4K Real-Time and Parallel Software Video Decoder for Multilayer HEVC Extensions

Wassim Hamidouche, *Member, IEEE*, Mickael Raulet, *Member, IEEE*, and Olivier Déforges

***Abstract*— Two High Efficiency Video Coding (HEVC) extensions, namely, the scalable HEVC (SHVC) extension and the multiview HEVC (MV-HEVC) extension, have been finalized in July 2014 by the Moving Picture Experts Group and Video Coding Experts Group. These two extensions enable additional features not covered in the first version of the HEVC standard such as spatial, fidelity, bitdepth, and color gamut scalability, as well as stereoscopic and multiview representations. In this paper, we propose a software parallel decoder architecture for the HEVC standard and its multilayer extensions, including SHVC and MV-HEVC extensions. The decoder consists of multiple instances of the *OpenHEVC* decoder, one instance to decode each layer with a communication between dependent layers to perform inter-layer predictions. The proposed multilayer HEVC decoder is parallel friendly and supports both wavefront parallelism to simultaneously process adjacent rows of the frame and frame-based parallelism to decode a set of temporal and spatial frames in parallel. Moreover, the most time-consuming operation introduced in the SHVC extension, namely, the resampling of the inter-layer reference picture in spatial scalability, is optimized in single instruction multiple data for x86 platform. We assess the complexity of the multilayer HEVC decoder with respect to the simulcast configuration. The multilayer decoder decoding two SHVC layers introduces in average 40%–71% additional complexity compared with the single layer HEVC decoder. Moreover, the low level optimizations with a hybrid parallel processing solution enable a real-time decoding of 4Kp60 enhancement layer on a 6-core Intel i7 processor running at 3.4 GHz.**

***Index Terms*— Decoder complexity, High Efficiency Video Coding (HEVC), low level optimizations and parallel processing, multilayer extensions, multiview HEVC (MV-HEVC), scalable HEVC (SHVC).**

I. Introduction

SCALABLE High Efficiency Video Coding (SHVC) and multiview High Efficiency Video Coding (MV-HEVC) are the scalable and multiview extensions [1] of the High Efficiency Video Coding (HEVC) standard [2], [3], respectively. These two extensions [4], [5] are finalized in July 2014 by the International Telecommunication Union (ITU-T) Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG) under partnerships known as Joint Collaborative Team on Video Coding (JCT-VC) and JCT on 3D video [6]. The SHVC extension defines tools to provide spatial, fidelity (SNR), bitdepth, and color (gamut) scalability, while the MV-HEVC extension enables efficient coding of 3D video content in multiview format. Concerning the SHVC extension, a number of scalable solutions [7]–[10] were proposed as a response to the SHVC call for proposal [11]. The approved solution uses all technologies defined in the HEVC standard; and is based on a multiloop coding approach, requiring the decoding of all intermediate reference layers. Moreover, the SHVC extension includes inter-layer predictions to take advantage of spatial correlations and further improve the coding efficiency. Adopting HEVC coding tools enables a gain around 50% in terms of subjective video quality with respect to the H.264/Advanced Video Coding (AVC) high profile [12]. In addition, inter-layer prediction in SHVC introduces additional rate–distortion gain of 15%–30% compared with a simulcast coding configuration[1] [13]. The MV-HEVC extension keeps the same coding concept than in AVC multiview extension Multiview Video Coding (MVC) [14], which is also based on multiloop coding framework. Moreover, the high level syntax has been designed in HEVC to be common to all HEVC multilayer extensions including SHVC and MV-HEVC extensions.

The SHVC encoder consists of L single layer HEVC encoders, one encoder to encode each layer with L the number of layers. The first layer ($l = 1$) represents a base quality of the video and decoding more layers allows to further enhance spatial or SNR quality of the video. To decode the layer $l = L$, all intermediate reference layers ($L - 1$) need to be fully decoded to perform inter-layer predictions. Moreover, in the case of spatial scalability, the decoded picture used for inter-layer prediction need to be upsampled and the related motion vectors (MVs) upscaled to match with the resolution of the enhancement layer being decoded. These additional operations introduced in the SHVC extension, namely, the decoding of the reference layers, the upsampling, and MVs upscaling operations in the case of spatial scalability, considerably increase the decoding complexity with respect to the simulcast configuration. Therefore, the complexity of the SHVC standard becomes a real issue especially when dealing with real-time decoding of high-resolution and high bitrate video (4k $\times$ 2k and 8k $\times$ 4k). The decoding of MV-HEVC bitstream is similar

Manuscript received November 15, 2014; revised March 16, 2015; accepted September 4, 2015. Date of publication September 22, 2015; date of current version January 6, 2015. This work was supported by the European Celtic-Plus project H2B2VS. This European project was partially funded by Finland, France, Spain, Switzerland, and Turkey. This paper was recommended by Associate Editor T. Wiegand.

The authors are with the IMAGE Group, Institut National des Sciences Appliquées de Rennes, Rennes 35708, France (e-mail: wassim.hamidouche@insa-rennes.fr; mickael.raulet@insa-rennes.fr; olivier.deforges@insa-rennes.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2478705

[1]Simulcast configuration consists in independently encoding of the video representations (views, layers). At the decoder side, only one representation (layer) is decoded with a conforming HEVC decoder.

to the decoding of the SHVC bitstream in SNR configuration since all views have the same resolution, and thus inter-view prediction is performed without the need of resampling.

This paper provides two main contributions. The first one consists in designing real-time and parallel decoder architecture for HEVC standard and its multilayer extensions including SHVC and MV-HEVC. The second contribution assesses the complexity of the multilayer HEVC decoder decoding SHVC video bitstream with respect to a single layer HEVC decoder. Several HEVC decoders have been presented in the literature and their performance assessed on different platforms [15], [16]. Bossen *et al.* [15] studied the complexity assessment of a single layer HEVC decoder, which remains quite similar to the complexity of the H.264/AVC decoder. The proposed decoder was optimized[2] for both x86 and Advanced RISC Machines (ARM) Neon architectures and compared with the HEVC reference software model (HM) decoder. Tikekar *et al.* [17] describe architectural optimizations for an HEVC video decoder chip. Optimizations were introduced to overcome with chip constraints including memory usage and power consumption. The decoder performs a real-time decoding of 2160p30 video sequences on a chip running at 200 MHz.

The proposed multilayer HEVC decoder is based on the open source project *OpenHEVC* [18], which implements an optimized HEVC decoder. The most time-consuming operation introduced in the SHVC extension, namely, the upsampling of the picture used for inter-layer prediction, is optimized in SIMD methods for x86 architecture. The experimental results showed that the optimized multilayer decoder decoding two SHVC layers introduces in average an additional complexity of 40%–71% compared with the single layer HEVC decoder. Moreover, the proposed decoder decoding two SHVC layers (two MV-HEVC views), on a single core, is four to five times faster, depending on the scalability configuration, than the SHVC (MV-HEVC) reference software models (SHM and HTM). The new upsampling operation introduced in the SHVC represents, after SIMD optimization, in spatial scalability configuration around 20% of the whole decoding time.

The multilayer HEVC decoder is parallel friendly and supports two levels of parallelism. The first level enables to simultaneously process separated regions of a frame through the high level parallel processing solutions defined in the HEVC standard including wavefront, slice, and tile [19]. As a second level of parallelism, the frame-based parallelism approach decodes a set of temporal and spatial (views) frames in parallel under the constraint that inter and inter-layer (inter-view) predictions are satisfied. We propose a hybrid parallelism solution combining the wavefront and the frame-based parallelism solutions to take advantage of both solutions and achieve a good tradeoff performance between speedup, frame latency, and memory usage. The low level optimizations jointly with the hybrid parallelism solution in the multilayer HEVC decoder enables a real-time decoding up to 4Kp60 in 2× spatial scalability configuration on a 6-core Intel i7 processor running at 3.4 GHz.

The rest of this paper is organized as follows. The principles of the HEVC standard and its multilayer extensions are provided in Section II. Section III describes the architecture of the HEVC multilayer decoder including low level optimizations and parallel processing solutions supported by the decoder. The performance of the optimized multilayer HEVC decoder is assessed and discussed in Section IV. Finally, Section V concludes this paper.

II. Related Work

A. Overview of the HEVC Standard

The HEVC standard can reach the same subjective video quality than its predecessor H.264/AVC at about half bitrate [12]. This gain is obtained thanks to the new tools adopted in the HEVC standard, such as quadtree-based block partitioning, large transform and prediction blocks, accurate intra-/inter-predictions, and the in-loop sample adaptive offset (SAO) filter [2]. The HEVC frame is partitioned into coding tree units (CTUs), each contains one luma coding tree block (CTB) and two chroma CTBs. Recursive subdivision of a CTU results in coding unit (CU) leaves with the corresponding coding blocks. The CU can be split into prediction units (PUs), a basic entity for intra- and inter-predictions, and recursively split into transform units, a basic entity for residual coding. The HEVC standard was designed with a particular attention to complexity, where several solutions have been defined to leverage multicore architectures and enhance both encoding/decoding frame rate (DFR) and frame latency (DFL). These high level parallel processing solutions including wavefront, independent slice, and tile [19] are briefly described in the following section.

1) Parallelism in HEVC: The three high level parallel processing solutions defined in the HEVC standard enables to process multiple regions of a single picture. The frame can be partitioned into one or many slices, mainly to increase the bitstream robustness against packet losses. The tile concept splits the picture into rectangular groups of CTUs, called tiles. The independent slices and tiles break the Context Adaptive Binary Arithmetic Coding (CABAC) and the intra-prediction dependencies and thus can be used for parallel encoding and decoding. However, intra-prediction limitation and resetting the CABAC probabilities decrease the coding performance in terms of rate distortion, especially for large number of tiles/slices per frame. Moreover, the in-loop filters cannot be performed in parallel at the tile/slice edges without additional control mechanism. The wavefront parallel processing (WPP) solution splits the frame into CTU rows [20]. In the WPP mode, the CABAC context is initialized at the start of each CTU row. The overhead caused by this initialization is limited, since the CABAC context at each CTU row is initialized by the CABAC context state at the second CTU of the previous CTU row. As shown in Fig. 1, the decoding of each CTU row can be carried out on separate threads with a minimum delay of two CTUs between adjacent CTU rows. Therefore, the wavefront dependencies require a delay of two CTUs between adjacent CTU rows, introducing parallelization inefficiency

[2]Optimized software refers in this paper to a code-source written in single instruction multiple data (SIMD) operations.

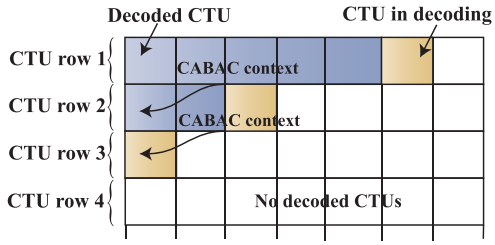

Fig. 1. Principle of the WPP approach in HEVC.

and requiring additional synchronization process between threads decoding adjacent CTU rows. These three high level parallel processing solutions depend on the bitstream, and can be used only when the independent slice, tile, and wavefront tools are enabled at the encoder side. The frame-level parallelism allows to simultaneously process multiple frames, whatever the coding configuration, under the restriction that the motion compensation dependencies are satisfied [21]. The frame-based parallelism also suffers from a number of limitations. The performance of the frame-based parallelism solution strongly depends on the coding structure and the ranges of MVs. Moreover, the frame-based parallelism improves the DFR but not the frame latency and it requires extra memory usage [22], [23], compared with tile and wavefront parallelism concepts [24].

B. Overview of the Multilayer HEVC Extensions

The multilayer HEVC extensions enable fidelity and spatial scalabilities through the SHVC extension and efficient coding of 3D multiview video content with the MV-HEVC extension. The multilayer syntax elements, mostly signaled at the video parameter set (VPS) header, are common to all HEVC extensions. These syntax elements provide information on the video layers such as the number of layers, and for each layer, resolution, bitdepth as well as the inter-layer dependencies. The multilayer HEVC encoder consists of L HEVC encoders, one encoder to encode each layer (view) with L the number of layers (views): one base layer (base view) and $L-1$ enhancement layers (no base views). In the case of SHVC spatial scalability, the base layer (BL) HEVC encoder encodes a downsampled version of the original video and feeds the first enhancement layer (EL) encoder with the decoded picture and its MVs. The enhancement layer encoder l ($l = 2, \ldots, L$) encodes a higher resolution video using the decoded picture from lower layer as an additional reference picture (included in the reference picture lists). The inter-layer reference picture is upsampled and it's MVs upscaled to match with the resolution of the layer being decoded. The upsampling operation is performed by a eight-tap interpolation filter for luma samples and four-tap interpolation filter for chroma filter [25]. Fig. 2 shows an example of the SHVC encoder encoding two layers in spatial scalability configuration. In the case of SNR scalability, the encoding process remains unchanged except that the picture used for inter-layer prediction is used without being upsampled and its MVs upscaled. As shown in Fig. 2, the outputs from the two encoders are multiplexed to form a conforming SHVC bitstream. The MV-HEVC extension with two views is similar to the SHVC encoding process in SNR configuration, where the

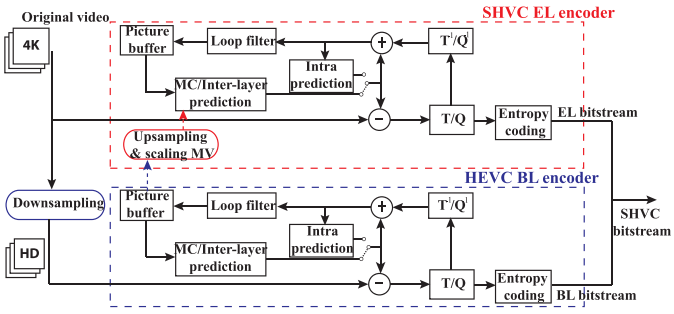

Fig. 2. Block diagram of the SHVC encoder encoding two spatial scalability layers.

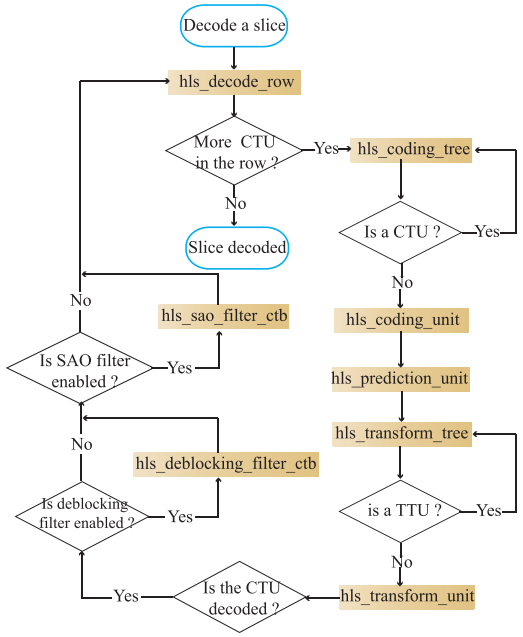

Fig. 3. Block diagram of the architecture of the *OpenHEVC* decoder (transform tree unit).

base layer is referred to the base view and the EL is referred to the no base view (or dependent view). The dependent view uses the base view as reference for inter-view predictions.

The HEVC standard version 2 defines two SHVC profiles (Scalable Main and Scalable Main 10) and one multiview profile (Multiview Main) [4]. The Scalable Main and Multiview Main profiles enable a base layer that conforms with the Main HEVC profile, while the Scalable Main 10 profile allows a base layer that conforms with the Main 10 HEVC profile.

III. Unified Decoder for HEVC Extensions

A. Decoder Architecture

The proposed unified HEVC decoder is based on the open source *OpenHEVC* project [18]. We developed the *OpenHEVC* decoder in C programming language on the top of the *FFmpeg* library [26]. The *OpenHEVC* decoder implements a conforming HEVC decoder and supports the three main profiles defined in the HEVC standard, namely, Main, Main 10, and Main Still Picture profiles [2]. Fig. 3 shows the block diagram of the *OpenHEVC* decoder. This decoder is based on the decoding of a CTU, where all decoding steps are performed at the level of a CTU. The decoder browses in raster scan the CTUs within the slice and calls the recursive

function *hls_coding_tree* to decode each CTU. This function browses the CUs within CTU in z-scan and calls for each CU the *hls_coding_unit* function. This one calls specific functions to perform inverse prediction and inverse transform operations. Once all CUs within the CTU are decoded, the decoder performs deblocking and SAO filters on the decoded CTU. The *OpenHEVC* decoder also supports the high level parallel processing solutions defined in the HEVC standard including wavefront, slice, and tile parallelism solutions. In addition, the *OpenHEVC* decoder supports the range extension profiles including monochrome, 4:2:2, and 4:4:4 enhanced chroma sampling structures and bitdepths in 10 and 12 bits. The decoder successfully passes the decoding of all range extensions conformance bitstreams of a bitdepth lower or equal than 12.

The proposed multilayer HEVC decoder consists of L instances of the *OpenHEVC* decoder, one instance to decode a layer (view), with L the number of layers (views). The *OpenHEVC* decoder enables to decode the base layer (view), since it conforms with the HEVC standard. Three main modifications were introduced into the *OpenHEVC* decoder to support the decoding of the enhancement layers in the SHVC extension and no base views in the MV-HEVC extension. First, the parser of the video headers was slightly changed to parse the new multilayer syntax elements signaling information of the SHVC and MV-HEVC layers. Second, two new functions were integrated to the decoder to perform the upsampling and MVs upscaling of the picture used for inter-layer prediction in SHVC spatial scalability configuration. In the case of fidelity scalability and MV-HEVC extension, the picture used for inter-layer prediction is only copied into the new inter-layer reference picture without being upsampled and its MVs upscaled. The upsampling and the MVs upscaling operations are performed by the enhancement layer decoder on the CTU including the PU used for inter-layer prediction. Finally, the reference picture lists at the enhancement layer decoder (no base view decoder) are differently managed to include inter-layer (inter-view) reference pictures.

The proposed multilayer decoder does not introduce additional complexity in the MV-HEVC extension, since all views need to be decoded to enable stereoscopic display capability. However, in the SHVC extension, the multilayer decoder introduces an extra complexity when compared with the simulcast configuration. The additional complexity consists in the decoding of the intermediate reference layers, and the inter-layer prediction including the upsampling and the MVs upscaling operations in the case of spatial scalability. In terms of memory usage, the multilayer decoder allocates extra memory related to memory of the *OpenHEVC* decoders decoding the intermediate layers as well as the memory of the inter-layer reference pictures. The additional complexity and memory introduced in the multilayer decoder for SHVC bitstreams are accurately assessed in this paper with respect to the simulcast configuration decoding a single layer.

B. Low Level Optimizations in the Multilayer HEVC Decoder

The source code of the *OpenHEVC* decoder is heavily optimized in SIMD methods. The most time-consuming operations including motion compensation, inverse transform, deblocking filter, and SAO filter are optimized in intrinsic instructions for x86 architecture. SSE2 up to AVX2 instructions are used to speed up the decoding functions in 8, 10, and 12 bitdepth configurations.

The most time-consuming operation introduced in the multilayer extensions is the upsampling of the picture used for inter-layer prediction in the SHVC extension. The complexity of the upsampling operation is similar to the complexity of the motion compensation in HEVC, which also uses a eight-tap (for half-sample positions) and four-tap interpolation filters for luma and chroma subpel positions, respectively [15]. The upsampling process of two spatial scalability ratios $2\times$ and $1.5\times$ is optimized in SIMD instructions in the multilayer HEVC decoder (SSE2 up to SSSE3 instructions are used). However, other ratios are not optimized and the C code version of the upsampling function is selected at run time.

In this section, we provide details on the SIMD optimizations of the upsampling process in $2\times$ spatial scalability configuration. In fact, the no optimized 1-D upsampling filter is performed in eight multiplications and seven additions for each luma component and four multiplications and three additions for the chroma component. The input and the output of the upsampling filter is 8-bits pixel in the Scalable Main profile and the intermediate buffer between horizontal and vertical filters is 16-bits precision. In $2\times$ spatial scalability, two eight-tap filters of different coefficients are used for odd and even pixel positions. In this implementation, 8 pixels are upsampled in parallel and the 1-D horizontal upsampling of 8 pixels is performed only by four multiplications and three additions. Fig. 4 shows in five steps the 1-D horizontal upsampling of 8 luma pixels. First, the original pixels are loaded in 128-bits streaming SIMD extensions (SSE) register with *_mm_loadu_si128* instruction. The second and the third steps set the pixels in appropriate positions in the register for multiplication with the filter coefficients using *_mm_unpackl_epi8* function. The fourth step performs multiplication and addition of the pixels with the filter coefficients using *_mm_maddubs_epi16* instruction. Finally, the last step performs three additions in 16-bits precision and then stores the final results (r_0) in an intermediate memory used as an input for the 1-D vertical filter.

C. Parallelism in the Multilayer HEVC Decoder

The *OpenHEVC* decoder supports the decoding in WPP solution. Therefore, each SHVC layer (MV-HEVC view) can be decoded in parallel through the wavefront parallelism approach. In this solution, the L SHVC layers (MV-HEVC views) are decoded in sequential order and the CTU rows of each layer are decoded in wavefront. We provide the analytical performance of the WPP solution. Let us consider x the number of CTU columns, y the number of CTU rows, and d the delay in terms of CTUs between two adjacent rows. The effective number of threads n used in the wavefront solution is given as

$$n = \min\left(\text{nb\_cpu\_threads}, \left\lfloor \frac{x}{d} \right\rfloor, y\right) \tag{1}$$

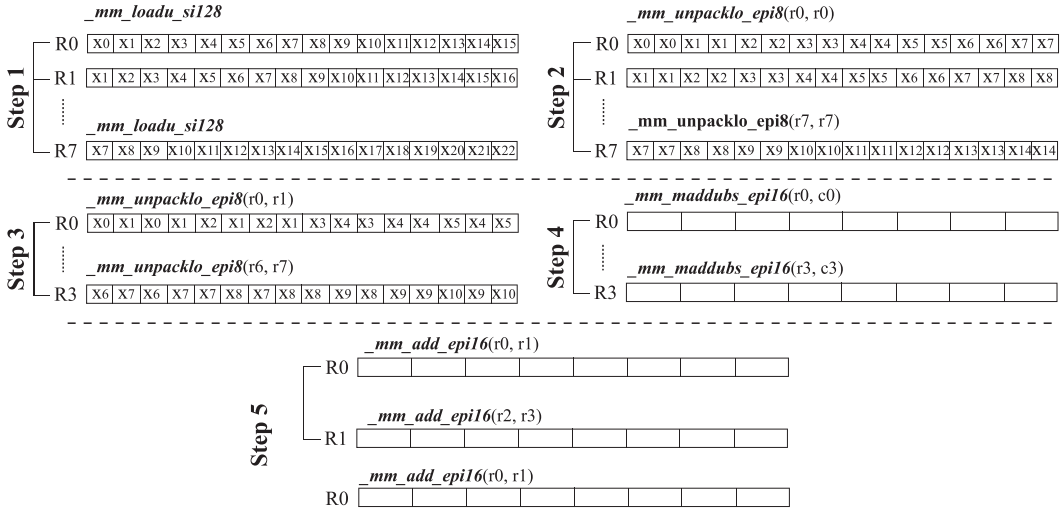

Fig. 4. 1-D horizontal luma upsampling filter in SSE instructions.

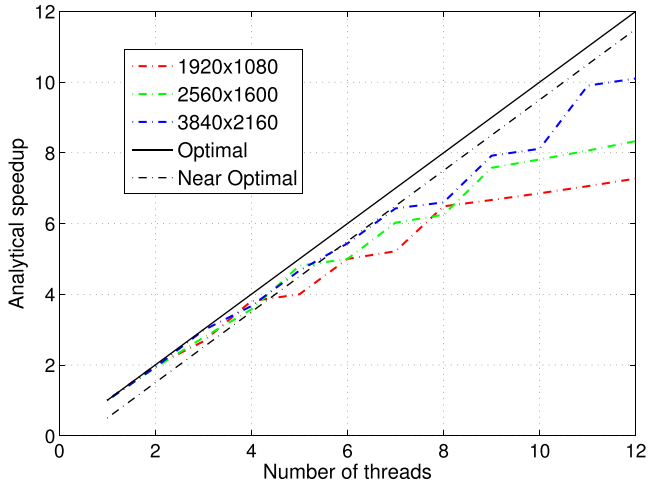

Fig. 5. Analytical speedup performance of the wavefront parallelism solution.

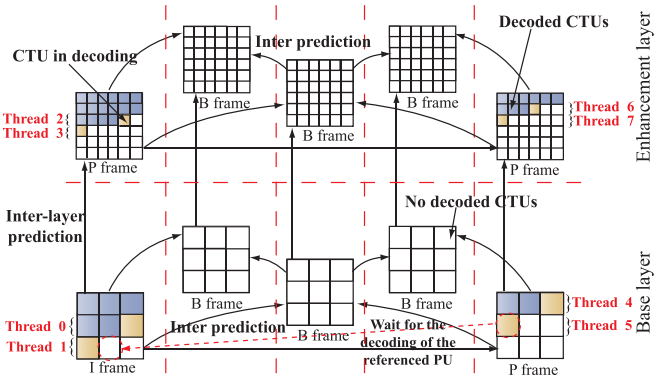

Fig. 6. Hybrid parallelism solution in the SHVC decoder decoding two spatial scalability layers.

where $d \in \mathbb{N}^+$ and nb_cpu_threads is the number of threads selected to decode the video sequence.

The analytical speedup γ is derived as

$$\gamma = \begin{cases} \dfrac{xy}{\frac{xy}{n} + d(n-1)}, & \text{if } \alpha = 0 \\ \dfrac{xy}{x\left\lceil \frac{y}{n} \right\rceil + d(\alpha - 1)}, & \text{if } \alpha \neq 0 \end{cases} \quad (2)$$

where $x, y, n \in \mathbb{N}^+$, and $\alpha = y \bmod n$.

The speedup of the WPP solution is equal to the number of CTUs of the frame (xy) divided by the number of CTUs decoded by each thread plus the additional delay required by the wavefront approach. When the number of CTU rows is multiple of the number of decoding threads, the delay of the wavefront solution at the decoding of the last CTUs is equal to $d(n-1)$. However, when the number of CTU rows is not multiple of the number of decoding threads, the delay at the end of the frame is equal to $d(\alpha - 1)$. Fig. 5 shows the analytical speedup of the WPP solution versus the number of threads for different video resolutions. We can note that the WPP performance decreases for a large number of threads, where the additional delay considerably increases. Moreover, the WPP solution efficiency increases with larger video resolution, for which the number of CTU rows and the number of CTUs within one row are higher.

Equation 2 does not consider the inactive threads waiting at the start and the end of decoding each frame. The waiting time of the inactive threads in terms of CTUs is computed as

$$\sigma = \begin{cases} d(n^2 - n), & \text{if } \alpha = 0 \\ d/2(n^2 - n + \alpha^2 - \alpha) + bx - d/2(b^2 + b), & \text{if } \alpha \neq 0 \end{cases} \quad (3)$$

where $b = n - \alpha$. We can note from 3 that a high value of b ($n \gg \alpha$ and $\alpha \neq 0$) considerably increases the inefficiency of the wavefront solution.

To overcome with the wavefront solution inefficiency, we propose a hybrid parallelism solution combining the WPP solution with the frame-based parallelism. The frame-based parallelism enables to decode in parallel a set of frames belonging to different layers (views) and different temporal representations within the same layer. These frames can be decoded in parallel under the restriction that inter and inter-layer (inter-view) predictions are satisfied, respectively. Moreover, the CTU rows of each frame are decoded in parallel while satisfying the wavefront dependencies. Fig. 6 shows the

TABLE I
CONFIGURATION OF THE EXPERIMENTS

System		Software	
Processor	Intel Core i7 E5-1650	Compiler	GCC-4.8.2
		OS	Ubuntu 12.04
ISA	X86-64	Kernel	3.13.0-37
Clock	3.4 GHz	SHVC	cff4b48a94
Level 3 cache	12 MB	release	based on SHM4.1
Cores	6		

hybrid parallelism approach in the SHVC decoder decoding two spatial scalability layers. In this configuration, four frames (two frames at each layer) are decoded in parallel and each frame is decoded in wavefront using two threads. In total, eight threads are used to decode the SHVC video sequence.

The hybrid parallelism solution takes advantage of both frame-based and wavefront solutions. Indeed, the wavefront solution is used at the near optimal region (corresponding to low number of threads in Fig. 5). The rest of the available threads can be used for frame-based approach to decode several spatial and temporal frames in parallel. Moreover, the inactive threads waiting for both wavefront and inter-prediction dependencies can be used by other frames waiting for available threads. This hybrid solution enhances both the speedup performance and the frame latency of the multilayer HEVC decoder.

An efficient synchronization mechanism is implemented in the multilayer HEVC decoder. This mechanism manages the communications between threads decoding different frames to ensure that the PU used for inter and inter-layer predictions is decoded at the reference frame. Otherwise, the decoding process waits until the PU required for the inter or inter-layer prediction is decoded. In the case of inter-layer prediction in spatial scalability, the EL decoder upsamples only the CTUs containing PUs used for inter-layer prediction. Moreover, the inter-layer reference frame is not removed from the decoded picture buffer (DPB), since the upper layer using this frame as reference is not fully decoded. The wavefront dependencies are also controlled by this mechanism through communications between threads decoding adjacent CTU rows. In fact, the decoding of a new CTU is performed only if $d = 2$, next CTUs in the previous row have been already decoded.

IV. RESULTS AND ANALYSIS

A. Experimental Configuration

The experiments were carried out on a computer fitted with a 6-core Intel i7 processor running at 3.4 GHz. Table I gives more details on the system configuration used in the experiments. For the SHVC coding configuration, we consider the common test conditions defined in the SHVC extension [29]. Two 3840 × 2160 video sequences including *CrowdRun* and *ParkJoy*, from the Sveriges Television AB high definition multiformat test set, were added to the test video sequences. Table II provides the configurations of the video sequences considered in the experiments. We use the scalable reference software model version 4.1 (SHM4.1) [27] to encode these video sequences in random access (RA) coding configuration,

TABLE II
VIDEO SEQUENCES CONSIDERED IN THE EXPERIMENTS

Class	Sequences	Resolution	Frame rate (Hz)	duration (second)
B	*Kimono*	1920x1080	24	10
	ParkScene		24	10
	Cactus		50	10
	BasketBallDrive		50	10
	BQTerrace		60	10
A	*Traffic*	2560x1600	30	5
	PeopleOnStreet		30	5
C	*CrowdRun*	3840x2160	30	5
	ParkJoy		30	5

TABLE III
QP CONFIGURATIONS

HEVC ext.	Configurations	BL QP	EL delta QP
SHVC	SNR	26, 30, 34, 38	-6, -4
	Spatial (2x & 1.5x)	22, 26, 30, 34	0, 2
	QP Single layer	20, 22, 24, 26, 28, 30, 32, 34, 36	
MV-HEVC	QP both views	25, 30, 35, 40	

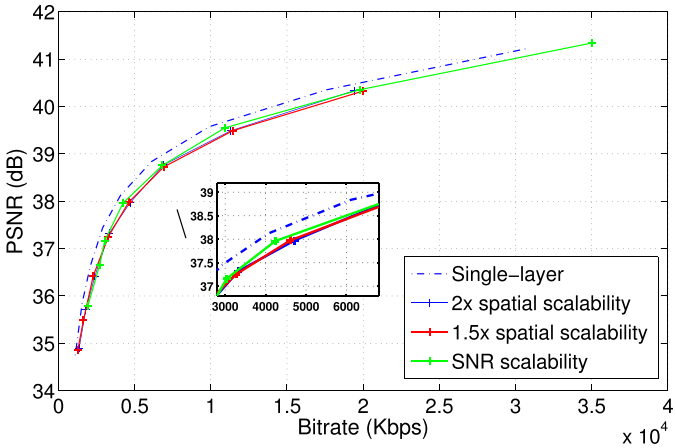

Fig. 7. Rate–distortion performance of the SHVC standard: RA configuration and class B video sequences.

two layers ($L = 2$) and different scalability configurations including fidelity (SNR) and spatial in two ratios 2× and 1.5×. The wavefront feature was enabled at both the base and the enhancement layers.

Concerning the MV-HEVC extension, we consider the 3D HEVC common test conditions defined in [30]. Eight multiviews (three views) of 10-s duration video sequences were encoded in RA configuration with the MV-HEVC reference software (HTM) version 12.0 [28]. The video sequences include three 1024 × 768 (*Balloons*, *Kendo*, and *Newspaper*) and five 1920 × 1088 (*Poznan_Hall2*, *Shark*, *GhostTownFly*, *Poznan_Street*, and *Undo_dancer*) sequences coded in four quantization parameter (QP) configurations. The QP values for both SHVC and MV-HEVC extensions are given in Table III.

Fig. 7 shows the rate–distortion performance of the SHVC standard in different scalability configurations. At a similar bitrate, the peak signal-to-noise ratio (PSNR) of the single layer (HEVC) configuration is slightly higher than the PSNR of the scalability configurations. The cost of the bitrate of the three scalable configurations with respect to the single layer configuration is assessed in Table IV in terms of Bjontegaard's

TABLE IV
BITRATE COST OF THE THREE SCALABLE CONFIGURATIONS COMPUTED IN BJONTEGAARD'S DIFFERENCE

Configurations	Class B	Class A	Class C
Spatial 2x	17.16%	18.60%	16.79%
Spatial 1.5x	16.63%	-	-
SNR	13.9%	15.79%	14.31%

difference. The scalable coding introduces in average between 13% and 19% overhead compared with the simulcast coding configuration.

B. Mono-Core SHVC Decoder

In this section, we provide the performance of the optimized multilayer HEVC decoder in comparison with the reference software decoders SHM, HTM, and the no optimized multilayer HEVC decoder. Table V provides the decoding time performance of the SHM reference software decoder and the HEVC multilayer decoder in different SHVC configurations for one particular QP value. The HEVC multilayer decoder reaches a real-time decoding for *Kimono*, *ParkScene*, and *Cactus* video sequences in all scalability configurations. Moreover, in the single-core configuration, the multilayer HEVC decoder decoding two layers is four to five times faster than the reference software decoder SHM. This high speed-up performance of the multilayer HEVC decoder (compared with the SHM software) is mainly achieved by the low level optimizations of both the decoding operations in *OpenHEVC* decoder (used to decode each layer) and the upsampling operation in the multilayer decoder.

On the other hand, we can note from Table V that the multilayer HEVC decoder introduces only 40% and 66% additional complexity with respect to a single-layer configuration in 2× and 1.5× scalability configurations, respectively. This additional complexity is mostly related to the decoding, the upsampling and the MVs upscaling operations of the BL, operations which are not performed in the simulcast configuration (only the EL is decoded). The decoding of the base layer is more complex in 1.5× and SNR configurations, since the base layer is higher resolution than the BL in 2× spatial scalability configuration. However, the reference software SHM introduces 57% and 92% additional complexity in 2× and 1.5× scalability, respectively. The low additional complexity of the multilayer HEVC decoder compared with the reference software decoder SHM is mainly achieved by SIMD optimizations of the upsampling operation in spatial scalability configurations.

Table VI gives the speedup achieved by the SIMD optimizations of the upsampling operation in the two considered spatial scalability configurations and all video sequences (average over all QP configurations). The SIMD optimization of the upsampling operation achieves in average a speedup around 1.6 in the spatial scalability configuration with two layers.

Fig. 8 shows the decoding time repartition of the multilayer HEVC decoder in different SHVC configurations. The inter-layer prediction in the case of spatial scalability represents in average 17%–20% of the whole decoding time. This

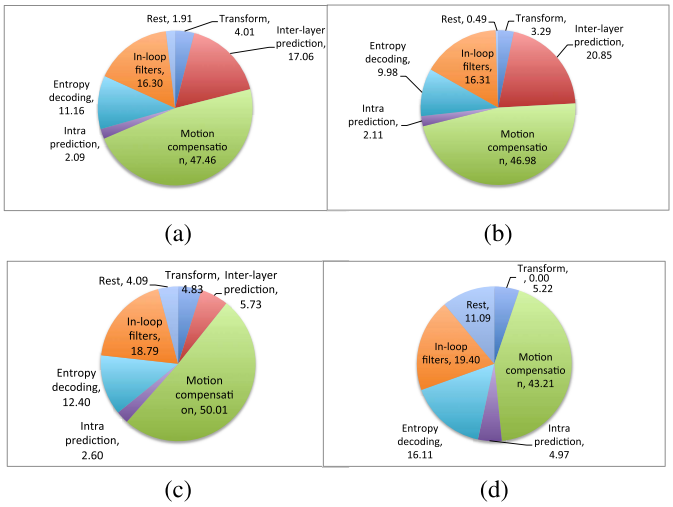

Fig. 8. Average time distribution (%) in the multilayer HEVC decoder: *BasketBallDrive*, RA, and all QP values. (a) 2× (7.4 s). (b) 1.5× (8.36 s). (c) SNR (10.91 s). (d) HEVC (5.87 s).

percentage is mainly related to the upsampling (after SIMD optimizations) of the base layer and the upscaling of its MVs operations. In the case of SNR scalability, the inter-layer operation represents only 7% of the whole decoding time, since the base layer samples and the corresponding MVs are only copied in the new inter-layer reference picture without being upsampled and upscaled, respectively.

Table VII shows the decoding time of the MV-HEVC video sequences decoded by the reference software decoder (HTM) and the multilayer HEVC decoder for all MV sequences at the lowest QP configuration (QP = 25). The multilayer HEVC decoder decoding two views achieves a real-time decoding with a decoding time less than 10 s for all MV video sequences, and the proposed decoder is in average five times faster than the reference software decoder HTM.

Table VIII shows the decoding performance of the multilayer HEVC decoder for three SHVC video sequences including *BasketballDrive*, *Traffic*, and *ParkJoy* in all scalability and QP configurations. The real-time decoding of *BasketballDrive* and *Traffic* video sequences is reached only at QP configuration higher than 28 in SNR scalability. However, the real-time decoding of *ParkJoy* video sequence at Ultra High Definition (UHD) resolution is not reached even at low bitrate (QP = 36, SNR). We can also note from Table VIII that the decoding time for the UHD *ParkJoy* video sequence is almost four times higher in high bitrate configuration (QP = 22) than in low bitrate configuration (QP = 36) in both 2× spatial and quality scalability. This significant increase in the decoding complexity is mainly related to the CABAC decoding which becomes at high bitrate, a bottleneck in the decoding process increasing both the DFR and the DFL especially for intra-coded frames. This shows the limit of the proposed decoder in single-core configuration to ensure real-time and low latency decoding of high-resolution video sequences coded at high bitrate. In the following section, we show how the hybrid parallelism solution can overcome with this limitation to enable higher DFR associated with lower DFL for video at high resolution and bitrate.

TABLE V

DECODING TIME PERFORMANCE OF THE SHM AND MULTILAYER HEVC DECODER: RA, $QP_{EL} = 26$, AND $QP_{BL} = 26$ AND 30 IN SHVC SPATIAL AND SNR SCALABILITY, RESPECTIVELY (NO OPTIMIZATIONS DOES NOT INCLUDE THE SIMD OPTIMIZATIONS)

Sequences	Decoding time (second)								
	Single-layer			SHVC					
				SNR		2x		1.5x	
	SHM	no opt. multi-layer HEVC	multi-layer HEVC	SHM	multi-layer HEVC	SHM	multi-layer HEVC	SHM	multi-layer HEVC
Kimono	11.75	8.06	2.31	21.67	4.4	19.32	3.44	22.13	3.93
ParkScene	12.22	8.22	2.75	22.28	5.1	20.39	3.98	23.38	4.62
Cactus	22.66	12.23	4.8	40.88	8.75	38.48	6.97	44.8	8.02
BasketBallDrive	27.17	17.68	5.62	48.39	10.31	44.03	8.16	50.7	9.29
BQTerrace	30.55	20.92	6.93	54.45	12.39	51.04	9.85	59.71	11.54
Traffic	12.93	8.23	2.77	24.01	5.21	22.86	4.15	-	-
PeopleonStreet	21.51	11.6	5.29	36.22	8.9	32.58	7.5	-	-
CrowdRun	47.13	31.67	11.38	76.72	18.29	68.76	15.45	-	-
ParkJoy	48.93	33.95	12.69	80.54	20.15	71.39	16.91	-	-
Speedup vs SHM	1	1.53	4.3	1	4.33	1	4.82	1	5.36
Complexity vs HEVC(%)	0	0	0	72	71	57	40	92	66

TABLE VI

DECODING TIME PERFORMANCE OF THE MULTILAYER HEVC DECODER: SHVC, RA, AND ALL QP (NO OPTIMIZATIONS DOES NOT INCLUDE THE SHVC RESAMPLING IN SIMD)

Sequences	2x			1.5x		
	no opt. multi-layer HEVC	multi-layer HEVC	Speedup	no opt. multi-layer HEVC	multi-layer HEVC	Speedup
Kimono	5.56	3.07	1.81	6.21	3.44	1.80
ParkScene	5.83	3.53	1.65	6.61	4.08	1.61
Cactus	10.99	6.45	1.70	12.28	7.31	1.68
BasketBallDrive	12.59	7.40	1.70	14.07	8.36	1.68
BQTerrace	14.9	9.2	1.62	16.88	10.75	1.57
Traffic	6.5	3.78	1.72	-	-	-
PeopleonStreet	10.41	6.51	1.59	-	-	-
CrowdRun	20.72	13.54	1.53	-	-	-
ParkJoy	22.04	14.69	1.49	-	-	-
Average	-	-	1.64	-	-	1.66

TABLE VII

DECODING TIME PERFORMANCE OF THE MULTILAYER HEVC DECODER: MV-HEVC, RA, AND QP = 25

Sequences	Decoding time (second)				
	View0		View0 + View1		
	HTM	multi-layer	HTM	Multi-layer	Speedup
Balloons	5.1	0.91	9.58	1.76	5.44
Kendo	4.87	0.84	9.15	1.66	5.49
Newspaper	3.96	0.69	7.47	1.39	5.36
Poznan_Hall2	7.3	1.3	13.7	2.48	5.52
Shark	14.5	3.09	22.79	4.87	4.67
GhostTownFly	12.73	2.74	19.54	4.31	4.53
Poznan_Street	8.77	1.76	16.11	3.28	4.89
Undo_dancer	12.25	2.8	21.32	4.84	4.4
Average	-	-	-	-	5.04

TABLE VIII

DECODING TIME PERFORMANCE OF THE MULTILAYER HEVC DECODER: SHVC, RA, AND ALL QP

Sequences	Decoding time (second)						
	BasketballDrive			*Traffic*		*ParkJoy*	
QP_{EL}	SNR	2x	1.5x	SNR	2x	SNR	2x
20	18.15	-	-	8.27	-	38.07	-
22	14.40	11.77	13.12	7.08	5.77	30.4	26.09
24	11.72	9.42	10.34	5.84	4.91	24.39	20.32
26	10.22	8.12	9.19	5.21	4.15	20.07	16.95
28	9.25	7.09	7.86	4.59	3.70	17.34	13.89
30	8.43	6.5	7.5	4.25	3.30	14.94	12.37
32	7.85	5.79	6.55	3.87	3.01	12.92	10.28
34	7.29	5.5	6.46	3.64	2.78	11.45	9.38
36	-	5.04	5.84	-	2.59	-	8.27
Average	10.91	7.4	8.36	5.34	3.78	21.2	14.69

C. Parallel Multilayer HEVC Decoder

Fig. 9 shows the speedup performance of the WPP implementation in the *OpenHEVC* decoder for different video resolutions. The speedup of the proposed implementation is close to the upper-bound performance of the WPP solution especially for a number of threads below 5. The performance of the proposed implementation slightly decreases for a number of threads higher than 4. This is mainly caused by the communications between threads decoding adjacent rows, which are not considered in the analytical performance, and increase with the number of threads. Moreover, some fractions of the code run in sequential, like reading the HEVC frames and parsing the slice headers. Thus, curves in Fig. 9 follow Amdahl's law [33], [34] drawing the theoretical speedup of parallel computing system when fractions of the system are sequential.

In the following, we compare the performance of three parallel decoding configurations $(n, m) = \{(6, 1), (1, 6), (3, 2)\}$ with n the number of threads decoding CTU rows in wavefront and m the number of frames decoded in parallel with the frame-based parallelism. The first configuration (6, 1) decodes the BL and the EL in sequential order and each frame in both

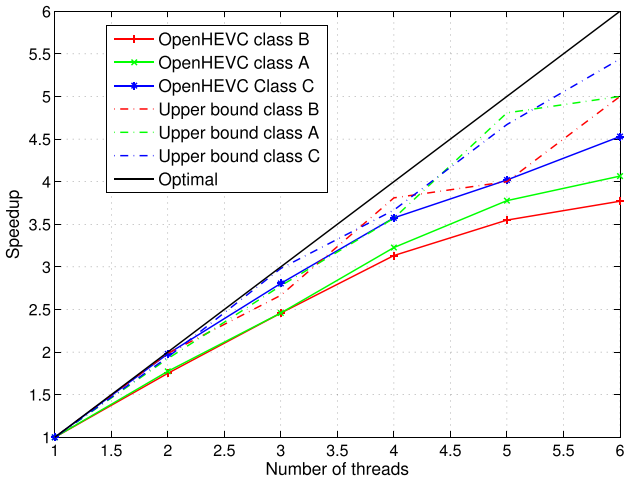

Fig. 9. Speedup performance of the wavefront solution: simulcast configuration at QP = 26.

layers is decoded in wavefront with six threads. The second configuration (1, 6) activates only the frame-based parallelism by decoding in parallel three BL frames and three EL frames. The last configuration (3, 2) activates the hybrid parallelism solution where the BL and the EL are decoded in parallel and frames in both layers are decoded in wavefront using three concurrent threads.

Table IX shows the performance of the three decoding configurations in terms of speedup, DFR, and DFL (decoding time by frame) for the three classes of the SHVC video sequences. The hybrid parallelism solution performs the highest speedup performance compared with the wavefront and frame-based parallelism solutions. This is because the wavefront parallelism is used at near optimal configuration with only three threads (see Fig. 5). The speedup is higher for a large resolution video sequence especially for class C video sequences. The high speed-up performance of the hybrid parallelism increases the DFR from 79 frames/s in a single-core configuration to reach 321 frames/s for class B video sequences in 2× spatial scalability configuration. The wavefront parallelism solution achieves the lowest latency, since the six threads are used to decode six CTU rows on each frame. However, the frame-based parallelism provides the highest latency since each frame is decoded by one thread, and the inter and inter-layer dependencies between frames decoded in parallel further increase the frame latency (versus single-core configuration). The hybrid parallelism solution decreases the frame latency thanks to the three threads used for wavefront decoding. Moreover, the decoding of the BL and the EL in parallel further decreases the SHVC frame latency.

Table X shows the DFR performance of the multilayer HEVC decoder decoding the multiview video sequences (at QP = 25) with the frame-based parallelism. The speedup performance is in average equal to 3, which enables to reach the DFR of 290 frames/s for *Poznan_Hall2* video sequence. This shows the capability of the HEVC multilayer decoder to decode in real-time HD multiview video sequences with a large number of views. Moreover, this high frame rate performance is associated with a low latency, since the different views are decoded in parallel (even when the wavefront, slice, and tile features are not enabled at the encoder).

Table XI shows DFR (frames/s) and decoding frame time (latency) the three video classes at high bitrate configuration (QP_{EL} = 22). The hybrid parallelism solution enables to overcome with the limit of the decoder in mono core at high resolution and high bitrate video. It enables to increase the DFR from 5 to 26 frames/s in SNR scalability (class C) with lower latency performance (173 ms instead of 195 ms). The proposed hybrid solution can reach the real-time decoding of UHD video at high bitrate on multicore architecture with more than six cores.

TABLE IX

PERFORMANCE OF THE MULTILAYER HEVC DECODER: SHVC, RA, ALL SEQUENCES, AND ALL QP VALUES

Configurations			Decoding configurations			
			(1, 1)	(6, 1)	(1, 6)	(3, 2)
Class B	Speedup	SNR	1	3.33	2.72	4.32
		×2	1	2.91	2.11	4.13
		×1.5	1	3.02	2.45	4.26
		HEVC	1	3.37	2.39	3.98
	Decoding frame rate (fps)	SNR	53	175	147	228
		×2	79	226	167	321
		×1.5	68	203	169	288
		HEVC	107	355	258	422
	Decoding time per frame (ms)	SNR	21.05	6.14	35.19	18.71
		×2	14.05	4.71	24.82	11.35
		×1.5	16.05	5.19	27.06	13.45
		HEVC	11.38	3.22	28.52	8.10
Class A	Speedup	SNR	1	4.1	3.32	4.53
		×2	1	3.93	2.71	4.5
		HEVC	1	4.97	2.55	4.44
	Decoding frame rate (fps)	SNR	24	97	78	107
		×2	34	130	89	149
		HEVC	47	197	136	203
	Decoding time per frame (ms)	SNR	47.54	11.26	66.54	42.99
		×2	34.15	18.97	47.33	26.74
		HEVC	26.12	5.91	50.18	16.58
Class C	Speedup	SNR	1	4.51	3.54	4.9
		×2	1	4.2	2.94	4.83
		HEVC	1	4.57	3.6	4.9
	Decoding frame rate (fps)	SNR	9	38	30	41
		×2	12	51	36	58
		HEVC	16	71	55	76
	Decoding time per frame (ms)	SNR	135.83	29.31	177.47	120.98
		×2	93.40	21.74	122.03	73.06
		HEVC	83.35	17.63	130.04	48.95

TABLE X

DFR PERFORMANCE OF THE MULTILAYER HEVC DECODER: MV-HEVC, RA, AND QP = 25

Sequences	Decoding frame rate (fps) View0 + View1		Speedup
	(1,1)	(1,6)	
ine *Balloons*	170	464	2.72
Kendo	180	547	3.03
Newspaper	215	477	2.21
ine *Poznan_Hall2*	80	290	3.6
Shark	61	182	2.97
GhostTownFly	57	190	3.28
Poznan_Street	76	222	3.04
Undo_dancer	51	168	3.31
Average	-	-	3.0

TABLE XI

DFR AND DFT PERFORMANCE OF THE MULTILAYER HEVC DECODER: SHVC, RA, AND $QP_{EL} = 22$

Class	Sca.	Bitrate (Mbps)	DFR (fps)		DFT (ms)	
			(1,1)	(3, 2)	(1, 1)	(3, 2)
B	SNR	35.01	35	159	29.02	25.10
	2x	19.39	43	188	23.76	18.2
	1.5x	20.01	38	172	26.83	21.11
	HEVC	17.61	59	245	17.58	12.37
A	SNR	37.81	17	79	62.43	56.4
	2x	27.48	20	95	51.72	39.57
	HEVC	23.08	29	132	37.33	23.29
C	SNR	140.06	5	26	195.44	173.26
	2x	144.47	5	30	168.09	124.06
	HEVC	130.03	7	37	137.19	77.67

TABLE XII

MEMORY USAGE IN THE *OpenHEVC* DECODER FOR THE THREE CONSIDERED VIDEO CLASSES A, B, AND C (SIMULCAST CONFIGURATION)

Class	Memory usage (Ko)			
	Local structure	Global structure without DPB	DPB with 4 pictures	Decoder memory
B	137	7094	24780	32012
A	137	13826	48288	62252
C	137	27904	97503	125545

D. Memory Usage Analysis

The *OpenHEVC* decoder is composed of two main structures: local structure and global structure. The local structure holds all memory blocks used for the decoding of a CTU. The global structure holds memory blocks used for the decoding of the video sequence such as the DPB as well as the VPS, sequence parameter set, and picture parameter set video headers. Table XII lists the memory size of the local and global structures in the *OpenHEVC* decoder in the Main profile for different video resolutions. The memory size of the local structure represents less than 1% of the decoder memory, while the global structure without the DPB represents more than 20% of the whole decoder memory with four frames in the DPB.

In mono-core configuration, the multilayer HEVC decoder decoding L layers allocates the memory of L *OpenHEVC* decoders including one DPB for each layer. In the wavefront parallelism, only the local memory is duplicated by thread decoding adjacent CTU rows. However, the frame-based parallelism allocates one *OpenHEVC* decoder (global + local structures) for each thread used for frame-based decoding, while the DPB is not duplicated and is shared between all decoders decoding the same layer. Fig. 10 shows the memory usage of the multilayer HEVC decoder in different parallel decoding configurations in 2× spatial scalability case. The decoder memory in wavefront parallelism configuration slightly increases, since only the local structure representing less than 1% of the whole decoder memory is duplicated by thread. Since in this configuration, the two decoders run in sequential n local structures are allocated for each decoder ($2 \times n$ local structures for the SHVC decoder). In the hybrid parallelism configuration, the BL and the EL decoders run in parallel ($m = 2$) and only one local memory is duplicated by additional thread (n local structures for the SHVC decoder)

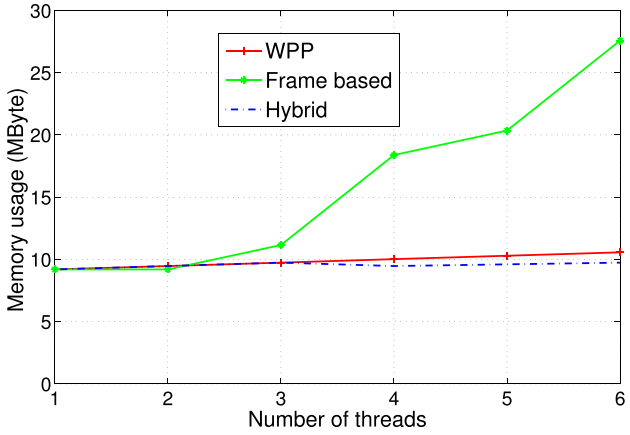

Fig. 10. Memory usage of the SHVC multilayer decoder without the DPB for different parallelism solutions (2× spatial scalability).

TABLE XIII

MEMORY USAGE (kBYTE) IN THE MULTILAYER HEVC DECODER WITHOUT THE DPB: SHVC, RA, AND ALL SEQUENCES

Configurations		Decoding configurations			
		(1, 1)	(6, 1)	(1, 6)	(3, 2)
B	×2	9222	10596	27666	9772
	×1.5	10351	11725	31053	21118
	SNR	14464	15838	43392	15013
	HEVC	7232	7919	43392	15013
A	×2	17652	19026	52956	18202
	SNR	27928	29302	8378	28478
	HEVC	13964	14651	83784	28478
C	×2	35273	36648	105820	35823
	SNR	56084	57458	168252	56634
	HEVC	28042	28729	168252	56634

resulting in lower memory than the wavefront configuration. However, the frame-based parallelism considerably increases the decoder memory, since both the global and the local structures are allocated by thread (m local and m global structures for the SHVC decoder), except the DPB, which is shared between all decoders of one layer. This considerably increases the memory allocation from ≈9 MBytes in single thread to ≈27 MBytes with using six threads $(n, m) = (1, 6)$. Table XIII gives the memory allocation of the multilayer HEVC decoder in different scalability and parallel decoding configurations.

V. CONCLUSION

In this paper, we presented a software decoder solution that supports 22 HEVC profiles including the three profiles of HEVC version 1, 16 range extension profiles among the 21 Rext profiles (up to 4:4:4, 12 bits), as well as the two scalable and one multiview profiles. The proposed multilayer HEVC decoder is based on the *OpenHEVC* decoder, which is an open-source implementation of the HEVC standard. The multilayer HEVC decoder is heavily optimized in SIMD methods for x86 architecture to reach real-time decoding requirements on a single-core platform. Moreover, the decoder architecture is designed to be parallel to leverage multicore platforms for higher DFR and lower frame latency performance.

We also investigated the complexity of the multilayer decoder decoding two SHVC layers in different scalability configurations: spatial and fidelity. The experimental results showed that the multilayer HEVC decoder decoding two layers introduces 40%–71% additional complexity in respect with the simulcast configuration. Moreover, the hybrid parallelism solution in the multilayer HEVC decoder provides a good tradeoff performance between speedup, latency, and memory usage. Moreover, the low level optimizations with the hybrid parallelism solution allow to reach a real-time decoding of 4Kp60 video sequences in 2× spatial scalability configurations on a 6-core Intel i7 processor running at 3.4 GHz. The first end-to-end video streaming demonstration using the proposed multilayer HEVC decoder under the Multimedia Open Source Project player was shown in various events including the 108th MPEG meeting in Valencia [32] and the demonstration section of the IEEE International Conference on Multimedia and Expo [31].

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 J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[2] 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.

[3] *High Efficiency Video Coding*, document Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.

[4] J. Chen, J. Boyce, Y. Ye, M. Hannuksela, G. J. Sullivan, and Y. K. Wang, *HEVC Scalable Extensions (SHVC) Draft Text 7*, document JCTVC-R1008, Sapporo, Japan, Jul. 2014.

[5] G. Tech, K. Wegner, Y. Chen, M. Hannuksela, and J. Boyce, *MV-HEVC Draft Text 9*, document JCT3V-I1002, Sapporo, Japan, Jul. 2014.

[6] J. Boyce *et al.*, *Edition 2 Draft Text of High Efficiency Video Coding (HEVC), Including Format Range (RExt), Scalability (SHVC), and Multi-View (MV-HEVC) Extensions*, document JCTVC-R1013, Sapporo, Japan, Jul. 2014.

[7] Z. Shi, X. Sun, and F. Wu, "Spatially scalable video coding for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1813–1826, Dec. 2012.

[8] P. Helle *et al.*, "A scalable video coding extension of HEVC," in *Proc. IEEE Conf. Data Compress.*, Mar. 2013, pp. 201–210.

[9] J. Chen *et al.*, "Scalable video coding extension for HEVC," in *Proc. IEEE Conf. Data Compress.*, Mar. 2013, pp. 191–200.

[10] Z. Zhao, J. Si, J. Ostermann, and W. Li, "Inter-layer intra mode coding for the scalable extension of HEVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 1636–1639.

[11] ISO/IEC-JTC1/SC29/WG11 and ITU-T-SG16, *Joint Call for Proposals on Scalable Video Coding Extensions of High Efficiency Video Coding (HEVC)*, document N12957 or ITU-T SG16 and VCEG-AS90, ISO/IEC JTC1/SC29/WG11 (MPEG), Stockholm, Sweden, Jul. 2012.

[12] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[13] V. Seregin, T. D. Chuang, Y. He, D. K. Kwon, and F. Le Leannec, *AHG Report: SHVC Software*, document JCTVC-L0011, Geneva, Switzerland, Jan. 2013.

[14] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Appl. Signal Process.*, vol. 2009, no. 8, Jan. 2009.

[15] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[16] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec. 2012.

[17] 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 J. Solid-State Circuits*, vol. 49, no. 1, pp. 61–72, Jan. 2014.

[18] *Open Source HEVC Decoder (OpenHEVC)*. [Online]. Available: https://github.com/OpenHEVC, accessed Jul. 2014.

[19] C. C. Chi *et al.*, "Parallel scalability and efficiency of HEVC parallelization approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1827–1838, Dec. 2012.

[20] G. Clare, F. Henry, and S. Pateux, *Wavefront Parallel Processing for HEVC Encoding and Decoding*, document JCTVC-F274, Turin, Italy, Jul. 2011.

[21] C. Meenderinck, A. Azevedo, B. Juurlink, M. Alvarez Mesa, and A. Ramirez, "Parallel scalability of video decoders," *J. Signal Process. Syst.*, vol. 57, no. 2, pp. 173–194, Nov. 2009.

[22] C. C. Chi, M. Alvarez-Mesa, J. Lucas, B. Juurlink, and T. Schierl, "Parallel HEVC decoding on multi- and many-core architectures: A power and performance analysis," *J. Signal Process. Syst.*, vol. 71, no. 3, pp. 247–260, 2013.

[23] B. Bross *et al.*, "HEVC real-time decoding," *Proc. SPIE, Appl. Digit. Image Process. XXXVI*, vol. 8856, p. 88561R, Sep. 2013.

[24] W. Hamidouche, M. Raulet, and O. Deforges, "Multi-core software architecture for the scalable HEVC decoder," in *Proc. IEEE ICASSP*, May 2014, pp. 7545–7549.

[25] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, *Scalable HEVC (SHVC) Test Model 4 (SHM 4)*, document JCTVC-O1007, Geneva, Switzerland, Nov. 2013.

[26] *FFmpeg: Open Source and Cross-Platform Multimedia Library*. [Online]. Available: http://www.ffmpeg.org, accessed Jul. 2014.

[27] *SHVC Reference Software Model (SHM)*. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svnSHVCSoftware/, accessed Jul. 2014.

[28] *3H HEVC Reference Software Model (HTM)*. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/, accessed Jul. 2014.

[29] V. Seregin and Y. He, *Common SHM Test Conditions and Software Reference Configurations*, document JCTVC-O1009, Geneva, Switzerland, Nov. 2013.

[30] K. Müller and A. Vetro, *Common Test Conditions of 3DV Core Experiments*, document JCT3V-G1100, 7th Meeting, San Jose, CA, USA, Jan. 2014.

[31] W. Hamidouche, G. Cocherel, J. Le Feuvre, M. Raulet, and O. Deforges, "4K real time video streaming with SHVC decoder and GPAC player," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICME)*, Chengdu, China, Jul. 2014, pp. 1–2.

[32] W. Hamidouche, J. Le Feuvre, and M. Raulet, *4K Real Time Streaming With SHVC Decoder and GPAC Player*, document JCTVC-Q0050, Valencia, Spain, Mar. 2014.

[33] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. Amer. Fed. Inf. Process. Soc. Conf.*, 1967, pp. 483–485.

[34] J. L. Gustafson, "Reevaluating Amdahl's law," *Commun. ACM*, vol. 31, no. 5, pp. 532–533, May 1988.

Wassim Hamidouche (M'15) received the Engineering degree in computer science from University of Sciences and Technologies of Algiers, Bab Ezzouar, Algeria, in 2006, and the master's degree in electrical engineering and the Ph.D. degree in signal and image processing from University of Poitiers, Poitiers, France, in 2007 and 2010, respectively.

He started his research work on realistic physical layer and joint source-channel coding solutions, including image and video transmission over wireless channels. From 2011 to 2012, he was a Research Engineer with Canon Research Centre, Rennes, France, where he was involved in video compression standard High Effciency Video Coding and its scalable extension, SHVC. Since 2013, he has been a Research Engineer with Institut d'Électronique et de Télécommunications de Rennes Laboratory, IMAGE Group, Institut National des Sciences Appliquées de Rennes, Rennes. His research interests include efficient real time and parallel architectures for the new generation video coding standard, multimedia transmission over heterogeneous networks, and multimedia content security.

Mickael Raulet (M'09) received the Ph.D. degree in electronic and signal processing from INSA, Villeurbanne, France, in collaboration with Mitsubishi Electric ITE, Rennes, France, in 2006.

He joined ATEME, Rennes, in 2015, where he is in charge of a research team on video compression. He was with Research Institute of Electronics and Telecommunications of Rennes, Rennes, until 2014, where he was a Researcher in rapid prototyping of video coding standards. He was the Project Leader of several French and European projects. Until 2014, he was also a member of IRT B-COM, Rennes, a new research institute. Since 2007, he has been involved in the ISO/IEC MPEG standardization activities as a Reconfigurable Video Coding Expert. He has authored three book chapters and more than 80 international conferences and journal papers. His research interests include dataflow programming, signal processing systems, and video coding.

Dr. Raulet served as a member of the Technical Committee of the Design and Implementation of Signal Processing Systems of the IEEE Signal Processing Society and the Circuits and Systems for Video Technology Editorial Board.

Olivier Déforges received the Ph.D. degree in image processing from Polytechnique Nantes, France, in 1995.

He joined the Department of Electronic Engineering, Institut National des Sciences Appliquées de Rennes (INSA), Scientific and Technical University, Rennes, France, in 1996. He has been involved with the ISO MPEG Standardization Group since 2007. He is currently a Professor with INSA. He is also a member of the Institute of Electronics and Telecommunications of Rennes, Rennes, and leads the IMAGE Team with the Institut d'Électronique et de Télécommunications de Rennes Laboratory (40 peoples). He has authored more than 130 technical papers. His research interests include image and video lossy and lossless compression, image understanding, fast prototyping, and parallel architectures.