Overview of the Low Complexity Enhancement Video Coding (LCEVC) Standard

Stefano Battista, Guido Meardi, Simone Ferrara, Lorenzo Ciccarelli, Florian Maurer, Massimo Conti, *Member, IEEE*, and Simone Orcioni, *Senior Member, IEEE*

***Abstract*— The Low Complexity Enhancement Video Coding (LCEVC) specification is a recent standard approved by the ISO/IEC JTC 1/SC 29/WG04 (MPEG) Video Coding. The main goal of LCEVC is to provide a standalone toolset for the enhancement of any other existing codec. It works on top of other coding schemes, resulting in a multi-layer video coding technology, but unlike existing scalable video codecs, adds enhancement layers completely independent from the base video. The LCEVC technology takes as input the decoded video at lower resolution and adds up to two enhancement sub-layers of residuals encoded with specialized low-complexity coding tools, such as simple temporal prediction, frequency transform, quantization, and entropy encoding. This paper provides an overview of the main features of the LCEVC standard: high compression efficiency, low complexity, minimized requirements of memory and processing power.**

***Index Terms*— LCEVC, MPEG, video coding, multi-layer video coding, low complexity.**

I. Introduction

THE Low Complexity Enhancement Video Coding (LCEVC) is a Video Coding standard finalized in November 2021 by the ISO/IEC JTC 1 working group formerly known as SC29/WG11 (MPEG) and currently as SC29/WG04 (MPEG Video Coding). The specification is officially named ISO/IEC IS 23094-2, also identified as MPEG-5 Part 2 [1]–[3].

The video coding standards have a history of over 30 years, with two main organizations active since the late 1980s decade: ITU-T SG16 (VCEG) and ISO/IEC JTC1 SC29 WG11 (MPEG) until 2020 and WG04 (MPEG Video Coding) since 2020. VCEG developed the recommendations H.261 in 1988 [4], and H.263 in 1996 [5], [6]. MPEG developed the International Standards MPEG-1 in 1993 [7], [8], MPEG-4 in 1999 [9], [10], and Essential Video Coding (EVC) in 2020 [11], [12]. Through the establishment of joint teams, including experts from both MPEG and VCEG, the two organizations developed also joint specifications, where the text is officially approved by both ISO as International Standard and ITU-T as Recommendation: MPEG-2 also known as H.262 in 1996 [13], [14], Advanced Video Coding (AVC) a.k.a. H.264 in 2003 [15], [16], High Efficiency Video Coding (HEVC) a.k.a. H.265 in 2013 [17], [18], and Versatile Video Coding (VVC) a.k.a. H.266 in 2021 [19], [20].

All these specifications represent three decades of evolution of video coding algorithms with the explicit goal of achieving ever increasing compression while maintaining the same subjective quality for the user. They share a common structure, since all of them use blocks of samples, either of fixed size or adaptive size depending on the picture content, and all of them exploit the spatial and temporal redundancy (intra and inter predictive coding), the prediction residual redundancy (transform coding), and finally the statistical redundancy (entropy coding).

In all the video coding standards listed above, from H.261 in 1988 to VVC in 2021, the bitstream is composed of a single-layer, since all pictures and all blocks forming a picture are processed and encoded in a single bitstream. VVC specifies from its first edition the possibility to use the multi-layer approach, but can still be used as a single-layer video codec, as specified in the Main 10 and Main 10 4:4:4 profiles, [19] Annex A.

Starting with MPEG-2/H.262 (1996) and H.263 (1996), multi-layer extensions, also denoted as scalable extensions, to the single-layer video coding algorithms have been developed. However, the most relevant specifications for scalable video coding were developed with AVC/H.264, named Scalable Video Coding (SVC) [21], defined in AVC Annex G, and finalized in 2008, and with HEVC/H.265, named Scalable High Efficiency Video Coding (SHVC) [22], defined in HEVC Annex H, and finalized in 2015. The main feature introduced with scalable video coding, as in SVC and SHVC, is the possibility to partition the video bitstream into several subsets, separating two or more layers in the temporal dimension (with a first layer at lower frame rate, e.g. 30 fps, and a second layer at higher frame rate, e.g. 60 fps), in the spatial dimension (e.g. with a first layer at lower resolution, e.g. 1920 × 1080, and a second layer at higher resolution, e.g. 3840 × 2160), or in the quality dimension (with a first layer at lower quality, and a second layer at higher quality). In all cases, the lower layer is a complete bitstream, sufficient to decode the video sequence at a lower quality, whatever the type of scalability: temporal, spatial, or quality scalability.

LCEVC is not designed for being an alternative to other existing and emerging video coding standards, like AVC,

Manuscript received 31 January 2022; revised 4 May 2022 and 30 May 2022; accepted 9 June 2022. Date of publication 13 June 2022; date of current version 28 October 2022. This work was supported in part by the Università Politecnica delle Marche and in part by V-Nova. This article was recommended by Associate Editor T. Lu. *(Corresponding author: Simone Orcioni.)*

Stefano Battista, Massimo Conti, and Simone Orcioni are with the Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy (e-mail: s.battista@univpm.it; m.conti@univpm.it; s.orcioni@univpm.it).

Guido Meardi, Simone Ferrara, Lorenzo Ciccarelli, and Florian Maurer are with V-Nova, London W2 6LG, U.K. (e-mail: guido.meardi@v-nova.com; simone.ferrara@v-nova.com; lorenzo.ciccarelli@v-nova.com; florian.maurer@v-nova.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3182793.

Digital Object Identifier 10.1109/TCSVT.2022.3182793

HEVC, EVC, or VVC, but rather for being a standalone toolset for enhancement of any other existing video codec. This is achieved working on top of other coding schemes, encoding the residual differences between a lower quality encoding and the original video. The LCEVC technology typically takes as input the decoded video at lower resolution and adds up to two enhancement sub-layers of residuals encoded with specialized low-complexity coding tools. Thus, LCEVC can be classified as a multi-layer video coding technology, but the main difference with existing scalable ones, like SVC and SHVC, lies in the fact that the added enhancement layers are completely independent from the base video. The most similar approach to LCEVC is represented by the use of an "external base layer" as defined in SHVC, [17] Annex H, where the enhancement layer specified by SHVC can be applied to a base layer specified by AVC, [15]. LCEVC generalizes this approach, allowing any base layer to be enhanced using the same LCEVC technology, completely agnostic of the codec used for the base video bitstream.

By design, LCEVC is based on a set of tools with low complexity, and it is intended to be efficiently and effectively implemented in software via existing hardware blocks in existing devices, such as Single Instruction, Multiple Data (SIMD) processors and Graphics Processing Units (GPU), for scaling and/or shading. With these design choices, LCEVC achieves a trade off in terms of optimal Rate Distortion (RD) performance, energy saving, and ease of implementation. The numerical results of such trade off, in terms of RD performance, are described in Section IX.

LCEVC has recently gained attention in the industrial and scientific community concerning its characteristic of improvement of the current codecs [23]–[25].

In [24] the authors report a comparison of LCEVC to AVC (in its implementation x264) and HEVC (in its implementation x265), when applied to High Dynamic Range (HDR) video sequences. The paper includes results for bitrate savings for LCEVC compared to the base codec, using the metrics PSNR, MS-SIM, and VMAF versus Bitrate.

In [25] the authors report a comparison of LCEVC to AVC (in its implementation x264) and HEVC (in its implementation x265), in the context of Live Gaming Video Streaming applications. The paper provides results for bitrate savings for LCEVC with respect to the base codec, in terms of PSNR, VMAF, and MOS versus Bitrate.

In [23] the authors provide an overview of the LCEVC specification and a comparison with AVC, HEVC, and VVC, in terms of PSNR, VMAF, and MOS versus Bitrate. The paper provides also an analysis of the complexity, reporting the encoding times of LCEVC with the base codec at quarter resolution versus the encoding times of the base codec at full resolution. Finally, the paper provides an analysis of the (low) correlation of bitrate savings with temporal complexity and the (high) correlation of bitrate savings with spatial complexity of the video sequences.

This paper is organized as follows. Section II provides a high level description of the LCEVC encoder, decoder, and bitstream structure. Sections III to VII analyze in detail the single processing blocks, namely: Upscaler, Predicted Residual, Temporal Prediction, Transformation, Quantization, and Entropy Coding. Section VIII provides a comparison of the complexity, in terms of processing time, of LCEVC with the base codec at quarter resolution versus the base codec at full resolution. Section IX reports results of LCEVC in terms of Rate Distortion performance. Finally, Section X summarizes the conclusions of the paper.

The following notation is used throughout the paper: bold letters are used for vectors and matrices, $|\ |$ indicates the magnitude operator, $\lfloor\ \rfloor$ the floor operator, $\mathrm{sgn}(\ )$ the sign function, $*$ the convolution, $\cdot$ the matrix product, and $\max(,)$ returns the maximum between the two arguments.

II. Overview

This section gives a high level description of the building blocks of the LCEVC video coding technology, for the encoder, the decoder, and the bitstream format.

The design of LCEVC foresees up to two sub-layers of enhancement to a base layer compressed video representation. The first layer (sub-layer 1) is optional and can be disabled by proper signaling in the LCEVC bitstream, while the second layer (sub-layer 2) is mandatory. Although the number of layers could be greater than two, the choice of using only one or two enhancement layers is based on empirical studies that show that adding further layers does not improve the overall performance of the multi-layer scheme, comparing the additional complexity introduced to the additional compression achieved.

A. Encoder

The general structure of an LCEVC encoder is depicted in Fig. 1. The encoding process can be divided into three main steps.

Firstly, the input sequence is downscaled using a non-normative downscaler. Depending on the chosen configuration, the downscaling can be applied up to two consecutive times. The video, now at a lower resolution than the input sequence, is fed into the base encoder (e.g., AVC, HEVC, EVC, VVC). This process is not further specified in LCEVC: any encoder that produces a decodable bitstream can be used. As explained in Section II-C, the base bitstream is included in the LCEVC bitstream. Using a normative upscaler, which allows the use of different and content-adaptive kernels as well as the optional encoder-signaled activation of a non-linear corrector called Predicted Residual, the upscaled base reconstruction is used as the input for the second step of the LCEVC encoding process. The enhancement sub-layer 1 (L-1) residuals are created by subtracting the downscaled input sequence and the base reconstruction. These residuals, which are typically sparse (e.g., sharp edges or fine details), are transformed, quantized and entropy encoded resulting in coefficient groups as discussed in Section II-C.

Some processing blocks specified in LCEVC take as input a small square block of samples, with size of either 2×2 or 4×4 samples: Temporal Prediction, Transform. The choice of the 2×2 or 4×4 block size can be done at the Encoder side, and signaled in the bitstream. Other processing blocks are

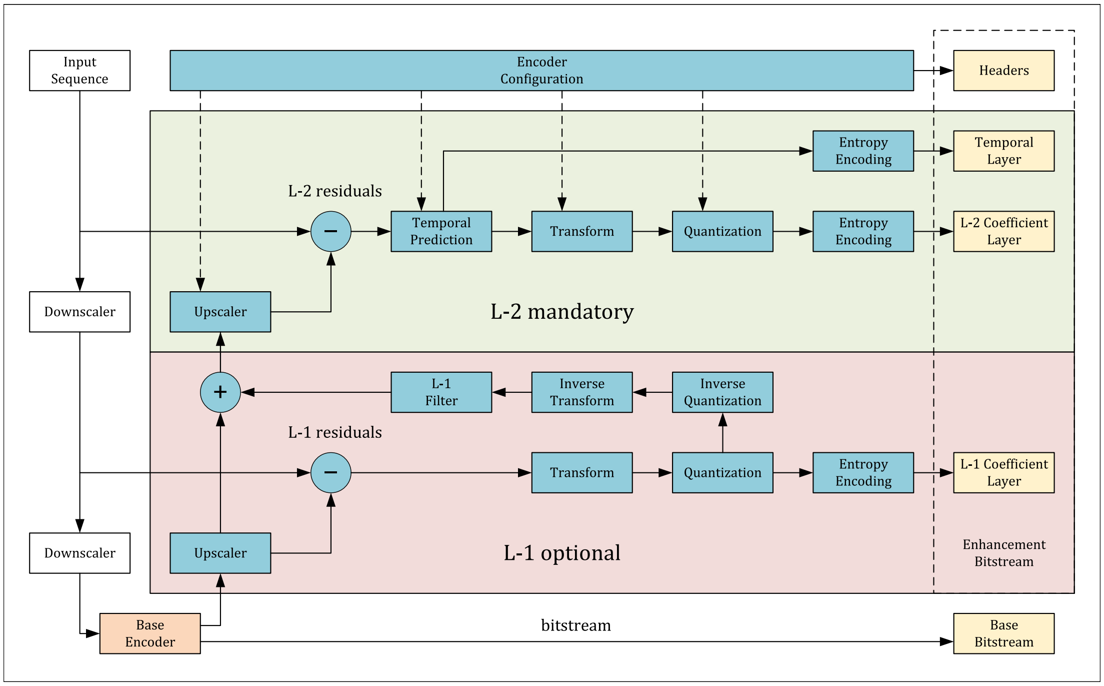


Fig. 1. Encoder block diagram.

applied directly to the whole matrix of samples or coefficients: Downscaler and Upscaler, Quantization, Entropy Coding.

The transform used in LCEVC has a simple structure and uses a small kernel, of size either 2×2 or 4×4. This allows to both efficiently code sparse information and parallelize the transforms, since individual blocks are not dependent on other blocks within a picture.

A linear quantizer, which may include an adaptive deadzone, is used to further process the transform coefficients.

The entropy encoder, which consists of a run-length encoder (RLE) and an optional prefix encoder (Huffman encoder), processes the quantized transform coefficients, and creates the coefficient groups for sub-layer 1.

The inverse processes of the quantization and transform are applied, with adaptive and optionally asymmetric dequantization, creating the sub-layer 1 reconstruction. Additionally, an L-1 filter can be added, which operates as a simple deblocking filter.

Finally, the sub-layer 1 reconstruction is upscaled to full resolution and subtracted from the original input sequence.

The such created enhancement sub-layer 2 (L-2) residuals are fed into the temporal prediction algorithm. LCEVC uses a zero-motion vector temporal scheme which operates on a block-by-block basis. The residuals from the previous picture are stored in a temporal buffer and are added to the L-2 residuals in case the temporal prediction module is activated. To reduce the signaling overhead in e.g. a fast-moving sequence, where this zero-motion vector scheme would likely not be beneficial, the temporal prediction can be disabled for a group of samples of size 32×32, as well as for an entire picture with a single bit. The temporal signaling, containing the information whether the temporal prediction is active for a specific transform block, is entropy encoded and included as a temporal layer in the LCEVC bitstream.

The sub-layer 2 residuals (after the temporal prediction is applied, when appropriate) are transformed, quantized, and encoded using the same tools as explained for enhancement sub-layer 1. The quantizer can use different quantization parameters for the two enhancement sub-layers L-1 and L-2, allowing to balance the impact of the two layers and to decide where to add more details.

The configuration of the Encoder and Decoder requires a set of parameters. Such parameters are set at the Encoder side, included in binary format in the Bitstream, and extracted by the Decoder. These parameters are handled by the Encoder Configuration and Decoder Configuration blocks, depicted graphically in Fig. 1 and Fig. 2. Besides the static information, like picture size, picture rate, bits per sample, the configuration includes parameters regarding the decision whether to use both sub-layer 1 and 2 or to use exclusively sub-layer 2, the block size for Temporal Prediction and Transform, the decision whether to use or not the L-1 filter, plus other configurable parameters that assume a default value if not explicitly specified in the Bitstream.

B. Decoder

The functionality of a normative LCEVC decoder is standardized in [1], and its structure is visualized in Fig. 2. As in the case of the encoder, described in the previous Section, three main steps are visible: the base decoding and the corrections in enhancement sub-layers 1 and 2.

The decoding process is highly parallelizable, and amenable to both SIMD- and GPU-accelerated processing. The base video decoder is independent from the LCEVC enhancement

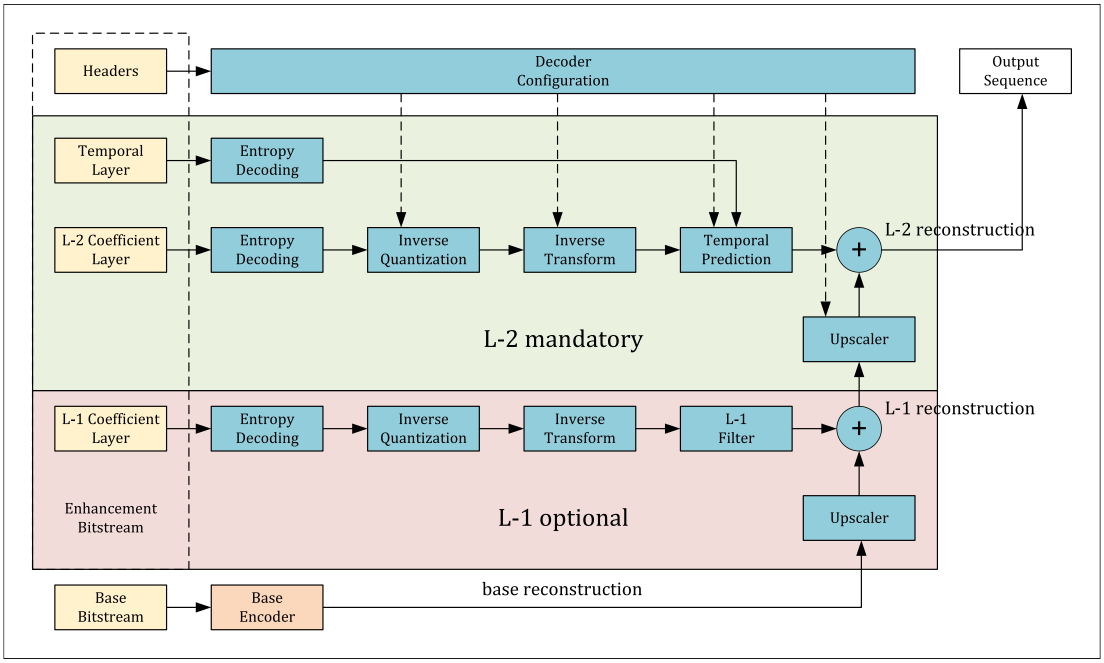


Fig. 2. Decoder block diagram.

part. Additionally, the two enhancement sub-layers can be reconstructed in parallel as well. No inter-block dependencies are present within a picture, making the inverse transform blocks independent from other blocks. If the temporal prediction is active, the enhancement sub-layer 2 reconstruction is dependent on the temporal buffer which stores residuals from the previous picture.

Using the same (or inverse) tools that have been described in Section II-A, the reconstructions from the three main decoding steps are achieved. Those are combined using two normative upscaling filters and additions resulting in the decoded output sequence.

C. Bitstream Structure

An LCEVC encoded bitstream is formed of two separate bitstreams, namely a base bitstream produced by a base encoder conformant to its associated specification (e.g. AVC, HEVC, EVC, VVC) and an enhancement bitstream (the LCEVC bitstream) produced by the enhancement encoder and conformant to the specification ISO/IEC 23094-2.

The LCEVC bitstream is structured in a specific order, depending on the chosen configuration parameters. A simplified structure is visualized in Fig. 3. The LCEVC encoder can be set to enhance all the three available planes (luma and chroma) or the luma plane, only. Within each encoded plane, up to two layers can be present. The first layer (enhancement sub-layer 1, L-1) is used to encode transformed and quantized residuals before applying the final upscaling. The second layer (enhancement sub-layer 2, L-2) is then added after the final upscaling, meaning at the same resolution as the overall output sequence. Each of these two sub-layers is split into coefficient groups. Depending on the chosen transform type, with a kernel size of 2×2 or 4×4 samples, either 4 or 16 coefficients groups are present within a sub-layer. Each coefficient group contains the corresponding transform coefficients.

Furthermore, temporal signaling is added as an additional coefficient group to enhancement sub-layer 2 if the temporal prediction is active. This group contains information whether residuals from the previous picture, stored in a temporal buffer, are used for prediction on a block-by-block basis.

III. Upscaler

The first processing step on the output pictures of the base decoder is the upscaling from the lower resolution at which the base decoder operates: for instance the base decoder may operate at 1920×1080 resolution while the LCEVC encoder operates at 3840×2160 resolution.

The upscaling process described in this section is based on two design choices, adopted to keep the implementation at low complexity, even in software, while at the same time providing a good quality of the upscaled picture.

The first design choice is limiting to 2 the upscaling factor in each dimension of the picture, to reduce the complexity of up-sampling filters.

The second design choice is limiting to 4 the number of taps in the poly-phase implementation of up-sampling, which was found to be a good compromise between computational complexity and picture quality.

Given the impulsive response of a one-dimensional up-sampling filter as $\boldsymbol{h} = [h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7]^T$, the vector representation of the up-sampling operation can be written as

$$\boldsymbol{F} = \boldsymbol{H} * ((\uparrow 2)\boldsymbol{B}) \tag{1}$$

where the $(\uparrow 2)$ operator is a dyadic up-sampling operator, and $\boldsymbol{H} = \boldsymbol{h} \cdot \boldsymbol{h}^T$ is the two-dimensional kernel. To avoid

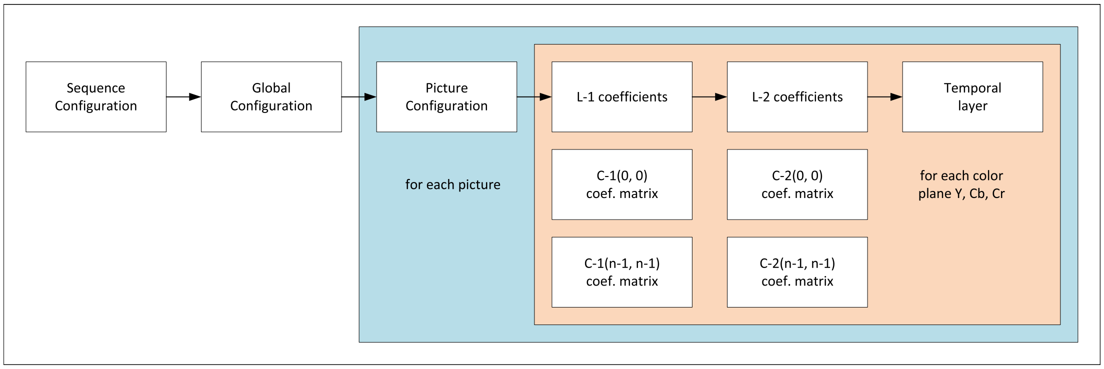


Fig. 3. LCEVC Bitstream structure.

the unnecessary multiplications of kernel coefficients by the input zeros inserted by the up-sampling operator, the multirate Noble identities can be used, obtaining the polyphase implementation of up-sampling operations. The kernel coefficients are divided according to their even or odd index position: $\boldsymbol{h}_\mathrm{e} = [h_0, h_2, h_4, h_6]$, $\boldsymbol{h}_\mathrm{o} = [h_1, h_3, h_5, h_7]$. Furthermore, since in our case the choice of coefficients is restricted to symmetrical kernels, i.e. $\boldsymbol{h}_\mathrm{o} = [k_0, k_1, k_2, k_3]$; $\boldsymbol{h}_\mathrm{e} = [k_3, k_2, k_1, k_0]$;, the poly-phase implementation can be realized with the cascade of mono-dimensional filtering as follows. The first pass is the horizontally up-sampling of the picture by a factor two

$$\begin{aligned} \boldsymbol{B}_\mathrm{e} &= (\uparrow 2)_{\mathrm{c\,e}} \left(\boldsymbol{h}_\mathrm{e}^T * \boldsymbol{B}\right) \\ \boldsymbol{B}_\mathrm{o} &= (\uparrow 2)_{\mathrm{c\,o}} \left(\boldsymbol{h}_\mathrm{o}^T * \boldsymbol{B}\right) \\ \boldsymbol{M} &= \boldsymbol{B}_\mathrm{e} + \boldsymbol{B}_\mathrm{o} \end{aligned} \tag{2}$$

where $(\uparrow 2)_{\mathrm{c\,e,o}}$ is a dyadic upsampling operator that inserts zeros at the even or odd positions in columns.

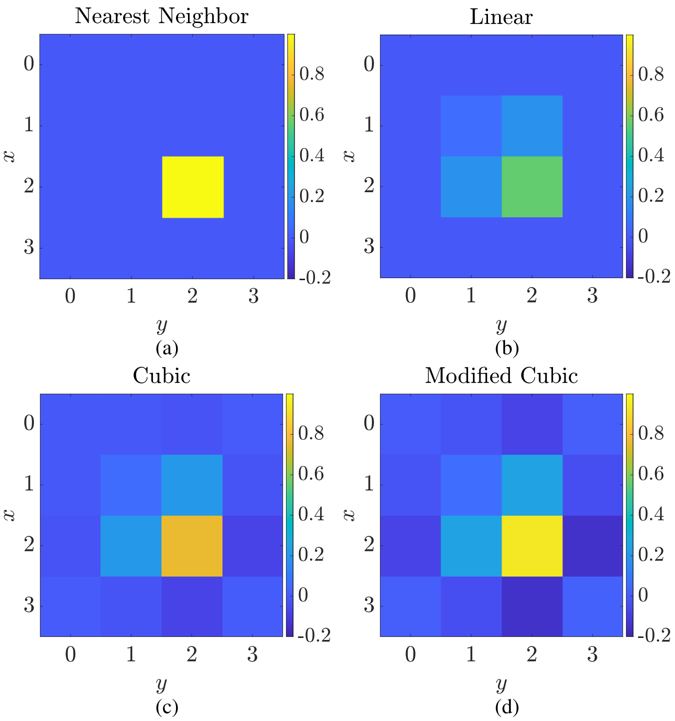
The second pass consists of vertically up-sampling the picture by a factor two,

$$\begin{aligned} \boldsymbol{M}_\mathrm{e} &= (\uparrow 2)_{\mathrm{r\,e}} \left(\boldsymbol{h}_\mathrm{e} * \boldsymbol{M}\right) \\ \boldsymbol{M}_\mathrm{o} &= (\uparrow 2)_{\mathrm{r\,o}} \left(\boldsymbol{h}_\mathrm{o} * \boldsymbol{M}\right) \\ \boldsymbol{F} &= \boldsymbol{M}_\mathrm{e} + \boldsymbol{M}_\mathrm{o} \end{aligned} \tag{3}$$

where $(\uparrow 2)_{\mathrm{r\,e,o}}$ is a dyadic upsampling operator that inserts zeros at the even or odd positions in rows. To apply the convolution operation at the picture boundaries, the picture matrix is extended outside the boundaries by replicating the last available sample.

Fig. 4 shows the values of one of the four poly-phase bidimensional kernels, resulting from the successive application of (2) and (3), $\boldsymbol{H}_{\mathrm{ee}} = \boldsymbol{h}_\mathrm{e} \cdot \boldsymbol{h}_\mathrm{e}^T$, for each of the four normative filters. The others kernels have the same values, rotated by multiples of $\pi/2$. Remember that in the final picture each kernel affects only one of the four up-scaled samples. These interpolation filters should limit the effect of the spectral input replicas, entering in the frequency band of the signal thanks to the up-sampling, so they are also known as anti-imaging filters.

For the upscaling operation, LCEVC defines a set of four normative filters, with a fixed scaling factor of 2 in the horizontal and vertical directions, named: “Nearest Neighbor”, “Linear”, “Cubic”, “Modified Cubic”. Their respective one-dimensional taps, on a scale of 16384, are shown in Table I.



(a)

(b)

(c)

(d)

Fig. 4. $\boldsymbol{H}_{\mathrm{ee}}$ filter coefficients.

TABLE I

UPSCALING FILTER COEFFICIENTS

Filter type	$16384 \times (k_0$	k_1	k_2	$k_3)$
Nearest Neighbor	0,	16384,	0,	0
Linear	0,	12288,	4096,	0
Cubic	−1382,	14285,	3942,	−461
Modified Cubic	−2360,	15855,	4165,	−1276

Fig. 5 shows the magnitude of the $\boldsymbol{H}$ filter, as defined in (1), resulting from the poly-phase implementation of 4 × 4 filter showed in Fig. 4. All have a low-pass behaviour, as required to an interpolation filter, with greater or lesser effectiveness in filtering the spectrum images.

Besides the four predefined filters specified by the standard and described above, LCEVC provides a mechanism for the

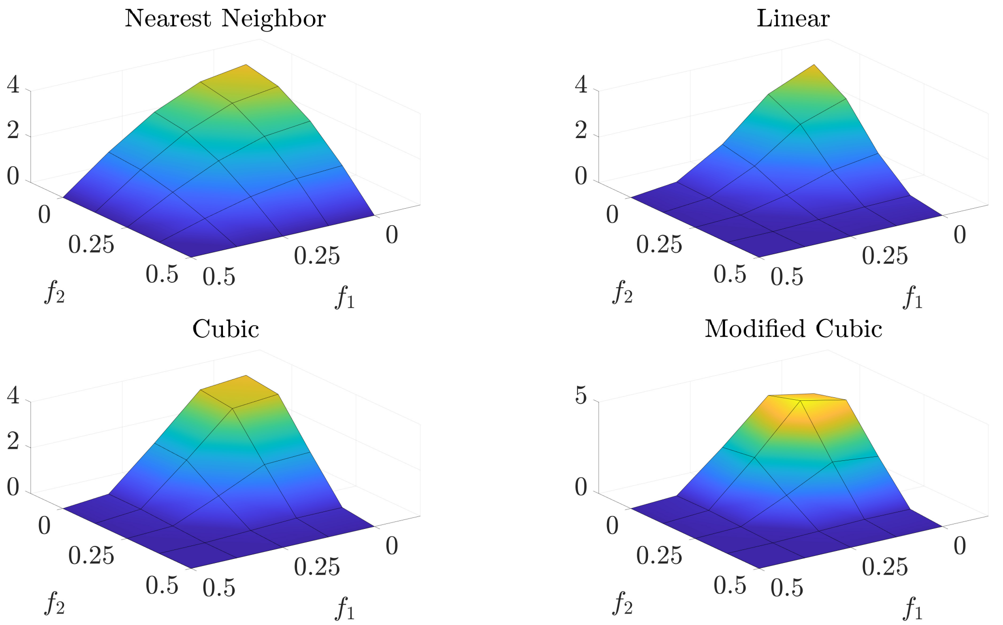


Fig. 5. Magnitude of the two dimensional DFT of the filter kernels ($\boldsymbol{H}$).

user to define a custom filter, with the only limitations that it shall be represented by four taps and implemented as a separable filter, as well as the predefined filters. When the custom filter is selected, the values of the four taps, are encoded in the bitstream using the same scale of 16384.

A. Predicted Residual

The processing block defined Predicted Residual consists of an adjustment of the sample values after up-scaling from the lower resolution to the higher resolution, typically by a factor 2 in the horizontal and vertical direction.

The adjustment is performed comparing the values of the base resolution picture samples $\boldsymbol{B}$ with the corresponding full resolution picture samples $\boldsymbol{F}$. The process consists in computing the difference between the base resolution samples and the average of the four full resolution samples, and use such modifier adding it to the four full resolution samples.

$$\boldsymbol{B}_{\mathrm{M}} = (\uparrow 2)(\boldsymbol{B} - (\downarrow 2)\boldsymbol{J}_4\boldsymbol{F}) \quad (4)$$

$$\boldsymbol{F}_a = \boldsymbol{F} + \boldsymbol{H}_{\mathrm{NN}} * \boldsymbol{B}_{\mathrm{M}} \quad (5)$$

where $\boldsymbol{J}_4$ is a 4×4 matrix of ones and $\boldsymbol{H}_{\mathrm{NN}}$ is Nearest Neighbor 8×8 filter.

IV. Temporal Prediction

The Temporal Prediction algorithm is extremely simple. The reconstructed residuals from a picture are stored in a temporal buffer. Based on a cost criterion implemented by the encoder, for each block of size 2×2 or 4×4, the block can be predicted from the temporal buffer or encoded without prediction.

The prediction is done for each block from its corresponding spatial position in the temporal buffer, so no Motion Estimation is performed and no Motion Vectors are encoded in the bitstream. Only one reconstructed picture is stored in the temporal buffer, so the prediction is always performed as a forward prediction from the previous picture in display order. Anyway, since the base reconstructed picture is needed to decode the corresponding LCEVC enhancement, the bitstream order of the LCEVC pictures is the same as the bitstream order of the associated Base pictures.

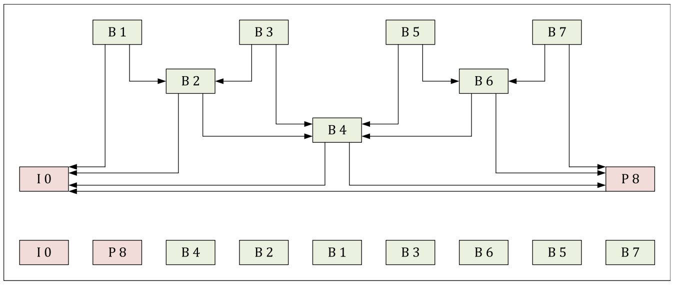


Fig. 6. Picture order (display and decoding) for the base bitstream.

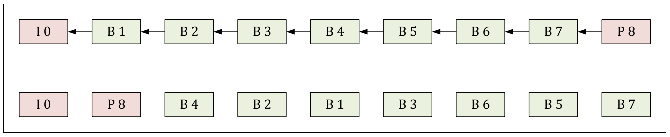


Fig. 7. Picture order (display and decoding) for the LCEVC enhancement bitstream.

Figs. 6 – 7 show the display order and the decoding order for the Base bitstream (e.g. AVC) and the LCEVC Enhancement bitstream, respectively. The upper part shows the display order with arrows representing the temporal prediction references, while the lower part shows the decoding order.

The Temporal Prediction information is then encoded to and decoded from the bitstream in a similar way to the transform coefficients, as described in Section VII, with the

$$DT = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{bmatrix}$$

Fig. 8. Direct transform matrix (DT).

$$IT = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{bmatrix}$$

Fig. 9. Inverse transform matrix (IT).

only exception that the information for each block is binary, to indicate whether Temporal Prediction is applied or not for that block.

The choice to use a very simple algorithm for Temporal Prediction, without any Motion Estimation and Motion Vectors, is based on the consideration that the input to the LCEVC encoding process consists of residuals, i.e. the coding errors between the upscaled base encoding and the full resolution original. For this reason, the input residuals to LCEVC benefit implicitly from the Motion Estimation performed by the Base codec, that removes the temporal correlation between the input pictures. Thus, the Temporal Prediction in LCEVC is only used when the residuals show a strong similarity to the co-located residuals of the preceding picture in display order. This is the case, for example, of a static part of the picture with high frequency details that benefits from the enhancement performed by LCEVC, like graphical overlay on natural video.

V. Transformation

A. Direct Transformation

The Direct Transformation, computes a 2×2 or 4×4 matrix of frequency coefficients from a 2×2 or 4×4 matrix of original residuals, i.e. samples in the picture domain. To define it as a matrix product, we need to transform the residual matrix $\boldsymbol{R}$ in a column vector and the coefficient matrix $\boldsymbol{C}$ in the same way:

$$r(iN+j) = \boldsymbol{R}(i,j) \quad i = 0\ldots N-1, j = 0\ldots N-1 \tag{6}$$

$$c(iN+j) = \boldsymbol{C}(i,j) \quad i = 0\ldots N-1, j = 0\ldots N-1 \tag{7}$$

where N is the dimension of the squared matrix of residuals.

Now we can write

$$\boldsymbol{c} = 2^{-N}\boldsymbol{DT}\cdot\boldsymbol{r} \tag{8}$$

where $\boldsymbol{DT}$ is a $2^N \times 2^N$ matrix of ±1 and it is represented in Fig. 8.

B. Inverse Transformation

In the same way, the Inverse Transformation, computing a 2×2 or 4×4 matrix of reconstructed residuals from a 2×2 or 4×4 matrix of coefficients, is defined as a matrix product:

$$\boldsymbol{r} = \boldsymbol{IT}\cdot\boldsymbol{c}. \tag{9}$$

The relationship between the Direct Transformation matrix ($\boldsymbol{DT}$) and the Inverse Transformation matrix ($\boldsymbol{IT}$) is that the latter is the inverse matrix of the former, except for a normalization factor

$$\boldsymbol{IT} = 2^N\boldsymbol{DT}^{-1}. \tag{10}$$

One notable property of the Direct Transform and Inverse Transform matrices is that, apart from the normalization factor, they are orthogonal, so

$$\boldsymbol{IT} = \boldsymbol{DT}^T. \tag{11}$$

The $\boldsymbol{IT}$ matrix is represented in Fig. 9.

The physical interpretation of the Inverse Transformation matrix is that each vertical vector represents a 4×4 component of the residuals triggered by a single $c^q(k)$ coefficient. Fig. 10 depicts such reconstructed residual patterns originated by each coefficient $\boldsymbol{C}(i,j)$, with $i = 0\ldots3$ and $j = 0\ldots3$.

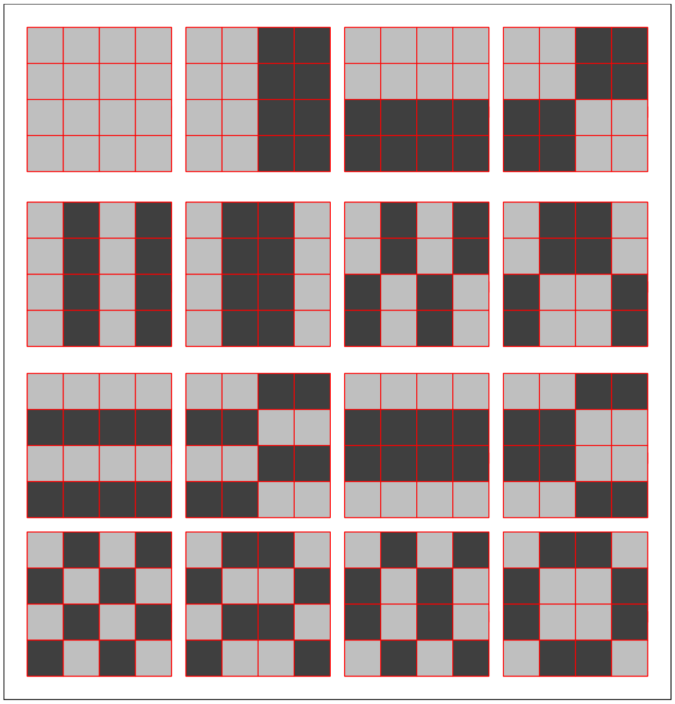


Fig. 10. Inverse transform bases (IT).

VI. Quantization

In the same way as other block based video coding standards, the main processing block of LCEVC that allows a precise control of the bitrate produced by the encoder is the Quantization block, with the specification of the quantizer stepwidth (SW), in a similar way to the "quantization parameter" (QP) defined in other MPEG standards. The modulation of the quantizer stepwidth allows a coarser quantization with higher values of SW resulting in a lower bitrate production, or a finer quantization with lower values of SW resulting in a higher bitrate. The following subsections describe in detail the algorithms for Direct Quantization and Inverse Quantization, and the formulae for deriving the different parameters used in the algorithms, with derivation from a single value of SW signaled in the bitstream, which is denoted as Original Stepwidth.

Concerning the number of bits used for the representation of samples, residuals, and transform coefficients, LCEVC operates as follows. The original samples, regardless of the original number of bits per sample, which can be 8, 10, 12, or 14, are internally represented with 15 bits. When computing the residuals, the representation is 16 bits, in the range $[-32768, 32767]$. Applying the Direct Transformation, no bits are added, since the result is divided by 2^N, so that the non quantized transform coefficients are represented with 16 bits. After the Direct Quantization, the quantized coefficients are clipped to 14 bits, in the range $[-8192, 8191]$.

Conversely, with the Inverse Quantization the dequantized coefficients are expanded and clipped to 16 bits. Applying the Inverse Transformation, 2 bits are added with the 2×2 matrix, and 4 bits are added with the 4×4 matrix. The resulting reconstructed residuals are then clipped to 16 bits.

It should be noted that, starting from the upscaled base picture, and adding Temporal Prediction and the reconstructed residuals after Inverse Transformation and Inverse Quantization, it is possible that the final result exceeds the 16 bits range $[-32768, 32767]$. This is prevented by clipping the results of the sum of the three components in the 16 bits range.

A. Direct Quantization

The algorithm for Direct Quantization can be described by the following formulae. The following symbols are used for the variables of the Direct Quantization equations:

- $c(k)$ for the input frequency Coefficient
- $c^q(k)$ for the corresponding quantized frequency Coefficient
- OSW for the Original Stepwidth
- DSW for the Direct Quantization Stepwidth
- DZ for the Dead Zone
- QMC for the Quantization Matrix Coefficient

The typical quantization formula used in block-based predictive-transform codecs, including MPEG ones,

$$c^q(k) = \mathrm{sgn}(c(k)) \left\lfloor \frac{|c(k)|}{\mathrm{DSW}} \right\rfloor \tag{12}$$

has already a dead zone around zero, since the quantization is obtained by rounding toward zero.

To increase the size of the dead zone around zero, an additional parameter DZ is subtracted from the absolute value of the input, checking that the resulting value is non negative:

$$c^q(k) = \mathrm{sgn}(c(k)) \left\lfloor \frac{\max(|c(k)| + \mathrm{DZ}, 0)}{\mathrm{DSW}} \right\rfloor \tag{13}$$

with output values clipped in the range $[-8192, 8191]$, that is in 14 bits. The DSW value is computed as

$$\mathrm{DSW} = \left\lfloor \frac{\mathrm{QMC}\,\mathrm{OSW}^2}{65536} \right\rfloor. \tag{14}$$

Thus, the actual denominator in the direct quantization formula has a quadratic relationship with the OSW value signaled in the bitstream.

The DZ value is computed as

$$\mathrm{DZ} = \left\lfloor \frac{\left(65536 - \left\lfloor \frac{A\,\mathrm{DSW}+B}{2} \right\rfloor\right)\mathrm{DSW}}{65536} \right\rfloor \tag{15}$$

with $A = 39$, $B = 126484$.

Thus, also the DZ used in the direct quantization formula to reduce the number of small transform coefficients to be encoded in the bitstream, has a quadratic relationship with the OSW value signaled in the bitstream, with negative values.

As an example, with the stepwidth signaled in the bitstream, i.e. the Original Stepwidth, assuming values of 1024, 2048, 3072, 4096, using a fixed value for the quantization matrix coefficient ($\mathrm{QMC} = 32$), the direct quantization formula parameters are reported in Table II.

TABLE II
EXAMPLE OF DIRECT QUANTIZATION FORMULA COEFFICIENTS

OSW	DSW	DZ
1024	512	−60
2048	2048	−1176
3072	4608	−6157
4096	8192	−19681

B. Inverse Quantization

The algorithm for Inverse Quantization can be described by the following formulae. The following symbols are used for the variables of the Direct Quantization equations:

- $c^q(k)$ for the input Quantized Coefficients
- $c^d(k)$ for the output Dequantized Coefficients
- OSW for the Original Stepwidth.
- DSW for the Direct Quantization Stepwidth.
- ISW for the Inverse Quantization Stepwidth.
- OO for the Original Offset.
- IO for the Inverse Quantization Offset.
- QMC for the Quantization Matrix Coefficient.

The inverse quantization formula is symmetric to the direct quantization one, taking into account that a Dead Zone value is subtracted from the original coefficient, and an Inverse Quantization Offset value is added to the dequantized coefficient.

$$c^d(k) = \operatorname{sgn}(c^q(k))(|c^q(k)|\,\mathrm{ISW} + \mathrm{IO}) \tag{16}$$

with output values clipped in the range [-32768, 32767], that is in 16 bits. The computation of the ISW value is given by

$$\mathrm{ISW} = \mathrm{DSW} + \left\lfloor \frac{(D - C\ \ln(\mathrm{DSW})\,\mathrm{DSW}^2)}{2147483648} \right\rfloor \tag{17}$$

with $C = 5242,\ D = 99614$. Which is equivalent to

$$\mathrm{ISW} = \mathrm{DSW} + \left\lfloor \frac{(99614 - 5242\ \ln(\mathrm{DSW}))\,\mathrm{DSW}^2}{2147483648} \right\rfloor. \tag{18}$$

Thus, ISW is derived from the DSW, with an adjustment that has a quadratic dependency on the same DSW.

The offset value is calculated as

$$\mathrm{IO} = \left\lfloor \left(\frac{2048}{4^{\mathrm{DOMF}}}\,\mathrm{OO} + \lfloor -5242\ \ln(\mathrm{DSW}) \rfloor + \lfloor 5242\ \ln(\mathrm{OSW}) \rfloor \right) \frac{\mathrm{DSW}}{65536} \right\rfloor \tag{19}$$

depending on the value of the DequantOffsetModeFlag (DOMF) signaled in the bitstream.

The actual value of the offset used in the inverse quantization formula depends on the offset value signaled in the bitstream, on DSW and on the logarithm of OSW and DSW.

As an example, with signaled stepwidth values of 1024, 2048, 3072, 4096, using a fixed value for the signaled offset (OO = 16), the inverse quantization formula coefficients are reported in Table III.

TABLE III
INVERSE QUANTIZATION FORMULA

OSW	ISW	IO
1024	513	268
2048	2064	1024
3072	4688	2239
4096	8439	3899

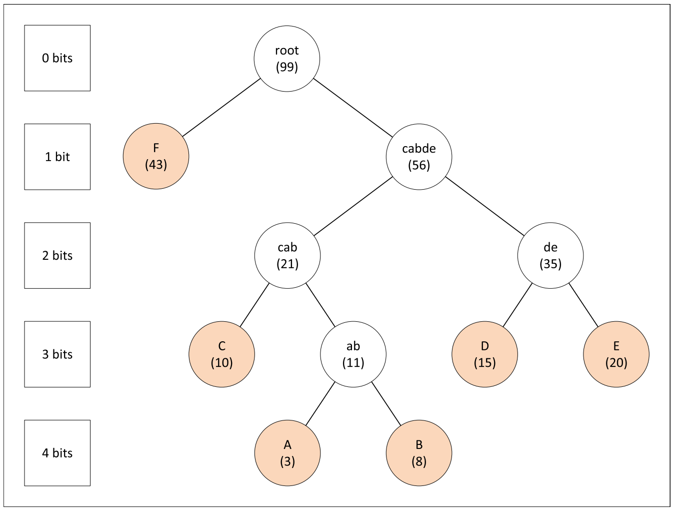


Fig. 11. Huffman tree construction.

VII. ENTROPY CODING

Prefix Coding is used to associate a variable length codeword (VLC) to each symbol of the transformed and quantized coefficients. The length of the codewords associated to all symbols in the codebook is assigned using a Huffman tree, built with the canonical Huffman algorithm. Then the binary codewords are computed with a deterministic algorithm, replicable at the encoder and decoder.

The following steps are iterated, until a single parent (root) node for the tree is created:

1) Order the nodes in a list, in order of increasing frequency.
2) Take the two symbols with lowest frequency and create a parent node that has the two nodes as children.
3) Assign the sum of frequencies of the children to the parent, add the parent to the nodes list and remove the children from the list.

Once the Huffman tree is completed, the number of branches needed to reach each leaf node, with each leaf corresponding to a symbol, determines the number of bits for the respective code word, that is the code length for that symbol.

Once the code lengths are assigned, the symbols are reordered in descending code length order and, when the length is the same, in ascending lexicographic order. Finally, binary codewords are assigned to the symbols, starting from the highest code length and from the lowest symbol, and starting from the all-zero codeword, in ascending order.

TABLE IV
LENGTH AND VALUES OF HUFFMAN CODEWORDS

Symbol	Code length	Code word
A	4	0000
B	4	0001
C	3	001
D	3	010
E	3	011
F	1	1

In the specific case with the six symbols depicted in Fig. 11, the code lengths and values assigned to each symbol are reported in Table IV.

The main innovative aspect of the entropy coding algorithm specified in LCEVC is the computation of an adaptive and optimized codebook for each set of symbols to be encoded. Both the encoder and the decoder implement the same deterministic algorithm to generate the codewords associated to the symbols from the frequency of the same symbols. Thus it is sufficient to transmit the symbol value and the corresponding codeword length for each set of symbols to be encoded and decoded.

Comparing LCEVC to the recent MPEG standards, LCEVC uses an adaptive VLC solution for entropy coding, as opposed to the Binary Arithmetic Coding (BAC) of the other specifications. This design choice is motivated by the lower computational complexity of VLC with respect to BAC, even though the compression performance is sub-optimal.

VIII. Processing Performance

The performance of the software implementation of LCEVC was compared to the performance of four different video codecs, namely HEVC, EVC, and VVC (specified by MPEG/VCEG), and AV1 (specified by AOM) [26]. For each video codec, the relevant Reference Software has been used to perform the encoding sessions, selecting the following software versions:

- HEVC: HM version 16.08 [27]
- EVC: TM version 01.00
- VVC: TM version 04.02 [28]
- AV1: release 01.00 [29]

The set of test sequences used to run the encoding sessions was limited to four video sequences from the MPEG test set. The encoding sessions have all been run with a fixed QP throughout the complete sequence of either 500 frames for the 50 fps sequences or 600 frames for the 60 fps sequences. To get comparable results, for each sequence the eight values of QP giving the best approximation of four target bitrates have been retained. The selected target bitrates are:

- 2.8, 4.0, 5.6, 8.0 Mbps for the lower resolution (HD) sequences
- 5.6, 8.0, 11.2, 16.0 Mbps for the higher resolution (UHD) sequences

Collecting the test data from the encoding sessions on the four test sequences with the four base codecs, the average execution time for each set of sequences was estimated, at the same resolution and the same frame rate. All the encoding sessions were performed on the same machine, a Windows Server 2012 with the following characteristics:

- CPU: intel Xeon E5-2690 at 2.90 GHz
- RAM: 32.0 GB
- OS: Windows Server 2012 R2 64 bit

From the average ratios of processing times for the three most recent codecs (EVC, VVC, AV1) with respect to HEVC, it is possible to estimate the complexity of the software implementation of the codecs:

- EVC has a complexity about 5 times higher than HEVC;
- VVC has a complexity about 10 times higher than HEVC;
- AV1 has a complexity about 17 times higher than HEVC (in the single pass configuration).

Concerning the processing time of LCEVC, the average times for the sequences of 500 frames and 600 frames are:

- for a full resolution of 1920 × 1080 around 70 s
- for a full resolution of 3840 × 2160 around 140 s

In all the above cases, including the Test Model for HEVC, the weight of the LCEVC processing is below 1% of the processing time of the base codec, so essentially negligible when computing the overall complexity. In summary, the saving in processing time with the configurations used for Base resolution plus LCEVC encoding compared to Full resolution encoding are:

- for HEVC Base with LCEVC compared to HEVC Full resolution, a factor 3.6;
- for EVC Base with LCEVC compared to EVC Full resolution, a factor 2.6;
- for VVC Base with LCEVC compared to VVC Full resolution, a factor 3.6;
- for AV1 Base with LCEVC compared to VVC Full resolution, a factor 3.1.

IX. Rate Distortion Performance

The goal of a video compression algorithm is the minimization of the bitrate required to achieve a given video quality, or, stated in a symmetrical way, the maximization of the video quality achieved for a given bitrate. To verify quantitatively the requirements on bitrate minimization or quality maximization, the approach adopted in the standardization process of ISO/MPEG and ITU-T/VCEG is the execution of a test campaign, finalized at determining the objective and subjective video quality, using well defined metrics.

For objective quality, the metrics adopted in the LCEVC verification tests are PSNR (Peak Signal to Noise Ratio) [30] and VMAF (Video Multi-method Assessment Fusion) [31]. For subjective quality, the metric adopted is MOS (Mean Opinion Score) [32].

The formal objective and subjective assessment of the LCEVC standard has been completed and approved in April 2021 at the 134th MPEG meeting. The official report on the LCEVC verification tests is published in the MPEG website as WG04 document N0076 [33].

LCEVC has been compared to four single-layer video coding technologies developed by MPEG and VCEG, specifically AVC/H.264, HEVC/H.265, EVC and VVC/H.266, that

TABLE V

OPERATIONAL POINTS FOR AVC WITH LCEVC TESTS

Parameter	UHD min	UHD max	HD min	HD max
BR AVC (Mbps)	2.5	17.2	0.6	3.7
BR LCEVC (Mbps)	1.7	12.2	0.6	2.4
PSNR AVC (dB)	31	42	31	42
PSNR LCEVC (dB)	32	41	30	40
VMAF AVC	47	99	65	94
VMAF LCEVC	48	99	69	92
MOS AVC	0.55	7.65	2.53	7.20
MOS LCEVC	2.95	7.48	3.02	7.48
EL/BL %	0	15	19	59

TABLE VI

OPERATIONAL POINTS FOR VVC WITH LCEVC TESTS

Parameter	UHD min	UHD max	HD min	HD max
BR VVC (Mbps)	0.2	9.4	0.3	3.5
BR LCEVC (Mbps)	0.3	7.7	0.4	2.3
PSNR VVC (dB)	31	43	35	45
PSNR LCEVC (dB)	31	41	31	40
VMAF VVC	66	99	83	97
VMAF LCEVC	69	99	83	96
MOS VVC	1.79	8.66	3.42	7.92
MOS LCEVC	2.50	8.79	3.91	7.96
EL/BL %	0	29	22	38

represent a set of single-layer coding technologies of increasing compression efficiency and at the same time increasing computational complexity.

While the in depth analysis of the objective and subjective MPEG Verification Tests results is beyond the scope of the paper, a summary of the operational points used for the test and the conclusions are reported here.

For each of the four video codecs, 6 sequences were encoded at full resolution with the single layer codec and at quarter resolution with the base codec adding the LCEVC enhancement: 4 sequences at 3840 × 2160 (UHD) resolution and 2 sequences at 1920 × 1080 (HD) resolution.

Four each video sequence, the test points were selected to span, as far as possible, the same range of objective quality, with particular care for the range of VMAF values, that give a better estimate of the range of subjective MOS quality.

Tables V and VI report, for the test with AVC and VVC respectively, the minimum and maximum values for the essential parameters: bit rate, PSNR, VMAF, MOS. Additionally, they report the minimum and maximum percentage of bit rate allocated to the LCEVC Enhancement bitstream with respect to the Base bitstream.

Comparing the full-resolution LCEVC-enhanced encoded sequences with the full-resolution single-layer encoded sequences, in the same range of subjective quality, the reported average bit rate savings are:

- 46% for UHD and 28% for HD for LCEVC enhancing AVC;
- 31% for UHD and 24% for HD for LCEVC enhancing HEVC;
- an overall benefit for LCEVC enhancing EVC and VVC.

X. CONCLUSION

The LCEVC standard was developed in the context of ISO/IEC MPEG Video Coding, with the explicit goal to provide a scheme for enhancement to any existing, or even future, single-layer video coding algorithm. It is a multi-layer scheme, but differs substantially from other scalable schemes defined by MPEG, such as SVC and SHVC, since it is designed to work in the samples and residuals domain, without any dependency from the base codec algorithm.

The two main goals of the LCEVC development were the high efficiency in terms of Rate Distortion performance, and the low complexity in terms of processing power and memory requirements. These goals have been verified through the test campaign performed following the established MPEG/VCEG approach.

The verification tests performed at the end of the standardization process confirm that bitrate savings can be achieved when using a single-layer base codec at quarter resolution in conjunction with LCEVC at full resolution, when comparing to the same base coding technology at full resolution. The subjective test results reported in [33] indicate bitrate savings of approximately 40% over AVC, 30% over HEVC, 15% over EVC and VVC, at the operational points used for these tests.

ACKNOWLEDGMENT

The authors would like to thank Prof. Lu Yu, the Chairperson of MPEG Video Coding (ISO/IEC JTC1 SC29 WG04) and also would like to thank Walt Husak, the Chairperson of the WG04 subgroup responsible for the development of LCEVC, for their leadership and their valuable support in the course of the standardization of LCEVC, between April 2018 and November 2021.

REFERENCES

[1] *ISO/IEC 23094-2—Information Technology—General Video Coding—Part 2: Low Complexity Enhancement Video Coding*, Standard ISO/IEC 23094-2:2021, Nov. 2021.

[2] *ISO/IEC 23094-3—Information Technology—General Video Coding—Part 3: Conformance and Reference Software for Low Complexity Enhancement Video Coding*, Standard ISO/IEC 23094-3:2021, 2022.

[3] F. Maurer, S. Battista, L. Ciccarelli, G. Meardi, and S. Ferrara, "Overview of MPEG-5 Part 2—Low complexity enhancement video coding (LCEVC)," *ITU J. ICT Discoveries*, vol. 3, no. 1, pp. 1–11, Jun. 2020.

[4] *ITU-T H.261—Video Codec For Audiovisual Services at p × 384 kbit/s*, document Rec. ITU-T H.261:1988, Nov. 1988.

[5] *ITU-T H.263—Video Coding for Low Bit Rate Communication*, document Rec. ITU-T H.263:1996, Mar. 1996.

[6] K. Rijkse, "H.263: Video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42–45, Dec. 1996.

[7] *ISO/IEC 11172-2—Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mbit/s—Part 2: Video*, Standard ISO/IEC 11172-2:1993, Aug. 1993.

[8] C. Fogg, D. J. LeGall, J. L. Mitchell, and W. B. Pennebaker, *MPEG Video Compression Standard*. Berlin, Germany: Springer, 1996.

[9] *ISO/IEC 14496-2—Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, Standard ISO/IEC 14496-2:1999, Dec. 1999.

[10] T. Ebrahimi and F. Pereira, *The MPEG-4 Book*. Hoboken, NJ, USA: Prentice-Hall, 2002.

[11] *ISO/IEC 23094-1—Information Technology—General Video Coding—Part 1: Essential Video Coding*, Standard ISO/IEC 23094-1:2020, Oct. 2020.

[12] K. Choi, J. Chen, D. Rusanovskyy, K.-P. Choi, and E.-S. Jang, "An overview of the MPEG-5 essential video coding standard," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 160–167, May 2020.

[13] *ISO/IEC 13818-2—Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video*, Standard ISO/IEC 13818-2:1996, May 1996.

[14] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. Hoboken, NJ, USA: Springer, 1996.

[15] *ISO/IEC 14496-10—Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, Standard ISO/IEC 14496-10:2003, Dec. 2003.

[16] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[17] *ISO/IEC 23008-2—Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, Standard ISO/IEC 23008-2:2013, Dec. 2013.

[18] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[19] *ISO/IEC 23090-3—Information Technology—Coded Representation of Immersive Media—Part 3: Versatile Video Coding*, Standard ISO/IEC 23090-3:2021, Feb. 2021.

[20] B. Bross *et al.*, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[21] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[22] J. M. Boyce, Y. Yan, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jul. 2016.

[23] S. Ferrara *et al.*, "MPEG-5 part 2: Low complexity enhancement video coding (LCEVC): Overview and performance evaluation," *Proc. SPIE*, vol. 11510, p. 44, Aug. 2020, doi: 10.1117/12.2569246.

[24] A. Jiménez-Moreno, L. Ciccarelli, R. Clucas, and S. Ferrara, "HDR video coding with MPEG-5 LCEVC," in *Proc. 1st Mile-High Video Conf.*, New York, NY, USA, Mar. 2022, pp. 25–31, doi: 10.1145/3510450.3517307.

[25] N. Barman, S. Schmidt, S. Zadtootaghaj, and M. G. Martini, "Evaluation of MPEG-5 Part 2 (LCEVC) for live gaming video streaming applications," in *Proc. 1st Mile-High Video Conf.* New York, NY, USA, Mar. 2022, pp. 108–109, doi: 10.1145/3510450.3517279.

[26] S. Battista, "Preliminary test results using LCEVC with current and next generation codecs," Int. Standardization Org., Geneva, Switzerland, Tech. Rep. MPEG-M051038, Oct. 2019. [Online]. Available: https://dms.mpeg.expert/doc_end_user/documents/128_Geneva/wg11/m51038-v4-M051038_V04.zip

[27] MPEG/VCEG. (2022). *HEVC Reference Software*. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[28] (2022). *VVC Reference Software*. [Online]. Available: https://jvet.hhi.fraunhofer.de/

[29] AOM. (2022). *AV1 Reference Software*. [Online]. Available: https://aomedia.googlesource.com/aom/

[30] S. Battista, M. Conti, and S. Orcioni, "Methodology for modeling and comparing video codecs: HEVC, EVC, and VVC," *Electronics*, vol. 9, no. 10, p. 1579, Sep. 2020, doi: 10.3390/electronics9101579.

[31] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *Netflix Tech. Blog*, vol. 6, no. 2, Jun. 2016. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

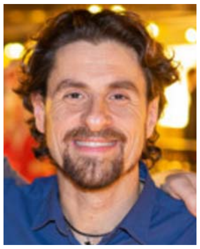
[32] *ITU-R BT.500—Methodologies for the Subjective Assessment of the Quality of Television Images*," document Rec. ITU-R BT.500:2019, Oct. 2019.

[33] *Verification Test Report on the Compression Performance of Low Complexity Enhancement Video Coding*, Standard ISO/IEC WG04 N0076, May 2021.

Stefano Battista received the Laurea degree in electronic engineering from the Università Politecnica delle Marche, Italy, in 1990. From 1991 to 1997, he was a Researcher with the Telecom Italia Laboratory, Video and Multimedia Group, Turin, Italy. His main activities have been on video coding, video analysis, 3D modeling, and multimedia systems, with a focus on standardization for multimedia communications. From 1997 to 1999, he was a System Engineer with the Advanced Systems Technology Group, STMicroelectronics, Agrate, Milan, Italy. His main activities were in the development of new architectures for multimedia consumer products, specifically on DVB/DVD platforms and their evolution toward interactive TV. From 1999 to 2018, he was a Researcher and a System Engineer with bSoft, Macerata, Italy, where he has been active in research and development. He has published several research papers on video coding and multimedia systems and contributed to books related to the fields of communications and multimedia. He actively participated in the standardization effort of MPEG, in particular to MPEG-4 reference software development, co-authoring part five of the standard. For this contribution to the standardization activity, he received an ISO Certificate of Appreciation. He contributed to co-operative research projects in the Third (RACE), Forth (ACTS), Fifth (IST), Sixth, and Seventh (ICT) Framework Programs of the European Commission. He worked as a consultant for several companies in Italy and abroad, and as an External Evaluator and a Reviewer for the European Commission.

Guido Meardi received the M.B.A. degree from MIT Sloan and the M.Sc. degree in computer engineering from the Politecnico di Milano. He was a Siebel Scholar with MIT Sloan for M.B.A. degree and an Intel Scholar with the Politecnico di Milano and The University of Texas at Austin for M.Sc. degree. He is currently a Co-Founder and the CEO of V-Nova, leading company in data compression and artificial intelligence. He is also a Keen Innovator, an Entrepreneur, and an Investor, with relevant business building experience and half a dozen exits. Having retained in-depth scientific and engineering expertise, he contributed to the foundational development work for the V-Nova Intellectual Property portfolio and is an Inventor or a Joint Inventor of several essential aspects of the coding standards MPEG-5 Part two Low Complexity Enhancement Video Coding (LCEVC) and SMPTE VC-6 ST-2117, with over 200 patents filed. Former Senior Partner at McKinsey, where he was the Head of the Organization and Operations Practices of the Mediterranean Complex, he also has a breadth of business experience and access to senior executives in a variety of industries and geographies, with well-established experience in telecoms, technology, healthcare, insurance, banking, automotive, aerospace, and defence. He led transformational projects in all continents, and he was instrumental in setting up some of McKinsey's own innovation-related business building activities.

Simone Ferrara received the Graduate degree in telecommunications engineering from the Politecnico di Milano and the M.Sc. degree in electrical engineering from Washington University in St. Louis, USA. He is currently a Senior Vice President at V-Nova, where he is responsible for the technology, IP and standardization strategy, including driving its execution across the company. He has been leading the development of MPEG-5 Part 2 Low Complexity Enhancement Video Coding and has contributed to many of the coding tools included in the standard. Prior to V-Nova, he worked in the telecommunications industry first as a Researcher and then as an IP Expert.



Lorenzo Ciccarelli received the Laurea degree in electronic engineering and telecommunication from the Università Politecnica delle Marche, Italy, in 1998. He received a final master's degree thesis on video coding and in particular working on algorithm optimization for H.263+ video codecs. From 1999 to 2006, he has been working on several aspects of video compression algorithm design and implementation participating and leading projects focused on developing codecs on different VLIW architecture mainly used for videoconference terminal and multiconference unit. In 2006, he moved to U.K. joining Ericsson SATTV (former Tandberg TV) Research and Development Department, where he had the opportunity to deepen his knowledge of the TV broadcasting side of the video coding gaining expertise in rate control and statmuxing while porting video compression algorithm on FPGAs. Between 2008 and 2014, he has been involved several projects to design software test models used to design, test and improve different algorithms based on MPEG2, AVC, and HEVC for large broadcasting systems based on multiple FPGA, DSPs, and CPUs. Between 2014 and 2016, he was leading the design of one of the first hardware implementation of a HEVC full UHD encoder based on hybrid x86 and FPGA architecture to then join BBC video coding RD, where he spent two years in improving internal video encoding testing platform (Turing Encoder) and working on different European funded projects. In 2018, he joined V-Nova Ltd., with the title of a Principal Research Engineer. During the last two years, he has been involved in the algorithm design and standardization process for MPEG5-Part 2 LCEVC.

Florian Maurer received the degree in electrical engineering, information technology, and computer engineering from RWTH Aachen University, Germany, and the Master of Science degree in 2020. He has focused his studies on signal processing with a particular interest in video coding and gained industry experience while working for a leading company in video compression. During his work, he is with the RWTH Institute of Communications Engineering chaired by Prof. Jens-Rainer Ohm, he acquired comprehensive knowledge of different video coding standards. He was involved in the standardization process of MPEG-5 Part 2—Low Complexity Enhancement Video Coding (LCEVC). His contributions helped to move LCEVC forward to the "International Standard (IS)" stage and he has taken part in the development of a reference implementation of LCEVC. Recently, he has also shown interest in graph signal processing (GSP) and is researching motion estimation in dynamic 3D point cloud sequences.

Massimo Conti (Member, IEEE) received the Graduate degree in electronics engineering from the University of Ancona, Italy, in 1987. He is currently an Associate Professor with the Dipartimento di Ingegneria dell'Informazione (DII), Università Politecnica delle Marche (UNIVPM), Ancona. His research activity in the field of microelectronics is mainly devoted to system level design of low power integrated circuits, design of energy harvesting systems, battery management systems, and vehicle to grid connection. He is the coauthor of more than 190 papers on international books, journals or conferences. Scopus: 158 publications, 856 citations, and H-index of 14. He is a Coordinator of European and National research projects. He is an editor of seven international books, a guest editor of three special issues, and the general chairperson of seven international conferences. Further more information visit the link (www.univpm.it/massimo.conti).

Simone Orcioni (Senior Member, IEEE) received the Laurea degree in electronics engineering from the Università degli Studi di Ancona, Italy, in 1992, and the Ph.D. degree in 1995.

From 1997 to 1999, he was a Post-Doctoral Fellow with the Università degli Studi di Ancona. In 2000, he became an Assistant Professor with the Università Politecnica delle Marche, teaching courses in analog and digital electronics, and publishing a text book. In 2017, he was a Guest Professor with the Ubiquitous Computing Laboratory (UC-Laboratory), HTWG Konstanz—University of Applied Sciences and a Lecturer of the course "Digital Signal Processing." Since 2021, he has been an Associate Professor with the Department of Information Engineering, Università Politecnica delle Marche. He has been working in statistical device modeling and simulation, analog circuit design, cyber-physical system simulation, and linear and nonlinear system identification. His current research interests include non-linear digital signal processing and power electronics for renewable energies.

Dr. Orcioni is a EURASPIP Member. He was a Guest Editor for *EURASIP Journal on Embedded Systems*, *Energy Research* (Frontiers), *Sensors* (MDPI), the program chair of three international conferences, an editor of four international books, and an inventor in two patents. He is currently a Guest Editor of *Mathematics* (MDPI).