Scalable Video Coding for Backward-Compatible 360° Video Delivery Over Broadcast Networks

Thibaud Biatek, Wassim Hamidouche, Pierre-Loup Cabarat, Jean-Francois Travers, Olivier Déforges

Scalable Video Coding for Backward-Compatible 360° Video Delivery Over Broadcast Networks

Thibaud Biatek, Wassim Hamidouche, Pierre-Loup Cabarat, Jean-François Travers, and Olivier Déforges

***Abstract*—Recently, the coding and transmission of immersive 360° video has been intensely studied. The technologies provided by standards developing organizations mainly address requirements coming from over-the-top services. The terrestrial broadcast remains in many countries the mainstream medium to deliver high quality contents to a wide audience. To enable seamless introduction of immersive 360° video services over terrestrial broadcast, the deployed technologies shall fulfill requirements such as backward compatibility to legacy receivers and high bandwidth efficiency. While bandwidth efficiency is addressed by existing techniques, none of them enables legacy video services decoding. In this paper, a novel scalable coding scheme is proposed to enable immersive 360° video services introduction over broadcast networks. The experiments show that the proposed scalable coding scheme provides substantial coding gains of 14.99% compared to simulcast coding and introduces a limited coding overhead of 5.15% compared to 360° single-layer coding. A real-time decoding implementation is proposed, highlighting the relevance of the proposed design. Eventually, an end-to-end demonstrator illustrates how the proposed solution could be integrated in a real terrestrial broadcast environment.**

***Index Terms*—Video coding, scalability, HEVC, SHVC, UHD, 4K, 360°, broadcast, broadband.**

I. Introduction

THE CODING and representation of 360° video contents have been widely investigated inside the ITU-T and ISO/MPEG Joint Video Exploration Team (JVET) or Joint Collaborative Team on Video Coding (JCT-VC). Projections and coding are investigated in JVET while signaling of 360° characteristics with High Efficiency Video Coding (HEVC) [1] is standardized in JCT-VC. More immersive approaches extending 360° up to 6 degrees of freedom (6DoF) are also explored in MPEG Immersive project (MPEG-I) [2]. Many standardization initiatives around Virtual Reality (VR) have emerged outside of Motion Picture Expert Group (MPEG), for example in Digital Video Broadcasting (DVB) [3] or 3rd Generation Partneship Project (3GPP) [4]. Beside those Standard Developing Organizations (SDO), the Virtual Reality Industry Forum (VRIF) establishes guidelines and recommendations for an inter-operable and stabilized ecosystem around VR [5].

The existing delivery methods are IP-streaming compatible since they mainly address Over-The-Top (OTT) requirements on latency and bandwidth adaptation. However, none of these techniques can be introduced in broadcast networks due to their lack of interoperability with legacy Integrated Receiver/Decoder (IRD). A wide range of legacy IRDs cannot benefit from a software update, typically most of the existing TV sets that are not updated anymore after years, for several reasons (complex manual operations, no IP connection, outdated warranty period). The naive solution usually adopted in broadcast to introduce a new service while ensuring backward compatibility with legacy receivers consists in simultaneous broadcasting both services. This simulcast solution is simple to deploy, however, it introduces significant bit-rate cost which increases the required bandwidth resources. On the other hand, the broadcast of single-layer 360 enables better coding efficiency but it is not backward compatible.

To enable seamless introduction of new 360° video services over broadcast network, this paper proposes a new coding technique which is backward compatible with legacy IRD. The proposed method is based on a novel multi-layer coding architecture suitable for broadcast deployment. The design of Scalable High efficiency Video Coding (SHVC) [6], [7] is enhanced to support 2D to 2D-360 scalability, with adapted inter-layer processing. One 2D view is extracted from a complete 360° field and coded as HEVC base-layer. Therefore, the Base Layer (BL) is backward compatible with legacy HEVC IRD. The reconstructed BL is processed with geometrical transforms and turned into a reference picture that can be used to encode the Enhancement Layer (EL) in a more efficient way than a single-layer HEVC.

The proposed method has been evaluated and shows significant coding gains of −14.99% compared to Simultaneous Broadcast (simulcast) with a limited coding overhead of 5.15% in average compared to single-layer 360° HEVC coding. In addition, a software implementation is proposed in the real-time HEVC decoder *OpenHEVC* making use of both parallelism in inter-layer processing and Single Instruction Multiple Data (SIMD) optimizations. To attest the practicability of the proposed approach, an end-to-end demonstrator

Manuscript received March 11, 2019; revised June 21, 2019; accepted August 6, 2019. This work was supported in part by the Two Clusters (Images & Réseaux and Cap Digital), in part by the French Government, and in part by the Two Local Councils (Region Bretagne and Region Ile-De-France). *(Corresponding author: Thibaud Biatek.)*

T. Biatek and J.-F. Travers are with the Headends and Services Expertise Center, TDF, 35510 Cesson-Sévigné, France (e-mail: jean-francois.travers@tdf.fr).

W. Hamidouche, P.-L. Cabarat, and O. Déforges are with the Institute of Electronics and Telecommunications of Rennes, CNRS UMR 6164, 35708 Rennes, France, and also with the VAADER Team, Univ Rennes, INSA Rennes, 35043 Rennes, France (e-mail: whamidou@insa-rennes.fr; pierre-loup.cabarat@insa-rennes.fr).

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

Digital Object Identifier 10.1109/TBC.2019.2941073

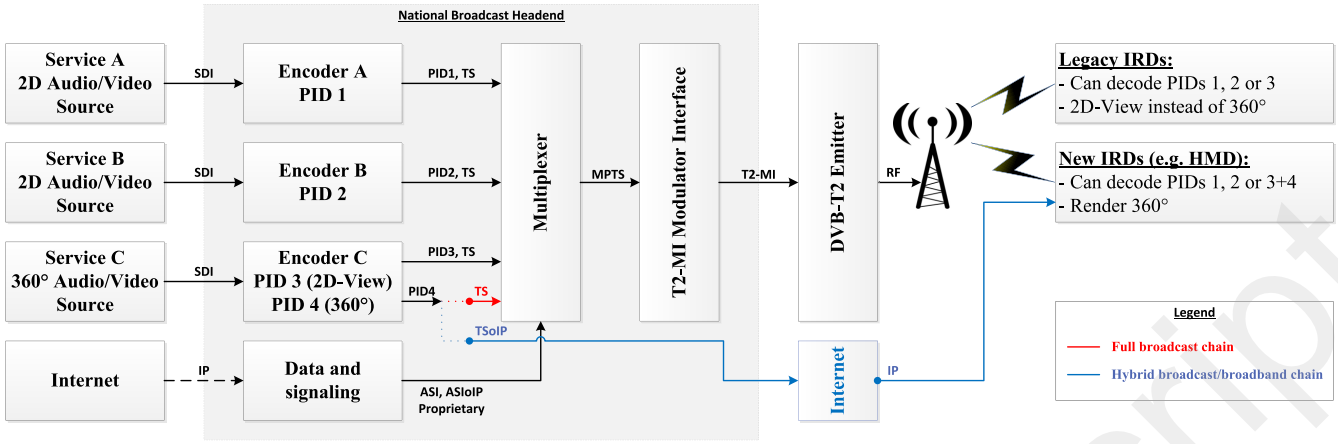


Fig. 1. Illustration of 3-services full broadcast and hybrid broadcast/broadband headend for introduction of 360° service.

presented during 2018 French Open is described. The demonstration shows that a bitstream encoded with the proposed coding scheme can be broadcasted over a classical Digital Terrestrial Television (DTT) architecture and properly decoded with both legacy TV set and a set-top-box respectively offering classical 2D or immersive experience with a remote controller.

The rest of this paper is organized as follows. The background and motivations of this work are given in Section II. Section III describes the proposed method and its implementation in the real-time *OpenHEVC* decoder. Section IV provides experimental results and performance of the proposed method, as well as discussion on software implementation aspects. Finally, an end-to-end demonstrator is presented in Sections V and VI concludes this paper.

II. Related Work and Motivations

A. Representation, Coding and Transmission of 360° Video

Usually, 360° video is captured with a rig of synchronized cameras that record overlapped areas of the surrounding environment. Then, the video sequences captured by each camera are stitched together to form a complete 360° planar representation. The recording devices can be either monoscopic or stereoscopic. Generally, the stitching stage delivers an equirectangular representation of the content that can be transformed in other format for encoding purpose.

Projection solutions from spherical field to 2D representation have been deeply explored in [8] for standardization purpose. Advanced padding methods have been investigated in [9], [10] to preserve texture continuity when motion compensation is performed across faces boundaries. Tools that improve coding efficiency for 360° video have also been studied. In [11], intra-coding is adapted to projection type while region-adaptive coding is performed in [12]. Adaptive quantization is explored in JVET [13] where the Quantization Parameter (QP) is adapted according to location into the 360° field. Rate-distortion optimization in the spherical domain has been explored in [14].

Concerning 360° video delivery scope, the widely deployed approach consists in dividing the 360° field into a tiled array which is dynamically delivered to the end-user according to its viewing angle [15]. This approach has been experimented using Dynamic Adaptive Streaming over HTTP (DASH) and MPEG media transport (MMT) streaming in [16] and [17], respectively. Moreover, multi-view and scalable coding has been investigated in [18]. In this approach, the signal is split into low-resolution primary views, almost non-overlapped and independently encoded with HEVC, and other high-resolution auxiliary views that overlapped with primary views are encoded by using combined Scalable and Multi-view HEVC extensions: SHVC and MV-HEVC [19], respectively. These approaches enable bandwidth saving by only transmitting required viewport, while full 360° field is transmitted in low-resolution to smoothly manage quick head movements.

B. Use-Cases

The introduction of new services on a broadcast platform requires specific features, such as backward compatibility and bandwidth efficiency. Fig. 1 illustrates the architecture of a full broadcast or hybrid broadcast/broadband headend for 360° service introduction. It can be observed that three audio/video services are delivered to the national broadcast headend: two classical 2D video services and one 360° immersive service. The services are encoded and multiplexed into a single MPEG-TS [20], [21] stream with embedded services. This stream is delivered to the T2-MI modulator interface. Then, the ready-to-broadcast stream is delivered to the DVB-T2 emitter that achieves the modulation and amplification stages before being transmitted to the antenna.

To deliver a backward-compatible video stream for newly introduced 360° service, the encoder C has to deliver a bitstream containing compatible versions for both VR and legacy IRDs, associated to a specific Packet Identifier (PID). To encode 360° video service into a multi-PID bitstream, the encoder C either operates in simulcast mode or uses an adapted multi-layered codec. Fig. 1 illustrates two ways of delivering content to receivers. In red, the full broadcast scenario is represented where all PIDs are transmitted on the air to receivers. This scenario requires to rescan services in the IRDs and to re-allocate the bitrate in the multiplex when 360° service is introduced. In blue, the hybrid broadcast/broadband

scenario is highlighted where PID enabling new service is separately sent to the receivers thanks to the broadband network. This hybrid scenario enables incremental services introduction while maintaining legacy services on the broadcast channel.

Considering these broadcast constraints, it appears from the literature that there is a lack that opens a room for improvements regarding coding and delivery of 360° in a broadcast ecosystem. In this paper, a novel multi-layered coding approach is proposed to introduce 2D backward-compatible 360° services in a full broadcast or hybrid broadcast/broadband network [22]. The proposed coding scheme is designed to be compatible with broadcast ecosystem and its requirements since it provides a bandwidth efficient multi-PID coding of 360° content that outperforms simulcast approach.

C. Motivations

This paper tackles the backward compatibility required by 360° video introduction over broadcast network. Literature shows that there is no coding solution that provides a 360° backward compatible stream. Indeed, the existing work addresses OTT delivery and does not address broadcast constraints. Although SHVC could be a potential solution, it only supports spatial, color-gamut and fidelity scalability. This paper proposes a novel coding solution, integrated to SHVC, that enables backward-compatible layered video coding suitable for broadcast delivery of 360° video services. The proposed solution has been implemented under a real time framework in *OpenHEVC* decoder showing a real time decoding of the EL for 360° service in 4K resolution, while the HD view-port is decoded on a classical TV receiver.

III. Proposed 360° Video Coding Scheme

A. Proposed Architecture

To enable backward compatibility with legacy IRDs, it is proposed to encode the 360° field in a scalable way. One or several views (N) are extracted from the 360° and encoded with a standard codec (e.g., HEVC), hence they are compatible with legacy HEVC IRDs. Each extracted view is encoded as a BL in a scalable coding scheme where it contributes to the encoding of the full 360° field EL. It must be noted that the extraction method includes projection processing that turns a 360° content into a viewable 2D scene.

The proposed encoding and decoding architecture is illustrated in Fig. 2 with two BL views ($N = 2$). It can be observed that two views are extracted from the 360° video source and are independently encoded with two HEVC encoders. In the same way as SHVC, reconstructed BL pictures are extracted from the Decoded Picture Buffer (DPB) and feed the EL encoder that achieves Inter Layer Processing (ILP) to construct the Inter Layer Reference (ILR) picture. The EL encoder takes advantage of the ILR to achieve higher coding efficiency.

The multi-PID stream is generated by multiplexing generated bitstreams and transmitted over the broadcast network. Once delivered, the PIDs #1 and #2 are decodable by all legacy HEVC IRDs providing backward-compatible views. The PID #3 contains 360° EL and requires the joint decoding of all BLs to enable proper reconstruction.

The proposed method requires specific ILP and adapted signaling design. Indeed, the ILR construction process must achieve specific geometrical transformations that may vary in time to improve the prediction efficiency. To handle this dynamic processing, information about projections has to be transmitted in the bitstream to the decoder. The proposed ILP and signaling are further described in the following sections.

In fact, the proposed architecture can be transposed to any standard codec (e.g., MPEG-4/AVC [23]) and to any number of BLs (N). In that case, the number of encoders should be duplicated to fit the number of expected BLs.

B. Proposed Inter-Layer Processing

The required ILP transforms the BL reconstructed pictures stored into the DPB in reference pictures that can be used in the EL encoder. Hence, the projection and location parameters used during viewport extraction have to be considered in the ILP stage, as well as the projection method used to represent the 360° field in the EL. In this paper, the particular case of Equi-Rectangular Projection (ERP) is considered, with viewport generation based on rectilinear projection [8]. In practice, other projection techniques could be considered in the proposed method for instance Cube-Map Projection (CMP).

To build the ILR picture, a three-steps processing is applied to convert viewport generated pictures into a 360° ERP field. The 360° and viewport resolutions are respectively (W_{360}, H_{360}) and (W_v, H_v). These steps are illustrated in Figure 4 where the proposed solution is illustrated in a typical scalable encoder.

1) Step 1 (Coordinates Mapping): The 2D sampling points in the destination space are mapped to coordinates into the source space. In this case, it is required to map coordinates into the ILR ERP field (m, n) to the viewport field (m', n') coded in the BL with $0 \leq n < H_{360}$, $0 \leq m < W_{360}$, $0 \leq n' < H_v$, $0 \leq m' < W_v$.

First, each coordinates (m, n) in the ERP field are converted into the normalized UV plane:

$$u = (m + 0.5)/W_{360}, \quad v = (n + 0.5)/H_{360}. \tag{1}$$

Then, the associated longitude and latitude coordinates $(\phi,\ \theta)$ in the spherical XYZ space are computed:

$$\phi = (u - 0.5) \times 2 \times \pi, \quad \theta = (0.5 - v) \times \pi. \tag{2}$$

The (X, Y, Z) coordinates into the spherical XYZ space are derived:

$$X = \cos(\theta) \times \cos(\phi),\ Y = \sin(\theta),\ Z = -\cos(\theta) \times \sin(\phi). \tag{3}$$

Second, the pivot coordinates into the XYZ space are converted into the viewport generated picture. Here, the viewport generation parameters signaled into the bitstream are used. Those parameters are: the viewport center coordinates defined by Yaw and Pitch rotation parameters (ϕ_c, θ_c), the horizontal and vertical Field Of View (FOV) dimensions (F_h, F_v) and

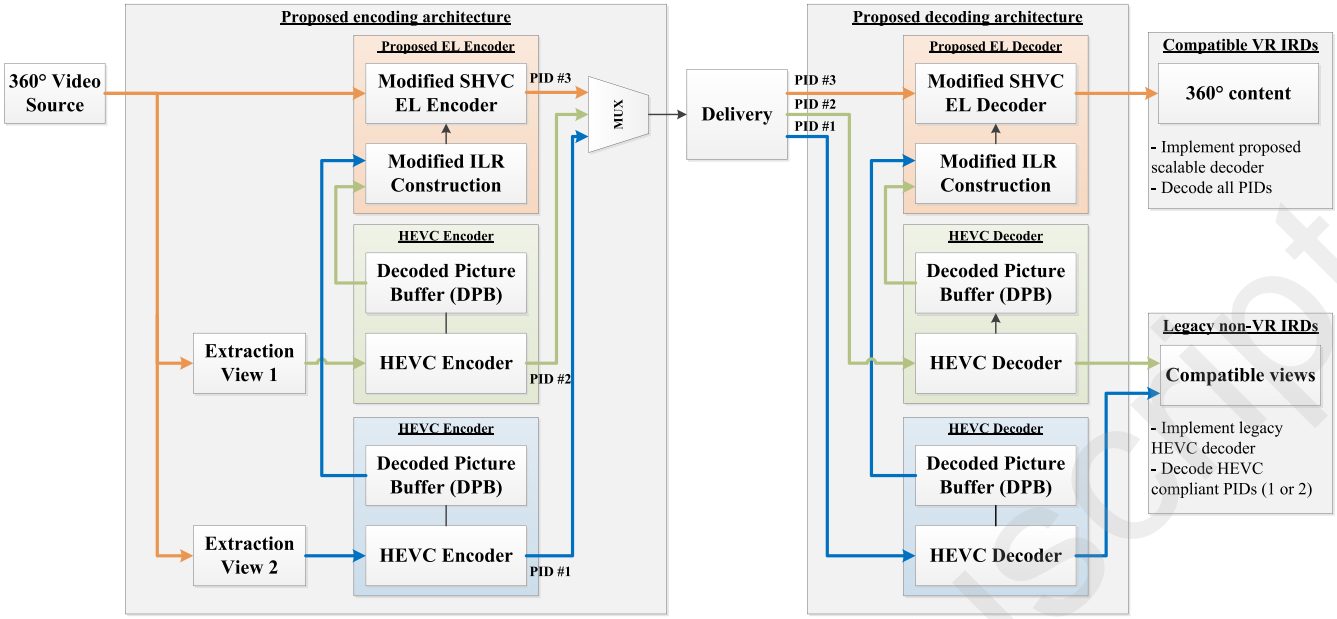
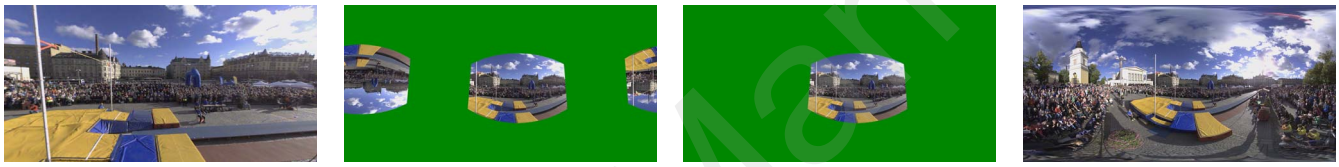


Fig. 2. Proposed multi-PID encoding and decoding architecture, in $N = 2$ configuration.



(a) Encoded BL (b) ILR before filtering (c) ILR after filtering (d) Encoded EL

Fig. 3. Illustration of proposed ILR processing for single-BL HD to 360° UHD layered coding.

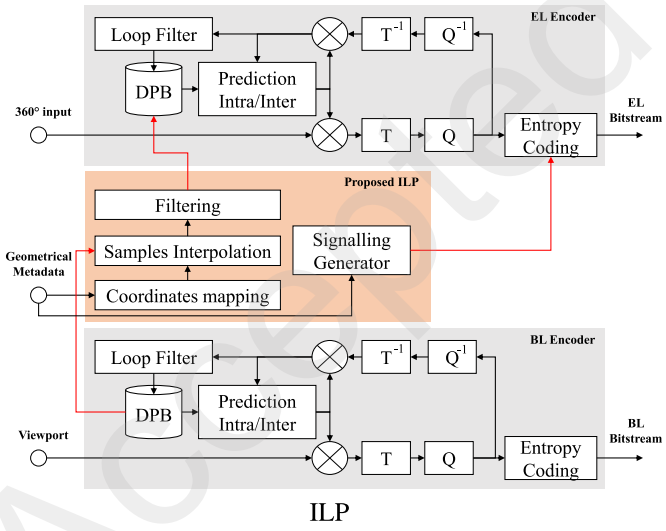


Fig. 4. Scalable encoder architecture with the proposed.

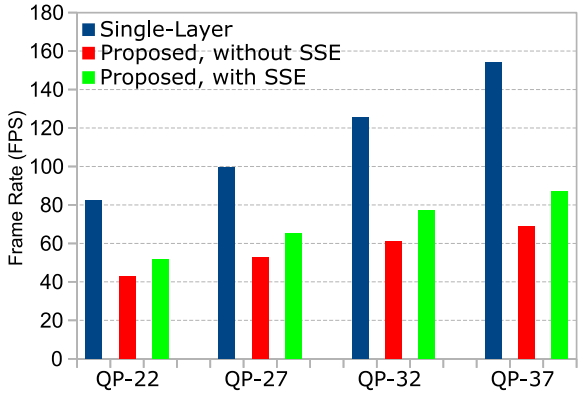


Fig. 5. Average decoding framerate comparison between single-layer and proposed method with and without SSE.

the viewport dimensions (W_v, H_v). The XYZ space coordinates are inverse rotated by:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R^{-1} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4}$$

where $\cdot$ is the matrix multiplication, R^{-1} is the inverse of the rotation matrix R, computed as follows:

$$R = \begin{bmatrix} c & -s \times \sin(\theta_c) & s \times \cos(\theta_c) \\ 0 & \cos(\theta_c) & \sin(\theta_c) \\ -s & -c \times \sin(\theta_c) & c \times \cos(\theta_c) \end{bmatrix} \tag{5}$$

with:

$$s = \sin(\phi_c + \pi/2), \quad c = \cos(\phi_c + \pi/2) \tag{6}$$

Then, the (x, y, z) coordinates are normalized:

$$x = x'/z', \quad y = y'/z', \quad z = 1. \tag{7}$$

The (m', n') viewport space coordinates are derived as follows:

$$m' = \frac{(x + \tan(F_h/2)) \times W_v}{2 \times \tan(F_h/2)} - 0.5,$$
$$n' = \frac{(\tan(F_v/2) - y) \times H_v}{2 \times \tan(F_v/2)} - 0.5. \tag{8}$$

Finally, a vectorized Look-Up-Table (LUT) P is updated for the current (m, n) position. The LUT P links the position (m, n) in the ILR with associated integerized position in the BL. In addition, the floating values of (m', n') are also stored to be used during interpolation stage. If positions (m, n) are not in the range of BL dimensions (W_v, H_v) a default value -1 is considered, otherwise the LUT is updated as follows:

$$P(m \times W_{360} + n) = \lfloor m' + 0.5 \rfloor \times W_v + \lfloor n' + 0.5 \rfloor. \tag{9}$$

The described processing has to be achieved twice during ILR construction, on both luma and chroma channels (for 4:2:0 case).

2) Step 2 (Samples Interpolation): The second step consists in interpolating ILR samples based on positions indexed in P. Only the pixels with value different from -1 are interpolated. Two Lanczos filters are used during this stage respectively 3-taps and 2-taps for luma and chroma channels. More information about interpolation processing is provided in [8].

3) Step 3 (Picture Filtering): Once interpolated, the ILR picture is filtered before being stored and used as a reference by the EL. Indeed, the periodic property of trigonometric functions used in the previously described coordinates mapping leads to a duplication of the interpolation window. In Fig. 3, the BL and EL pictures are illustrated together with ILR before and after filtering. It is noticed that the filtering process removes projected-BL duplication. The final ILR picture is ready to be used as reference for EL coding.

C. Proposed Signaling Design and Integration in SHVC

The proposed ILP requires signaling of projection parameters in the bistream for proper reconstruction at the decoder side. First, projections used in BL (rectilinear projection) and EL (ERP) must be signaled. Second, the viewport-generation parameters must be signaled which includes the Yaw and Pitch parameters as well as the horizontal and vertical FOV dimensions. They can be transmitted once in the Sequence Parameter Set (SPS) header for the entire sequence, but also dynamically in the Picture Parameter Set (PPS) in case of a moving viewport. In this paper, a SPS level signaling is considered since static viewports are encoded in the BLs.

The proposed method is implemented in the SHVC reference software 16.10 [24], in a straightforward way. The motion vector mapping process used in SHVC is deactivated. The resampling process used to construct the ILR is replaced by the proposed processing, which simply constructs the ILR picture as proposed in this paper. The mode selection process remains unchanged, and the signaling design is also implemented. The software is also modified to handle several separated BLs encoding (e.g., $N = 2$ BL views plus one 360° EL).

The integration into a typical SHVC encoder is illustrated in Figure 4 with the signalling generator and all ILP steps.

D. Real-Time Implementation in OpenHEVC

To validate the targeted use-cases in a real-time environment, the proposed method is implemented in the *OpenHEVC* decoder [25]. *OpenHEVC* has been selected since it already implements real-time SHVC decoding in spatial, quality, bit-depth and codec scalability [26]. To achieve real-time decoding in SHVC configuration, *OpenHEVC* takes profit of both inter layer parallelism and SIMD optimizations of the most time consuming decoding operations including the up-sampling filters in the case of spatial scalability [25].

As depicted in the previous subsection, the ILR interpolation process will not only depend on the Yaw, Pitch and FOV dimensions but also on the frame dimension. Since the frame dimension is specified into the SPS, the mapping process needs to be updated when a new SPS is received by the decoder.

The LUTs generation for the coordinates mapping is an expensive process since it relies on trigonometric operations on floating point values with double precision and it has to be performed on each pixel in the ILR ERP field. It is worth noticing that if the received SPS does not change, the LUTs generation can be skipped, so in the case of a static viewport it has only to be performed once at the decoder initialization. Since LUTs optimization with SIMD instructions is immediate, much of the optimization relies on ILR interpolation. The ILR interpolation is based on Lanczos filters that shall be applied every time the ILR prediction is used. Processing optimization was carried out along two axes:

- Avoid non-mandatory and heavy operations
- Use vector parallelism operations.

1) Removal of Non-Mandatory Operations: It is observed that if an area on the ILR frame is not required as predictor in the EL frame, the Lanczos interpolation can be avoided. In order to do so, the Lanczos interpolation filter is only applied as soon as a Prediction Unit (PU) in the EL requires local ILR picture data. It has been chosen to work on a Coding Tree Unit (CTU)-level granularity since it avoids multiple processing of small PU. The CTU is then marked as available for the Motion Compensation Prediction (MCP) process and the resulting 360 ILR Lanczos reference CTU is written onto the ILR reference frame. Therefore, once a CTU is processed it becomes available to other PUs taking their reference inside. Hence, the filtering process is skipped when superfluous in the decoding process while keeping overlapping filtering operations on CTU borders at a reasonable amount.

2) Usage of Vector Parallelism: It appears that getting along with SIMD operations to perform the Lanczos interpolation is feasible. The Streaming SIMD Extensions (SSE) instructions set targeting $\times 86$ platforms has been selected. However, it is worth noticing that the proposed method could be ported to Advanced Vector Extension (AVX) instructions with minimal effort. Targeting other architecture such as ARM would not be an issue either. The use of SIMD on this

TABLE I
CODING PERFORMANCE OF PROPOSED METHOD VERSUS SIMULCAST AND 360° SINGLE-LAYER FOR $N = 1$ AND $N = 2$

(a) Bitrate savings versus simulcast.

Sequence	N	BD-Rate[YUV]		
		PSNR	WS-PSNR	S-PSNR-NN
AerialCity	1	-14.89%	-14.90%	-14.89%
	2	-22.41%	-22.74%	-22.74%
Balboa	1	-12.84%	-12.69%	-12.68%
	2	-25.05%	-24.98%	-24.97%
BranCastle	1	-7.76%	-6.42%	-7.23%
	2	-17.54%	-16.42%	-17.09%
Broadway	1	-12.98%	-12.95%	-13.15%
	2	-26.30%	-26.39%	-26.36%
Harbor	1	-23.27%	-23.40%	-23.41%
	2	-31.63%	-31.75%	-31.78%
Gaslamp	1	-15.78%	-15.72%	-15.89%
	2	-29.16%	-28.79%	-28.95%
KiteFlite	1	-17.64%	-17.84%	-17.80%
	2	-26.47%	-26.87%	-26.82%
Landing2	1	-26.48%	-25.87%	-25.87%
	2	-32.26%	-31.86%	-31.83%
PoleVault	1	-8.92%	-8.41%	-8.40%
	2	-21.46%	-21.27%	-21.22%
Trolley	1	-9.34%	-9.52%	-8.55%
	2	-21.48%	-21.71%	-21.77%
Mean	1	-14.99%	-14.77%	-14.89%
	2	-25.38%	-25.28%	-25.35%

(b) Coding overhead versus 360° single-layer.

Sequence	N	BD-Rate[YUV]		
		PSNR	WS-PSNR	S-PSNR-NN
AerialCity	1	4.51%	4.46%	4.47%
	2	7.27%	6.73%	6.73%
Balboa	1	8.78%	8.95%	8.96%
	2	11.43%	11.52%	11.53%
BranCastle	1	2.84%	4.33%	3.43%
	2	10.91%	12.45%	11.46%
Broadway	1	8.91%	8.92%	8.66%
	2	12.07%	11.89%	11.95%
Harbor	1	6.92%	6.76%	6.74%
	2	11.01%	10.83%	10.80%
Gaslamp	1	4.54%	4.61%	4.40%
	2	7.88%	7.89%	7.64%
KiteFlite	1	2.89%	2.64%	2.69%
	2	5.41%	4.82%	4.89%
Landing2	1	6.87%	7.73%	7.73%
	2	11.10%	11.73%	11.78%
PoleVault	1	2.63%	3.24%	3.26%
	2	7.13%	7.47%	7.54%
Trolley	1	2.58%	2.36%	2.33%
	2	6.14%	5.84%	5.76%
Mean	1	5.15%	5.40%	5.27%
	2	9.04%	9.12%	9.01%

type of filter raises two issues. First, the non-linear coordinates mapping between source and destination samples. More specifically, the direct neighbor of a destination sample does not necessarily uses the direct neighbor of its reference sample neighbor. Also, the interpolation filter coefficients also depend on the sample location. Each sample has to be processed independently since intermediate results cannot be reused in a deterministic way. Hence, LUTs are used to map the position of each destination sample to its center reference for interpolation as well as the corresponding filter coefficients location. Second, boundary conditions are problematic since the BL reference picture is supposed to be taken from the BL DPB that prevent optimization from using edge emulation techniques. Moreover, the result output has to be masked in a non linear way before writing in the ILR picture, to ensure the MCP can be processed safely. To overcome these issues, it is chosen to discard SIMD operations reverting to sequential operations when the interpolation process encounters a border condition. In order to quickly alternate between SIMD and per pixel template operation, a binary map is used to detect whether a border condition is detected or not.

IV. EXPERIMENTAL RESULTS

The performance of the proposed solution is assessed in two ways. First, from a coding prospective to check that the solution addresses bandwidth efficiency problematic. Second, in terms of decoding capabilities to validate that the design is realistic from an implementation point of view.

A. Experimental Setup

The experiments are conducted using sequences of the JVET Common Test Conditions (CTC) for 360° video [27]. The 8K sequences are down-sampled to a UHD (4K) resolution and two non-overlapped HD viewports V1 and V2 are extracted to be used as BLs:

- V1: Yaw = 20°, Pitch = −5°, hFOV=110° and vFOV = 80°
- V2: Yaw = −100°, Pitch = −5°, hFOV = 110° and vFOV = 80°

Encoding is carried out in Random Access (RA) configuration, according to JCT-VC SHM CTC that uses four fixed QP values $QP \in \{22, 27, 32, 37\}$. As similar definition is encoded on each layer, a zero delta-QP is chosen between BL and EL. Two encoding setups are evaluated, respectively with $N = 1$ and $N = 2$ BLs. The coding performance –provided in Table I– is measured with Bjøntegaard Delta Bit Rate (BD-BR) metric [28] in comparison with simulcast (i.e., separate layers encoding) and single-layer encoding of the 360° layer. The WS-PSNR and S-PSNR-NN distortion metrics recommended by JVET for 360° performance evaluation [27] are used beside of Peak Signal to Noise Ratio (PSNR). A positive BD-BR value means a coding overhead while a negative value refers to a bitrate reduction.

The real-time decoding is evaluated on a Intel Core I7-6820HQ running at 2.70 GHz, on a Linux platform with a version 4.4.0-141-generic of the kernel as distributed by in Ubuntu version 16.04.5 LTS. In order to get rid of the Operating System (OS) performance variation, measurements result from averaged values of ten decoding processes. The presented results were obtained using 12 threads in inter layer frame parallel configuration. Both decoding runtime and achieved frame-rate were measured.

Three decoding experiments have carried out. First experiment aims at measuring Simulcast single-layer performance. Second experiment evaluates the proposed 360° ILR filtering

TABLE II
MEASURED DECODING FRAME RATES. (a): SINGLE-LAYER. (c) AND (b): WITH AND WITHOUT SIMD OPTIMIZATIONS, RESPECTIVELY

Sequences	QP-22			QP-27			QP-32			QP-37		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
AerialCity	51.87	31.61	35.71	95.61	46.66	58.48	142.94	59.17	79.16	175.14	69.44	89.02
Balboa	74.22	37.97	44.38	99.51	43.67	54.55	114.16	48.23	62.76	92.11	51.37	70.26
BranCastle	33.63	24.53	27.47	44.66	30.77	35.97	55.84	39.06	46.44	89.49	43.42	54.45
Broadway	72.97	35.34	42.31	67.56	41.90	52.54	101.00	47.69	62.76	136.65	51.37	69.44
Gaslamp	144.11	74.07	87.46	170.09	85.23	106.76	196.35	96.15	115.83	222.41	102.04	127.66
Harbor	136.75	59.64	76.53	167.07	68.97	89.82	196.62	75.95	102.04	225.06	85.96	113.64
KiteFlite	92.07	44.64	55.05	84.69	52.91	67.42	117.95	59.52	78.13	154.31	65.08	79.37
Landing2	55.33	32.09	39.06	66.38	38.86	49.02	74.84	44.25	57.47	117.22	49.18	65.22
PoleVault	48.65	28.85	33.78	63.45	38.71	47.69	91.91	53.00	65.08	141.08	65.50	83.80
Trolley	112.73	62.11	72.99	138.64	77.92	87.72	163.41	89.82	103.09	188.69	106.01	118.58
Min	**33.63**	**24.53**	**27.47**	**44.66**	**30.77**	**35.97**	**55.84**	**39.06**	**46.44**	**89.49**	**43.42**	**54.45**
Max	**144.11**	**74.07**	**87.46**	**170.09**	**85.23**	**106.76**	**196.62**	**96.15**	**115.83**	**225.06**	**106.01**	**127.66**
Average	**82.23**	**43.09**	**51.48**	**99.77**	**52.56**	**65.00**	**125.50**	**61.28**	**77.28**	**154.22**	**68.94**	**87.14**

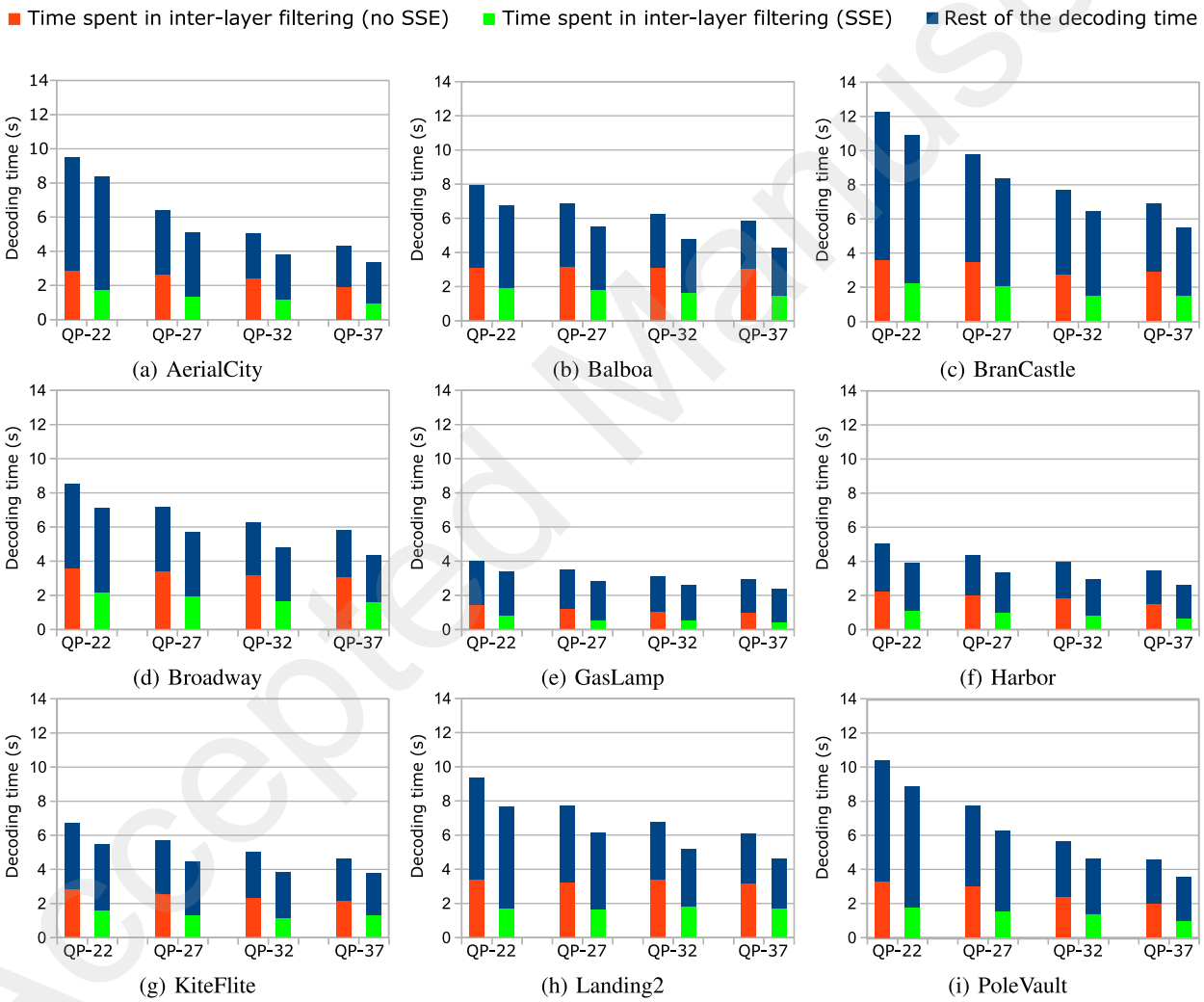


Fig. 6. Ratio of time spent in 360° ILR filtering with and without SSE optimizations compared to whole decoding time.

method implemented using sequential instructions, while the third experiment assesses performance when taking advantage of SSE optimization. The ILR filtering operations were switched on/off in second and third experiments to retrieve the relative times spent into 360° ILR filtering.

The decoding frame rates are provided in Table II. Fig. 5 compares reached frame-rate for all experiments. Fig. 6 summarizes the decoding times and the times spent in 360° ILR for each sequence with and without SSE optimization. Finally, Table III outlines the speedup in the 360° ILR filtering process brought by the SSE implementation.

TABLE III
AVERAGE SPEEDUP ENABLED BY SIMD OPTIMIZATIONS

Sequences	QP-22	QP-27	QP-32	QP-37
AerialCity	1.62	1.95	2.11	1.96
Balboa	1.58	1.75	1.86	2.05
BranCastle	1.57	1.68	1.80	1.91
Broadway	1.64	1.75	1.90	1.96
Gaslamp	1.74	2.31	2.04	2.44
Harbor	2.01	1.99	2.19	2.25
KiteFlite	1.79	1.91	2.03	1.62
Landing2	1.98	1.98	1.85	1.88
PoleVault	1.86	1.94	1.77	1.99
Trolley	1.79	1.61	1.84	1.75
Min	**1.57**	**1.61**	**1.77**	**1.62**
Max	**2.01**	**2.31**	**2.19**	**2.44**
Average	**1.76**	**1.89**	**1.94**	**1.98**

B. Encoding Performance

The complete coding performance is provided in Table I. The bitrate savings enable by the proposed method against simulcast are provided in Table Ia while the coding overhead compared to single-layer encoding is presented in Table Ib.

Overall, it is observed that consistent coding gains are obtained compared to simulcast coding. The PSNR-based BD-BR shows gains of -14.99% and -25.38% respectively for $N = 1$ and $N = 2$ BLs. This substantial gain is confirmed by WS-PSNR and S-PSNR-NN based BD-BR. It is observed that some sequences perform less than other, for instance *Bran-Castle*, *Pole-Vault* and *Trolley*. Those particular sequences are static and the selected views are not centered on moving and hard to encode areas, thus inter-layer prediction is less efficient which leads to lower coding enhancement. The coding overhead compared to single-layer encoding is limited with $+5.15\%$ and $+9.04\%$ for $N = 1$ and $N = 2$ BLs, respectively. This performance remains similar with all 360° related distortion metrics (WS-PSNR and S-PSNR-NN).

In average, adding a second BL leads to a BD-BR additional enhancement of -10% against simulcast and $+5\%$ less bitrate overhead compared to single-layer which is reasonable. It is observed that BLs contribute in an unbalanced way in some cases, for example in *Landing2* sequence where introducing additional BL improves coding gains from -26.48% to -32.26% versus simulcast coding.

C. Decoding Performance

At first glance, Fig. 5 shows that the achieved average decoding frame rates is already high enough to be qualified as real time. The worst case is related to the second experiment at the lowest QP which performs over 30 FPS (native sequence framerate). The implementation of SSE optimization in the third experiment raises the average decoding framerate over 50 FPS. Without any surprise, the decoding frame rates are higher with an increase of QP in all experiments. This is expected since lower QP leads to smaller PUs, but also to a lower amount of skipped Transform Unit (TU)s and more coefficients to be read and decoded by the Context-Adaptive Binary Arithmetic Coding (CABAC) engine. Thus, this makes the decoding process heavier for high bitrates.

It is noticed that the proposed 360° scalable content decoding performs poorly against the Simulcast Single Layer. This is also expected since SHVC requires an additional HEVC decoding process for the BL and inter-layer filtering, which explains the frame rates differences illustrated in Table II.

It can be observed from Fig. 6 that the decoding bottleneck does not seem to come from the 360° ILR filtering process and seems more related to classical HEVC decoding process. It appears from Fig. 6 that the time spent into 360° ILR filtering is more sensitive to the sequence characteristics than on the QP. Indeed, the decoding times is relatively stable which indicates that inter-layer prediction is applied to the same PUs locations, which varies few with regards to the QP. An other interesting fact is that the sequences spending the less time in inter-layer processing are those using a static camera, namely *Gaslamp*, *Harbor*, *KiteFlite*, *PoleVault* and *Trolley*. This can be explained by the fact that in those cases the motion compensation prediction is preferred to inter-layer prediction since it does not require complex motion description into the 360° ERP domain.

The speedup brought to the ILR filtering by using SSE is provided in Fig. 6 and summarized in Table III. The speedup values spread from 1.57 for *BranCastle* at QP -22 to 2.44 for *GasLamp* at QP -37. The average speedup tends to slowly increases from 1.76 to 1.98 according to the QP. The differences in speedup might be explained by the fact that SSE implementation does not apply to BL borders. According to the number of time the decoder faces a BL borders when 360° ILR is required the speedup might vary slightly because the same sequential function is used between both implementations instead of the optimized one. The more the 360° ILR is used on the BL borders, the lower the speedup will be between both experiments.

V. END-TO-END DEMONSTRATOR

The proposed approach has been demonstrated during the 2018 French Tennis Open. The setup is illustrated in Fig. 7.

A one minute sequence was recorded offline and produced in 4K equirectangular format. In addition, a static HD view was extracted, covering the whole tennis court.

The encoding was carried out with the proposed method, implemented into the SHVC reference software, as described in Section III-C. Several bitstreams were created using fixed QP, as described in Section IV. The targeted DVB-T2 modulation profile is detailed in Table IV and enables an effective bandwidth of 34.908 Mbps. Empirically, the bitstream generated with a QP value of 22 is selected because it fits into the T2 profile, with an overall average bitrate of 30 Mbps. The bitstream is then packaged using GPAC [29] producing a DVB compliant MPEG-TS file ready to be broadcasted.

A laptop **1*a*** is used together with a DekTec DTU-215 DVB-T2 modulator **1*b*** [30] to achieve modulation stage and output in RF format. The prepared MPEG-TS file is used with DekTec modulation software that output in the RF channel using a carrier frequency of 514.166MHz. The RF channel is distributed to two separate paths, as illustrated in the Fig. 7.

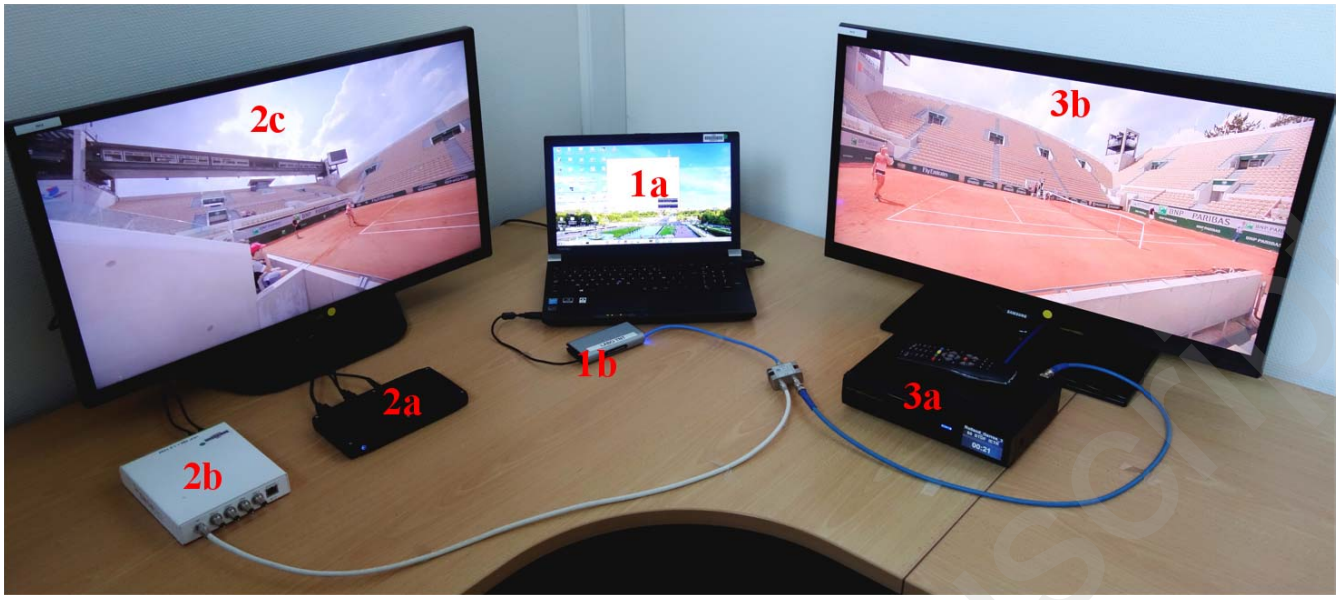


Fig. 7. End-to-end demonstrator of backward compatible 360° SHVC content broadcast, **1*a***) laptop streamer, **1*b***) DekTec DTU-215 DVB-T2 modulator, **2*b***) TestTree Referee-II receive , **2*c***) 360°/SHVC decoder, **3*a***) legacy receiver, **3*b***) legacy HD display.

TABLE IV
TARGETED DVB-T2 PROFILE

Parameter	Value
iFFT	32k
Modulation	256-QAM-R
Code-rate	2/3
LDPC length	64800 bits
Pilot pattern	PP6
Frame length	58
Guard interval	1/32
Resulting bitrate	34.908Mbps

The first receiver is the one compatible with the proposed solution. For practical reason, the receiver is split into two devices. The RF signal is demodulated with a TestTree Referee-II receiver **2*b*** [31]. This receiver is managed from the Intel NUC **2*a*** [32] and output the MPEG-TS stream locally (loopback), in multicast UDP format. A version of GPAC supporting the real-time decoding described in Section III-D is installed on the Intel NUC. The UDP-multicasted bistream is decoded by GPAC that supports 360° navigation with a USB remote control. The video output is displayed on the TV set **2*c*** through HDMI connection, and viewer can navigate in the content with a remote controller.

The second receiver is the one for which a backward-compatible experience is expected. The RF signal is received by a Dreambox DM-920 Set-Top-Box which is both DVB-T2 and HEVC compatible [33]. After channels scanning, the one located on 514.166 Mhz carrier is detected and properly displayed on the TV set ***3b*** through HDMI.

As expected, the produced 360° video service can be received by legacy and new compatible receivers. From a quality-of-experience prospective, the legacy service is displayed with an HD resolution and a slight fisheye effect that gives an immersive feeling to legacy viewers. The user equipped with the adapted receiver can navigate into the content and benefits from high quality immersive experience.

VI. CONCLUSION

In this paper, a new layered video coding approach is proposed for broadcast delivery of immersive 360° contents. The proposed approach is backward compatible with existing legacy receivers through its base layers and also addresses new 360° receivers with its enhancement layer. From coding performance perspective, it appears that the proposed method significantly outperforms simulcast approach with a limited overhead compared to 360° single-layer coding. The proposed solution can be considered as non normative scalability for 360 video in SHVC and can be proposed for the standardisation of the scalable extension of next generation video coding standard Versatile Video Coding (VVC) as a normative solution.

A real-time decoding implementation is also proposed. The additional complexity required by the method is leveraged by exploiting parallelism and SIMD instruction sets. From results, it appears that optimization in filtering process enables a significant speedup that makes real-time decoding possible.

Eventually, an end-to-end demonstrator is proposed to show the compatibility of the proposed method with classical broadcast infrastructure. Based on a DVB compliant MPEG-TS stream, the service is modulated using DVB-T2 and delivered to two receivers. The first one is a legacy DVB-T2/HEVC compatible set-top-box that only decodes the HEVC base-layer and renders backward-compatible signal. The second one is a compatible set-top-box that is able to decode the full bitstream and renders the immersive experience.

The proposed solution fulfills broadcast requirements regarding backward compatibility and bandwidth efficiency and is a practical candidate for DTT deployment of 360° video services.

ACKNOWLEDGMENT

This work has been achieved within the Convergence TV collaborative research project.

REFERENCES

[1] 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.

[2] J. Boyce, "MPEG-I future directions," in *Proc. SMPTE*, Oct. 2018, pp. 1–12.

[3] "CM1706: DVB study mission on virtual reality," CM-VR, Digit. Video Broadcast, Geneva, Switzerland, Rep. CM-1706, Oct. 2018.

[4] Technical Specification Group Services and System Aspects, "Virtual reality (VR) media services over 3GPP," 3rd Gener. Partnership Project, Sophia Antipolis, France, Rep. TR 26.918, Sep. 2017.

[5] *VRIF Guidelines*, VR Ind. Forum, Fremont, CA, USA, 2018. [Online]. Available: http://www.vr-if.org/guidelines/

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

[7] 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.

[8] Y. Ye, E. Alshina, and J. Boyce, "JVET-G1003: Algorithm description of projection format conversion and video quality metrics in 360lib version 4," Joint Video Exploration Team, Turin, Italy, Rep. JVET-G1003, Jul. 2017.

[9] Y. He, Y. Ye, P. Hanhart, and X. Xiu, "Geometry padding for motion compensated prediction in 360 video coding," in *Proc. Data Compression Conf. (DCC)*, Apr. 2017, p. 443.

[10] L. Li, Z. Li, X. Ma, H. Yang, and H. Li, "Co-projection-plane based 3-D padding for polyhedron projection for 360-degree video," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 55–60.

[11] R. G. Youvalari, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 525–528.

[12] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, "360 degrees video coding using region adaptive smoothing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Quebec City, QC, Canada, Sep. 2015, pp. 750–754.

[13] Y. Sunn and L. Yu, "JVET-G1006: EE3 adaptive QP for 360 video," Joint Video Exploration Team, Turin, Italy, Rep. JVET-G1006, Jul. 2017.

[14] Y. Li, J. Xu, and Z. Chen, "Spherical domain rate-distortion optimization for 360-degree video coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 709–714.

[15] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," in *Proc. IEEE Virtual Reality (VR)*, Los Angeles, CA, USA, Mar. 2017, pp. 423–424.

[16] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming based on MPEG-DASH SRD," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, San Jose, CA, USA, Dec. 2016, pp. 407–408.

[17] Y. Hu, S. Xie, Y. Xu, and J. Sun, "Dynamic VR live streaming over MMT," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2017, pp. 1–4.

[18] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Standard-compliant multiview video coding and streaming for virtual reality applications," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, San Jose, CA, USA, Dec. 2016, pp. 295–300.

[19] G. Tech, Y. Chen, K. Müller, J. R. Ohm, A. Vetro, and Y. K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[20] *Generic Coding of Moving Pictures and Associated Audio Information—Part 1: Systems*, ISO/IEC Standard 13818-1, 2018.

[21] K. Park, Y. Lim, and D. Y. Suh, "Delivery of ATSC 3.0 services with MPEG media transport standard considering redistribution in MPEG-2 TS format," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 338–351, Mar. 2016.

[22] J. L. Feuvre, N. Viet-Than-Trung, W. Hamidouche, M. Patrick, and D. Pascal, "A test bed for hybrid broadcast broadband services," in *Proc. Media Synchronization Workshop*, Jun. 2015, pp. 1–4.

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

[24] Y. Ye and V. Seregin, "JCTVC-W1013: Reference software for scalable HEVC (SHVC) extensions draft 4," Joint Collaborative Team Video Coding, San Diego, CA, USA, Rep. JCTVC-W1013, Feb. 2016.

[25] W. Hamidouche, M. Raulet, and O. Déforges, "4K real-time and parallel software video decoder for multilayer HEVC extensions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 169–180, Jan. 2016.

[26] P. Cabarat, W. Hamidouche, and O. Déforges, "Real-time and parallel SHVC hybrid codec AVC to HEVC decoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 3046–3050.

[27] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "JVET-H1030: JVET common test conditions and evaluation procedures for 360° video," Joint Video Explor. Team, Macau, China, Rep. JVET-H1030, Oct. 2017.

[28] G. Bjøntegaard, "VCEG-M33: calculation of average PSNR differences between RD-curves," Video Coding Exp. Group, Austin, TX, USA, Rep. VCEG-M33, Apr. 2001.

[29] J. Le Feuvre, C. Concolato, and J.-C. Moissinac, "GPAC: Open source multimedia framework," in *Proc. 15th ACM Int. Conf. Multimedia (MM)*, 2007, pp. 1009–1012. [Online]. Available: http://doi.acm.org/10.1145/1291233.1291452

[30] *DTU-215 Leaflet*, Dektec Digit. Video BV, Hilversum, The Netherlands, Sep. 2011.

[31] *Datasheet: REFEREE II*, Enensys Technol., Rennes, France, 2018.

[32] *Product Brief: Intel NUC Kit NUC6i7KYK*, Intel Corporat., Santa Clara, CA, USA, 2016.

[33] *Dreambox dm920*, Dream Property GmbH, Lünen, Germany, 2016. [Online]. Available: https://dreambox.de/en/dm920-ultrahd