OVERVIEW OF MPEG-5 PART 2 – LOW COMPLEXITY ENHANCEMENT VIDEO CODING (LCEVC)

Florian Maurer[1], Stefano Battista[2], Lorenzo Ciccarelli[3], Guido Meardi[3], Simone Ferrara[3]
[1]RWTH Aachen University, Germany, [2]Università Politecnica delle Marche (UNIVPM), Italy, [3]V-Nova, UK

Abstract *– This paper provides an overview of MPEG-5 Part 2 Low Complexity Enhancement Video Coding (LCEVC), a novel video coding standard from the MPEG ISO Working Group. The codec is designed for use in conjunction with existing video codecs, leveraging specific tools for encoding "residuals", i.e. the difference between the original video and its compressed representation. LCEVC can improve compression efficiency and reduce the overall computational complexity using a small number of specialized enhancement tools. This paper provides an outline of the coding structure of encoder and decoder, coding tools, and an overview of the performance of LCEVC with regard to both compression efficiency and processing complexity.*

Keywords – Low Complexity Enhancement Video Coding (LCEVC), MPEG-5 Part 2, multi-resolution video coding, video compression.

1. INTRODUCTION

This paper provides an overview of MPEG-5 Part 2 Low Complexity Enhancement Video Coding (LCEVC), a new video coding standard developed by MPEG that is scheduled to be published as ISO/IEC 23094-2 [1].

Rather than being a replacement for existing video coding schemes, LCEVC is designed to leverage existing (and future) codecs to enhance their performances whilst reducing their computational complexity. It is not meant to be an alternative to other codecs, but rather a useful complement to any codec.

This is achieved by a combination of processing an input video at a lower resolution with an existing single-layer codec and using a simple and small set of highly specialized tools to correct impairments, upscale and add details to the processed video.

2. COMMERCIAL REASONS FOR THE STANDARD

LCEVC was driven by several commercial needs put forward to MPEG by many leading industry experts from various areas of the video delivery chain, from vendors to traditional broadcasters, from satellite providers to over-the-top (OTT) service providers and social media [2].

Service providers work with complex ecosystems. They make choices on codecs based on various factors, including maximum compatibility with their existing ecosystems, costs of deploying the technology (including royalty rates), etc. Sometimes they are forced to make certain choices. Whichever is the case, changing codecs cannot be done without relevant up-front investments and large amounts of time. Accordingly, having the possibility to upgrade an ecosystem without the need to replace it completely and still having the freedom to select a base codec of their choice is an important option that operators need to have.

Further, service operators, small and big alike, are increasingly concerned about the cost of delivering a growing number of services, often using decentralized infrastructures such as cloud-based systems or battery-powered edge devices. The need to increase the overall efficiency of video delivery systems must also be balanced with the seemingly conflicting needs to upgrade video resolutions and consume less power.

Finally, the "softwarization" of solutions across the technological spectrum has brought up the need to have also codec solutions which do not necessarily require a bespoke dedicated hardware for operating efficiently, but rather can operate as a software layer on top of existing infrastructures and deliver the required performances.

LCEVC seeks to solve the above issues by providing a solution that is compatible with existing (and future) ecosystems whilst delivering it at a lower computational cost than it would be otherwise possible with a tout-court upgrade.

Aside from rapidly improving the efficiency of legacy workflows, LCEVC can also improve the business case for the adoption of next-generation codecs, by combining their superior coding efficiency with significantly lower processing requirements.

3. KEY TECHNICAL FEATURES

LCEVC deploys a small number of very specialized coding tools that are well suited for the type of data it processes. Some of the key technical features are highlighted below.

3.1 Sparse residual data processing

As further shown in Section 5, the coding scheme processes one or two layers of residual data. This residual data is produced by taking differences between a reference video frame (e.g., a source video) and a base-decoded upscaled version of the video. The resulting residual data is sparse information, typically edges, dots and details which are then efficiently processed using very simple and small transforms which are designed to deal with sparse information.

3.2 Efficient use of existing codecs

The base codec is typically used at a lower resolution. Because of this, the base codec operates on a smaller number of pixels, thus allowing the codec to use less power, operate at a lower quantization parameter (QP) and use tools in a more efficient manner.

3.3 Resilient and adaptive coding process

The scheme allows the overall coding process to be resilient to the typical coding artefacts introduced by traditional discrete cosine transform (DCT) block-based codecs. The first enhancement sub-layer (L-1 residuals) enables us to correct artefacts introduced by the base codec, whereas the second enhancement sub-layer (L-2 residuals) enables us to add details and sharpness to the corrected upscaled base for maximum fidelity (up to lossless coding). Typically, the worse the base reconstruction is, the more the first layer may contribute to correct. Conversely, the better the base reconstruction is, the more bit rate can be allocated to the second sub-layer to add the finest details.

3.4 Agnostic base enhancement

The scheme can enhance any base codec, from existing ones (MPEG-2, VP8, AVC, HEVC, VP9, AV1, etc.) to future and under-development ones (including EVC and VVC). The reason is that the enhancement operates on a decoded version of the base codec in the pixel domain, and therefore it can be used on any format as it does not require any information on how the base has been encoded and/or decoded.

3.5 Parallelization

The scheme does not use any inter-block prediction. The image is processed by applying small (2x2 or 4x4) independent transform kernels over the layers of residual data. Since no prediction is made between blocks, each 2x2 or 4x4 block can be processed independently and in a parallel manner. Moreover, each layer is processed separately, thus allowing the decoding of the blocks and decoding of the layers to be done in a largely parallel manner.

4. BITSTREAM STRUCTURE

The LCEVC bitstream contains a base layer, which may be at a lower resolution, and an enhancement layer consisting of up to two sub-layers. The following section briefly explains the structure of this bitstream and how the information can be extracted.

While the base layer can be created using any video encoder and is not specified further in the LCEVC standard, the enhancement layer must follow the structure as specified. Similar to other MPEG codecs [3][4], the syntax elements are encapsulated in network abstraction layer (NAL) units which also help synchronize the enhancement layer information with the base layer decoded information. Depending on the position of the frame within a group of pictures (GOP), additional data specifying the global configuration and for controlling the decoder may be present.

The data of one enhancement picture is encoded into several chunks. These data chunks are hierarchically organized as shown in Fig. 1. For each processed plane (nPlanes), up to two enhancement sub-layers (nLevels) are extracted. Each of them again unfolds into numerous coefficient groups of

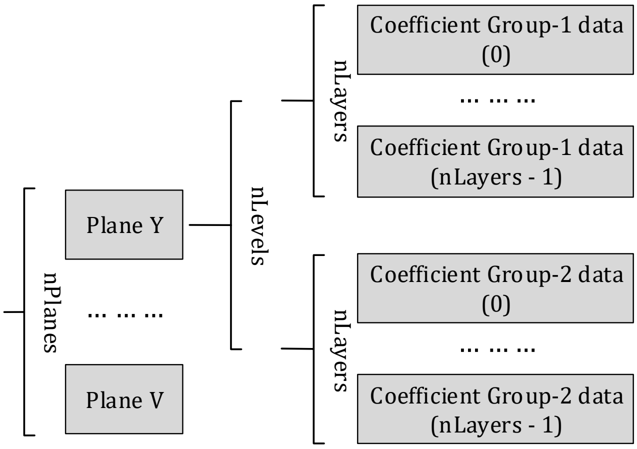


Fig. 1 – Encoded enhancement picture data chunk structure

entropy encoded transform coefficients. The amount depends on the chosen type of transform (nLayers). Additionally, if the temporal prediction is used, for each processed plane an additional chunk with temporal data for Enhancement sub-layer 2 is present.

5. CODING STRUCTURE

5.1 Encoder

The encoding process to create an LCEVC conformant bitstream is shown in Fig. 2 and can be depicted in three major steps.

5.1.1 Base codec

Firstly, the input sequence is fed into two consecutive downscalers and is processed according to the chosen scaling modes. Any combination of the three available options (2-dimensional scaling, 1-dimensional scaling in the horizontal direction only or no scaling) can be used. The output then invokes the base codec which produces a base bitstream according to its own specification. This encoded base is included as part of the LCEVC bitstream.

5.1.2 Enhancement sub-layer 1

The reconstructed base picture may be upscaled to undo the downscaling process and is then subtracted from the first-order downscaled input sequence in order to generate the Layer 1 (L-1) residuals. These residuals form the starting point for the encoding process of the first enhancement sub-layer. A number of coding tools, which will be described further in the following subsection, process the input and generate entropy encoded quantized transform coefficients.

5.1.3 Enhancement sub-layer 2

As a last step of the encoding process, the enhancement data for Layer 2 needs to be generated. In order to create the residuals, the coefficients from Layer 1 are processed by an in-loop decoder to achieve the corresponding reconstructed picture. Since Layer 1 might have a different resolution than the input sequence, the reconstructed picture is processed by an upscaler, again depending on the chosen scaling mode. Finally, the residuals are calculated by a subtraction of the input sequence and the upscaled reconstruction.

Similar to Layer 1, the samples are processed by a few coding tools. Additionally, a temporal prediction can be applied on the transform coefficients in order to achieve a better removal of redundant information. The entropy encoded quantized transform coefficients of Layer 2, as well as a temporal layer specifying the use of the temporal prediction on a block basis, are included in the LCEVC bitstream.

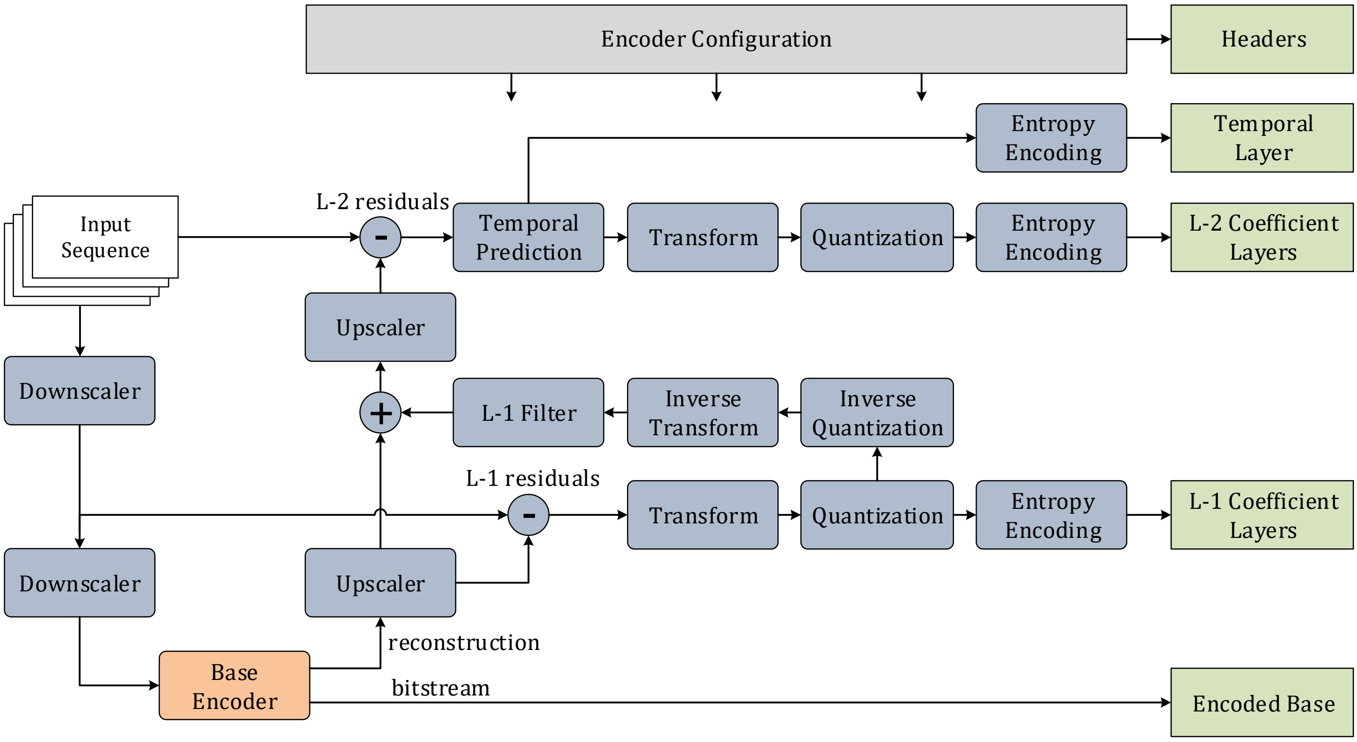


Fig. 2 – Structure of an LCEVC encoder

5.2 Decoder

For the creation of the output sequence, the decoder analyses the LCEVC conformant bitstream. As can be seen in Fig. 3, the process can again be divided into three parts.

5.2.1 Base codec

In order to generate the *Decoded Base Picture (Layer 0)* the base decoder is fed with the extracted base bitstream. According to the chosen scaling mode, this reconstructed picture might be upscaled and is afterwards called *Preliminary Intermediate Picture*.

5.2.2 Enhancement sub-layer 1

Following the base layer, the enhancement part needs to be decoded. Firstly, the coefficients belonging to Sub-layer 1 are decoded using the inverse tools compared to the encoding process. Additionally, an L-1 filter might be applied in order to smooth the boundaries of the transform block. The output is then referred to as *Enhancement Sub-Layer 1* and is added to the preliminary intermediate picture which results in the *Combined Intermediate Picture*. Again, depending on the scaling mode, an upscaler may be applied and the resulting *Preliminary Output Picture* has then the same dimensions as the overall output picture.

5.2.3 Enhancement sub-layer 2

As a final step, the second enhancement sub-layer is decoded. According to the temporal layer, a temporal prediction might be applied to the dequantized transform coefficients. This *Enhancement Sub-Layer 2* is then added to the *Preliminary Output Picture* to form the *Combined Output Picture* as a final output of the decoding process.

6. CODING TOOLS

6.1 Down- and upscaler

Two non-normative downscalers can be used to downscale the input sequence to a lower resolution. The downscaling can be done either in both vertical and horizontal directions, only in the horizontal direction or alternatively cannot be applied. Two upscalers are available reconstructing the sequence at a higher resolution. One of four specified upscaling kernels can be used.

6.2 Transform

LCEVC allows the usage of two different transforms. Both operate with a small kernel of size 2x2 or 4x4. In case the upscaling process is perfomed in the horizontal direction only, the transform kernels are slightly modified to better reflect the preceding 1-dimensional upscaling.

6.3 Quantization

The transform coefficients are quantized using a linear quantizer. The linear quantizer may use a dead zone whose size changes relative to the quantization step.

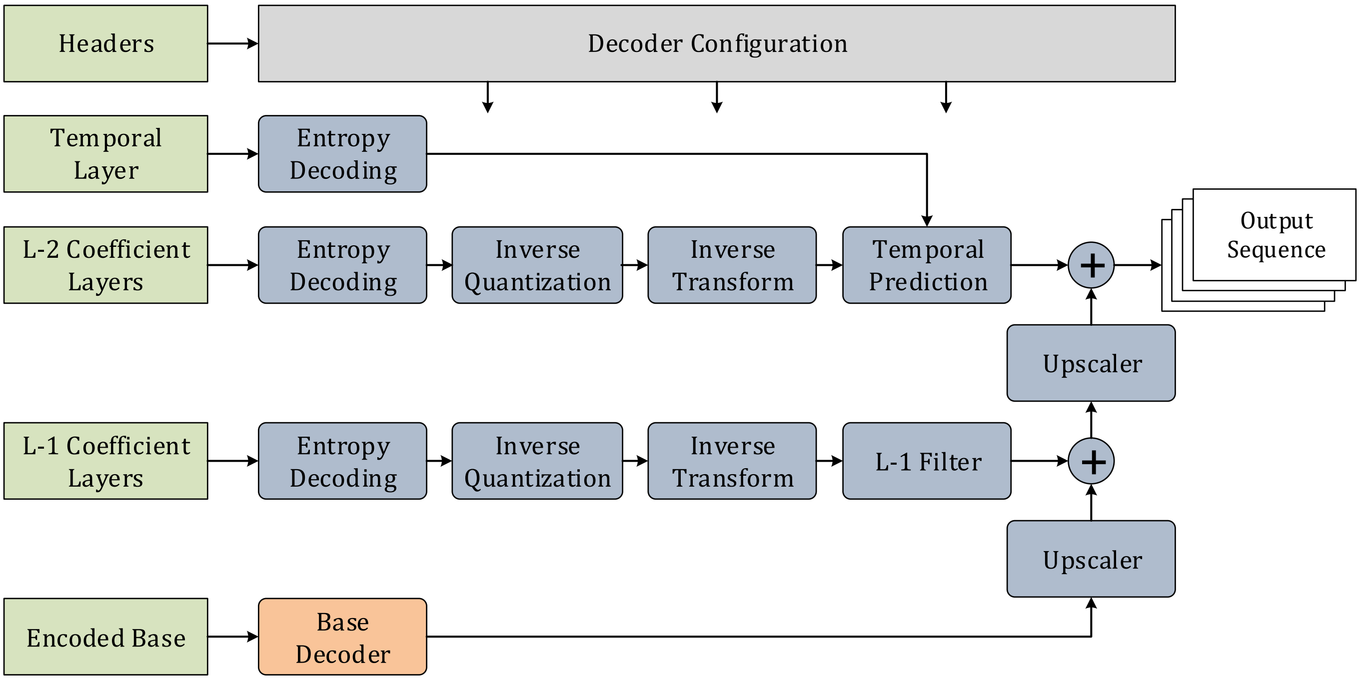


Fig. 3 – Structure of an LCEVC decoder

6.4 L-1 filter

The L-1 filter, whose concept is similar to a simple deblocking filter, can be applied on the L-1 residuals if the transform with the larger kernel size (4x4) is used. The residuals on the outer boundary of the transform block are multilplied with a coefficient between 0 and 1. The value of these coefficients can be signalled independently for edges and corners.

6.5 Temporal prediction

The temporal prediction uses a zero-motion vector prediction with a temporal buffer which stores residuals from the previous frame only. The decision, where to use temporal prediction, is done on a transform block basis. Additionally, an entire tile of 32x32 residuals can be signalled to be used without temporal prediction, reducing the signalling overhead for, e.g., a fast moving part of the sequence.

6.6 Entropy encoding

The two coefficient layers and the temporal layer are processed independently by an entropy encoder before the encapuslation into the bitstream. The entropy coding process consists of two components: a Run Length Encoder (RLE) and a Prefix Coding encoder. Aditionally, it is possible to only use the RLE for the entire data within a coefficient group.

7. PERFORMANCE RESULTS

In order to show that LCEVC is a low-complexity scheme enabling performance improvements over a given base codec, several tests were run on LCEVC using H.264/AVC and H.265/HEVC as video coding standards for the base codec.

7.1 Experimental set-up

The tests have been performed using two different software implementations.

7.1.1 Reference implementation

A first set of tests has been performed using the LCEVC reference implementation in its current version LTM 4.0 [5]. LTM 4.0 uses as a base codec either reference implementation JM 19.0 for H.264/AVC [6] or HM 16.18 for H.265/HEVC [7].

7.1.2 Commercial implementation

A second set of tests has been performed using a commercial LCEVC implementation provided by V-Nova, which uses as a base codec x264 for H.264/AVC and x265 for H.265/HEVC. This set of tests has been performed using a very slow preset for both encoders. The same settings have been used when encoding x264/x265 at full resolution as an anchor and at quarter resolution as a base codec for LCEVC. The default constant rate factor (CRF) quality setting was used for both x264 and x265, both at full resolution and at a quarter resolution.

7.2 Objective and subjective metrics results

The performance results are shown using the two objective metrics Peak Signal-to-Noise Ratio (PSNR) and Video Multimethod Assessment Fusion (VMAF) [8][9]. The latter is operated using the default model (v0.6.1) for HD sequences and the 4K model (4k_v0.6.1) for UHD sequences.

All the results compare a full resolution video encoded using the anchor codec (H.264/AVC or H.265/HEVC) against a full resolution video encoded using LCEVC. When comparing against an H.264/AVC anchor, the LCEVC would use H.264/AVC as the base codec. When comparing against an H.265/HEVC anchor, the LCEVC would use H.265/HEVC as the base codec.

Two video data sets have been used, Set A for HD and Set B for UHD resolutions. The sequences of each data set are listed in Table 1. Each sequence has been tested at four different operating points.

Table 1 – Test sequences

Seq.	Sequence name	Resolution	Frame rate
A1	Campfire	3840x2160	30
A2	ParkRunning3	3840x2160	50
A3	FoodMarket4	3840x2160	60
A4	Fortnite (Part 1)	3840x2160	60
B1	Cactus	1920x1080	50
B2	BasketballDrive	1920x1080	50
B3	RitualDance	1920x1080	60
B4	EuroTruck Simulator	1920x1080	60

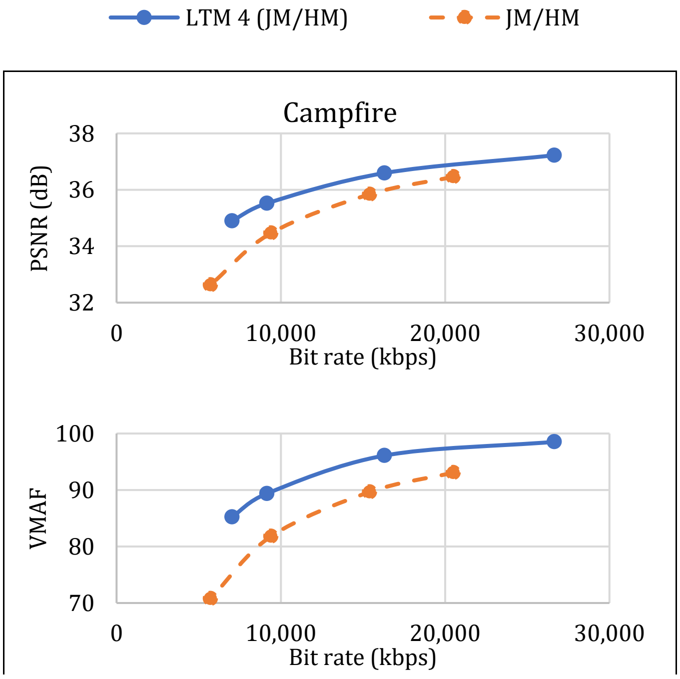
Table 2 provides the average coding performances of LCEVC using the Bjøntegaard metric (BD-rate) [10]. The encodes are performed using the reference implementation as described in Section 7.1.1.

Table 2 – Coding performance comparison of LCEVC (LTM 4.0) over AVC/HEVC anchor (JM/HM)

Video data set (base & anchor codec)	**PSNR**	**VMAF**	**MOS**
Avg. A (H.264/AVC)	−36.46%	−43.44%	−44.98%
Avg. A (H.265/HEVC)	−8.77%	−24.34%	−43.05%
Avg. B (H.264/AVC)	12.40%	−20.68%	−31.98%

Subjective tests using formal Mean Opinion Scores (MOS) according to ITU-R BT.2095 [11] were performed by the GBTech Laboratories under the supervision of Vittorio Baroncini, Chair of MPEG Test Group. The results highlight greater MOS benefits than what has been suggested by objective metrics. The results per sequence are also shown in Table 2 and were calculated using the Bjøntegaard metric. The better correlation of VMAF with subjective MOS results is likely due to VMAF having been designed to assess the visual quality of encodes at different resolutions with the "convex hull" methodology [12], and thus in the presence of spatial scaling. PSNR is known to be less reliable at approximating subjective assessments when comparing different codecs, especially in the presence of scaling [13][14].

Figures 4 to 11 show the rate-distortion (RD) curves for each sequence in the data set A using the reference implementation of LCEVC, as well as H.264/AVC and H.265/HEVC as the base and anchor codecs.



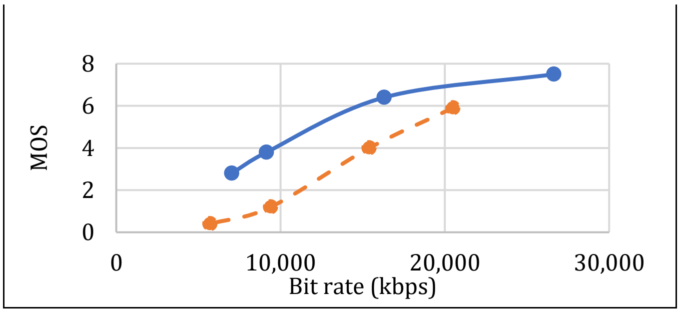


Fig. 4 – RD-curves for sequence A1 using H.264/AVC as base and anchor codec

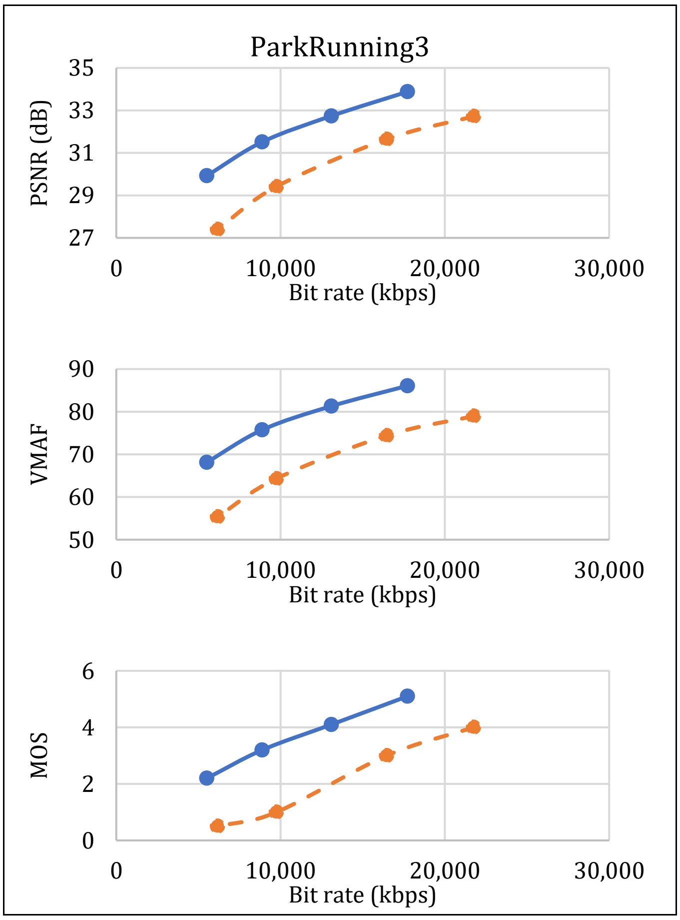
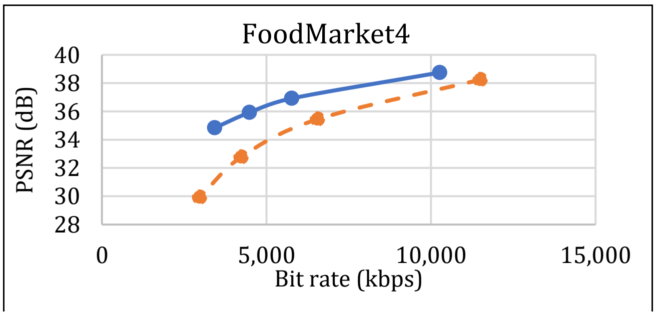


Fig. 5 – RD-curves for sequence A2 using H.264/AVC as base and anchor codec



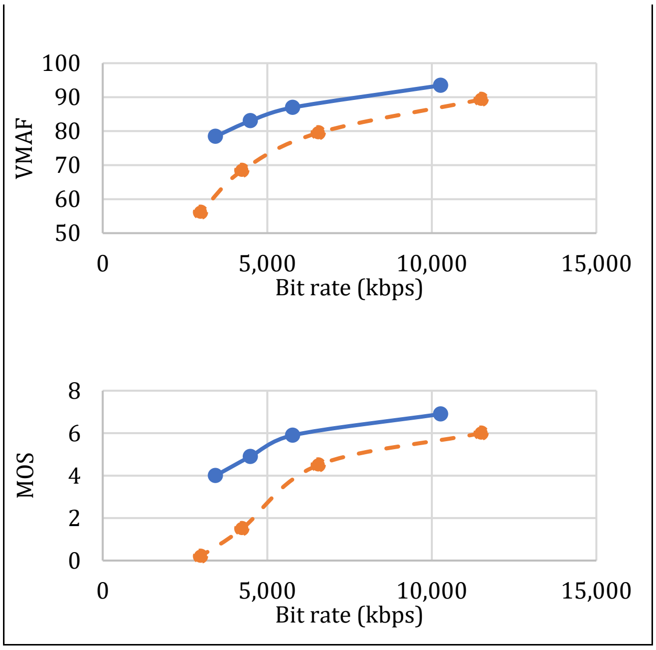


Fig. 6 – RD-curves for sequence A3 using H.264/AVC as base and anchor codec

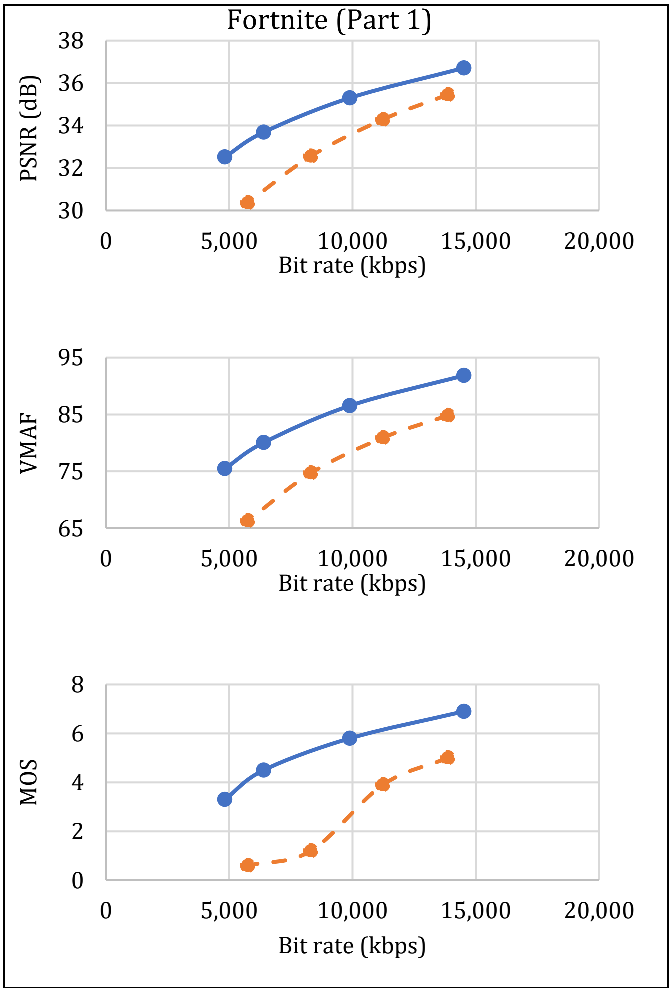


Fig. 7 – RD-curves for sequence A4 using H.264/AVC as base and anchor codec

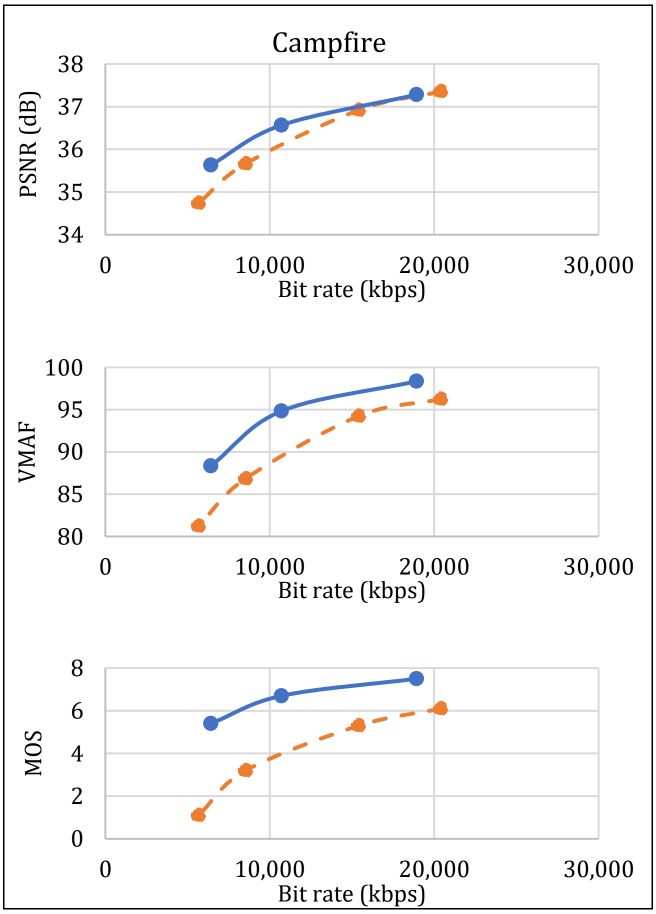
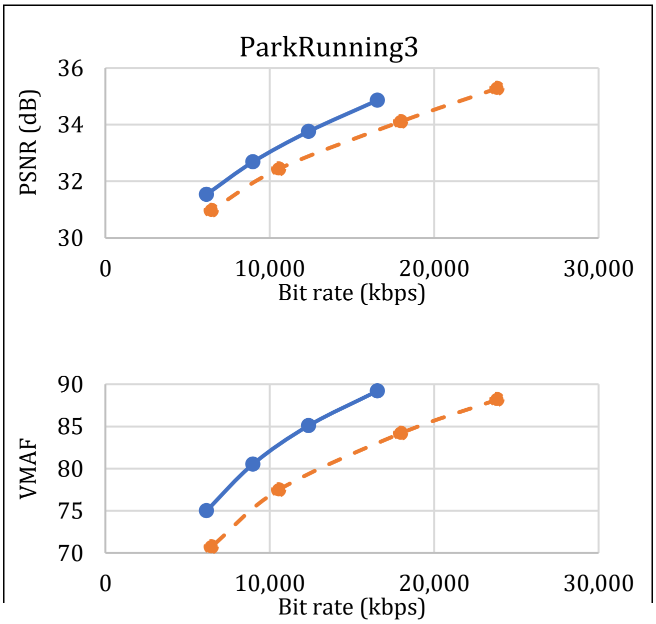


Fig. 8 – RD-curves for sequence A1 using H.265/HEVC as base and anchor codec



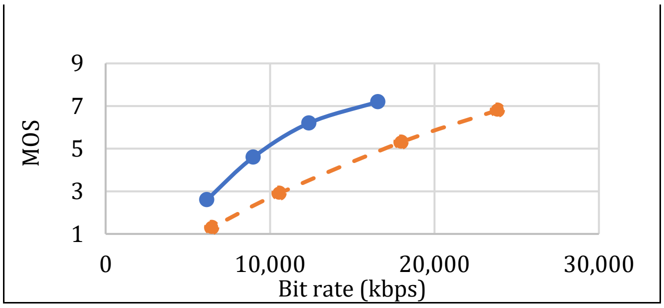


Fig. 9 – RD-curves for sequence A2 using H.265/HEVC as base and anchor codec

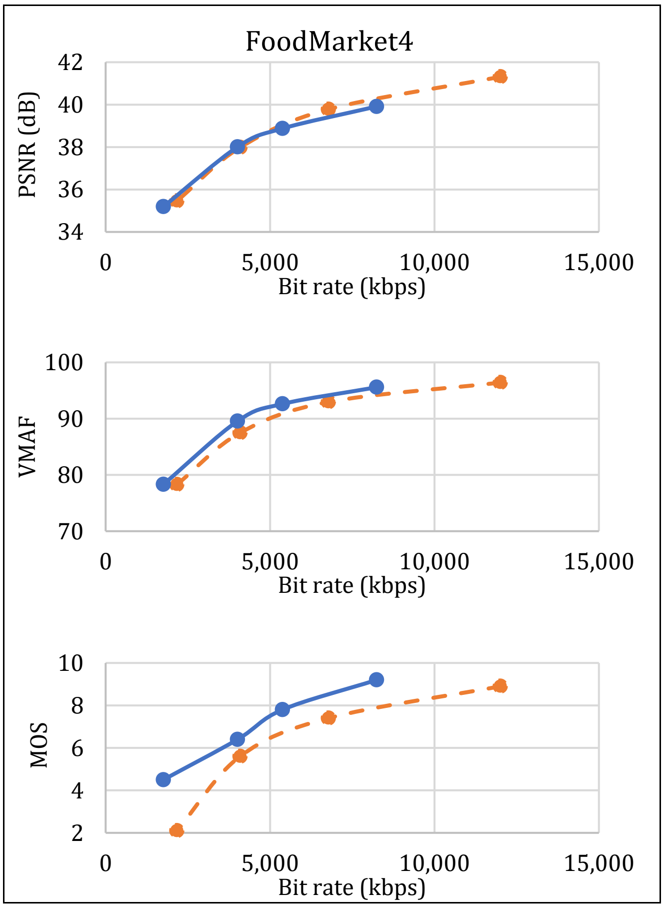
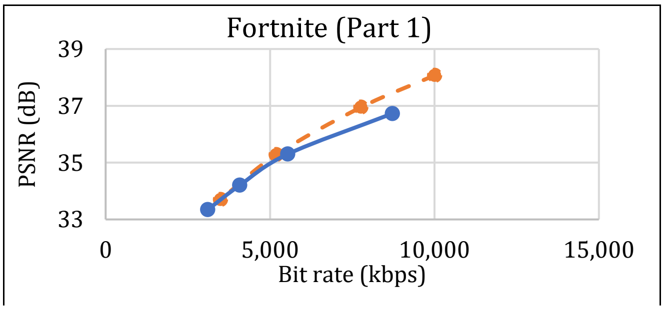


Fig. 10 – RD-curves for sequence A3 using H.265/HEVC as base and anchor codec



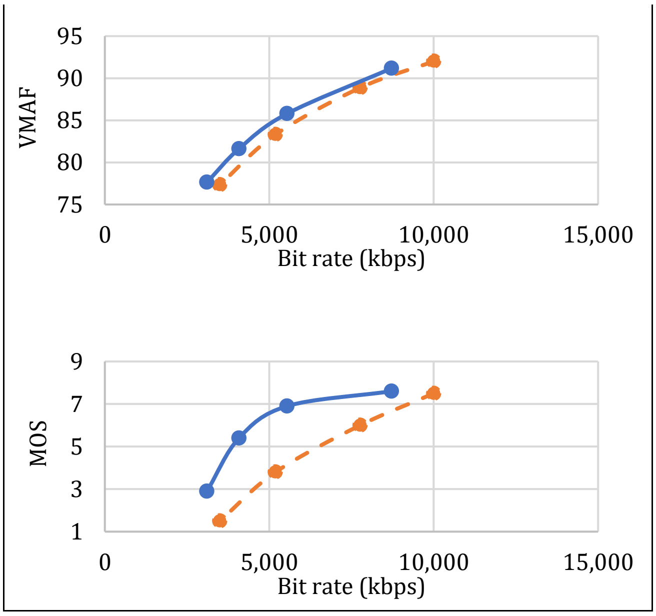


Fig. 11 – RD-curves for sequence A4 using H.265/HEVC as base and anchor codec

Results show that LCEVC provides a significant improvement over the respective anchor codecs, in terms of both objective and subjective metrics.

In addition to the previous experiment, the video data set A has been encoded using the reference implementations of LCEVC, H.264/AVC and H.265/HEVC at higher bit rates. Namely, the following four QPs have been used to encode the anchors: 27, 32, 37 and 42. The resulting BD-rates are reported in Table 3. By way of example, Fig. 12 highlights the RD-curves for objective metrics of the sequence 'ParkRunning3' over the extended bit rate range for H.264/AVC and LCEVC over H.264/AVC. When comparing Table 3 with Table 2 and Fig. 12 with Fig. 5, it can be seen that the performance is consistent also across an extended bit rate range.

Table 3 – Coding performance comparison of LCEVC (LTM 4.0) over AVC/HEVC anchor (JM/HM) – extended bit rate range

Video data set (base & anchor codec)	**PSNR**	**VMAF**
Avg. A (H.264/AVC)	−31.97%	−42.80%
Avg. A (H.265/HEVC)	−5.45%	−23.55%

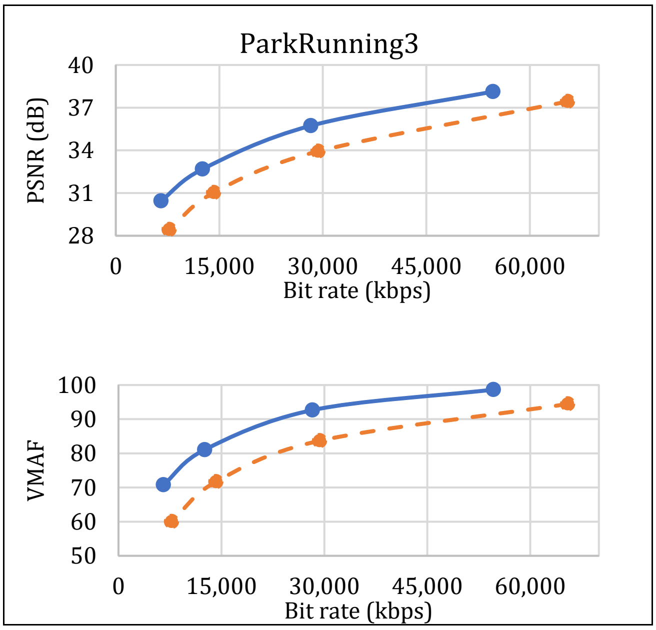


Fig. 12 – RD-curves for sequence A2 using H.264/AVC as base and anchor codec – extended bit rate range

Table 4 gives an overview of the coding performances of LCEVC using the commercial implementation as described in Section 7.1.2. The average BD-rates are calculated for the same two data sets as used in the previous experiment.

Table 4 – Coding performance comparison of LCEVC (commercial implementation) over AVC/HEVC anchor (x264/x265)

Video data set (base & anchor codec)	**PSNR**	**VMAF**
Avg. A (H.264/AVC)	−41.54 %	−44.91 %
Avg. A (H.265/HEVC)	−15.30 %	−21.34 %
Avg. B (H.264/AVC)	−13.94 %	−30.93 %

Table 5 – Coding performance comparison of LCEVC (commercial implementation) over AVC/HEVC anchor (x264/x265), variable bit rate (crf)

El Fuente HD data set (136 sequences)		**PSNR**	**VMAF**
H.264/ AVC	% of sequences with LCEVC BD-rate < 0%	96%	100%
	Average LCEVC BD-rate	−33.09%	−48.67%
	Median LCEVC BD-rate	−36.99%	−49.10%
H.265/ HEVC	% of sequences with LCEVC BD-rate < 0%	61%	99%
	Average LCEVC BD-rate	0.52%	−29.54%
	Median LCEVC BD-rate	−8.52%	−30.27%

To further validate LCEVC benefits with a larger data set, Table 5 includes the results achieved when encoding the Netflix El Fuente test set (which includes 136 video sequences) at 1080p, using x264/x265 with the "veryslow" preset as anchor and base encoders. LCEVC provides BD-rate benefits for the vast majority of clips (100% and 99% for VMAF, 96% and 61% for PSNR), as well as average and median BD-rates consistent with the previous tests, confirming the statistical significance of results.

The sequences from the El Fuente test set have been further encoded at different resolutions to form a boundary called convex hull [15]. The resulting RD-curves for two exemplary sequences are shown in Fig. 13 and Fig. 14. Specifically, sequence #125 is a sequence in the median range, whereas sequence #54 is a sequence above the median range. The light blue and red curves each indicate an encoding at a different resolution at six operating points using LCEVC and x264, respectively. The resolutions span from 360p up to 1080p. Based on these individual encodings, the convex hull is generated. It is shown in a dark blue or red color depending on the underlying codec. When comparing the convex hull of LCEVC and x264, VMAF BD-rate differences of −39.33% for sequence #54 and −44.32% for sequence #125 were achieved.

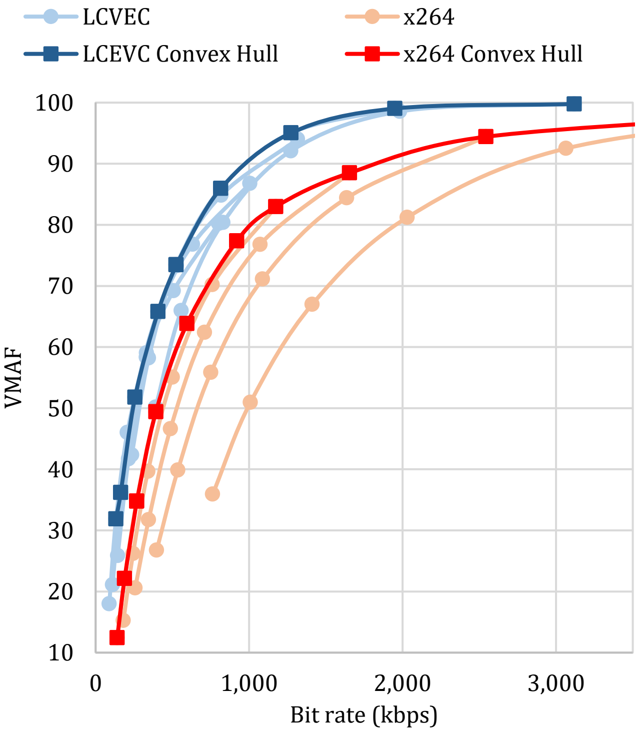


Fig. 13 – RD-curve showing the convex hull of LCEVC and x264 for El Fuente sequence #54

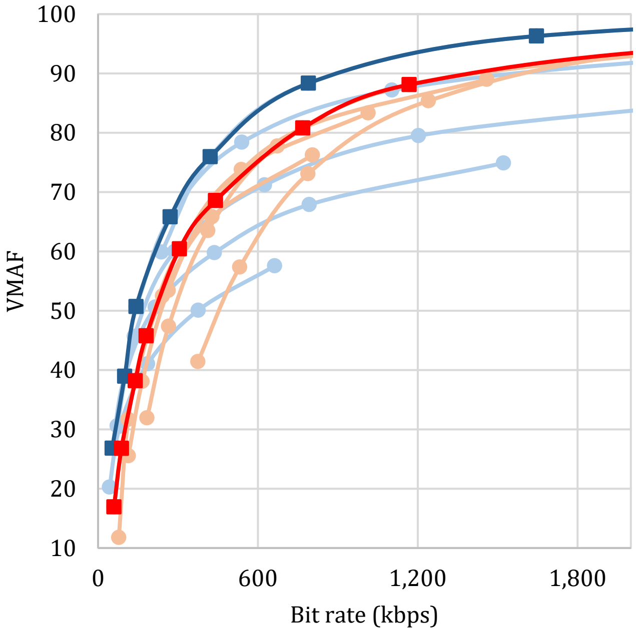


Fig. 14 – RD-curve showing the convex hull of LCEVC and x264 for El Fuente sequence #125

Fig. 15 compares two cropped screenshots taken from the above-metioned exemplary sequence #54. The left one shows an encoding using x264 at a bit rate of 2654 kbps while the image on the right was encoded using LCEVC at a bit rate of 2051 kbps.

Fig. 15 – Cropped screenshots from an exemplary El Fuente sequence (left: x264 @ 2654 kbps, right: LCEVC @ 2051 kbps)

7.3 Processing time performances

Processing complexity considerations based upon encoding/decoding times are best made on real-world implementations, since reference implementations have received diverse levels of code optimization: for instance, the HM encoder (reference implementation of HEVC) is faster than the JM encoder (reference implementation of AVC), despite HEVC actually being a more complex codec than AVC and real-world HEVC implementations being slower than real-world AVC implementations. It should be noted that the current LTM 4.0 has not been optimized to improve processing time, particularly at the decoder side.

Accordingly, processing times for LCEVC were measured using the commercial implementations of LCEVC, H.264/AVC and H.265/HEVC, as described in Section 7.1.2. The encodes and decodes have been performed on a common platform (Intel i9-8950HK @ 2.9GHz).

For each full resolution, the same sequences mentioned in Section 7.2 were used.

Table 6 reports the average timings for each resolution for both anchors and LCEVC.

Table 6 – Relative encoding and decoding times for LCEVC vs. anchors (anchor ≙ 100%)

Base & anchor codec	Resolution	Encoder time	Decoder time
H.264/AVC	UHD	32.99%	81.88%
H.265/HEVC	UHD	34.44%	64.24%
H.264/AVC	HD	51.48%	96.72%

As can be seen, the encoding time for LCEVC is between circa 30% and 50% of the encoding time required for the anchors depending on base encoder and resolution. On the decoding side, LCEVC requires between circa 60% and 95% of the decoding time required for the anchors depending on base decoder and resolution. The low complexity of LCEVC allows power-efficient implementations of the codec via software, also at relatively high levels of the software stack. As discussed in Section 3.5, LCEVC processing is highly parallelizable due to certain characteristics of the scheme. The tools are designed to minimize the number of operations required as well as the interdependency between them, making efficient use of available general-purpose hardware acceleration, including SIMD, GPUs or DSPs, either alternatively or in conjunction.

8. CONCLUSION

The results in this paper confirm that LCEVC successfully achieves the objectives set-out in the MPEG requirements document [16], namely that:

– when enhancing an n-th generation MPEG codec (e.g., AVC), compression efficiency for the aggregate stream is appreciably higher than that of the n-th generation MPEG codec used at full resolution and as close as possible to that of the (n+1)-th generation MPEG codec (e.g., HEVC) used at full resolution, at bandwidths and operating conditions relevant to mass market distribution; and

– encoding and decoding complexity for the aggregate full resolution video (i.e., base plus enhancement) shall be comparable with that of the base encoder or decoder, respectively, when used alone at full resolution.

LCEVC requirements were driven by commercial needs put forward to MPEG by many leading industry experts, highlighting the imporance of making available a similar coding tool.

As such, LCEVC seems capable of satisfying important commercial needs and, in combination with its relative ease of deployment, have a rapid impact on multiple segments of the video delivery landscape.

REFERENCES

[1] "Draft Text of ISO/IEC DIS 23094-2, Low Complexity Enhancement Video Coding", ISO/IEC JTC1/SC29/WG11 N18986, Brussels, BE, Jan. 2020.

[2] "Market needs and requirements for software-based capability extension on top of existing and future video codecs", in ISO/IEC JTC1/SC29/WG11 M44832, Macao, CN, Oct. 2018.

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

[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand (2013), "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, Jan.

[5] "LTM: Test model of low complexity enhancement video coding", MPEG Git, 2020. [Online]. Available: http://mpegx.int-evry.fr/software/MPEG/LCEVC/LTM.

[6] K. Suehring, "H.264/AVC Software", Fraunhofer Heinrich Hertz Institute. [Online]. Available: http://iphome.hhi.de/suehring/.

[7] "High Efficiency Video Coding (HEVC)", Fraunhofer Heinrich Hertz Institute. [Online]. Available: https://hevc.hhi.fraunhofer.de/.

[8] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara (2016), "Toward A Practical Perceptual Video Quality Metric", Netflix Technology Blog, 06 Jun. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[9] Z. Li et al. (2018), "VMAF: The Journey Continues", Netflix Technology Blog, 25 Oct. [Online].
Available: https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12.

[10] G. Bjøntegaard (2001), "Calculation of average PSNR differences between RD-curves", ITU-T VCEG-M33.

[11] Recommendation ITU-R BT.2095-1, (2017), "Subjective assessment of video quality using expert viewing protocol".

[12] I. Katsavounidis and L. Guo (2018), "Video codec comparison using the dynamic optimizer framework," Applications of Digital Image Processing XLI, vol. 10752, Sep.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004), "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr.

[14] J.-S. Lee, F. De Simone, T. Ebrahimi, N. Ramzan, and E. Izquierdo (2012), "Quality assessment of multidimensional video scalability," IEEE Communications Magazine, vol. 50, no. 4, pp. 38–46, Apr.

[15] A. Aaron, Z. Li, M. Manohara, J. De Cock and D. Ronca (2015), "Per-Title Encode Optimization", Netflix Technology Blog, 14 Dec. [Online].
Available: https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2.

[16] "Requirements for Low Complexity Video Coding Enhancements", ISO/IEC JTC1/SC29/WG11 N18098, Macao, CN, Oct. 2018.