Video quality evaluation and testing verification of H.264, HEVC, VVC and EVC video compression standards

Shreyanka Subbarayappa¹,² and K R Rao³

¹Electronics and Communication Engineering Department, Ramaiah University of Applied Sciences, Bangalore, KA 560054, India

²Electrical Engineering Department, University of Texas at Arlington, Arlington, TX 76019, U.S.A

Abstract. This paper is a comparative analysis with respect to video compression, coding efficiency with respect to performance and quality of the compressed video of Advanced Video Coding (H.264), High Efficiency Video Coding (HEVC), Versatile Video Coding (VVC) and MPEG-5 Essential Video Coding (EVC). The approach used for coding and comparative analysis is by keeping the video quality (QP) and group of pictures (GOP) constant for all the four codecs. Performance with respect to quality and coding efficiency is tabulated with respect to objective quality metrics analysis like (PSNR, M-SSIM, Bitrate, compression ratio) and subjective quality analysis. From both these techniques, it is observed that the VVC is superior to EVC, HEVC and H.264 in terms of compression. Therefore, more bitrate savings is observed for VVC which is 15% to 45% more when compared to EVC. Due to the complexity of the encoder block, VVC takes almost twice the time taken to encode EVC and HEVC taking the least time to encode the video bit streams. The conclusion of the quality of video keeping system parameters constant, showed VVC performing better than EVC, followed by HEVC and then H.264 for both objective and subjective analysis. Time taken to decode the encoded sequences is almost same with very less delta difference between the video codecs. The codec configurations with algorithmic enhancements is also explained in this paper.

1. Introduction

Innovations in the communication systems and technology are growing tremendously and the growth seen is unimaginable in the last forty years. In multimedia communication systems, technology has transformed from analog television to digital television in the video domain. Mobile phones are known as smart phones as they are used, not only to make voice calls, but also used to send emails, video calls, transfer data, GPS, taking pictures and so on. Due to the widespread user applications, compression on data becomes important to save system resources. Video has occupied 80% of all major traffic of data transfer in 2019 and is expected to cover 85% by 2021 [2]. Video is also continuously increasing in size and quality from standard-definition (SD) video to ultra-high definition (UHD – 4K, 8K and 12K) video. More data or size in video requires higher transmission bandwidth or more disk space to store, which is very expensive. This drives into the betterment in compression and hence a demand for a new video codec. Several algorithms are implemented to achieve compression of Image or Video with respect to the user’s demand on the quality of output as well as for applications it is used for. These algorithms working together are classified in terms of codecs. These video codecs are developed to target specific
application usages. Advances in video compression technology are used to reduce the utilization of the system resources like processing time, memory use, network bandwidth and battery life. This is possible by reducing the complexity of the video codecs without compromising on the output video quality. There are two separate streams in the next generation video coding development work. VVC driven by Joint Video Exploration team (JVET) and EVC driven by Moving Picture Expert Group (MPEG). These two codecs are the extended versions with respect to the advances in compression technology to HEVC.

Versatile Video Coding (VVC) standard which was finalized on July 6th, 2020 by the JVET team of ITU-T SG 16 WP 3 [11] and ISO/IEC JTC 1/SC 29/WG 11 [10] is due by October 2020. The latest software version of VVC available is VTM-8.0 [14]. This codec promises 40% more compression than its predecessor (HEVC) with having the same perceptual quality of video [6]. VVC supports lossless and subjectively lossless compression with resolutions varying from 4K to 16K along with 360 videos. It supports video contents of 8 to 16 bit depth with chroma formats ranging from 4:2:0 to 4:4:4. Some applications of VVC are high dynamic range (HDR) video, Multiview coding, still picture coding, panoramic formats. Complexity of the codec was observed to be 10 times more than that of HEVC, but that is waived off with the better quality of the video output with more compression achieved.

Essential Video Coding (EVC) standard which is under development by the MPEG team of ISO/IEC JTC 1/SC 29/WG 11 is due by end-2020. The latest software version of EVC available is ETM-4.0 [15]. The coding efficiency of EVC is almost like HEVC with slightly better video quality than HEVC. It is developed with some licensing conditions, i.e. royalty-free for the baseline profile and with IPR for the main profiles.

High Efficiency Video Coding (HEVC) also known as H.265 standard was developed by Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP-3 and ISO/IEC JTC1/SC29/WG11 and published the first version in June 2013. The next version included applications like range extensions (RExt), Multiview extensions (MV-HEVC) and scalability extensions (SHVC) and published in 2015. The coming versions applications included extensions in 3D video and screen content coding (SCC) which were published in the 1st quarter of 2017. HEVC promised in offering 30%-50% better compression of video with respect to H.264 and having the same perceptual quality of the video [1]. It supported video contents up to 8K resolution with bit depth from 8 to 12 bits. The latest software version of HEVC is HM-16.0 [13].

Advanced Video coding (AVC) also known as H.264 was developed by Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6) and published in 2003. H.264 codec is the first codec to support HD videos and with a very large video compression possibility with good perceptual output video quality. In today’s time, it is the most commonly used video format with 81% [19] of the video industries using it for distribution, compression and recording of the video content. With the later years of its initial development, some new features were added to the codec, i.e. fidelity range extensions, multiview video coding, scalable video coding, 3D-AVC and MFC stereoscopic coding. H.264 promised 50% better bitrate efficiency when compared to MPEG-2 part 2 [5]. It supported contents of bit depth 8 to 14 bits and chroma formats from 4:2:0 initially in baseline profile to 4:4:4 at the later stage for high profile. This codec is known for its broad application range like digital video compression at low-bitrate, internet streaming to broadcasting, and digital cameras for nearly lossless coding. The latest software version of AVC is JM-19.0 [12].

2. Structure of H.264, HEVC, VVC and EVC Codecs

2.1. H.264 structure [5]

The H.264 codec structure starts with a block for video frame to be divided into maximum 16x16 macroblocks. The first frame of the video undergoes an intra prediction of Transform coding which is 2 Dimensional DCT (2-D DCT) [7] [20] followed by scaling and quantization. These quantized transform coefficients are given to the entropy block where we select either CABAC or CAVLC as entropy coding techniques to get the bit-stream of the 1st frame [8]. Simultaneously we have the decoder block within the encoder block which decodes the present frame by performing inverse quantization and inverse transform and stores the frame for performing motion estimation, which is needed for the next frame. A
debloating filter is added at the decoder section within the encoder block to smooth the ringing effect produced by 2-D DCT in the output video. There are 9 intra prediction modes with 5 luma prediction modes and 4 chroma prediction modes. The encoder framework of H.264 is analyzed in Figure 1.

![Figure 1. H.264 video coding encoder framework][5][9]

2.2. HEVC Structure [3]

The HEVC codec is a hybrid motion compensated codec with the block structure splitting in the Quadtree format. Coding Tree Unit (CTU) for each block ranges up to 64X64. The codec constitutes total 35 intra prediction modes. There are two types of transform coding in HEVC, i.e. DCT-II and DST-VII. Transform block sizes vary from 8x8, 16x16 to 32x32. The Inter prediction is based on the hierarchical weighted prediction of the prediction unit (PU) level motion vector prediction. There are two loop filters within the encoder framework. One being the deblocking filter and the other being the sample adaptive offset (SAO) filter which removes the artifacts produced by transform and quantization technique. CABAC is the only entropy coding technique used in HEVC. The encoder framework of HEVC is analysed in Figure 2.

![Figure 2. HEVC video coding encoder framework][1][4]

2.3. VVC Structure

The VVC codec is the next generation video codec which also follows the hybrid motion compensated algorithm. Block structure partition is performed by Quad Tree Binary Tree (QTBT) and Ternary Tree (TT) structure which is new in VVC. The CTU’s has maximum size up to 256x256. The inter prediction is performed by cross component linear model (CCLM) and having 65 intra prediction modes as enhanced intra mode coding. Mode dependent non-separable secondary transform and adaptive multiple core transform are the two types of transform coding. Transform block sizes vary from 4x4, 8x8, 16x16, 32x32 and 64x64. Inter prediction includes the new affine motion prediction and everything else in that
part is similar to HEVC. In the filter block, Adaptive loop filter is the new addiction along with deblocking filter and sample adaptive offset filter (SAO) of HEVC. Modified version of CABAC is the new technique at entropy block. The encoder framework of VVC is analysed in Figure 3.

![Figure 3. VVC video coding encoder framework [14]](image)

### 2.4. EVC Structure

The EVC codec by the MPEG team uses quad tree (QT), binary and ternary tree (BTT) and split unit coding order (SUCO) for block structure partition with each CTU sizing up to maximum 128x128. There are two profiles in EVC, naming baseline profile and high profile to select the coding techniques. Inter prediction has Uni/Bi prediction, Affine motion prediction, adaptive motion vector resolution (AMVR), merge with motion vector difference (MMVD), history-based motion vector prediction (HMVP) and decoder side motion vector refinement. There are 33 Intra prediction modes. The transform block has DCT-II for baseline profile and Adaptive transform selection (ATS) in main profile to select between DCT VIII and DST VII. The transform size ranges from 4x4, 8x8,16x16, 32x32, 64x64 and 128x128. In-Loop filter like Deblocking filter (DBF) along with Hadamard transform domain filter (HTDF), Adaptive Loop Filter (ALF), Advanced deblocking filter is added to remove the artifacts caused by transform and quantization for smoothing the video frames. At the entropy block, there is a choice for 2-D Run Length coding, Arithmetic coding engine, context modelling and advanced coefficient coding (ADCC) based on the profile chosen for encoding. The encoder framework of EVC is analyzed in Figure 4.

![Figure 4. EVC video coding encoder framework [18]](image)

### 3. Test Configurations

All the video codecs were tested for the same content with GOP = 16 (BBBBBBBBBBBBBI) and QP = 10, 20, 30, 40 and 50. The system used is Dell Inspiron with i7 processor/ Windows 10/ 64 bit. The test sequences used for testing is described in Table 1.
Table 1. Test sequence description [16][17]

| Test sequence     | Resolution | Number of frames | Original size (kB) | Source       |
|-------------------|------------|------------------|--------------------|--------------|
| Akiyo             | 176x144    | 300              | 11138              | Xiph.org     |
| SpeedBag          | 640x360    | 120              | 40500              | Xiph.org     |
| Shield2           | 640x360    | 120              | 40500              | Xiph.org     |
| MobileCalendar    | 720x486    | 360              | 246038             | Xiph.org     |
| Stockholm         | 1280x720   | 604              | 815400             | Xiph.org     |
| FourPeople        | 1280x720   | 601              | 811350             | Xiph.org     |
| Beauty            | 1920x1080  | 600              | 1822500            | UVG          |
| SunBath           | 3840x2160  | 600              | 3645000            | UVG          |
| Bhosphorus        | 3840x2160  | 600              | 7290000            | UVG          |

3.1. H.264 [12]
Software : Joint Model (JM) – version 19.0
Encoder Commandline : lencod.exe -d encoder_main.cfg
Decoder Commandline : ldecod.exe

In the encoder commandline, -d indicated the configuration file to be added which is in the directory location ‘JM\bin’. In the testing mode, encoder_main.cfg file is edited according to the parameter preferences like, ‘InputFile’ location on the system to be added. Input/output content parameters are edited in the source width/height and accordingly output width/height. QPI/P/BSlice are changed according to the QP value needed for testing. Other parameters to be edited in the configuration file “encoder_main.cfg” is shown.

#== encoder_main.cfg File ===============

| Parameter          | Value                              |
|--------------------|------------------------------------|
| InputFile          | "C:\Desktop\akiyo.yuv"             |
| FramesToBeEncoded  | 300                                |
| FrameRate          | 100.0                              |
| SourceWidth        | 176                                |
| SourceHeight       | 144                                |
| OutputWidth        | 176                                |
| OutputHeight       | 144                                |
| IntraPeriod        | 16                                 |
| QPISlice           | 10                                 |
| QPPSlice           | 10                                 |
| NumberBFrames      | 15                                 |
| QPBSlice           | 10                                 |

After the encoder, the encoded binary file ‘test.264’ is obtained in the same folder location. After the decoder, the output file is ‘test.yuv’ which is the reconstructed video. Main profile is used in testing H.264 for the analysis and comparison purpose.

3.2. HEVC [13]
Software : Hybrid Model (HM) – version 16.0
Encoder Commandline : TAppEncoder.exe -c encoder_randomaccess_main.cfg -c akiyo.cfg
Decoder Commandline : TAppDecoder.exe -b str.bin -o dec.yuv -d 8

In the encoder commandline, -c indicated the configuration file to be added which is in the directory ‘cfg’ and ‘cfg\per-sequence’. The cfg\per-sequence location file is to be created by the user under any name as the content file to be tested and includes the content parameters. Since the content to be tested
in this example is akiyo, ‘cfg\per-sequence’ file is named as ‘akiyo.cfg’ as the secondary configuration file. In our testing, the main configuration file in ‘cfg\’ location that is ‘encoder_randomaccess\main.cfg’ parameters is edited. The values of ‘IntraPeriod’, ‘GOPSize’ to 16 and ‘QP’ parameters are edited as shown in VVC software parameters. New ‘cfg/per-sequence’ file to be added with parameters under the name ‘akiyo.cfg’ is shown.

#======== akiyo.cfg File I/O ===============

| Parameter       | Value       | Description                               |
|-----------------|-------------|-------------------------------------------|
| InputFile       | C:\Desktop\akiyo.yuv | Input content destination                 |
| InputBitDepth   | 8           | Input content bit-depth                   |
| InputChromaFormat | 420        | Ratio of luminance to chrominance samples of the input content |
| FrameRate       | 120         | Frame Rate per second of the content      |
| FrameSkip       | 0           | Number of frames to be skipped in input content |
| SourceWidth     | 176         | Input content frame width                 |
| SourceHeight    | 144         | Input content frame height                |
| FramesToBeEncoded | 300      | Number of frames to be coded              |
| Level           | 4.1         |                                           |

After the encoder, the encoded binary file ‘str.bin’ is obtained in the same file location. After the decoder, the output file is ‘dec.yuv’ which is the reconstructed video. Random access main profile is used in HEVC testing for the analysis and comparison purpose.

3.3. VVC [14]
Software: VVC Test Model (VTM) – version 8.0
Encoder Commandline: EncoderApp.exe -c encoder_randomaccess_vtm.cfg -c akiyo.cfg
Decoder Commandline: DecoderApp.exe -b str.bin -o dec.yuv -d 8

VVC open source software is a test version which is still under development. The software at the encoder side uses two configuration files. The second file is the content file which is similar as HEVC as ‘akiyo.cfg’. The first config file is the main file names as ‘encoder_randomaccess_vtm.cfg’ which specifies the parameters to be changed in order to get the output as desired as shown.

#======== encoder_randomaccess_vtm.cfg File ===============

| Parameter      | Value       | Description                               |
|----------------|-------------|-------------------------------------------|
| IntraPeriod    | 16          | # Period of I-Frame ( -I = only first)    |
| GOPSize        | 16          | # GOP Size (number of B slice = GOPSize-1) |
| QP             | 10          | # Quantization parameter(0-51)            |

After the encoder, the encoded binary file ‘str.bin’ is obtained in the same folder location. After the decoder, the output file is ‘dec.yuv’ which is the reconstructed video. Random access main profile is used in test VVC for the analysis and comparison purpose.

3.4. EVC [15]
Software: EVC Test Model (ETM) – version 4.0
Encoder Commandline: evca_encoder.exe -i akiyo.yuv -q 10 -w 176 -h 144 -p 16 -f 300 -z 120 -d 8 -o str.bin -config encoder_randomaccess.cfg
Decoder Commandline: evca_decoder.exe -i str.bin -o dec.yuv

EVC software used for testing is a test model and this codec is still under development. In the Encoder commandline the following notations denotes:
‘-i’ - path to the input file
‘-q’ - quantization parameter
‘-w’ - source width
For every test run, we need to change the ‘-q’ value in the encoder commandline with respect to QP value for each run. The configuration file ‘encoder_randomaccess.cfg’ is added in the command line with no edits needed in the file. After the encoder, the encoded binary file ‘str.bin’ is obtained in the same file location. After the decoder, the output file is ‘dec.yuv’ which is the reconstructed video. Random access main profile is used in EVC testing for the analysis and comparison purpose.

4. Results and Summary
It is very essential to analyze the codec implementation and perform a comparative study looking at the performance of the codec and output quality of the encoded bitstream by both objective metric and subjective metric. Three different methods were used for the analysis of the codec implementation. They are as specified below,

4.1. Objective Quality Metric [21]:
The first method is an objective quality metric comparative study between the codecs in terms of PSNR, SSIM, Bitrate, time taken to encode, time taken to decode and compression size for each of the test sequence specified in Table 1 and keeping the QP constant to 32, GOP = 16 (IBBBBBBBBBBBBBBB).

4.2. Rate Distortion:
The second method is the rate distortion or RD method where the rate distortion graph is obtained for a content with different QP values (10, 20, 30, 40 and 50) and same GOP =16 (IBBBBBBBBBBBBBBB) for all four codecs.

4.3. Subjective Quality Analysis:
The third method is the subjective quality analysis where the comparisons of the codecs are observed by the visual analysis of the videos. The rate distortion method outputs for QP values (10, 20, 30, 40 and 50) are observed frame by frame and the distortion noted for codecs H.264, HEVC, VVC and EVC.

| Input Video File | PSNR (dB) | SSIM (YUV) | MSE (YUV) | Bitrate (kbps) | Encoding Time (s) | Decoding Time (s) | Compressed o/p video size (kB) |
|------------------|-----------|------------|-----------|----------------|-------------------|-------------------|-----------------------------|
| Akiyo            | 36.80     | 0.94       | 15.55     | 106.39         | 323.238           | 0.093             | 39                          |
| SpeedBag         | 41.05     | 0.97       | 5.89      | 141.63         | 1873.631          | 0.451             | 104                         |
| Shield2          | 34.09     | 0.88       | 28.74     | 862.01         | 1292.539          | 0.668             | 632                         |
| MobileCalendar   | 34.88     | 0.92       | 24.45     | 2486.94        | 8073.856          | 4.539             | 5465                        |
| Stockholm        | 36.10     | 0.89       | 19.90     | 2027.63        | 38512.408         | 10.091            | 7475                        |
| Fourpeople       | 39.67     | 0.95       | 8.67      | 401.52         | 27572.912         | 8.137             | 1473                        |
Table 3. Objective quality results of HEVC encoder and decoder software implementation

| Input Video File | PSNR (dB) | SSIM (YUV) | MSE (YUV) | Bitrate (kbps) | Encoding Time (s) | Decoding Time (s) | Compressed o/p video size (kB) |
|------------------|-----------|------------|-----------|----------------|------------------|-------------------|-----------------------------|
| Akiyo            | 39.250    | 0.98945    | 7.728     | 130.922        | 49.35            | 0.454             | 40                          |
| Shields2         | 35.947    | 0.98599    | 16.54     | 347.445        | 231.14           | 0.693             | 255                         |
| Stockholm        | 35.812    | 0.98276    | 17.06     | 4222.370       | 5501.18          | 11.149            | 2595                        |
| Beauty           | 38.6891   | 0.97482    | 8.795     | 2238.458       | 2127.03          | 23.354            | 1367                        |
| SunBath          | 42.921    | 0.99132    | 3.319     | 12965.12       | 37967.48         | 54.871            | 3957                        |
| Bosphorus        | 39.250    | 0.98945    | 7.728     | 130.922        | 49.35            | 0.454             | 40                          |

Table 4. Objective quality results of VVC encoder and decoder software implementation

| Input Video File | PSNR (dB) | SSIM (YUV) | MSE (YUV) | Bitrate (kbps) | Encoding Time (s) | Decoding Time (s) | Compressed o/p video size (kB) |
|------------------|-----------|------------|-----------|----------------|------------------|-------------------|-----------------------------|
| Akiyo            | 40.2677   | 0.9879     | 97.8196   | 83.3984        | 446.137          | 1.099             | 26                          |
| Shields2         | 36.7821   | 0.9890     | 218.2678  | 192.6293       | 1727.356         | 1.279             | 142                         |
| Stockholm        | 36.1117   | 0.9690     | 254.7011  | 2356.0132      | 22901.051        | 19.528            | 1448                        |
| Beauty           | 38.9595   | 0.9967     | 132.2072  | 1691.5568      | 101368.613       | 45.738            | 1033                        |
| SunBath          | 43.8083   | 0.9990     | 43.2886   | 7955.6192      | 168480.477       | 104.085           | 2978                        |
| Bosphorus        | 42.0861   | 0.9830     | 64.3559   | 890.18         | 192677.252       | 202.441           | 3260                        |

Table 5. Objective quality results of EVC encoder and decoder software implementation

| Input Video File | PSNR (dB) | SSIM (YUV) | Bitrate (kbps) | Encoding Time (s) | Decoding Time (s) | Compressed o/p video size (kB) |
|------------------|-----------|------------|----------------|-------------------|-------------------|-----------------------------|
| Akiyo            | 41.43     | 0.995      | 21.262         | 154.576           | 1.518             | 39                          |
| Shields2         | 37.92     | 0.993      | 348.658        | 1114.427          | 1.581             | 256                         |
| Stockholm        | 38.99     | 0.983      | 709.307        | 17186.5           | 17.895            | 2615                        |
| Beauty           | 39.601    | 0.972      | 2079.912       | 40601.449         | 110.111           | 1270                        |
| SunBath          | 45.6415   | 0.999      | 11141.700      | 94910.641         | 98.6              | 3401                        |

The objective quality metric comparison as mentioned in point one in section 4, using metrics like PSNR, SSIM, Bitrate, encoding time, decoding time and compressed output size is shown for each codec H.264, HEVC, VVC and EVC for different contents in Table 2, 3, 4 and 5. H.264 was not tested for 4K resolution videos as it does not support 4K content. These codecs were tested keeping QP=32 and GOP = 15 for random access main profile for HEVC, VVC and EVC and main profile for H.264. Keeping the quality QP constant, it was observed that the bitrate savings had a lot of variations across codecs. Bitrate for contents were observed less for VVC followed by EVC, HEVC and then H.264. Bitrate savings also concludes that encoded compression size was achieved more for VVC followed by EVC, HEVC and then H.264. This is achieved due to the various advanced algorithms in VVC and EVC compared to HEVC and H.264, like increased CTU size, advanced entropy coding, advanced motion estimation and compensation blocks. The encoding time for each content was observed to be more for VVC then followed by EVC and then HEVC as seen in Figure 5. This is due to the extensive algorithmic part CTU partitioning, motion estimation techniques, more prediction modes observed in VVC and EVC when compared to HEVC and H.264. Time consumed to decode the encoded video for different codecs were almost very close, for each content with less time difference between them.
Figure 5. Encoding time for each content for HEVC, VVC and EVC

The second point in section IV performs a comparative study using rate distortion method. Here RD graph is obtained testing one content with different QP size (10, 20, 30, 40 and 50) on all four codecs (H.264, HEVC, VVC and EVC). Other parameters of the codecs are kept constant like GOP, all filters ‘ON’ and main profile. For each testing scenario, PSNR and Bitrate is tabulated, and graph is obtained. Figure 6 shows the rate distortion graph of stockholm content keeping other codec parameters constant. It is observed from the graph that VVC performs better with better quality and bitrate savings, followed by EVC, HEVC and H.264 codecs.

Figure 6. Rate distortion graph for Stockholm content

The third point in section IV is subjective quality analysis. The outputs of rate distortion tests are view in a media player and quality degradation is observed frame by frame. Fig. 7 shows the quality degradation in encoded output video for QP=50 and shield2 content. It was observed from the red marked area that VVC showed better quality, followed by EVC, HEVC and then H.264. This analysis was observed for other contents from Table 1 and the results for quality analysis remained same.
5. Conclusion
The codec analysis using the three methods of verification concluded by providing the same results in terms of codec performance with respect to four codec software implementation. The testing verified that VVC showed better bitrate savings for the encoded bit sequence, 30% – 50% better when compared to HEVC codecs keeping the video quality constant. EVC showed 10% – 30% better bitrate savings when compared to HEVC keeping the video quality constant. HEVC providing 50% better bitrate savings than H.264 for same output video quality was verified.

Complexity of the codec is another domain which opens for research to improve the codec time performance. VVC as the codec with software version 8.0 showing more time to encode the sequence when compared to EVC, HEVC, H.264. The time taken was almost double the time taken for EVC to encode a 4K resolution video. This is due to the increase in prediction modes which is almost double the modes in EVC and HEVC. This opens for avenues of research in reducing the complexity of the codec. VVC and EVC brings in a lot of applications usages to the market like supporting 4K to 8K streaming video and HDR video support. VVC and EVC also adds the development in the field of transcoding video codecs and multiplexing with other audio codecs.

References
[1] Subbarayappa S and Rao K R 2019 Overview and extensions of the high efficiency video coding (HEVC) and beyond (Versatile Video Coding) International Journal of Emerging Technology and Advanced Engineering 9(12) 70-97
[2] M. Lister, 37 Staggering Video Marketing Statics for 2018, World stream online advertising made easy, June 30, 2020. Accessed on: July 5th, 2020. [Online] Available: https://www.wordstream.com/blog/ws/2017/03/08/video-marketing-statistics#:~:text=45%25%20of%20people%20watch%20more,online%20marketers%20use
%20video%20content.
[3] Sullivan G J, Ohm J R, Han W J and Wiegand T 2012 Overview of the high efficiency video coding (HEVC) standard IEEE Transactions on circuits and systems for video technology 22(12) 1649-68
[4] Ohm J R, Sullivan G J, Schwarz H, Tan T K and Wiegand T 2012 Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC) IEEE Transactions on circuits and systems for video technology 22(12) 1669 - 84
[5] Wiegand T, Sullivan G J, Bjontegaard G and Luthra A 2003 Overview of the H.264/AVC video coding standard IEEE Transactions on circuits and systems for video technology 13(7) 560-76
[6] Topiwala P, Krishnan M and Dai W 2018 Performance comparison of VVC, AV1 and HEVC on 8-bit and 10-bit content Applications of Digital Image Processing XLI International Society for optics and Photonics 10752 107520V
[7] Ochoa D H and Rao K R 2019 Discrete Cosine Transforms CRC Press
[8] Kwon S K, Tamhankar A and Rao K R 2006 Overview of H.264/MPEG-4 part 10 Journal of Visual Communication and Image Representation 17(2) 186-216
[9] Richardson I E 2004 H. 264 and MPEG-4 video compression: video coding for next-generation multimedia John Wiley and Sons
[10] ISO Website: Accessed on July 9th, 2020 [Online] Available: https://www.iso.org/home.html
[11] ITU-T Website: Accessed on July 9th, 2020 [Online] Available: https://www.itu.int/en/ITU-T/Pages/default.aspx
[12] JM Software: Accessed on July 9th, 2020 [Online] Available: https://github.com/shihuade/JM.git
[13] HM Software: Accessed on July 9th, 2020 [Online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/trunk/
[14] VTM Software: Accessed on July 9th, 2020 [Online] Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.git
[15] ETM Software: Accessed on July 9th, 2020 [Online] Available: https://gitlab.com/MPEG-5/ETM
[16] Test videos: Accessed on July 9th, 2020 [Online] Available: https://media.xiph.org/video/derf
[17] Test videos: Accessed on July 9th, 2020 [Online] Available: http://ultravideo.cs.tut.fi/
[18] Choi K et al 2019 MPEG-5: essential video coding standard Applications of Digital Image Processing XLII International Society for optics and photonics 11137 1113710
[19] Topiwala P, Krishnan M and Dai W 2019 Performance comparison of VVC, AV1 and EVC Applications of Digital Image Processing XLII International Society for optics and photonics 11137 1113715
[20] Subbarayappa S 2012 Implementation and analysis of directional discrete cosine transform in H.264 for baseline profile M.S. thesis Electrical Engineering Dept. University of Texas at Arlington
[21] Hore A and Ziou D 2010 Image quality metrics: PSNR vs. SSIM 20th International conference on pattern recognition 2366-69