary video codecs is the VP8 codec [9]-[11], which was developed privately by On2 Technologies® Inc. that in turn, was later acquired by Google® Inc. Based on VP8, Google® Inc. started the development of its successor VP9 [12], [13] in 2011, which was recently announced to be finalized [14].

However, up to now little is known about the coding efficiency of VP9, especially in comparison to the two latest representatives of ITU-T and ISO/IEC video coding standards, i.e., H.264/MPEG-AVC and H.265/MPEG-HEVC. In order to provide that sort of information in a reproducible and reliable form, this paper presents experimental results of such a comparison along with a discussion of the selected software implementations, the choice of coding parameters, and the corresponding evaluation setup.

This paper is organized as follows. In the next section, the selected representative encoders are introduced. Section *III* contains a description of the test methodology and evaluation setup. Then, the detailed experimental results are presented in Section *IV*, and this paper is concluded in Section *V*.

## II. Selected Encoder Implementations

In this section, a brief overview of the selected representative encoders is presented.

### A. H.264/MPEG-AVC Encoder

For evaluating H.264/MPEG-AVC, an open H.264/MPEG-AVC encoder implementation - the x264 encoder was selected [15]-[18]. The first version of the x264 encoder was released in 2006, and since then, it has proven to be very fast, efficient, and reliable. Particularly, due to its flexible trade-off between coding efficiency and computational complexity, it was widely adopted in many network-based applications. Currently, the x264 video encoder is considered to be one of the most popular encoders for H.264/MPEG-AVC-based video coding [15].

The x264 encoder has a two-pass run option, which refers to a multi-pass rate control [17], [18]. At the first pass, a file with the detailed statistic data about every input frame is generated. In turn, at the second pass, this information is used to improve the encoder rate-distortion performance. According to [18], by employing the above-mentioned two-pass run, an average of about 7% decrease in bit rate is achieved for the same video quality (this bit-rate decrease was also approved in authors' initial experiments).

Therefore, the authors found the x264 encoder to be one of the best representatives of publicly available H.264/MPEG-AVC-based encoding implementations. Particularly, the authors used the latest version of the x264 encoder, i.e., the "r2334" version, which was released on May 22, 2013 [15]-[17].

### B. H.265/MPEG-HEVC Encoder

For H.265/MPEG-HEVC-based encoding [19], [20], the HM reference software encoder [21] was selected, since it is currently considered as the most popular available encoder implementation. Particularly, the authors selected the latest reference model 10 (HM 10.0) for conducting their performance evaluation.

### C. VP9 Encoder

As already noted above, the final VP9 bitstream format and its corresponding encoder were released by Google® Inc. per June 12, 2013 [14]. The VP9 encoder has a two-pass run option, similarly to x264, which results in the improved rate-distortion performance, and which was also used in our experiments, as further explained in the next section.

## III. Test Methodology and Evaluation Setup

For performing the detailed performance analysis and in order to be as fair as possible due to the significant difference in the capabilities of the individual encoders, the authors of this paper used very similar settings for all tested encoders, i.e., for the HM reference software, VP9, and x264 video encoders. Below, the test methodology and the evaluation setup are explained in detail. Particularly, in *Sub-Section III.A,* the HM reference software configuration is discussed, followed by the discussion of VP9 and x264 configurations, in *Sub-Section III.B*. Then*,* an overview of the performed Bjøntegaard-Delta Bit-Rate (BD-BR) measurements is presented in *Sub-Section III.C*.

### A. HM Reference Software Configuration

For the HM reference software encoder [21], a Random Access (RA) configuration was selected, since it provides better results in term of coding efficiency compared to the Low Delay configuration [22]. The Group of Picture (GOP) size was set to 8 pictures, and the Intra Period was set to 24, 32, 56, and 64 pictures for 24, 30, 50, and 60 fps video contents, respectively. Also, Hierarchical *B pictures* were used with a Quantization Parameter (QP) increase of 1 (i.e., the quantization step size increase of 12% [22]) between each temporal level. Also, the coding order was set to 0, 8, 4, 2, 1, 3, 6, 5, 7. It is noted that the above test conditions were selected similarly to the test conditions presented in [6] and [22]. For selecting additional encoding parameters, the authors used the "*CFG 16*" configuration, which was presented in [23], and which was proven to be optimal both from coding efficiency and computational complexity points of view. Table I below summarizes the above-mentioned HM reference software encoder [21] configuration.

TABLE I. Settings for the HM Reference Software Encoder

| CODING OPTIONS | CHOSEN PARAMETER |
|---|---|
| Encoder Version | HM 10.0 |
| Profile | Main |
| Reference Frames | 4 |
| R/D Optimization | Enabled |
| Motion Estimation | TZ search |
| Search Range | 64 |
| GOP | 8 |
| Hierarchical Encoding | Enabled |
| Temporal Levels | 4 |
| Intra Period | 1 sec |
| Deblocking Filter | Enabled |
| Coding Unit Size/Depth | 64/4 |
| Transform Unit Size (Min/Max) | 4/32 |
| TransformSkip | Enabled |
| TransformSkipFast | Enabled |
| Hadamard ME | Enabled |
| Asymmetric Motion Partitioning (AMP) | Enabled |
| Fast Encoding | Enabled |
| Fast Merge Decision | Enabled |
| Sample adaptive offset (SAO) | Enabled |
| Rate Control | Disabled |
| Internal Bit Depth | 8 |

### B. VP9 and x264 Configuration

The VP9 and x264 configuration settings are presented in Table II below. It should be noted that since there is currently no official VP9 specification as well as no VP9 encoder manual, the authors used both the VP9 two-pass best-quality settings recommended by leading VP9 senior developers [24], [25], as well as the most recommended VP8 two-pass best-quality settings [9], which are denoted in Table II as *Configuration 1*, *Configuration 2*, and *Configuration 3*. However, it should be noted that using all these three different settings led to substantially the same performance results.

TABLE II. SELECTED SETTINGS FOR THE VP9 AND X264 ENCODERS

| CODEC | VP9 | x264 |
|---|---|---|
| **Versions** | **Defined as Final [14] v1.2.0-3088-ga81bd12 of June 12, 2013** | **Most Recent [16]: r2334 of May 22, 2013** |
| **2-pass best-quality recommended settings of [24]** *Configuration 1* | --good --cpu-used=0 --threads=0 --profile=0--lag-in-frames=25 --min-q=$QP --max-q=$QP --cq-level=20 --end-usage=0--auto-alt-ref=1 --passes=2 --kf-max-dist=$IntraPeriod --kf-min-dist==$IntraPeriod --drop-frame=0--static-thresh=0 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --arnr-maxframes=7 --arnr-strength=5 --arnr-type=3 --sharpness=0--undershoot-pct=100 --codec=vp9 | --qp $QP --profile high --pass 2 --direct auto --tune psnr --ref 4 --preset placebo --b-pyramid strict --keyint=$IntraPeriod --min-keyint=$IntraPeriod --open-gop --weightp 2 --level 5.1 |
| **2-pass best-quality recommended settings of [25]** *Configuration 2* | --codec=vp9 --passes=2 --good --cpu-used=0 --auto-alt-ref=1 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --lag-in-frames=25 --kf-min-dist=$IntraPeriod --kf-max-dist=$IntraPeriod --static-thresh=0 --min-q=$QP --max-q=$QP --arnr-maxframes=7 --arnr-strength=5 --arnr-type=3 | *The above-mentioned "preset placebo" command is defined as[15]-[17]:* --bframes 16 --b-adapt 2 --direct auto --slow-firstpass --no-fast-pskip --me tesa --merange 24 --partitions all --rc-lookahead 60 --ref 16 --subme 11 --trellis 2 |
| **2-pass best-quality recommended settings of [9]** *Configuration 3* | -p 2 -t 4 --best --end-usage=vbr --auto-alt-ref=1 --minsection-pct=5 --maxsection-pct=800 --lag-in-frames=16 --kf-min-dist==$ IntraPeriod --kf-max-dist==$ IntraPeriod --token-parts=2 --static-thresh=0 --drop-frame=0 --min-q=$QP --max-q=$QP | |

The reader is referred to [9], [17] for obtaining more detailed information with regard to all VP9 and x264 commands, respectively, as presented in Table II.

The IntraPeriod interval as well as the QP values in the above VP9 and x264 configuration were set to be similar to those used for running the HM 10.0 encoder in order to be consistent (they are presented as $IntraPeriod and $QP, respectively). By such a way, it is ensured that *I-pictures* are inserted in regular time intervals (which are the "1 sec." intervals), and at exactly the same time instances [6].

Also, it should be noted that both VP9 and x264 encoder configurations were tuned for the best PSNR values. Further, it should be noted that the VP9 command line executed either with "--target-bitrate " command (i.e., specifying the target bit-rate) or without it, led to similar results in all case.

### C. *Bjøntegaard-Delta Bit-Rate Measurements*

As rate-distortion (R-D) performance assessment, the authors used a Bjøntegaard-Delta bit-rate (BD-BR) measurement method for calculating average bit-rate differences between R-D curves for the same objective quality (e.g., for the same $PSNR_{YUV}$ values) [26], where *negative BD-BR values* indicate actual bit-rate savings.

The authors used R-D curves of the combined luma (Y) and chroma (U,V) components, while the combined $PSNR_{YUV}$ value were calculated as a weighted sum of the PSNR values per each picture of each individual component [6], i.e., of $PSNR_Y$, $PSNR_U$, and $PSNR_V$:

$$PSNR_{YUV} = 6 \bullet PSNR_Y + PSNR_U + PSNR_V)/8 \quad (1)$$

As a result, using the combined $PSNR_{YUV}$ and bit-rate values as an input to the BD-BR measurement method enables to determine a single average difference in bit-rate that considers the tradeoffs between luma and chroma component fidelity [6].

## IV. EXPERIMENTAL RESULTS

For obtaining experimental results, most of the test sequences were selected according to the common HM test conditions [27], as presented in Table III. The authors mainly focus here on *Classes A, B, E* and *F*, which mostly relate to higher resolution video content. Further, for each of these video sequences, four quantization parameter (QP) values were selected: 22, 27, 32, and 37, which are the QP values used for the I-frame coding of the HM [27]. For simplicity, 150 frames of each sequence were tested. Also, the tests were carried out on computers with Intel Core i5 CPU, 2.4 GHz, 4GB RAM.

TABLE III. TEST VIDEO SEQUENCES

| Class | Sequence Name | Resolution | Frame rate |
|---|---|---|---|
| *A* | Traffic | 2560x1600 | 30fps |
| *A* | PeopleOnStreet | 2560x1600 | 30fps |
| *B* | Kimono | 1920x1080 | 24fps |
| *B* | ParkScene | 1920x1080 | 24fps |
| *B* | Cactus | 1920x1080 | 50fps |
| *B* | BQTerrace | 1920x1080 | 60fps |
| *B* | BasketballDrive | 1920x1080 | 50fps |
| *E* | FourPeople | 1280x720 | 60fps |
| *E* | Johnny | 1280x720 | 60fps |
| *E* | KristenAndSara | 1280x720 | 60fps |
| *F* | BaskeballDrillText | 832x480 | 50fps |
| *F* | ChinaSpeed | 1024x768 | 30fps |

Figure 1 below presents R-D curves of HEVC, x264, and VP9 encoders, along with corresponding HEVC bit-rate savings for two typical examples of tested sequences.

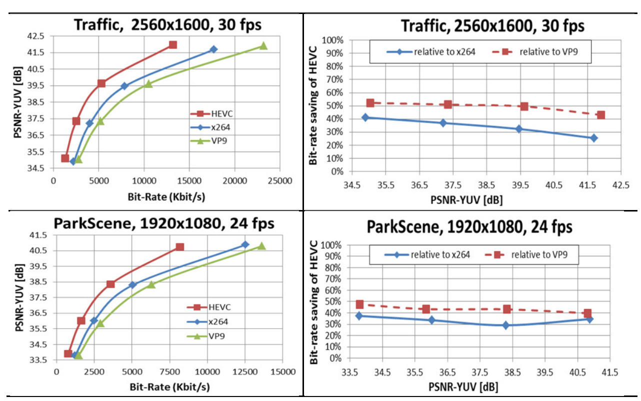


Figure 1. R-D curves and corresponding bit-rate saving plots for several typical examples of tested sequences.

As it is clearly seen from Figure 1, the HEVC encoder [21] provides significant gains in terms of coding efficiency compared to both VP9 and x264 encoders. In addition, Table IV presents detailed experimental results, including the calculated BD-BR savings [26].

TABLE IV. HEVC BIT-RATE SAVINGS (INCL. BD-BR SAVINGS) FOR EQUAL $PSNR_{YUV}$ (COMPARED TO VP9 AND X264 ENCODERS)

| | ***HEVC vs. VP9 (in %)*** | | | | | ***HEVC vs. x264 (in %)*** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sequences/QPs** | **22** | **27** | **32** | **37** | **BD-BR** | **22** | **27** | **32** | **37** | **BD-BR** |
| Traffic | 43.0 | 49.5 | 51.0 | 52.3 | **-50.1** | 25.4 | 32.4 | 36.9 | 41.1 | **-38.2** |
| PeopleOnStreet | 16.8 | 25.7 | 27.4 | 35.7 | **-26.4** | 27.8 | 23.6 | 28.7 | 31.7 | **-24.9** |
| Kimono | 23.0 | 29.8 | 36.1 | 44.9 | **-33.1** | 36.1 | 40.9 | 44.8 | 47.6 | **-41.2** |
| ParkScene | 39.8 | 43.2 | 43.4 | 47.5 | **-44.9** | 34.6 | 29.1 | 33.5 | 37.3 | **-32.9** |
| Cactus | 18.6 | 43.5 | 51.3 | 52.3 | **-45.3** | 23.6 | 33.6 | 37.8 | 39.3 | **-39.6** |
| BQTerrace | 18.3 | 40.9 | 58.1 | 58.4 | **-49.3** | 32.4 | 47.6 | 47.5 | 51.8 | **-47.3** |
| Basketball Drive | 15.9 | 31.0 | 34.1 | 40.8 | **-32.1** | 34.4 | 40.5 | 46.0 | 49.4 | **-45.0** |
| FourPeople | 43.9 | 45.6 | 47.6 | 52.0 | **-47.1** | 34.8 | 27.4 | 31.8 | 35.7 | **-34.2** |
| Johnny | 44.7 | 51.9 | 51.0 | 51.5 | **-52.2** | 55.1 | 49.8 | 51.2 | 51.8 | **-47.9** |
| KristenAnd Sara | 41.2 | 49.8 | 50.8 | 52.4 | **-49.5** | 39.3 | 40.2 | 43.0 | 45.7 | **-41.9** |
| Baskeball DrillText | 37.8 | 41.9 | 49.1 | 50.4 | **-45.4** | 40.2 | 44.3 | 47.7 | 48.1 | **-43.4** |
| ChinaSpeed | 35.8 | 40.6 | 45.3 | 53.8 | **-44.2** | 30.2 | 35.7 | 39.6 | 39.8 | **-34.8** |
| **Averages** | **31.6** | **41.1** | **45.4** | **49.3** | **-43.3** | **34.5** | **37.1** | **40.7** | **43.3** | **-39.3** |
| **Total Average** | **41.9** | | | | **-43.3** | **38.9** | | | | **-39.3** |

The average BD-BR savings of the HEVC encoder relative to VP9 and x264 encoders are *43.3%,* and *39.3%,* respectively. As it is also observed from Table IV, the bit-rate savings, on average, are increasing along with an increase of quantization parameters for both VP9 and x264 encoders. Table V below provides a full summary of the BD-BR results, where *negative BD-BR values* indicate bit-rate savings in contrast to positive values, which indicate the required overhead in bit-rate to achieve the same $PSNR_{YUV}$ values.

TABLE V. SUMMARIZED BD-BR EXPERIMENTAL RESULTS

| ***CODEC*** | **HEVC** | **x264** | **VP9** |
|---|---|---|---|
| **HEVC** | | **-39.3%** | **-43.3%** |
| **x264** | **66.4%** | | **-6.2%** |
| **VP9** | **79.4%** | **8.4%** | |

As shown in Table V, the x264 encoder achieves an average gain of 6.2% in terms of BD-BR savings compared to VP9. Also, in order to achieve the same $PSNR_{YUV}$ values of HEVC, when employing VP9 and x264 encoders, the BD-BR overhead of 79.4% and 66.4%, respectively, is required. It is noted that since the fitting of R-D curves slightly differs when fitting the R-D curve of one encoder to another and *vice versa*, the product $(100 + b_1)(100 + b_2)$ for each pair $(b_1, b_2)$ of corresponding BD-BR values (e.g., x264 vs. VP9 and VP9 vs. x264) is approximately equal to 10.000.

Furthermore, Table VI below presents detailed encoding run times as an indication of computational complexity involved for each of the tested encoders. Note, however, that all three encoders represent different degrees of software optimizations.

TABLE VI. ENCODING RUN TIMES FOR EQUAL $PSNR_{YUV}$

| | ***HEVC vs. VP9 (in %)*** | | | | ***VP9 vs. x264 (in %)*** | | | |
|---|---|---|---|---|---|---|---|---|
| **Sequences/QPs** | **22** | **27** | **32** | **37** | **22** | **27** | **32** | **37** |
| Traffic | 708 | 625 | 580 | 576 | 15168 | 16365 | 17448 | 17692 |
| PeopleOnStreet | 104 | 929 | 856 | 869 | 9866 | 11105 | 11880 | 11551 |
| Kimono | 1047 | 948 | 850 | 801 | 10220 | 12231 | 13821 | 14777 |
| ParkScene | 691 | 638 | 587 | 578 | 11724 | 15296 | 16365 | 17706 |
| Cactus | 761 | 626 | 594 | 591 | 10307 | 13365 | 14795 | 15247 |
| BQTerrace | 799 | 588 | 517 | 507 | 8223 | 9987 | 12384 | 13837 |
| Basketball Drive | 872 | 779 | 738 | 714 | 8983 | 10987 | 11480 | 12651 |
| FourPeople | 630 | 635 | 619 | 629 | 13506 | 16438 | 17557 | 18480 |
| Johnny | 644 | 649 | 679 | 749 | 9945 | 11791 | 13082 | 13869 |
| KristenAndSara | 686 | 701 | 700 | 733 | 11018 | 12717 | 12996 | 13759 |
| Baskeball DrillText | 833 | 764 | 712 | 672 | 11745 | 13238 | 14350 | 15691 |
| ChinaSpeed | 1158 | 1032 | 885 | 774 | 9522 | 11470 | 13610 | 16246 |
| **Averages** | **822** | **743** | **693** | **683** | **10852** | **12916** | **14148** | **15126** |
| **Total Average** | **735.2** | | | | **13260.3** | | | |

As can be seen from Table VI, the typical encoding times of the VP9 encoder are around 130 times higher than those measured for the x264 encoder. On the other hand, when compared to the H.265/MPEG-HEVC reference encoder implementation, the VP9 encoding times are lower by a factor of 7.35, on average.

## V. CONCLUSION

A performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders was presented. According to the experimental results, the coding efficiency of VP9 was shown to be inferior to both H.264/MPEG-AVC and H.265/MPEG-HEVC with an average bit-rate overhead at the same objective quality of 8.4% and 79.4%, respectively. Also, it was shown that the VP9 encoding times are larger by a factor of more than 100 compared to those of the x264 encoder.

## REFERENCES


[1] Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video, 1994 :ITU-T and ISO/IEC JTC 1.

[2] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.13, no.7, pp.560-576, Jul. 2003.

[3] H.264/AVC Software Coordination, JM Reference Software, Online: http://iphome.hhi.de/suehring/tml

[4] B. Bross, “An overview of the next generation High Efficiency Video Coding (HEVC),” in “Next Generation Mobile Broadcasting”, (ed. David Gómez-Barquero), CRC Press, 2013.

[5] ITU-T, Recommendation H.265 (04/13), Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of Moving Video, High Efficiency Video Coding, Online: http://www.itu.int/rec/T-REC-H.265-201304-I.

[6] J. Ohm, G.J. Sullivan, H. Schwarz, T.K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—including High Efficiency Video Coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol. 22, no.12, pp.1669-1684, Dec. 2012.

[7] B. Li, G. J. Sullivan, and J. Xu, “Comparison of compression performance of HEVC Draft 9 with AVC high profile and performance of HM9.0 with temporal scalability characteristics,” JCTVC-L0322, 12th JCT-VC meeting, Geneva, Switzerland, Jan. 2013.

[8] M. Horowitz, F. Kossentini, N. Mahdi, S. Xu, H. Guermazi, H. Tmar, B. Li, G. J. Sullivan, J. Xu, “Informal subjective quality comparison of video compression performance of the HEVC and H.264/MPEG-4 AVC standards for low-delay applications,” *Proc. SPIE 8499, Applications of Digital Image Processing XXXV*, 84990W, Oct. 15, 2012.

[9] WebM™: an open web media project, VP8 Encode Parameter Guide, 2-Pass Best Quality VBR Encoding, Online: http://www.webmproject.org/docs/encoder-parameters/#2-pass-best-quality-vbr-encoding

[10] E. Ohwovoriole, and Y. Andreopoulos, "Rate-Distortion performance of contemporary video codecs: comparison of Google/WebM VP8, AVC/H.264 and HEVC TMuC," *Proc. London Communications Symposium (LCS),* Sep. 2010, pp. 1-4

[11] J. Bankoski, P. Wilkins, X. Yaowu, "Technical overview of VP8, an open source video codec for the web," *Multimedia and Expo (ICME), 2011 IEEE International Conference on* , pp.1,6, 11-15 Jul. 2011.

[12] Chromium® open-source browser project, VP9 source code, Online: http://git.chromium.org/gitweb/?p=webm/libvpx.git;a=tree;f=vp9;hb=aaf61dfbcab414bfacc3171501be17d191ff8506

[13] J. Bankoski, R. S. Bultje, A. Grange, Q. Gu, J. Han, J. Koleszar, D. Mukherjee, P. Wilkins, and Y. Xu, “Towards a next generation open-source video codec,” *Proc. SPIE 8666, Visual Information Processing and Communication IV*, Feb. 21, 2013, pp. 1-13.

[14] Paul Wilkins, Google® Groups "WebM Discussion", Online: https://groups.google.com/a/webmproject.org/forum/?fromgroups#!topic/webm-discuss/UzoX7owhwB0

[15] Projects from VideoLAN™, x264 software library and application, Online: http://www.videolan.org/developers/x264.html

[16] x264 free library/codec, 32-bit, 8-bit depth version r2334 (May 22, 2013), Online: http://www.divx-digest.com/software/x264.html

[17] x264 Settings, Online: http://mewiki.project357.com/wiki/X264_Settings

[18] x264 Codec Capabilities Analysis, x264 Codec Parameters Comparison, YUVsoft Corp., Online: http://www.yuvsoft.com/pdf/x264_parameters_comparison.pdf

[19] G.J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1649-1668, Dec. 2012.

[20] B. Bross, W.-J. Han, J.-R. Ohm, G. Sullivan, Y.-K. Wang, and T. Wiegand "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent)," JCT-VC, Doc. JCTVC-L1003. Geneva, Switzerland, Jan. 2013.

[21] HEVC Reference Software, Online: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[22] P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, “Subjective quality evaluation of the upcoming HEVC video compression standard,” *Proc. SPIE 8499, Applications of Digital Image Processing XXXV*, Oct. 15, 2012, pp. 1-13.

[23] G. Correa, P. Assuncao, L. Agostini, L.A. da Silva Cruz, "Performance and computational complexity assessment of High-Efficiency Video encoders," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol.22, no.12, pp.1899-1909, Dec. 2012.

[24] Ronald Bultje, Google® Groups "WebM Discussion", Online: https://groups.google.com/a/webmproject.org/forum/?fromgroups#!topic/webm-discuss/xopTll6KqGI

[25] John Koleszar, Google® Groups "Codec Developers", Online: https://groups.google.com/a/webmproject.org/forum/#!msg/codec-devel/yMLXzaohONU/m69TbYnEamQJ

[26] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves”, *ITU-T Q.6/SG16 VCEG 13th Meeting*, Document VCEG-M33, Austin, USA, Apr. 2001.

[27] F. Bossen, “Common HM test conditions and software reference configurations,” document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013.