



4K Real Time Software Solution of Scalable HEVC for Broadcast Video Application

Ronan Parois, Wassim Hamidouche, Pierre-Loup Cabarat, Mickaël Raulet,
Naty Sidaty, Olivier Déforges

► To cite this version:

Ronan Parois, Wassim Hamidouche, Pierre-Loup Cabarat, Mickaël Raulet, Naty Sidaty, et al.. 4K Real Time Software Solution of Scalable HEVC for Broadcast Video Application. IEEE Access, 2019, 7, pp.46748-46762. 10.1109/ACCESS.2019.2904196 . hal-02077600

HAL Id: hal-02077600

<https://hal.science/hal-02077600>

Submitted on 26 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

4K Real Time Software Solution of Scalable HEVC for Broadcast Video Application

Ronan Parois, Wassim Hamidouche, Pierre-Loup Cabarat, Mickael Raulet, Naty Sidaty, and Olivier Déforges

Abstract—Scalable High efficiency Video Coding (SHVC) is the scalable extension of the High Efficiency Video Coding (HEVC) standard. SHVC enables spatial, quality, bit-depth, color gamut and codec scalability. The architecture of the SHVC encoder is based on multiple instances of the HEVC encoder where each instance encodes one video layer. This architecture offers several advantages of being modular and close to the native HEVC coding block scheme. However, the close-loop SHVC architecture requires the complete decoding of the reference lower-layer frames to decode a higher quality layer, which considerably increases the complexity of both encoder and decoder processes. In this paper, we propose an end-to-end 4K real time SHVC solution, including both software encoder and decoder, for video broadcast applications. The SHVC codec relies on low level optimizations for specific Intel x86 platform and parallel processing to speed-up the encoding and decoding processes. The proposed encoder enables a real time processing of 4Kp30 video in 2x spatial scalability on the 4x10-cores Intel Xeon processor (E5-4627V3) running at 2.6 GHz. In addition, the SHVC decoder enables to decode, respectively, the lower quality layer in full HD (1920x1080p30) resolution, on ARM Neon mobile platform, and the enhancement layer in UHD (3840x2160p30), on a laptop fitted, with 4 cores Intel i7 processor running at 2.7 GHz. Finally, experimental results have shown that the proposed solution can reach a high rate-distortion performance close to the reference SHVC reference software Model (SHM) with a speed-up of 37 and 66 in Intra and Inter coding configurations.

Index Terms—Scalable Video Coding, HEVC, SHVC extension, real time video codecs.

I. INTRODUCTION

Nowadays, with the gross consumption of video contents, these latter are stored and delivered in several formats, such as resolution, frame rate, quality, bitdepth and codec in order to cover a wide range of users requirements. These needs consist in the available bandwidth, memory and codec: display, computing and energy capabilities as well as the content quality. However, encoding and delivering the video in all these specifications considerably increases both storage and bandwidth resources. The Scalable High efficiency Video Coding (SHVC) extension [1], [2], has been designed by the Joint Collaborative Team on Video Coding (JCT-VC) as the Annex H of the High Efficiency Video Coding (HEVC) standard [3] to encode the video in several layers (formats). SHVC is based on the HEVC standard and supports spatial, quality, bitdepth, color-gamut and codec scalability. The SHVC extension leverage

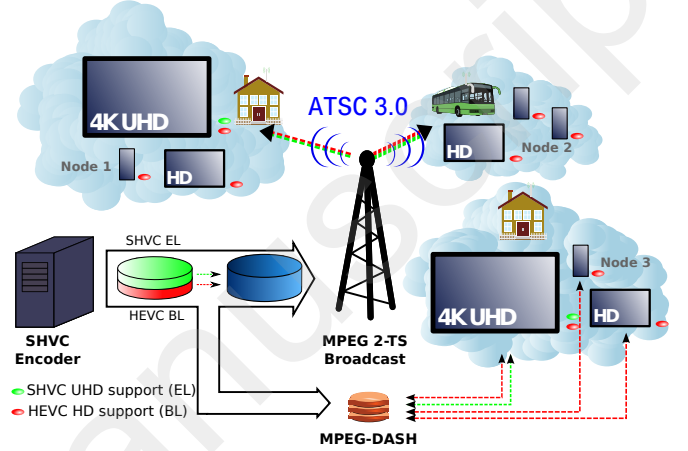


Fig. 1: ATSC3.0 broadcasting scenario with SHVC codec.

inter-layer predictions to improve the Rate Distortion (RD) performance by up to 30% under the Common Test Conditions (CTC) [4], compared to the simulcast coding configuration, which consists in independent HEVC encodings of a same video in various formats. This gain can be further enhanced with an optimal bitrate allocation strategy between SHVC layers, as proposed in [5], [6]. Compared to the previous Scalable Video Coding (SVC) extension [7], SHVC offers two main advantages. First, the coding architecture of SHVC remains simple based on the core HEVC standard with inter-layer prediction requiring only high level changes. Second, SHVC has been released only one year and half after HEVC. These two advantages hasten time-to-market after adoption and support its deployment in more video applications, not restricted to video conferencing services as for SVC.

Advanced Television Systems Committee (ATSC) 3.0 has considered several broadcasting scenarios for which SHVC has been identified as a serious candidate solution for video coding [8]. Fig. 1 illustrates one ATSC3.0 broadcasting scenario where the SHVC encoder encodes the video in two layers, HD to UHD (2x) spatial resolutions, and then are broadcast in Motion Picture Expert Group 2 - Transport Stream (MPEG-2 TS) within two Physical Layer Pipes (PLP) [9], [10]. The end-user receiving the two layers decodes either the Base Layer (BL) for HD quality or both layers for UHD quality, depending on its display, energy and mobility configurations.

The close-loop architecture of the SHVC extension requires the decoding of all reference layer frames to encode/decode a higher quality layer frame. This increases both encoding and decoding complexities compared to a single-layer coding

Dr. Wassim Hamidouche, Pierre-Loup Cabarat, Dr. Naty Sidaty and Pr. Olivier Déforges are with the Institute of Electronic and Telecommunication of Rennes (IETR) UMR 6164, VAADER team, INSA Rennes e-mail: firstname.lastname@insa-rennes.fr.

M. Ronan Parois and Mickael Raulet are with ATEME Rennes, 14 G rue du Pâtis Tatelin, 35700 Rennes, e-mail: r.parois@ateme.com and m.raulet@ateme.com.

configuration. Moreover, additional processing is introduced by the SHVC extension to rescale the reference frames used by the Enhancement Layers (EL) for inter-layer predictions. In this paper, we propose a complete solution for 4K real time SHVC codec including both SHVC encoder and decoder software. The proposed SHVC encoder, called SHVC-ATEME Encoder (SHVC-AE), is based on the professional *ATEME* HEVC software encoder HEVC-ATEME Encoder (HEVC-AE) [11]. The SHVC decoder is based on the open source real time *OpenHEVC* decoder [12]. The most time consuming coding/decoding operations are optimized with Single Instruction Multiple Data (SIMD) methods for x86 platforms.

The down-sampling (encoder side) and up-sampling (both encoder and decoder) operations, required for spatial scalability in SHVC encodings, are also optimized to speed-up and minimize the delay introduced by these operations. The HEVC high level parallel processing solutions including tile, slice and wavefront [13] are supported by the HEVC *ATEME* encoder and the *OpenHEVC* decoder. The encoding layers and the down/up-sampling functions are pipelined and processed in parallel to take advantage of multicore platforms and further minimize the end-to-end delay. The encoding solution enables a real time processing of 3840×2160 p 30 fps video on 4×10 -cores Intel Xeon processor (E5-4627V3) running at 2.6 GHz. The decoder enables a real time decoding of the BL in full HD (1920×1080 p) resolution on mobile Advanced RISC Machine (ARM) platform and the enhancement layer in UHD (3840×2160 p) at 30 fps on a laptop fitted with a 4 cores i7 Intel processor running at 2.7 GHz.

The rest of this paper is organized as follows. Section II provides details on the SHVC extension and the existing implementations of the HEVC and its scalable extension. The architecture of the SHVC encoder/decoder are provided in Section III. The performance in terms of coding efficiency and speed of the SHVC codec are provided and discussed in Section IV. Section V depicts the complete end-to-end SHVC demonstration in broadcast environment. Finally, Section VI concludes the paper.

II. RELATED WORKS

A. SHVC extension

The SHVC extension [2] enables several types of scalability not supported by SVC such as color-gamut and bit depth. These two scalability enable to switch from Standard Dynamic Range (SDR) to High Dynamic Range (HDR) formats within one bitstream [14]. SHVC defines high level syntax elements mostly at the level of Video Parameter Set (VPS) header. These syntax elements provide information on the video layers such as the number of layers, and for each layer: resolution, bit depth and the inter-layer dependencies. The SHVC encoder architecture consists of L HEVC encoders in a single encoder to encode each layer with L the number of layers: one BL and $L - 1$ ELs. In the case of SHVC spatial scalability, the BL HEVC encoder encodes a down-sampled version of the original video and feeds the first EL encoder with the decoded picture and its Motion Vectors (MVs). The

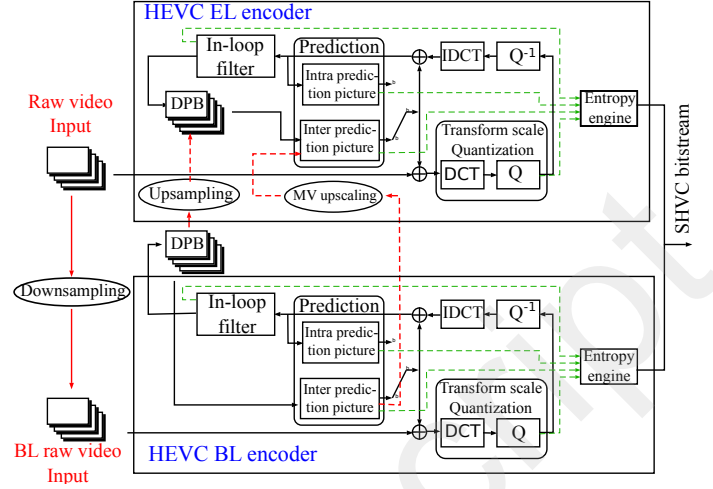


Fig. 2: Block diagram of the SHVC encoder by encoding two spatial scalability layers.

BL is the first ($l = 1$) and encodes the lowest resolution of the video. The EL layer encoder l ($l = 2, \dots, L$) encodes a higher resolution video with using the decoded picture from a lower layer as an additional reference picture (included in the reference picture lists). The inter-layer reference picture is up-sampled and its MVs up-scaled to match with the resolution of the layer being encoded. The up-sampling operation is standard operation performed by a 8-tap and 4-tap interpolation filters for luma and chroma samples, respectively. The down-sampling operation carried-out to produce the lower resolution video is not standard and can be considered as pre-processing operation. Fig. 2 shows a block diagram of the SHVC encoder encoding two layers in spatial scalability configuration. In the case of quality scalability (same resolution), the encoding process remains unchanged except that the picture used for inter-layer prediction is used without being up-sampled and its MVs up-scaled. As shown in Fig. 2, the outputs from the two encoders are multiplexed to form one bitstream that conforms to SHVC.

The HEVC standard version 2 defines two SHVC profiles: Scalable Main and Scalable Main 10 [2]. The Scalable Main enables a BL that conforms with the Main HEVC profile, while the Scalable Main 10 profile allows a BL that conforms with the Main 10 HEVC profile. The 4th HEVC version defines four more scalable profiles for BL in monochrome format with 8, 12 and 16 bitdepth (Scalable Monochrome, Scalable Monochrome 12, Scalable Monochrome 16) and one Scalable Main 4:4:4 profile that conforms to the Main 4:4:4 HEVC profile.

B. Real time video codecs

In this section we give a brief description on the existing SVC, HEVC and SHVC encoder and decoder solutions. The software *openSVC* decoder [15] has been developed to offer an open source real time decoder solution of the SVC extension. It was developed in C language and supports the Scalable Baseline profile library offering all tools to deal with

spatial, temporal and fidelity scalability. The *openSVC* decoder achieves a speed-up up to 50 times faster than the SVC reference software decoder Joint Scalable Video Model (JSVM) [16]. Authors in [17] proposed a SVC video encoder dedicated to HD video conferencing applications. This encoder combines slice-level parallelism for frame encoding with block-level parallelism for the up-sampling and interpolation filter processes. The baseline encoder is optimized in SIMD using Streaming SIMD Extensions (SSE)2 instructions. The parallel encoder enables, on a 8 cores Intel Xeon E5-2687W processor running at 3.1 GHz, to encode a 720p30 video in real time at different bitrates. The slice partitioning introduces a slight loss in rate-distortion coding efficiency.

Recently, several hardware [18]–[22] and software [12], [23]–[26] HEVC decoders have been developed. The hardware solutions offer a fast HEVC decoder implementation enabling real time decoding of 4Kp60 [19] and even 8Kp60 [21] with a very low energy consumption performance [20]. On the other hand, software HEVC decoder implementations offer flexibility, fast time-to-market and are well suited for quick adaptation to standard evolutions. In addition, software decoder can be easily optimized for several platforms, not dedicated to video processing, including Intel x86 [12] and ARM/Neon [26] using SIMD instructions.

There are a number of hardware [27] and software [11], [28]–[31] implementations of the HEVC encoder. The two open source software HEVC encoders, *Kvazaar* and *x265*, enable a real time encoding of 4K videos, with using both parallel processing (frame, tile and wavefront) and low level optimizations through SIMD instructions. In addition, these solutions use algorithmic optimizations to avoid the full rate-distortion search, especially at the level of quad-tree partitioning and intra prediction. These algorithmic optimizations enable encoding complexity reduction at the expense of bitrate increase [32], [33].

For SHVC encoder, authors in [34] leverage the existing correlation between layers to select the Coding Unit (CU) size at the EL by restricting the CU depth range to reduce the encoding complexity for quality scalability. This method skips some specific depth levels which are rarely used in the previous frame and neighboring CUs to further reduce the full search set and decrease the coding complexity with similar RD performance as the original SHVC encoder. Work in [35] propose a method to predict CU modes based on the co-located CU within the reference quality layer. This solution enables up to 51% complexity reduction while maintaining the overall quality of the original SHVC coding. Finally, authors in [36] developed an efficient Coding Tree Unit (CTU) decision method by combining a temporal-spatial searching order algorithm at the BL and a fast inter-layer searching algorithm at the EL to speed-up the SHVC encoding.

The major drawbacks of the SHVC solutions, mentioned above [34]–[38] is the absence of real time character. In fact, the complexity reduction opportunities offered by these solutions are around 50%, corresponding to a speedup of 2 of the reference SHVC reference software Model (SHM) encoder. However, to reach real time encoding of 4K resolution video with the SHM encoder in spatial scalability a speedup of

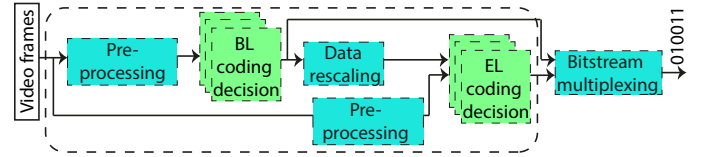


Fig. 3: SHM encoder processing.

40 to 80 is required depending on the coding configuration (Intra/Inter). In addition, these solutions, that use coding decisions of the BL encoder at the EL encoder, can not be integrated in the context of professional encoders since as depicted in the SHVC extension only the decoded BL frame and associated MVs shall be available at the EL encoder without the coding decisions. To cope with this inconvenience, we propose an end-to-end solution that takes into account the real-time character, imperative for broadcast application. Hence, in this paper we focus on the software implementation for real time SHVC encoder and decoder on multi-core Intel x86 platform. The SHVC encoder is based on the professional *ATEME* core encoder, which includes SIMD instructions for Intel x86 platform, algorithmic optimizations and parallelism. The real time SHVC decoder is based on the core HEVC decoder, *openHEVC*, which optimizes the most time consuming operations in SIMD for x86 platform and takes advantage of multicore processor to speed-up the decoding process through tile, wavefront and frame parallelisms. To the best of our knowledge, there is no SHVC codec, except the SHM [39] developed by the JCT-VC, to evaluate the proposed algorithmic contributions. In addition, this latter is not dedicated to real time processing.

The SHM encoder enables a high rate-distortion performance since it relies on the full search rate-distortion optimization at the expense of coding speed. Moreover, SHM does not include low level optimizations neither uses parallel processing. Fig. 3 illustrates the sequential architecture of the SHM encoder encoding two video layers. First, the input video is pre-processed, which corresponds to down-sampling in the spatial scalability, and then encoded with the BL encoder. The decoded BL frame is then processed by the *data rescaling* block to rescale BL output data. This block performs up-sampling operation of the decoded frame and MV up-scaling in spatial scalability. Finally, the EL encoder block encodes the original video with using the decoded BL input as an additional reference frame. We can notice that the SHM software performs these four main encoding operations in sequential order which would increase both the encoding time and end-to-end latency compared to a fully pipelined architecture on multi-core platform. The proposed real time SHVC codec is compared in this paper to the SHM codec in terms of rate-distortion performance for the encoder, speed-up and processed frames per second (fps) for both encoder and decoder.

III. PROPOSED REAL TIME SHVC CODEC

A. Real time SHVC decoder

The SHVC decoder consists of multiple instances of the *OpenHEVC* HEVC decoder, where each instance decodes

one SHVC layer. In the proposed architecture, the SHVC pixel's up-sampling and MV up-scaling operations are carried-out at the block level by the EL decoder. This architecture enables both fast and low latency decoding since only blocks used as reference are up-sampled in spatial scalability and efficient parallel decoding is performed between layers. The up-sampling operation that consists in 8-tap filter for Luma and 4-tap filter for chroma components are optimized in SIMD instructions for Intel x86 and embedded ARM Neon processors. Moreover, the most complex HEVC decoding operations including Discrete Cosine Transform (DCT)/Discrete Sine Transform (DST) transforms and Motion Compensation filters are optimized in the core HEVC decoder (*OpenHEVC*) for these two platforms. The *OpenHEVC* decoder supports the wavefront, tile and frame-based parallel processing solutions enabling to decode CTU rows, tiles and frames in parallel, respectively. The wavefront and tile parallel processing in the core *OpenHEVC* decoder can be activated for all SHVC layers when these two tools are enabled by the encoder, respectively. The frame based parallel decoding mechanism in the core *OpenHEVC* decoder has been extended to support parallel decoding of frames from different layers.

Fig. 4 illustrates the frame based parallel decoding of two SHVC video layers encoded in $2\times$ spatial scalability. In total, six frames (three at each layer) are decoded in parallel with inter and inter-layer control mechanisms to ensure that the block used as reference is available (already decoded) to perform inter and inter-layer predictions. In the case where the block used as reference is not available (not yet decoded), the threads of the depending blocks wait until the reference block is decoded. Therefore, once one thread completes the decoding of the block, it wakes up all threads waiting for this block. Moreover, while the EL frame is not fully decoded, the reference BL frame is not released since it can be used as reference by the EL decoder. The proposed decoder supports several scalability including spatial, quality, color gamut, bitdepth and codec with BL coded by the Advanced Video Coding (AVC) standard [40].

B. Real time HEVC encoder

The proposed SHVC encoder (SHVC-AE) relies on the core HEVC software encoder (HEVC-AE) developed by *ATEME*. As for the SHVC decoder, SHVC-AE instantiates multiple instances of the core HEVC-AE to encode the SHVC layers. The software HEVC-AE is also optimized in SSE2 instructions to speed-up, on Intel platform, the main HEVC coding operations including Intra prediction, motion compensation filters, DCT/DST transforms and in-loop filters. The encoding steps in the HEVC-AE involving video acquisition, pre-processing, Group of Pictures (GOP) construction and coding decision are pipelined as illustrated in Fig. 5. The first step manages the video acquisition from a file or from an external device (camera, Serial Digital Interface (SDI) card). Then, the pre-processing step adapts the input source video format to the encoder input format including color conversion and bitdepth adaptation. The GOP construction module affects to each picture a specific Picture Order Count (POC). Then,

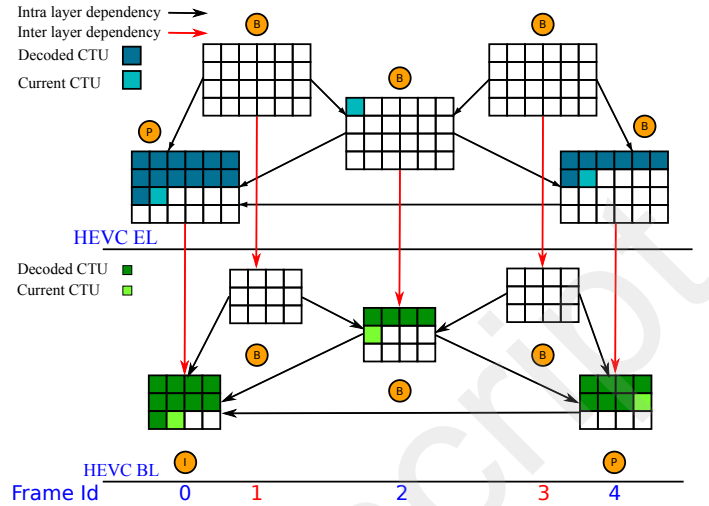


Fig. 4: Frame-based parallel decoding in the scalable *OpenHEVC* decoder.

the bitrate estimation module estimates the bitrate allocated to each picture to follow the target bitrate for the highest video quality. This step may introduce a latency depending on the GOP configuration since all pictures of the GOP are required. Finally, the coding decision step performs the rate-distortion minimisation over a pre-defined set of HEVC coding configurations ending up with the most efficient coding tools within the considered set:

$$\{C_k^*\}_{k=1}^M = \arg \min_{\{\vec{C}_k\}_{k=1}^M} \sum_{i=1}^M (J_{i|\vec{C}_i}) \quad (1)$$

where M is the number of coding parameters, \vec{C}_k the set of all coding configurations tested for the coding parameter k and J is the RD cost to minimize computed by Equation (2) with λ , D and R are the Lagrangian parameter, the distortion and the bitrate, respectively.

$$J = D + \lambda \cdot R. \quad (2)$$

The number of configurations H to be tested by the encoder is equal to the $M - 1$ multiplications between the number of configurations of the N parameters expressed as follows:

$$H = \prod_{i=1}^M \dim_K (\vec{C}_i). \quad (3)$$

The coding decision is the most complex step within the HEVC-AE pipeline. We can notice in Fig. 5 that the coding decision takes more than one real time cycle. To support a real time encoding the duration of this step should be lower than the duration of one frame (real time cycle equal to $\frac{1}{\text{video fps}}$ in second)

Three different optimizations are carried-out to reduce the coding decision duration to fill within a real time cycle. The first one consists in SIMD optimization of the most complex coding operations including Intra prediction, motion compensation filters, DCT/DST and in-loop filters. The second optimization consists in the definition of restricted sets of

TABLE I: Coding configurations of *FILE*, *LIVE* HD and *LIVE* UHD HEVC-AE setups.

Coding configurations	<i>FILE</i> setup	<i>LIVE</i> HD setup	<i>LIVE</i> UHD setup
CU sizes	64x64, 32x32, 16x16, 8x8	32x32, 16x16, 8x8	32x32, 16x16
PU partitions	2Nx2N, 2NxN, Nx2N, NxN	2Nx2N	2Nx2N
TU sizes	32x32, 16x16, 8x8, 4x4	16x16, 8x8	16x16, 8x8
Intra modes	DC, Planar, 27 directions	DC, Planar	DC, Planar
Inter modes	11	3	3
Number of reference frames by list	8	1	1
Split mode on PU sizes	16x8, 8x16, 8x8	-	-

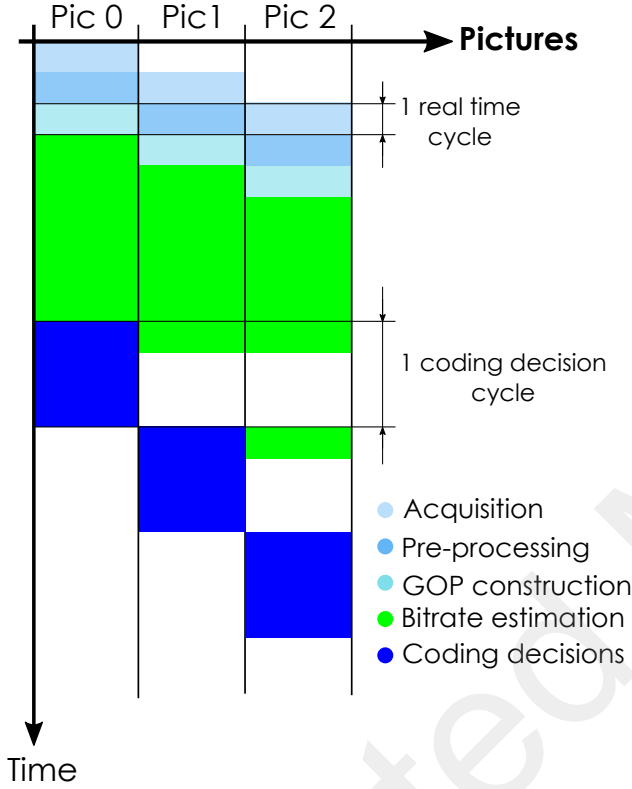


Fig. 5: Pipeline of the encoding steps in the HEVC-AE.

coding configurations to be tested by the encoder. This optimization enabled to define three coding setups named *FILE*, *LIVE* HD and *LIVE* UHD. The *FILE* setup considers a large set of coding configurations targeting a high video quality at the expense of coding speed performance, while the *LIVE* HD and *LIVE* UHD setups test reduced coding configurations favouring coding speed to fulfil real time requirements of HD and UHD resolutions, respectively. TABLE I gives the tested coding tools sets for *FILE*, *LIVE* HD and *LIVE* UHD setups. The complexity reduction of HEVC encoders has been widely investigated in the literature [33], the derivation of these setups is not investigated in this paper, which focuses more on the parallel and optimized software implementation of the SHVC codec.

The third optimization considers parallel processing at different levels of the encoder to take advantage of multi-core platforms. The core HEVC-AE supports the Tile parallel processing defined in the HEVC standard. HEVC-AE can

process in parallel multiple independent rectangular regions (Tiles) of one frame. This will speed-up the coding decision step at the expense of slight coding performance loss caused by Tile partitioning. The Tile parallel processing will be activated only in *LIVE* setups.

The second level of parallelism, called CTU-parallelism, enables to process the CTU rows of the frame in parallel. The CTU-parallelism is different from the wavefront parallelism proposed in HEVC [13] in the way that the entropy engine is not initialized at each CTU row. This improve the coding efficiency of the Context-Adaptive Binary Arithmetic Coding (CABAC) engine, since it is not initialized, at the expense of memory increase. In fact, the CTU-parallelism performs all encoding operations of the CTU rows in wavefront except the CABAC which is performed in sequential order once the coding of all CTU rows is completed. This solution increases the memory usage since all coding decisions are stored and then processed by the CABAC engine once the last CTU of the previous row is encoded (CABAC context is available).

The third level of parallelism is performed between the coding decision steps of different frames. Several frames are encoded in parallel where the main process manages the inter-frame dependencies ensuring that the block used as reference is available within the reference frame. The main process (manager) launches the frame encodings in parallel (threads) and manages all communications between concurrent threads. The frame-based parallelism speed-up the encoding process without impacting the coding quality and then can be activated as for the CTU-parallelism parallelism in *FILE*, *LIVE* HD and *LIVE* UHD setups.

C. Real time SHVC encoder

The SHVC-AE creates multiple instances of the core HEVC-AE to encode the SHVC layers. The support of the SHVC standard introduces two new operations to the core HEVC-AE: down-sampling and up-sampling operations. The down-sampling operation enables, in spatial scalability, to build from the source video the frames to be encoded by the BL encoder, while the up-sampling operation creates, from the decoded BL frame, the frame used by the EL encoder as reference for inter-layer predictions. Fig. 6 shows the pipeline of the coding steps in the SHVC-AE encoding two layers. The down-sampling and up-sampling operations illustrated in red and pink colors are performed by the BL and EL encoders, respectively. To perform inter-layer prediction, the EL requires the coding information from the BL. This means the beginning

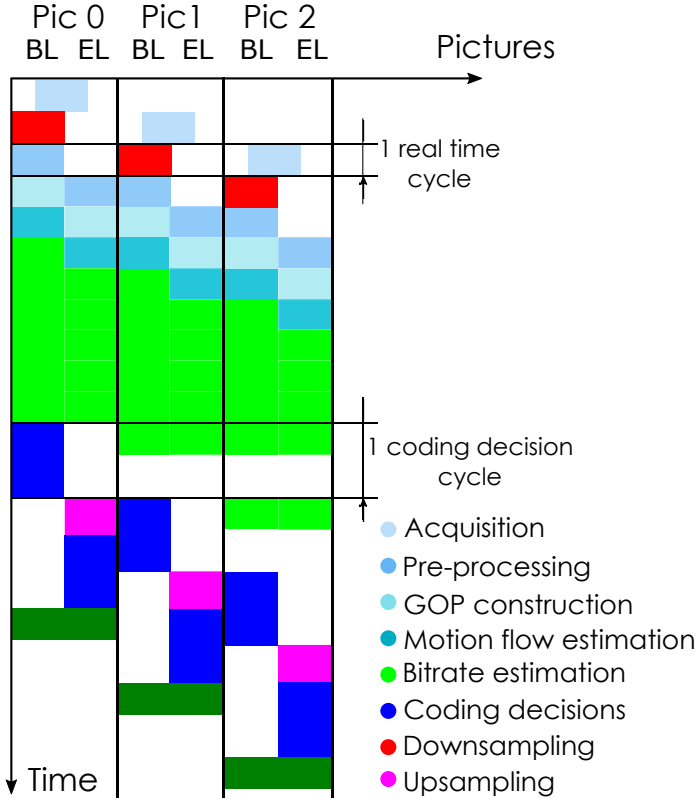


Fig. 6: Pipeline of the encoding steps in the SHVC-AE.

of the coding decision on the EL needs to be synchronized with the end of the coding decision on the BL. Therefore, a latency of three cycles is introduced corresponding to the down-sampling step, the inter-layer synchronization and the up-sampling step. We can also notice from Fig. 6 that the durations of both up-sampling and down-sampling operations are higher than one real time cycle. Two optimizations are proposed to speed-up these operations including SIMD optimization and parallelism. The up-sampling is standard operation and consists in 8 tap and 4 tap filters for luma and chroma components, respectively. The down-sampling is not standard operation and is also carried-out in this paper with 8 tap and 4 tap filters for luma and chroma components. These two operations are performed with a convolution product between the pixels and the filter coefficients:

$$s_m = \sum_{i=-\lceil W/2 \rceil-1}^{\lceil W/2 \rceil} c_i + \lceil W/2 \rceil - 1 \cdot p_{m+i} \quad (4)$$

with c_i is the filter coefficients, p_m the pixel value at position m , s_m the output of the filter at position m and W the size of the filter. In this paper, W is equal to 8 and 4 for Luma and Chroma components, respectively.

The 2D convolution product requires 8 multiplications and 7 additions in horizontal and vertical directions. SSE3 instructions define several functions to perform arithmetic operations on registers of sizes 64 and 128 bits. The 8 tap filter for 8 luma positions (pixels) can be performed only by 4 multiplications (`_mm_maddubs_epi16`) and three additions (`_mm_add_epi16`) on 64 bits and 128 bits for 8 and 10

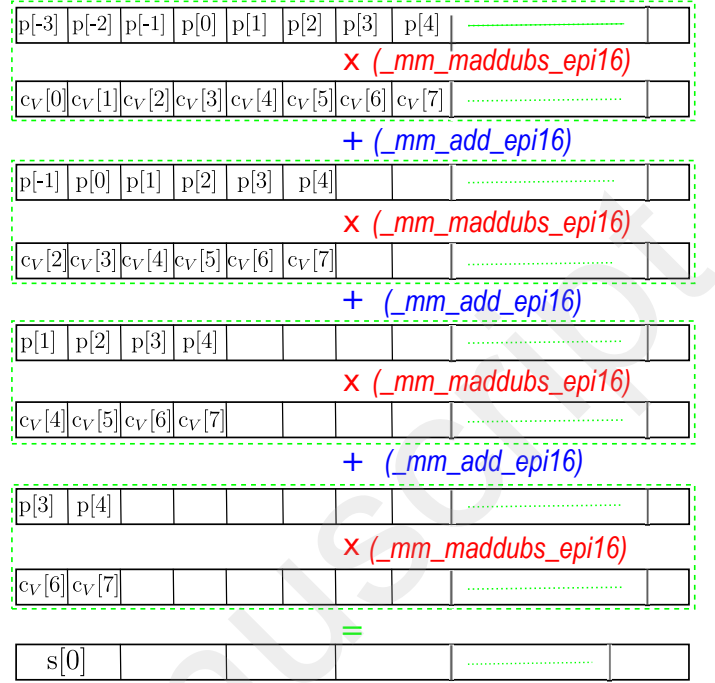


Fig. 7: Convolutional product optimized in SSE instructions with buffer sizes of 64 and 128 bits.

bitdepth, respectively. Fig. 7 illustrates horizontal 8 tap filters performed by SSE3 instructions. The down-sampling and up-sampling operations can also be conducted in parallel on multi core processors. To optimize the memory access, we propose to process the three color components in parallel. Moreover, the frame of each component is partitioned in four horizontal regions (two for chroma) of similar height equal to the frame height / 4 (frame height / 2 for chroma) and the width of the frame. These four regions are also processed in parallel resulting in 8 threads (4 for luma and 4 for chroma) running in parallel for both up-sampling and down-sampling processes. It should be noted that this partitioning does not have an impact on the RD performance since it is only used for parallel processing of the up-sampling process.

TABLE II gives the coding gains in terms of Bjøntegaard Delta Bit Rate (BD-BR) metric [41], [42] of the SHM with respect to single-layer coding configuration (ie. EL coded with SHM versus EL coded with HEVC). It provides the bitrate reduction of the EL when inter-layer prediction is activated on only I slices, I and P slices and I, P and B slices in Random Access (RA) coding configuration illustrated in Fig. 4. We can notice from TABLE II that inter-layer prediction on I slices brings 33 % of the total SHVC gain whereas I slices represent only 1 % to 4 % of the slices in RA bitstream depending on the video frame rate (fps). The P slices representing between 8 % and 11 % in RA bitstream bring 36% of the total SHVC gain. Finally, the B slices representing around 88 % of the slices in RA bitstream bring on average 30 % of the total SHVC gain. We can use these statistics to reduce the SHVC-AE complexity with a slight impact on the coding gain. We propose in this paper

TABLE II: Coding gains in terms of BD-BR of I, P and B slices with the SHM in RA coding configuration.

Videos	Gain with I-P and B slices	Gain with I-P slices	Gain with I slices
<i>Traffic</i>	−26.80 %	−24.10 %	−18.00 %
<i>PeopleOnStreet</i>	−45.10 %	−20.60 %	−7.70 %
<i>Kimono1</i>	−42.70 %	−29.34 %	−14.50 %
<i>ParksScene</i>	−27.70 %	−23.0 %	−15.80 %
<i>Cactus</i>	−28.00 %	−18.60 %	−6.60 %
<i>BasketballDrive</i>	−31.50 %	−14.60 %	−2.90 %
<i>BQTerrace</i>	−8.20 %	−7.00 %	−1.90 %
Average	−30.00 %	−19.61 %	−9.63 %
Gain vs. total gain	100 %	69.38 %	33.03 %

to disable inter-layer prediction in the SHVC-AE on the B slices of the highest temporal layer since these frames are not used as reference in inter prediction and bring the lowest SHVC gain (frames id 1 and 3 in Fig. 4). This optimization concerns only RA coding configuration enabling to speed-up the coding process since up-sampling is not performed on B slices of the highest temporal layer. Moreover, this technique also enables to decrease the decoder complexity since the B slices of the highest temporal layer are not up-sampled. In fact, the proposed SHVC decoder architecture up-samples only blocks used as reference for inter-layer prediction.

The SHVC-AE inherits the parallelism from the core HEVC-AE. The SHVC-AE encoders each layer in parallel and can also use Tile parallelism when running in *LIVE* setups. The CTU-parallelism can also be used on each Tile or on the whole frame to speed-up the video coding processing in both *FILE* and *LIVE* setups since it does not reduce the compression performance. Moreover, the frame-based parallelism is also extended to process in parallel the BL and EL of several frames. The main manager can launch the encoding in parallel of several SHVC frames with synchronisation between concurrent encodings. It should be noted that the BL and EL of one frame are always processed in sequential order and only other operations in the pipeline are carried-out in parallel between layers of one frame. For the SHVC-AE, we define three setups: *FILE* and *LIVE* that use the *FILE* and *LIVE* UHD single-layer encoder setups at both layers, respectively as well as *LIVE+* setup that uses *LIVE* HD setup on the BL and *LIVE* UHD one on the EL. TABLE III summarizes the activated coding tools in the three considered setups for the proposed SHVC-AE.

IV. RESULTS AND PERFORMANCE EVALUATION

A. Experimental setup

The experimental tests for the SHVC-AE have been carried out on a 4×10-cores Intel Xeon processor (E5-4627V3) running at 2.6 GHz. Several test video sequences from the SHVC CTC and 4-EVER French collaborative project (*Brest*), described in TABLE IV, have been considered in this study. These videos are encoded with the SHVC reference software (SHM) encoder and the proposed *ATEME* SHVC

TABLE III: Configurations of the SHVC-AE.

Videos	SHVC-AE <i>FILE</i>	SHVC-AE <i>LIVE+</i>	SHVC-AE <i>LIVE</i>
HEVC-AE BL setup	<i>FILE</i>	<i>LIVE</i> HD	<i>LIVE</i> UHD
HEVC-AE EL setup	<i>FILE</i>	<i>LIVE</i> UHD	<i>LIVE</i> UHD
Tiles	OFF	ON	ON
CTU-parallelism	ON	ON	ON
parallelism			
Frame-based parallelism	ON	ON	ON
Disable ILP on B-slices RA	ON	ON	ON

TABLE IV: Test video sequences.

Sequence	Name	Resolution	Frame rate (fps)	number of frames
UHD1	<i>PeopleOnStreet</i>	3840×2160	30	150
UHD2	<i>Brest</i>	3840×2160	60	600
A1	<i>Traffic</i>	2560×1600	30	150
A2	<i>PeopleOnStreet</i>	2560×1600	30	150
B1	<i>Kimono1</i>	1920×1080	24	240
B2	<i>ParksScene</i>	1920×1080	24	240
B3	<i>Cactus</i>	1920×1080	50	500
B4	<i>BasketballDrive</i>	1920×1080	50	500
B5	<i>BQTerrace</i>	1920×1080	60	600

encoder (SHVC-AE) in three setups *FILE*, *LIVE* and *LIVE+*. The videos are encoded in 2× spatial scalability, eight Quantization Parameter (QP)s: $(QP_{BL}, QP_{EL}) \in \{(22, 22), (22, 24), (26, 26), (26, 28), (30, 30), (30, 32), (34, 34), \text{ and } (34, 36)\}$ and three GOP coding configurations: All Intra (AI), Low Delay P (LD, P) and RA. The performance of the SHVC-AE is assessed in terms of coding speed in fps, speed-up compared to SHM and rate-distortion with respect to SHM and HEVC-AE single-layer using the BD-BR metric [41], [42]. The proposed real time SHVC decoder is assessed in terms of decoding frame rate in fps and speed-up compared to the reference SHM decoder. The performance of the decoder is carried out on two platforms: laptop fitted with 4-core Intel i7-6820HQ CPU for both layers and octacore Exynos 5410 System on Chip (SoC) for the BL resolution. This SoC is based on the big.LITTLE configuration including a cluster of 4 ARM Cortex-A15 cores and a cluster of 4 ARM Cortex-A7 cores. The Tile parallelism activates in *LIVE* setups splits the video frame in 4 tiles (2×2) of the same size.

B. The proposed SHVC-AE *FILE*

TABLE V gives the performance of the SHVC-AE *FILE* in terms of BD-BR with respect to the reference SHM encoder in AI coding configuration. The first column shows the bitrate saving of the SHVC-AE EL with respect to HEVC-AE encoding the EL in single-layer configuration. The inter-layer prediction in the proposed SHVC-AE enables on average 36 % bitrate reduction while SHM enables 34.5%. The inter-layer prediction is more efficient in SHVC-AE than in the SHM since the two single encoders in SHM are

TABLE V: BD-BR performance of the SHVC-AE EL in comparison with HEVC-AE, SHM EL with HM and SHVC-AE EL with SHM EL in AI coding configuration.

Sequences	BD-BR		
	SHVC-AE vs HEVC-AE	SHM vs HM	SHVC-AE vs SHM
A1	-42.6 %	-40.2 %	3.9 %
A2	-46.7 %	-44.1 %	3.2 %
B1	-50.5 %	-48.7 %	7.6 %
B2	-35.8 %	-34.2 %	5.1 %
B3	-31.9 %	-31.1 %	7.1 %
B4	-26.1 %	-25.7 %	10.7 %
B5	-18.4 %	-17.6 %	7.5 %
Average	-36.0 %	-34.5 %	6.4 %

more efficient in terms of compression than the HEVC-AEs encoding the two layers. This lower coding performance is mainly caused by the restrictions of coding tools set in the core HEVC-AE *FILE*. Therefore, SHVC-AE uses more inter-layer prediction compared to the SHM which has more efficient Intra coding tools used to encode the two layers. The last column in TABLE V shows that the reference SHM encoder outperforms the proposed SHVC-AE by 6.2 % on average in terms of BD-BR, which is mainly caused by restrictions in the *FILE* setup.

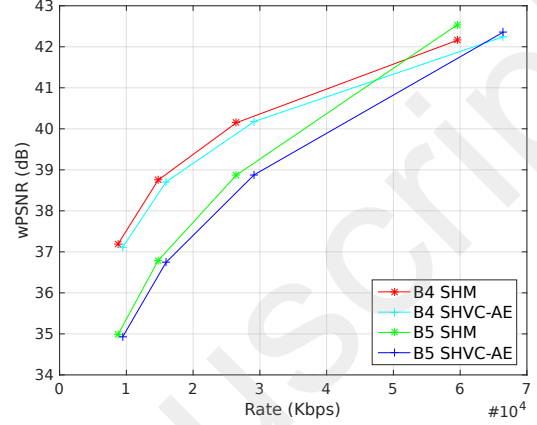
TABLE VI gives the performance of the SHVC-AE *FILE* in terms of BD-BR in comparison with the reference SHM encoder in LD. P coding configuration. The inter-layer prediction enables a bitrate saving of 30.4 % on average while SHM reference encoder reaches 53.6 %. This difference is mainly introduced by the restriction on intra and inter coding tools in the proposed SHVC-AE. Moreover, the restriction on inter coding tools also impacts the inter-layer prediction efficiency since the same tools are used for both inter and inter-layer predictions.

TABLE VII gives the performance of the SHVC-AE in terms of BD-BR in comparison with the reference SHM encoder in RA coding configuration. In this coding configuration, the inter-layer prediction enables a bitrate reduction on average of 18.5 % and 24.6 % for SHVC-AE and SHM encoders, respectively. As in RA configuration, the loss in coding efficiency of the SHVC-AE compared to the reference SHM encoder is mainly caused by restricted coding tools in the *FILE* configuration. In addition, disabling the inter-layer prediction for the B slices of the highest temporal layer also decreases the inter-layer gain impacting the global coding performance of the SHVC-AE.

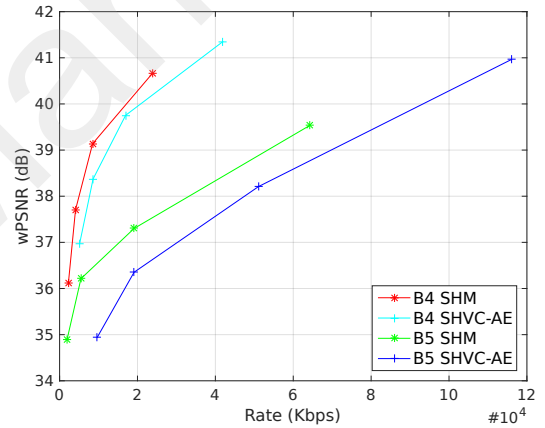
We can also notice from these results that the gain brought by the inter-layer prediction depends on the characteristics of the video sequence including spatial and temporal informations as well as its resolution.

Fig. 8 shows the weighted PSNR ($wPSNR = (6 \cdot YPSNR + UPSNR + VPSNR)/8$) performance versus the bitrate of the proposed SHVC-AE and SHM encoder in the three coding configurations for *BasketballDrive* and *BQTerrace* video sequences. The difference between the curves of

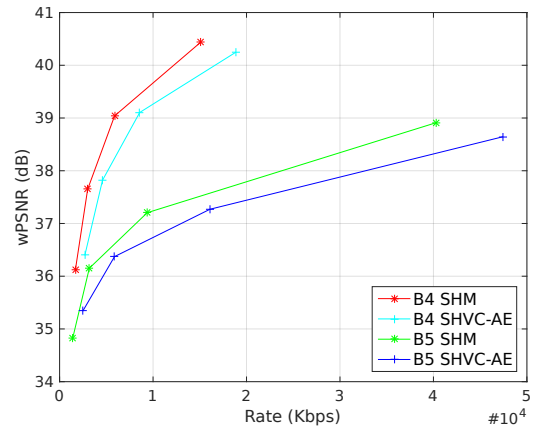
the two encoders remains similar at the four printed bitrates. Moreover, this difference is higher in LD. P configuration at all bitrates which explain the high bitrate loss in LD. P configuration especially for *BQTerrace* video (B5).



(a) AI configuration



(b) LD. P configuration



(c) RA configuration

Fig. 8: Rate-distortion performance of the SHM and SHVC-AE encoders using three GOP coding configurations, in *FILE* setup, for *BasketballDrive* (B4) *BQTerrace* (B5) videos.

TABLE VIII gives the speed-up performance of the SHVC-

TABLE VI: BD-BR performance of the SHVC-AE EL in comparison with HEVC-AE, SHM EL with HM and SHVC-AE EL with SHM EL in LD. P coding configuration.

Sequences	BD-BR		
	SHVC-AE vs	SHM vs	SHVC-AE vs
	HEVC-AE	HM	SHM
A1	-34.4 %	-69.3 %	166.5 %
A2	-37.0 %	-49.1 %	48.9 %
B1	-40.5 %	-54.9 %	61.6 %
B2	-29.3 %	-59.8 %	109.1 %
B3	-28.9 %	-59.2 %	126.2 %
B4	-24.2 %	-38.1 %	50.3 %
B5	-19.5 %	-44.5 %	130.8 %
Average	-30.4 %	-53.6 %	99.0 %

TABLE VII: BD-BR performance of the SHVC-AE EL in comparison with HEVC-AE, SHM EL with HM and SHVC-AE EL with SHM EL in RA coding configuration.

Sequences	BD-BR		
	SHVC-AE vs	SHM vs	SHVC-AE vs
	HEVC-AE	HM	SHM
A1	-19.1 %	-21.7 %	31.0 %
A2	-27.7 %	-36.6 %	39.2 %
B1	-30.2 %	-38.1 %	43.4 %
B2	-17.1 %	-22.5 %	33.1 %
B3	-15.0 %	-22.3 %	35.9 %
B4	-12.5 %	-24.8 %	45.8 %
B5	-8.2 %	-6.3 %	59.2 %
Average	-18.5 %	-24.6 %	41.1 %

AE compared to both single-layer HEVC-AE and the SHM encoder in the three considered GOP configurations. The speed-up of an encoder of encoding time $EC1$ with respect to the reference encoder of encoding time $EC2$ is computed as follows:

$$Sp = \frac{EC2}{EC1} \cdot 100 \% \quad (5)$$

The speed-up of the SHVC-AE in AI configuration is on average around 44 % compared to the Single Layer (SL) encoding. The SHVC-AE is almost two times slower than the HEVC-AE encoding the equivalent EL. The complexity of the SHVC-AE, with respect to the single layer HEVC-AE, is caused by the additional processing introduced by the SHVC extension including the up-sampling and down-sampling operations as well as the encoding of the BL. For LD. P and RA coding configurations, the speed-up is on average 90 % and 98 %, respectively. The complexity increase versus single-layer encoder is significantly reduced in these two inter coding configurations since both single-layer and scalable encoders use inter predictions. The slight complexity increase is related in these configurations to the BL encoding as well as up-sampling and down-sampling operations. On the other hand, the speed-up of the proposed SHVC encoder with respect to SHM is on average equal to 3750, 6690 and 6650 in the three considered configurations. The different optimizations and parallel processing introduced at the level of the core HEVC-AE and its scalable extension in the *FILE* setup enable to speed-up the encoder by 37 times in AI configuration and

TABLE VIII: speed-up (Sp) performance in % of the SHVC-AE compared to both single-layer SHVC-AE and SHM encoder

Sequences	SHVC-AE vs HEVC-AE			SHVC-AE vs SHM		
	AI	LD. P	RA	AI	LD. P	RA
A1	43	88	96	5190	6390	7060
A2	39	89	99	5280	9110	10200
B1	34	94	98	2600	6460	6060
B2	40	88	97	3950	5940	54.5
B3	46	91	98	3100	6200	5870
B4	55	92	99	3040	6700	6280
B5	51	90	98	3110	6010	5660
Average	44	90	98	3750	6690	6650
SHVC-AE frame rate (fps)	1.33	0.96	0.98	-	-	-

TABLE IX: BD-BR performance (%) of the SHVC-AE *LIVE* and *LIVE+* in comparison with the single-layer HEVC-AE *LIVE*

Seq.	SHVC-AE <i>LIVE</i> vs			SHVC-AE <i>LIVE+</i> vs		
	HEVC-AE <i>LIVE</i>			HEVC-AE <i>LIVE</i>		
	AI	LD. P	RA	AI	LD. P	RA
UHD1	-53.9%	-43.3%	-33.5%	-55.8%	-45.3%	-35.2%
UHD2	-31.0%	-21.0%	-1.4%	-32.1%	-21.7%	-1.8%
A1	-49.8%	-36.7%	-18.9%	-51.4%	-37.9%	-19.8
A2	-53.9%	-42.1%	-32.5%	-56.4%	-44.8%	-35.1
B1	-51.1%	-35.3%	-30.2%	-55.9%	-38.3%	-29.3
B2	-40.9%	-28.2%	-13.3%	-42.4%	-28.9%	-13.8
B3	-40.7%	-32.7%	-17.7%	-43.1%	-34.4%	-18.3
B4	-34.6%	-24.6%	-15.5%	-39.5%	-29.0%	-17.8
B5	-29.5%	-20.0%	-1.8%	-31.0%	-21.2%	-3.4%
Av.	-42.8%	-31.5%	-18.3%	-45.3%	-33.5%	-19.4%

66 times in the two inter coding configurations LD. P and RA. Therefore, the *FILE* setup of the proposed SHVC-AE enables a high RD performance with an efficient use of the inter-layer prediction and an interesting speed-up compared to the SHM encoder. The last row of TABLE VIII gives the average encoding frame rate in fps of the proposed SHVC-AE in *FILE* setup. We can notice that the frame rate is around 1 fps in the three coding configurations which is far from real time performance. Therefore, this setup can be used only for offline encoding on the cloud to reach a high video quality but it does not enables real time encoding of live HD/UHD video broadcasting.

C. The proposed SHVC-AE *LIVE* & *LIVE+*

To reach a real time performance, we proposed two setups of the SHVC-AE: *LIVE* which uses *LIVE* UHD setup of the single-layer encoder HEVC-AE for both layers, and *LIVE+* setup which uses *LIVE* HD setup of the single-layer encoder for the BL and *LIVE* UHD setup for the EL. TABLE IX provides the BD-BR performance of the SHVC-AE *LIVE* and *LIVE+* in comparison with the single-layer HEVC-AE *LIVE* UHD. The average results show that the SHVC-AE benefits well from inter-layer prediction in both *LIVE* and *LIVE+* setups with a BD-BR

TABLE X: BD-BR performance of the SHVC-AE *LIVE* and *LIVE+* in comparison with SHVC-AE *FILE* .

Seq.	SHVC-AE <i>LIVE</i> vs SHVC-AE <i>FILE</i>			SHVC-AE <i>LIVE+</i> vs SHVC-AE <i>FILE</i>		
	AI	LD. P	RA	AI	LD. P	RA
UHD1	24.9 %	39.4 %	49.2 %	18.8 %	28.9 %	36.4 %
UHD2	25.4%	37.3%	55.9%	20.9%	32.2%	49.9%
A1	23.3 %	33.3 %	50.2 %	17.0 %	25.8 %	40.7 %
A2	28.4 %	44.6 %	54.0 %	21.3 %	32.5 %	39.5 %
B1	24.8 %	38.2%	49.2 %	14.7 %	26.5 %	40.1 %
B2	14.6 %	26.6 %	45.5 %	11.1 %	21.0 %	36.9 %
B3	23.7 %	37.4 %	66.8 %	18.2 %	29.2 %	54.9 %
B4	41.0 %	58.8 %	69.2 %	24.7 %	41.7 %	55.9 %
B5	20.7 %	39.8 %	98.6 %	16.6 %	34.5 %	88.1 %
Av.	25.2 %	39.5 %	59.8 %	18.1 %	30.3 %	49.2 %

savings of 42.8 %, 31.5 % and 18.3 % in the three coding configurations for *LIVE* setup and 45.3 %, 33.5 % and 19.4 % for *LIVE+* setup in comparison with a single-layer HEVC-AE in *LIVE* UHD setup encoding the EL. Therefore, the *LIVE+* setup of the SHVC-AE enables to benefit more from the inter-layer prediction compared to the *LIVE* setup since the BL is of higher quality when encoded in *LIVE* HD single-layer encoder.

TABLE X gives the BD-BR performance of the SHVC-AE *LIVE* and *LIVE+* in comparison with SHVC-AE *FILE* . We can notice that the restrictions in *LIVE* UHD setup at both layers (SHVC *LIVE* setup) and *LIVE* UHD setup at the only EL (SHVC *LIVE+* setup) significantly reduce the rate-distortion performance by 25.2 %, 39.5 %, 59.8 % and 18.1 %, 30.3 %, 49.2 % respectively in the three coding configurations. The SHVC-AE in *LIVE+* setup has higher performance than *LIVE* setup enabled by the higher efficiency of the BL encoder in *LIVE* HD setup resulting in more efficient inter-layer predictions.

TABLE XI gives the encoding frame rate performance of the SHVC-AE in *LIVE* and *LIVE+* setups. We can notice that the two *LIVE* and *LIVE+* setups enable almost the same coding frame rate performance. In fact, the additional complexity of the *LIVE+* setup is introduced by the *LIVE* HD setup at the BL which represents less than 10% of the whole scalable encoder complexity in LD. P and RA coding configurations. Moreover, we can also notice that both configurations enable real time encoding of all considered video sequences even with a 3840×2160p 30 fps format.

D. The proposed OpenHEVC decoder

The decoding frame rate (in fps) performance of the *OpenHEVC* decoder on 4 cores i7 laptop is provided in TABLE XII for three encoders including SHVC-AE *FILE* and *LIVE+* setups and SHM in three coding configurations: AI, LD. P and RA. We can notice that on average the decoder reaches a real time decoding of 3840×2160p 30 fps videos in inter configurations (LD. P and RA) and higher than 60

TABLE XI: Encoding frame rate performance of the SHVC-AE *LIVE* and *LIVE+* .

Seq.	Encoding time in fps					
	SHVC-AE <i>LIVE</i>			SHVC-AE <i>LIVE+</i>		
	AI	LD. P	RA	AI	LD. P	RA
UHD1	50.1	41.8	41.6	50.1	42.4	41.6
UHD2	46.3	41.8	43.6	46.3	41.9	43.6
A1	94.5	75.5	73.1	94.4	75.5	73.0
A2	95.6	79.2	79.3	94.2	78.9	79.0
B1	190.3	140.8	149.9	189.1	140.1	148.7
B2	172.6	134.7	144.1	169.3	134.0	142.6
B3	178.1	140.9	151.4	178.5	140.0	150.5
B4	181.7	139.3	147.8	180.3	139.4	147.2
B5	162.7	132.1	145.8	163.7	131.8	144.7
Av. UHD	48.2	41.8	42.6	48.2	42.1	42.6
Av. A	95	77.3	76	94.3	77.2	76
Av. B	177	137.5	147.8	176.8	137	146.7

fps and 107 fps for videos of classes A and B (LD. P and RA), respectively. The decoding performance is on average slightly higher for SHM bitstream since the reference encoder decreases the bitstream size compared to the proposed *ATEME* encoders leading to lower complexity at the decoder side. For high bitrate configurations (min), the real time decoding is not reached for videos in UHD and 2K (class A) resolutions. The RA coding configuration leads to the fastest decoding performance since this configuration enables the highest RD coding performance and the up-sampling operation is not performed on the highest temporal B slices.

TABLE XIII provides the decoding frame rate of the proposed *OpenHEVC* decoder decoding the BL on ARM mobile platform for bitstreams encoder with the SHVC-AE *LIVE* and *FILE* in the three coding configurations. We can notice that the decoder enables real time decoding of the BL in full HD resolution (1920×1080p) for RA configuration on embedded ARM platform. Moreover, the BL is decoded in real time for videos of classes A and B in the three coding configurations.

V. REAL TIME UHD HDR VIDEO DEMONSTRATION

The proposed encoder and decoder enable real time processing of UHDp30 video sequences on the considered platforms for RA coding configuration. This allows the integration of the solution into a broadcast channel context. In our demonstration, as illustrated in Fig. 9, we consider a broadcast context composed of a camera or a streamer, SHVC encoder, SHVC decoder and both a UHD HDR compliant TV screen and a smartphone with an HD SDR screen. This set-up simulates a stream application with an end-to-end transmission on cable and network.

First, the camera or the streamer sends the captured uncompressed video to the SHVC encoder through an SDI link. SDI is a standard enabling to transfer uncompressed video on cable. The SDI link retained for the experiment is composed of 4×3G SDI cables allowing a 12 Gbps maximum bit-rate. To send uncompressed UHD contents in 4:2:2 format, a bitrate amount inferior to 10 Gbps is required, as expressed on the

TABLE XII: Decoding frame rate performance of the *OpenHEVC* decoder on 4 cores i7 laptop decoding SHVC bitstreams encoded by SHM encoder and SHVC-AE in *LIVE* and *LIVE+* setups.

Seq.	Decoding time in fps											
	SHVC-AE <i>FILE</i>			SHVC-AE <i>LIVE</i>			SHVC-AE <i>LIVE+</i>			SHM		
	AI	LD. P	RA	AI	LD. P	RA	AI	LD. P	RA	AI	LD. P	RA
UHD1	26	30	40	26	35	45	30	32	37	26	26	40
UHD2	20	30	49	24	35	46	24	33	45	28	28	44
A1	50	79	123	54	88	112	55	83	112	56	71	123
A2	48	59	77	58	65	79	52	67	80	55	48	73
B1	138	170	227	85	117	155	86	112	147	162	139	226
B2	89	135	187	71	110	145	64	112	150	105	113	198
B3	99	149	233	70	98	167	68	104	168	115	143	215
B4	120	149	205	70	109	151	80	109	133	142	125	196
B5	87	123	214	60	104	179	60	101	177	100	119	198
Av. UHD	23	30	44	25	35	45	27	33	41	27	27	42
Av. A	49	69	100	56	77	96	53	75	96	56	60	98
Av. B	107	145	213	71	107	159	71	108	155	125	128	207
Max. UHD	41	55	80	44	59	71	43	57	68	51	43	73
Max. A	84	134	187	90	125	156	80	126	148	99	105	177
Max. B	204	271	406	120	169	273	120	172	268	240	235	344
Min. UHD	10	13	18	12	15	22	11	14	21	11	13	19
Min. A	21	30	45	25	34	40	28	39	47	23	27	47
Min. B	32	38	71	31	35	71	26	31	66	33	46	69

TABLE XIII: Decoding frame rate performance of the *OpenHEVC* decoder decoding the BL encoded by HEVC-AE in *FILE* and *LIVE* setups on mobile ARM platform

Seq.	Decoding time in fps							
	HEVC-AE BL <i>FILE</i>			HEVC-AE BL <i>LIVE</i>				
	AI	LD. P	RA	AI	LD. P	RA		
UHD1	20	27	33	22	27	34		
UHD2	18	27	40	18	29	41		
A1	36	59	78	42	54	71		
A2	33	48	57	38	53	61		
B1	62	123	105	64	137	108		
B2	56	103	88	60	106	95		
B3	52	100	93	59	109	96		
B4	57	101	99	63	126	96		
B5	50	90	102	57	101	109		
Av. UHD	19	27	36	20	28	37		
Av. A	34	53	67	40	53	66		
Av. B	55	103	98	61	116	101		
Max. UHD	26	36	53	29	41	53		
Max. A	49	79	101	55	79	81		
Max. B	67	167	140	74	166	146		
Min. UHD	13	19	24	13	19	24		
Min. A	26	34	42	29	37	45		
Min. B	41	63	74	46	68	80		

following calculations:

Bit-rate UHD1 ($3840 \times 2160p$ 30 *fps*) = $3840 \times 2160 \times 30 \times 2 \times 10 = 4.976$ Gbps

Bit-rate UHD2 ($3840 \times 2160p$ 60 *fps*) = $3840 \times 2160 \times 60 \times 2 \times 10 = 9.953$ Gbps

The device used to receipt the uncompressed video on the encoder side is the DTA-2174 produced by Dektec [43] which allows a $4 \times 3G$ SDI reception. Each uncompressed frame sent on the SDI link is in a V210 format. The V210 format

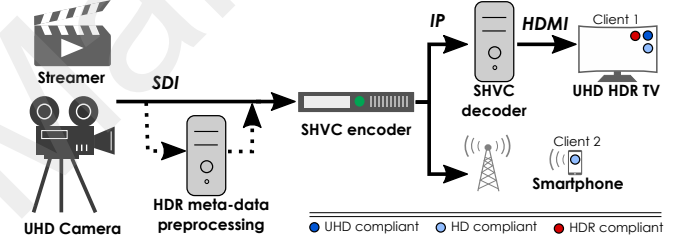


Fig. 9: SHVC Streaming in Real-Time Context.

consists of a 4:2:2 representation with 10 bits per pixels and each pixel of Luma (Y) and chroma (U and V) packed in a sequence such as: $U_0, Y_0, V_0, Y_1, U_2, Y_2, V_2, Y_3, \dots$. Once the DTA-2174 receipts a frame, this last one is converted to a 4:2:0 planar representation before encoding. As a result, on the input side of the encoder, the DTA-2174 is added and a conversion from V210 to 4:2:0 planar representation is processed. The DTA-2174 is embedded on the 4×10 cores Intel Xeon processor (E5-4627V3) where the SHVC-AE is also integrated. In the case of HDR coding, the uncompressed video is first sent to a preprocessing device before the encoder. This preprocessing device enables to produce the meta-data used for HDR displays. It can be used for different HDR technologies such as:

- Perceptual Quantizer (PQ) proposed by the Society of Motion Picture and Television Engineers (SMPTE) in specification ST-2084 defining a transfer function enabling HDR displays with 10 bits per pixel and a BT.2020 color-gamut,
- HDR10 exploiting PQ and specification SMPTE ST-2086 defining information transfer for color calibration in HDR displays with static size of meta-data.
- HDR10+ proposed by *Samsung* and *Amazon Video* ex-

exploiting specification SMPTE ST-2094-40 and enhancing HDR10 with dynamic size of meta-data.

- Dolby Vision proposed by *Dolby* similar to HDR10+ (using PQ and SMPTE ST-2094-40) but with luminosity adaptation for HDR displays on TV.
- Hybrid Log Gamma (HLG) proposed by the British Broadcasting Corporation (BBC) and Japan Broadcasting Corporation (NHK) in specification ARIB STD-B67 defining another transfer function for HDR displays with 10 bits per pixel and a BT.2020 color-gamut,
- SL-HDR1 proposed by *STMicroelectronics*, *Philips International* and *Technicolor* in specification ETSI TS 103 433 relying on SMPTE ST-2087, ST-2086, ST-2094-20 and ST-2094-30 with dynamic size of meta-data sends in a Supplemental Enhancement Information (SEI) message.

In the proposed application, we only use the SL-HDR1 for backward compatibility SDR-HDR but other HDR technologies can be employed for all layers. The SL-HDR1 meta-data is added to the SDI messages in ancillary data packets. Once received by the DTA-2174, the meta-data is put in an SEI message and passes through the encoding process.

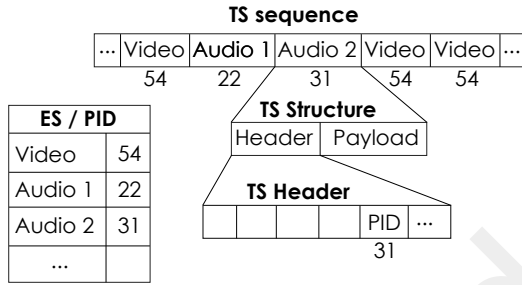


Fig. 10: Schematic of the MPEG-2 TS packets structure.

Then, the SHVC-AE, embedded on the 4×10 cores Intel Xeon processor (E5-4627V3), processes the received frame as explained in Section III-C. Once the encodings are performed, the SHVC bitstream is packed in MPEG-2 TS packets and sent to the decoder through an Internet Protocol (IP) link. As illustrated in Fig. 10, the MPEG-2 TS packet is composed of a payload containing the encoded bitstream, also called Elementary Stream (ES), and a header containing information on the payload. This information concerns, for instance, the type of transmitted data which can be video but also audio or subtitles... The type of data is identified thanks to the syntax element called Packet Identifier (PID). In the broadcast environment, there are two main specifications:

- Digital Video Broadcasting (DVB) used in Africa, Europe, Middle East, Oceania and South Asia,
- ATSC used in North America and South Korea.

They rely on standards such as HEVC for video coding or MPEG-2 TS for IP transmission. In the case of SHVC, they define different PID specifications: DVB recommends different PID for each scalable layer while ATSC recommends a single PID for the video ES. Our solution supports both solution and the default configuration uses the ATSC recommendation.

The MPEG-2 TS packets are then transferred to the *OpenHEVC* decoder through cables and to a smartphone through

network. For the display on TV, the *OpenHEVC* decoder is integrated to the GPAC player as proposed in [44] to manage the reception of MPEG-2 TS packets. Both BL and EL are decoded to enable UHD display. Once decoded, UHD frames are finally sent to the UHD HDR TV through a High-Definition Multimedia Interface (HDMI) link. If present, the SEI message containing information for HDR display passes through the decoder and are employed by the TV. Otherwise, the TV displays the UHD content in SDR. On the other hand, the smartphone receives the MPEG-2 TS packets through network and process only the BL for an HD display. The SEI message containing the information for HDR display are not employed by the smartphone and only SDR can be displayed.

The real time end-to-end video transmission of UHD SDR contents (3840×2160 pixels) at 30 fps with 10 bits per pixel was experimented and demonstrated in [45] and [46] for codec scalability. We improve this demonstration by adding HDR support on all layers. In this demonstration, the SHVC-AE only realizes a spatial scalability with 10 bits per pixel and BT.2020 color-gamut on both layers. The backward compatibility between SDR and HDR is enabled by the SL-HDR1 technology.

VI. CONCLUSION

In this paper we have proposed a complete software implementation solution of the scalable extension of the HEVC standard. This solution includes both SHVC encoder and decoder based, respectively, on the core professional HEVC encoder (HEVC-AE) and the open source real time HEVC decoder (*OpenHEVC*). Several optimizations have been integrated into the proposed scalable HEVC encoder (SHVC), resulting in three setups of the encoder *FILE*, *LIVE* and *LIVE+*. The SHVC-AE in *FILE* setup enables to reach a high rate-distortion performance close to the reference SHM with a speed-up of 37 and 66 in Intra and Inter coding configurations. The SHVC-AE in *LIVE* and *LIVE+* setups enables real time encoding performance of 3840×2160p 30 fps video with an efficient inter-layer prediction. The complete solution, including the SHVC-AE and scalable *OpenHEVC* decoder, enables a real time encoding/decoding of 3840×2160p30 videos on multi-core Intel Xeon platform. Moreover, the scalable *OpenHEVC* decoder enables to decode the BL in HD resolution on ARM mobile platform.

Several improvements on the SHVC-AE can be investigated as future works. First, the proposed encoder can be extended to support the encoding of more than two layers (N layers). In addition, it would be interesting to investigate the performance of the encoder with other types of scalability including quality, bit-depth, color and codec. Finally, more algorithmic optimizations can be performed to improve the coding efficiency of the HEVC-AE encoder, especially in Inter coding configuration.

REFERENCES

- [1] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

- [2] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramanian, "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, Jan. 2016.
- [3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6316136>
- [4] V. Seregin and Y. He, "Common SHM test conditions and software reference configurations." CTVC document JCTVC-Q1009, Apr. 2014.
- [5] T. Biatek, W. Hamidouche, J.-F. Travers, and O. Deforges, "Optimal Bitrate Allocation in the Scalable HEVC Extension for the Deployment of UHD Services," *IEEE Transactions on Broadcasting*, vol. 62, no. 4, pp. 826–841, Dec. 2016.
- [6] X. HoangVan, J. Ascenso, and P. Pereira, "Improving SHVC performance with a joint layer coding mode." Shanghai: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2016, pp. 1145–1149.
- [7] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4317636>
- [8] "ATSC Standard: Video HEVC (A/341)," May 2017.
- [9] Y. L. Kyungmo Park and D. Y. Suh, "Delivery of ATSC 3.0 Services With MPEG Media Transport Standard Considering Redistribution in MPEG-2 TS Format," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 338–351, Mar. 2016.
- [10] *Generic Coding of Moving Pictures and Associated Audio Information – Part 1: Systems*, ISO/IEC 13818-1.
- [11] R. Parois, W. Hamidouche, J. Vieron, M. Raulet, and O. Deforges, "Efficient Parallel Architecture for a Real-time UHD Scalable HEVC Encoder," in *European Signal Processing Conference (EUSIPCO)*, Aug. 2017.
- [12] W. Hamidouche, M. Raulet, and O. Deforges, "4k Real-Time and Parallel Software Video Decoder for Multilayer HEVC Extensions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 169–180, Jan. 2016. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7273890>
- [13] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel Scalability and Efficiency of HEVC Parallelization Approaches," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1827–1838, Oct. 2012.
- [14] P. Bordes, P. Andrivon, X. Li, Y. Ye, and Y. He, "Overview of color gamut scalability," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1580–1594, July 2017.
- [15] M. Blestel and M. Raulet, "Open SVC decoder: a flexible SVC library." ACM International Conference on Multimedia, Oct. 2010, pp. 1463–1466.
- [16] HHI, "Joint Scalable Video Model JSVM." <https://www.hhi.fraunhofer.de/en/departments/vca/research-of-h264avc/jsvm-reference-software.html>.
- [17] S. Sanz-Rodriguez, M. Mauricio, T. Mayer, and T. Schierl, "A parallel H.264/SVC encoder for high definition video conferencing," *Signal Processing: Image Communication - Elsevier*, vol. 30, pp. 89–106, Jan. 2015.
- [18] P.-T. Chiang, Y.-C. Ting, H.-K. Chen, S.-Y. Jou, I.-W. Chen, H.-C. Fang, and T.-S. Chang, "A QFHD 30-frames/s HEVC Decoder Design," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 724–735, Mar. 2015.
- [19] M. Abeydeera, M. Karunaratne, G. Karunaratne, K. De Silva, and A. Pasqual, "4k Real-Time HEVC Decoder on an FPGA," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 236–249, Jan. 2016.
- [20] C.-C. Ju, "A 0.2n/pixel 4k 60fps Main-10 HEVC decoder with multi-format capabilities for UHD-TV applications," in *IEEE European Solid State Circuits Conference (ESSCIRC)*, Italy, Sep. 2014.
- [21] D. Zhou, "An 8k H.265/HEVC Video Decoder Chip With a New System Pipeline Design," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 113–126, Jan. 2017.
- [22] M. Tikekar, C.-T. Huang, C. Juvekar, V. Sze, and A. P. Chandrakasan, "A 249-Mpixel/s HEVC Video-Decoder Chip for 4k Ultra-HD Applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 61–72, Jan. 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6636099>
- [23] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.
- [24] C. C. Chi, M. Alvarez-Mesa, J. Lucas, B. Juurlink, and T. Schierl, "Parallel HEVC Decoding on Multi- and Many-core Architectures," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 247–260, Jun. 2013.
- [25] "Libde265 decoder," in <https://github.com/strukturag/libde265>, 2017.
- [26] C. C. Chi, M. Alvarez-Mesa, B. Bross, B. Juurlink, and T. Schierl, "SIMD Acceleration for HEVC Decoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 841–855, May 2015.
- [27] Qualcomm, "Snapdragon 810 processor product brief," 2014.
- [28] MulticoreWare, "x265 HEVC Encoder / H.265 Video Codec." <http://x265.org/>, 2017.
- [29] Vantrix, "F265 Open Source HEVC/H.265 Project." <http://vantrix.com/f-265-2/>, 2017.
- [30] UltraVideoGroup, "Kvazaar HEVC Encoder." <http://ultravideo.cs.tut.fi/#encoder>, 2017.
- [31] M. Viitanen, A. Koivula, A. Lemmetti, J. Vanne, and T. D. Hmlinen, "Kvazaar HEVC encoder for efficient intra coding," in *Circuits and Systems (ISCAS)*, 2015 IEEE International Symposium on, Lisbon, Portugal, May 2015.
- [32] A. Mercat, F. Arrestier, W. Hamidouche, M. Pelcat, and D. Menard, "Energy Reduction Opportunities in an HEVC Real-Time Encoder," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2017 IEEE International Conference on. IEEE, 2017.
- [33] G. Correa, P. Assuncao, L. Agostini, and L. A. d. S. Cruz, *Complexity-Aware High Efficiency Video Coding*. Cham: Springer International Publishing, 2016. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-25778-5>
- [34] X. Li, M. Chen, Z. Qu, J. Xiao, and M. Gabbouj, "An effective CU size decision method for quality scalability in SHVC," *ACM Journal Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8011–8030, Mar. 2017.
- [35] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "An Encoder Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 664–674, Sep. 2016.
- [36] C.-C. Wang, Y.-S. Chang, and K.-N. Huang, "Efficient Coding Tree Unit (CTU) Decision Method for Scalable High-Efficiency Video Coding (SHVC) Encoder," in *Recent Advances in Image and Video Coding*, Nov. 2016.
- [37] W.-J. Chiang, J.-J. Chen, and Y.-H. Tsai, "A fast shvc coding scheme based on base layer co-located cu and cross-layer pu mode information," in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2017, pp. 381–386.
- [38] H. R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, "Online-learning-based mode prediction method for quality scalable extension of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2204–2215, Oct 2017.
- [39] HHI, "SHVC Reference Software (SHM)." https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/, 2017.
- [40] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "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. 2003. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1218189>
- [41] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April*, 2001.
- [42] G. Bjontegaard, "Improvements of the bd-psnr model," *ITU-T SG16 Q*, vol. 6, p. 35, 2008.
- [43] Dektec Digital Video BV, "DTA-2174 Leaflet," June 2015.
- [44] P. Cabarat, W. Hamidouche, O. Deforges, M. Raulet, and J. L. Feuvre, "4k real-time video streaming in hybrid codec scalability shvc configuration," in *IEEE Conference on Design and Architectures for Signal and Image Processing (DASIP2016, Demo session)*, Rennes, France, 2016.
- [45] R. Parois, W. Hamidouche, E. Mora, M. Raulet, and O. Deforges, "Demo: UHD Live video streaming with a real-time scalable HEVC encoder," in *Conference on Design & Architectures for Signal & Image Processing (DASIP)*, Oct. 2016.
- [46] P. Cabarat, W. Hamidouche, and O. Dforges, "Real-time and parallel shvc hybrid codec avc to hevc decoder," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 3046–3050.