Panorama view with spatio-temporal occlusion compensation for 3D video coding

Muhammad Shahid Farid, *Student Member, IEEE,* Maurizio Lucenteforte, *Member, IEEE,* and Marco Grangetto, *Senior Member, IEEE*

***Abstract*—The future of novel 3D display technologies largely depends on the design of efficient techniques for 3D video representation and coding. Recently, multiple view plus depth video formats have attracted many research efforts since they enable intermediate view estimation and permit to efficiently represent and compress 3D video sequences. In this paper we present Spatio-Temporal Occlusion compensation with Panorama view (STOP), a novel 3D video coding technique based on the creation of a panorama view and occlusion coding in terms of spatio-temporal offsets. The panorama picture represents most of the visual information acquired from multiple views using a single virtual view, characterized by a larger field of view. Encoding the panorama video with state of the art HECV and representing occlusions with simple spatio-temporal ancillary information STOP achieves high compression ratio and good visual quality with competitive results with respect to competing techniques. Moreover, STOP enables free view point 3D TV applications whilst allowing legacy display to get a bi-dimensional service by using a standard video codec and simple cropping operations.**

***Index Terms*—3D-TV, 3D video coding, HEVC, AVC, Depth image based rendering**

I. Introduction

HUMAN visual perception of depth is achieved in 3D display devices by rendering two or more scene views simultaneously. There has been much innovation in 3D displays in the last few years achieving promising levels of visual immersivity. Popular stereoscopic displays in the market show two images of the same scene from slightly different viewing angles that are discriminated through active or passive glasses. The more recent autostereoscopic technology removes the hurdle of glasses by exploiting the principles of parallax barrier or lenticular arrays that allow the user to discriminate a pair of stereoscopic pictures depending on the relative position between the display and the viewer. These autostereoscopic displays use several views, usually up to 50, to generate the perception of depth and to enable the so called free viewpoint TV, where the viewer can roam around these views changing his view-point.

The technological evolution of 3D displays clearly needs to be accompanied by the design of efficient 3D video representations and coding formats able to achieve high level of compression, given the huge amount of data that need to be managed. This is of paramount importance in the context of broadcasting, where the precious radio spectrum must be used as parsimoniously as possible. Moreover, the broadcasting scenario involves other strict constraints represented by presence of legacy devices and backward compatibility issues [1].

Recently, many pilot commercial 3D television (3DTV) services have been launched, following the requirements of the first phase of the 3DTV specified by the DVB [2]. Current technical solutions are based on the so called frame compatible formats [3] that are used to feed stereoscopic display sacrificing the resolution of the individual views.

Up to date the most successful 3D video coding format is represented by the multiview extension (MVC) of the widespread AVC standard [4]; MVC exploits the spatial redundancy among several views adopting the usual block based compensation mechanism and it is backward compatible with any AVC decoder. One of the shortcomings of MVC is the fact that the compression bitrate increases linearly with the number of views [5], making it unfit in the context of free-viewpoint TV. In response to the new requirements novel 3D video formats have emerged, e.g. Depth Enhanced Stereo (DES) [6], Layered Depth Video [7], [8] and Multiview video plus depth (MVD) [9], where pictures are coupled with a per pixel depth map that represents the distance of every pixel from the imaging plane. Such formats enable depth image-based rendering (DIBR) for generation of intermediate views and are catalysing many research efforts in the area of 3D video compression [10].

In this paper we proposed a Spatio-Temporal Occlusion compensation with Panorama view (STOP) technique to efficiently represent and encode MVD formats. The major contributions of the paper are in the following.

- A panorama picture of the scene is created as a mean to capture most of the 3D redundancy of the scene. To this end multiple views are fused in a single panorama picture characterized by a larger viewing angle embracing all the available multiple views.
- DIBR is exploited to extract any desired intermediate view starting from a single panorama view plus depth picture.
- The panorama video undergoes standard video encoding allowing legacy 2D devices to obtain the corresponding 2D pictures by standard decoding and simple cropping.
- The image quality guaranteed by STOP on the decoder side is improved by adding an ancillary compressed bit-stream representing spatio-temporal offsets that are

The authors are with Università degli Studi di Torino, Dipartimento di Informatica, Torino, Italy. M.S. Farid is also with University of the Punjab, Lahore, Pakistan.
E-mail: {first.last}@unito.it

exploited to recover dis-occluded details after DIBR. An accurate and automatic mechanism for the selection of the dis-occluded areas that will be visually relevant on the decoder side is proposed, achieving excellent rate/quality trade-offs.

The proposed STOP 3D video representation approach jointly achieves many goals that are of paramount importance in the 3D video broadcasting framework, namely:

- *2D compatibility*: the proposed STOP codec supports stereoscopy and autostereoscopy while allowing standard 2D service for existing legacy displays. In particular, STOP exploits simple pre/post processing and can be used along with any existing standard video codec. By signaling the coordinates of the 2D image to the decoder, legacy devices can enjoy 2D service by simply cropping the panorama picture. As an example, the cropping area can be signaled in the compressed bitstream by extending the syntax of supplemental enhancement information messages defined in AVC/HEVC [10].
- *Efficient compression*: by using HEVC STOP is capable to encode three views plus depths at a bitrate that is only 1.5 times that of a standard 2D video.
- *Low complexity*: The computational cost of STOP approaches that of single view video plus depth coding. Our experiments show that the panorama picture is usually less than 10% larger than the original picture; this characteristic significantly reduces the coding time compared to simulcast coding and competing MVD coding approaches.
- *Promising visual quality*: High visual quality of coded and synthesized videos is guaranteed with values of SSIM that are almost the same as those guaranteed by simulcast of multiple views plus depths.
- *FTV application*: Free-view point TV applications can be easily implemented on the receiver side. To achieve stereoscopy one additional view can be obtained by projecting the central view to the left (or right) viewpoint through DIBR plus occlusion filling using spatio-temporal offsets. For large field of view autostereoscopy, additional intermediate views can be estimated using DIBR.

The rest of the paper is organized as follows. In Sect. II related work is briefly discussed. In Sect. III the construction of the proposed panorama view is presented, followed by Sect. IV that describes the overall STOP technique based on panorama view representation of the scene and spatio-temporal occlusion/dis-occlusion estimation and coding. Sect. V shows the experimental results obtained on a standard dataset to evaluate the performance of the proposed coding tool. In Sect. VI our conclusions are drawn.

II. Related work

In this section we briefly recall some related works that help in contextualizing our contributions. In particular, we mention current research efforts in the fields of 3D video coding and DIBR algorithms related to our proposal.

As already mentioned in Sect. I pilot 3D TV services are based on simple frame compatible solution: the most well-known are the top-bottom and side by side approaches that simply place two views of a stereoscopic video into a single picture by stacking them vertically or horizontally [3]. Another frame compatible approach [11] reduces the overall pixel resolution in each dimension but provides some advantages such as potential compatibility with 2D display. The frame compatible formats allow one to use a standard high definition video to carry a stereoscopic pair by clearly sacrificing the spatial resolution of the individual views.

Various video coding formats have been proposed to efficiently encode multiple views required for 3DTV. The MVC extension of AVC must be recalled as the first efficient mean to encode multiple views. MVC shows much better compression ratio as compared to its standard ancestors. However, there are a couple of limitations: MVC bitrate increases linearly with the number of views [5] and produces blur effect when it is used to encode depth [12].

Recently, video plus depth formats have attracted much attention. Using the depth of the scene one can generate intermediate viewpoints using DIBR techniques [13]–[16]. Consequently, many works have been devoted to the compression of depth and its exploitation for view prediction [17]–[19]. Since 2011 MPEG has issued a call for proposal on 3D video coding technology [20] and in 2012 an ISO/ITU collaborative effort [21] has been started to cope with novel evolutions of the 3D video. The Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) is pursuing the standardization of 3D extensions of both AVC (3D-AVC) [22] and the novel High Efficiency Video Coding (3D-HEVC) [10]. The first achievement is represented by Multiview High Efficiency Video Coding (MV-HEVC), where the same approach used by MVC on top of AVC has been adopted to encode multiple views. Inter-view prediction has been implemented by including the inter-view pictures in the reference lists used for prediction. The average compression gain of MV-HEVC over HEVC Simulcast and MVC is reported upto 30% and 50%, respectively. The design of novel and efficient 3D coding tool exploiting MVD formats is the main focus of the current efforts in JCT-3V [23]–[25]. 3D-HEVC is expected to exploit new tools and coding mode capable to improve significantly the compression performance. The base view and corresponding depth map are encoded by unmodified HEVC; the dependent views and depths exploit additional tools e.g. disparity compensated prediction, inter-view prediction of motion parameters, inter-view prediction of residual data, etc. The 3D-HEVC is emerging as the state of the art 3D coding tool and provides 50% and 20% bit rate savings over HEVC Simulcast and MV-HEVC respectively [23].

In [26] a new multiview video representation is introduced, where foreground objects are extracted from the scene and represented as discrete surfaces using a Monte-Carlo method. The obtained surfaces can be easily reconstructed and warped to any virtual camera position. One limitation of [26] is the intrinsic assumption of a static background. Finally, a 3D coding technique sharing some idea with STOP is the one proposed in [27]; this latter has been used in our experiments as a reference.

In STOP we propose a spatio-temporal approach to conceal dis-occlusions produced by DIBR. Many techniques have been proposed in the past to handle the dis-occlusions.

The most popular methods to recover occlusion are based on *inpainting* techniques. These latter can be categorized into four major classes: i) texture-based inpainting, ii) patch-based inpainting, iii) depth guided inpainting, and iv) spatio-temporal methods. In first class methods, the missing region is filled using the color and texture information of the surrounding pixels [28]–[31]. In patch-based inpainting the missing region is filled with similar patches that can be generally found in the neighboring area in the same image. A patch around the hole is selected and searched for in the rest of the image; then the best match is used to recover the occlusion [32]–[34]. In depth based inpainting either the depth map is inpainted first and then used to find (by 3D warping) the pixels that can cover the holes or the recovered depth map is used in conjunction with texture-based or patch-based inpainting [35]–[38].

Finally, other techniques in literature aim at filling the DIBR holes by maintaining a background sprite. If the background is static or changing slowly it can be estimated and exploited to fill the disocclusions [39]–[43]. Few other techniques that used spatial and temporal information to fill the large holes may be found in [44]–[46].

III. Panorama Representation

Let us consider a horizontal set-up where the scene is captured by n cameras with co-planar and parallel camera vectors and using the same focal length f. We denote as V_i, $i = 1, \ldots, n$ the i-th view image (moving from left to right), where $V_i(u, v)$ represents a pixel with row $u = 1, \ldots, H$ and column $v = 1, \ldots, W$, being $W \times H$ the image resolution. In the following we always assume horizontal camera arrangement and rectified images. We further consider the availability, thanks to stereo matching algorithms or physical measurements, of dense depth maps. We denote as D_i the depth map associated to the view V_i, with $D_i(u, v)$ the depth of the pixel (u, v) of the i-th view.

Fig. 1 shows an example of multiview camera setup comprising 3 cameras with horizontal shifts. The distance between the viewpoints b_i of the different cameras, termed as baseline, is usually limited to guarantee a smooth 3D experience, e.g. when using an autostereoscopic display. As graphically represented in Fig. 1, this yields a high overlap among the field of views of the different cameras, that in turn can be recognized in terms of redundancy in the collected images.

At very high level one may imagine to summarize all the collected information using a single virtual camera with focal point in an intermediate reference position and a field of view larger than the ones of the single cameras. Unfortunately, because of occlusions and dis-occlusions, e.g. background pixels that appear behind foreground objects, it is not possible to compact all the information acquired by several cameras into a single image plane of the virtual intermediate camera. Nonetheless, in practical settings with limited baseline, the creation of the virtual intermediate view, that we termed *panorama view*, represents a viable approach for representing a set of multiview images, compactly. Finally, using DIBR techniques one shall be able to estimate the required intermediate views.

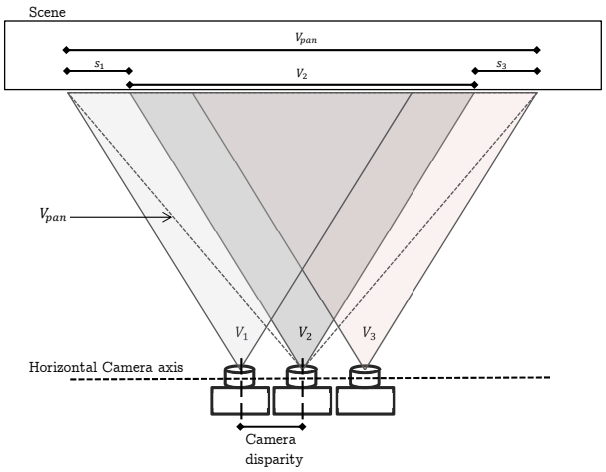


Fig. 1. A multiview camera setup with horizontal arrangement.

A. Background on Depth Based Image Rendering

DIBR process consists in reprojecting the pixels of a set of acquired images (typically 2) into the image plane of an intermediate virtual view, exploiting the depth information and the knowledge of the camera intrinsic and extrinsic matrices.

Let V_s be the source view with corresponding depth D_s and viewpoint b_s. Let b_t the position of a target view V_t with disparity $b = b_t - b_s$ from the source view. Assuming the horizontal camera setup, V_t can be obtained applying horizontal shifts[1] to the pixels of V_s. Given $V_s(u, v)$ this latter can be mapped to the corresponding $V_t(u, v + \delta_{s,t}(u, v))$ with

$$\delta_{s,t}(u, v) = \frac{bf}{D_s(u, v)} \tag{1}$$

being the shift required to map pixel (u, v) in the coordinate system of the image plane of the target camera. Clearly, the shift depends on the depth of each pixel and generally requires sub-pixel precision. DIBR techniques compete in designing the best solutions to overcome a series of issues that arise when shifting pixels, e.g. recovering from occlusion and dis-occlusion, high quality virtual image re-sampling, robustness to depth estimation errors, etc.

Finally, we recall that in practical cases the depth is mapped onto its inverse $1/D_s$ represented over 255 levels, usually termed as quantized disparity. Denoting as d_s the 255 levels image used to represent the quantized disparity, one can compute the depth map as:

$$D_s(u, v) = \frac{1}{\frac{d_s(u,v)}{255}\left(\frac{1}{Z_n} - \frac{1}{Z_f}\right) + \frac{1}{Z_f}} \tag{2}$$

where Z_n and Z_f are the minimum and maximum depth, respectively.

B. Panorama View Generation

Without loss of generality we assume that the number of camera n is odd. Let V_m be the intermediate view,

[1] In the general case with arbitrary camera positions both horizontal and vertical shifts must be taken into account determining the so called image warping process

$V_1, \cdots, V_{m-1}$ the views on its left side and $V_{m+1}, \cdots V_n$ the views on the right side. As discussed earlier, the camera views share a large portion of their field of view due to their limited disparity. When moving from V_m towards V_{m-1} a new part of the scene shows up on the left side of V_{m-1} because of camera translation, whereas another part disappears to the right. Moreover, some occlusions and dis-occlusions will occur, especially around the edges of foreground objects. For the sake of the panorama view construction we make the assumption that no occlusions and dis-occlusions are to be taken into account. We will remove this limiting assumption later in Sect. IV.

Under this simplifying assumption any view V_i can be obtained from the intermediate V_m with the exception of a band of pixels (appearing on the right or on the left side), that in the following we refer to as the slit s_i. It is worth noting that the slits get larger while moving farther away from the intermediate position. Fig. 1 shows a multi views setup with 3 cameras where one can select the central view V_2 as a reference. As an example, it can be noted that V_3 shares a large portion of image plane except for a small slit s_3 on the right side. It is worth pointing out that other views on the right side, e.g. $V_4, V_5, \ldots$ would yield larger slits $s_4, s_5, \ldots$, including s_3. In general, we can state that:

$$\begin{array}{ll} s_i \subseteq s_{i-1} & \text{if } 1 \leqslant i < m; \\ s_{i+1} \subseteq s_i & \text{if } m < i \leqslant n; . \end{array}$$

Hence, a panorama view V_{pan} with size $W' \times H$ can be generated concatenating s_1, V_m and s_n so as the create a single picture collecting most of the information in the scene. The same approach can be followed to create a single panoramic depth D_{pan} that carries the information required to synthesize any desired intermediate view.

The panorama width $W' > W$ clearly depends on the size of the left and right slits, that in turns depends on the maximum shift that a pixel can undergo when moving from the leftmost/rightmost view to the reference view. The maximum shift varies with the scene depth and possibly changes as a function of time within the video sequence. Nonetheless, we can easily assume that the minimum depth z_n of the considered scene is known; since the maximum shift is produced by the scene point that is closest to camera one can upperbound all shifts by

$$\Delta = \left\lceil \frac{(b_m - b_1)f}{z_n} \right\rceil \tag{3}$$

As a consequence, we set the panorama width to be $W' = W + 2\Delta$, i.e. the panorama view contains the pixels $V_{pan}(u, v)$ with $u = 1, \ldots, H$, and $v = 1, \ldots, W + 2\Delta$.

Two alternative strategies can be used to create V_{pan} and D_{pan} images, respectively. In the following we focus on the construction of the panorama view; the same process can be followed to build the corresponding depth map, as well.

1) Panorama construction via Inward Projection: In inward projection approach the left view V_1 and the right view V_n are warped to the central view V_m; the pixels of V_1 and V_n that are warped outside image plane of the central view are those that forms the slits s_1 and s_n.

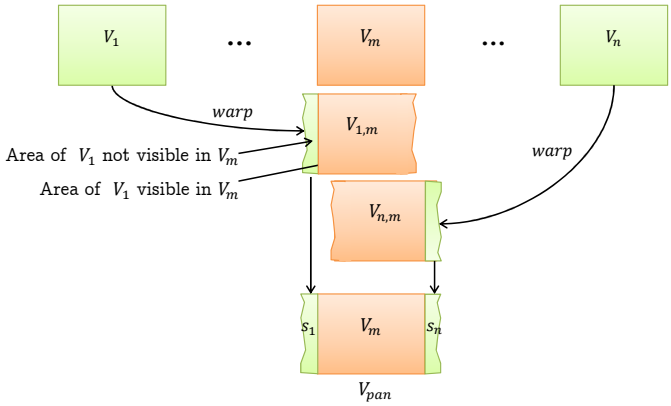


Fig. 2. Panorama construction process using inward projection approach.

The panorama view can be constructed as follows:

$$V_{pan}(u, v) = \begin{cases} V_{1,m}(u, v - \Delta) & \text{if } v \leqslant \Delta \\ V_m(u, v - \Delta) & \text{if } \Delta < v \leqslant W + \Delta \\ V_{n,m}(u, v - \Delta) & \text{if } v > W + \Delta \end{cases} \tag{4}$$

being $V_{i,m}(u, v)$, $i = 1, n$ the pixel synthesized by warping V_i according to the point of view of V_m. In particular, the panorama will collect the warped pixels $V_{1,m}(u, v)$ with $v < 1$, i.e. falling outside the field of view of V_m and those of $V_{n,m}(u, v)$ with $v > W$. For this reason in previous equation we assume that $V_{1,m}(u, v)$ is defined also for column coordinates that are 0 or negative and $V_{n,m}(u, v)$ takes values also when $v > W$. Clearly, during the warping process more than one pixel of V_1 (or V_n) may be shifted to the same coordinate (u, v); in such case only the pixel closest to the camera is copied in the panorama view. Moreover, shifts are not guaranteed to map on integer coordinates and proper interpolation is required as in common DIBR algorithms. This process is schematically shown in Fig. 2.

2) Panorama construction via Outward Projection: The slits' pixels can be identified with a process that is the dual of the previous one; indeed, one can warp the reference view V_m according to the viewpoint of V_1 and record the pixel coordinates of V_1 that cannot be covered shifting pixels of V_m. Under our assumption of absence of occlusions/disocclusions this process permits to identify the pixel in slit s_1. The same result is obtained for s_n by warping V_m towards V_n.

Let $H_i(u, v)$ be a binary map representing the coordinates of view V_i that cannot be obtained warping V_m. In this case we can form a panorama picture by setting:

$$V_{pan}(u, v) = \begin{cases} V_1(u, v') & \text{if } v \leqslant \Delta, H_1(u, v') = 1, \\ & v' = v - \Delta + \mathrm{rt}(H_1(u, \cdot)) \\ V_m(u, v - \Delta) & \text{if } \Delta < v \leqslant W + \Delta \\ V_n(u, v') & \text{if } v > W + \Delta, H_n(u, v') = 1 \\ & v' = v - \Delta - \mathrm{lt}(H_n(u, \cdot)) \end{cases} \tag{5}$$

where $\mathrm{rt}(H_1(u, \cdot))$ returns the rightmost column of the u-th row where H_1 is equal to 1 and $\mathrm{lt}(H_n(u, \cdot))$ returns the leftmost column of the u-th row where H_n is equal to 1. According to previous equation the pixels $V_1(u, v')$ that cannot be warped from V_m to V_1 are copied on the left side of

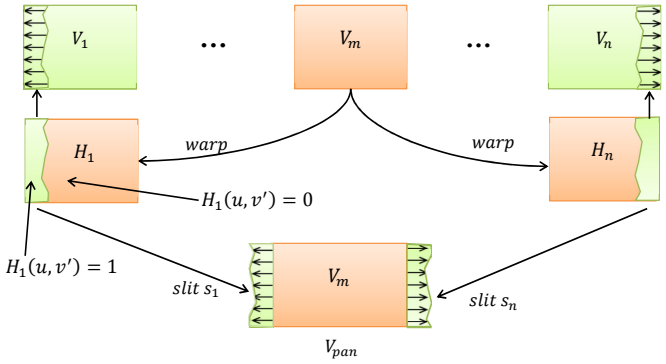


Fig. 3. Panorama construction process using outward projection approach.

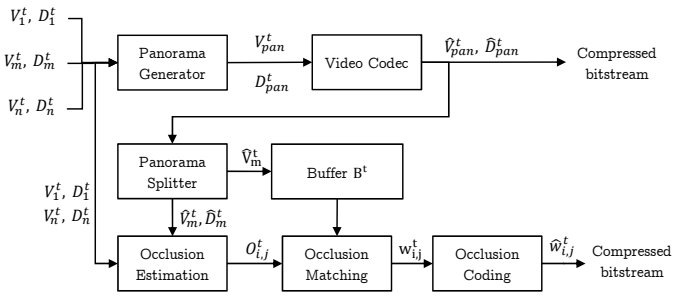


Fig. 4. STOP Encoder

V_{pan} row by row. Analogously, pixels $V_n(u,v')$ that cannot be warped from V_m to V_n are appended to the right side of the panorama row by row.

The process is graphically show in Fig. 3. In the following we termed this panorama construction process as the outward projection approach. It can be noted that in this case the right edge of s_1 is not guaranteed to be straight; for simplicity the panorama image is constructed aligning each row of s_1 on the right so as to place s_1 and V_m side by side. The same process is adopted for s_n aligning each row on the left.

3) Comparison of inward projection and outward projection approach: Both inward projection and outward projection approaches can be used to build the panorama view. The two approaches have the same computational complexity as both of them require to carry out two image re-projections, i.e. from V_1 and V_n to V_m in one case and from V_m to V_1 and V_n in the other case.

It is important to recall that using the inward projection method the slits pixels are taken from the synthesized views $V_{1,m}$ and $V_{n,m}$, whereas in the outward projection approach those pixels are copied from the original views V_1 and V_m. On the receiver side, the panoramic view V_{pan} can be re-projected on any given intermediate view. As an example, the pixels of V_{pan} can be warped according to the projection vector of V_1 to estimate the leftmost view. Since view synthesis requires resampling and interpolation and may be affected by depth quantization or depth estimation errors, image warping will produce a virtual view $V_{pan,1}$ that is an estimate of V_1. Clearly, our goal is to get the best possible approximation of V_1. In this light, the outward projection approach exhibits a slight advantage in terms of the image quality obtained by warping the slits of V_{pan}. Indeed, in the outward projection solution the slits pixels are obtained from the original views and are warped only once to get a given virtual view. On the contrary, in the inward projection approach slit pixels are warped twice, first for the construction of the panorama and then for rendering virtual views. Since warping potentially impairs the image quality it is clearly better to limit its usage. Nonetheless, this observation applies only to the slits pixels (a very limited area of the panorama in presence of limited camera baseline) and does not represents a major issue on the whole image quality. In the following of the paper we always assume to use the outward projection approach.

IV. The STOP Video Coding Method

In the following a method is proposed to represent (at the encoder side) and recover (at the decoder side) the occlusion and dis-occlusion areas[2] that have been neglected during the creation of the panoramic view. This approach will allow us to design STOP, based on standard coding of the panorama view and depth, followed by a novel mechanism for handling occlusions.

The proposed solution aims at representing every occluded patch by encoding its spatio-temporal offset pointing to the best approximation of the missing area that can be found in pictures that are already available on both encoder and decoder. This goal can be achieved by exploiting the fact that occluded pixels of a given frame are likely to be visible in the subsequent or preceding frames, thanks to camera and/or objects movements; alternatively, missing areas can be concealed by means of similar areas that can be found in the same frame.

A. Panorama Encoder and Decoder

Before presenting the details of the procedure used to cope with occlusions we need to provide a high level description of the whole encoding and decoding process.

In Fig. 4 the block diagram of the encoder is shown. The first stage consists in the creation of the panorama view and depth, taking the original views and depths as input. Without loss of generality, in the following, a camera setup with 3 views and 3 depth will be considered as it is assumed in Fig. 4.

Then, the panorama view and depth are encoded with any existing technique, e.g. AVC or HEVC. In the following, we need to enrich previous notation by adding the temporal index of a frame of a video sequence, e.g. V_{pan}^t refers to the t-th panorama view of the video sequence. The output of the first stage are the corresponding compressed bitstream and the decoded panorama view and depth, $\hat{V}_{pan}^t$ and $\hat{D}_{pan}^t$, $t = 1, 2, \ldots$. The reconstructed views and depths can be used to compute on the encoder side the occlusions that will be experienced by the decoder when generating a given view. The reconstructed central view $\hat{V}_m^t$ and corresponding depth $\hat{D}_m^t$ are trivially extracted from $\hat{V}_{pan}^t$ and $\hat{D}_{pan}^t$ removing the slits as described in Sect. III. In the proposed encoder DIBR is used to compute the occlusions that show up when

[2]In the following, for conciseness, we use the term occlusion to identify both occlusion and dis-occlusion

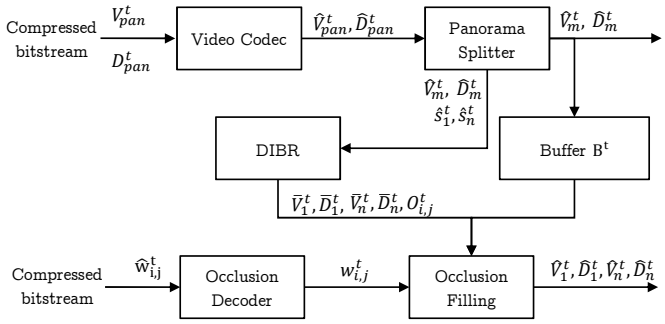


Fig. 5. STOP Decoder

estimating the leftmost and rightmost views, respectively. The best approximation of each occlusion is then searched in the decoded frames $\widehat{V}^t_m$ and the corresponding spatio-temporal offset is encoded and sent as ancillary data as detailed in Sect. IV-B.

On the receiver side, the various components of the compressed bitstream are parsed and decoded. Backward compatibility towards legacy 2D devices can be easily obtained by letting them decode the standard panorama video that can be simply displayed after cropping the central view. On the contrary 3D receivers use the panorama view and depth to estimate $\widehat{V}^t_1, \widehat{D}^t_1$ using DIBR. Thanks to the presence of the slits the area of the view appearing on the left hand side is obtained. The remaining occlusions are filled using the spatio-temporal offset information provided by the encoder. Those areas are simply copied from frames that have already been decoded; indeed, we observed that in the proposed spatio-temporal hole filling no post-processing such as boundary matting is useful because the patches are very small in size and they usually belong to nearby frames. Therefore additional processing would increase the complexity of the technique without carrying significant quality improvements. Finally, the rightmost view and depth can be estimated with the same process. Given $\widehat{V}^t_1, \widehat{V}^t_m$ and $\widehat{V}^t_n$ (and corresponding depth) DIBR can be used to estimate any desired intermediate view. Fig. 5 shows the block diagram of proposed decoder.

B. Occlusion matching and coding

As already anticipated, the occlusions that are not taken into account during the construction of the panorama image are coded separately, by sending a reference to a patch of a picture (that is already available at the decoder) that can be used to approximate the missing area. The occlusion coding process can be divided into 3 steps:

1) occlusion patches selection and segmentation
2) occlusion offset estimation
3) occlusion offset coding

On the decoder side, the same occlusion selection and segmentation algorithm used by the encoder is run to identify the same set of parches that can be recovered copying pixels from the areas of the picture referenced by the decoded occlusion offsets.

1) Occlusion selection: The encoder aims at selecting and encoding only the occlusions that potentially have a sensible impact on the rendered visual quality. The most significant occlusions are clearly obtained when using $\widehat{V}^t_{pan}, \widehat{D}^t_{pan}$ to render $\widehat{V}^t_1$ and $\widehat{V}^t_n$.

By means of a standard DIBR algorithm the encoder is able to identify the area of the target images $\widehat{V}^t_i$, $i = 1, n$ that cannot be recovered with pixels of $\widehat{V}^t_{pan}$ (usually referred to as holes). The occlusions of the i-th view at time t are organized into a list of connected components, e.g. by using the flood-fill algorithm to label all the pixels of the binary image representing the holes left by DIBR. Let $\mathrm{O}^t_{i,j}$ refers the j-th connected component, i.e. patch of occluded pixels, of the i-th rendered view at time t of the video sequence and $|\mathrm{O}^t_{i,j}|$ its size measured in number of pixels.

To limit the coding overhead the occlusions with size $|\mathrm{O}^t_{i,j}| \leqslant \tau_S$ are simply discarded. On the decoder side such occlusions can be recovered by standard inpainting [31], [47].

It is worth pointing out that, because of the horizontal camera setup, some very tiny vertical cracks will show up. These holes are generally only few pixels wide and can be recovered efficiently by inpainting. As a consequence, we add a criterion to drop occlusion patches that have a very skewed aspect ratio. In particular, we define the following ratio

$$r(\mathrm{O}^t_{i,j}) = \frac{|\mathrm{O}^t_{i,j}|}{|\mathrm{O}_{i,j}|^v} \tag{6}$$

between the patch size and its vertical length $|\mathrm{O}_{i,j}|^v$; this latter can be simply computed as $|\mathrm{O}_{i,j}|^v = \max_{row} \mathrm{O}^t_{i,j} - \min_{row} \mathrm{O}^t_{i,j} + 1$, where $\max_{row}$ and $\min_{row}$ evaluates the maximum and minimum row indexes in a patch, respectively. Then, patches with $r(\mathrm{O}^t_{i,j}) \leqslant \tau_R$ are discarded for further processing.

The size of the occlusions generated by an object depends on its size and depth. If a large object is close to the camera, then an occlusion of large size, especially in the vertical direction, is to be expected. To easy the following task of occlusion matching it is preferred to segment large occlusions into smaller patches. Indeed, smaller patches are likely to represent more homogeneous areas that in turn are matched more easily in the surrounding images (both temporally and spatially). Moreover small patches are more robust to distortions introduced by camera and/or object movements. As a consequence, if $|\mathrm{O}_{i,j}|^v > \tau_L$, the corresponding occlusion is segmented into smaller chunks (with height less or equal to τ_L) cutting it horizontally.

To avoid cluttering the notation we keep using the symbol $\mathrm{O}^t_{i,j}$ to identify the occlusions surviving the pruning process described above.

2) Occlusion matching: The next step consists in the search for the best match of every occlusion $\mathrm{O}^t_{i,j}$ given the views that have already been decoded at the receiver. This task is very similar to motion estimation techniques adopted in standard video codecs, based on block matching. Nonetheless, in our encoder we avoid using blocks and we aim at estimating the best match of every occlusion patch $\mathrm{O}^t_{i,j}$ by keeping its original shape. As for motion estimation we use the Sum of Absolute Differences (SAD) as a computationally

effective metric of similarity. Clearly, our method can be easily extended to the case of more complex metrics that take into account other image features such edges, orientations and color [48]–[51]. It is worth pointing out that the occlusions can be matched also in the depth domain; this possibility is not exploited by our encoder from one hand to limit the computational cost, and from the other hand because depth images offer less details and are usually error prone due to disparity estimation errors.

The occlusion matching consists in the following minimization:

$$\mathbf{w}_{i,j}^t = (x,y,k) = \underset{x,y,k:B_k \in \mathcal{B}^t}{\operatorname{argmin}} \sum_{u,v \in \mathrm{O}_{i,j}^t} |V_i^t(u,v) - B_k(u-x, v-y)| \tag{7}$$

where $\mathcal{B}^t$ is a buffer of decoded images selected for the search at time t, (x,y) represents the spatial offset and k references a particular decoded image $B_k \in \mathcal{B}^t$. The vector $\mathbf{w}_{i,j}^t = (x,y,k)$ represents the spatio-temporal offset to the best occlusion match and will be encoded as described in the following section.

Clearly, the search range must be optimized so as to trade-off accuracy of the matching and the computational cost. In our implementation we include in $\mathcal{B}^t$ the central view $\hat{V}_m^j$ of the decoded panorama pictures, with $j = t-\theta, \ldots, t+\theta$, i.e. we consider a symmetrical time window around the target view at time t. The larger θ, the higher the chance to cover an occlusion exploiting camera and/or object motion. Furthermore θ must be selected taking into account the video encoder settings, e.g. the GOP structure, and the delay constraint of the application.

Clearly, the above selected set of reference pictures is not enough to recover occlusions generated by stationary objects in a sequence without camera motion (in the following we terms them *static occlusions*). To overcome this issue we can include in $\mathcal{B}^t$ some virtual views where all occlusions are inpainted. These virtual views can be generated by both encoder and decoder warping the decoded panorama view and depth. As an alternative, a set of key frames of the leftmost and rightmost views can be encoded[3], e.g. with the same intra period used to encode V_{pan}, D_{pan} or just when a scene-cut [52]–[54] is detected. The second option incurs some bitrate overhead but generally yields higher image quality. In general, we let $\mathcal{B}^t$ include ω supplementary reference pictures that serve to match static occlusions. In conclusion, $\mathcal{B}^t$ contains $2\theta+1$ temporal references and ω spatial references.

As far as the search in the spatial domain (x,y) one can optimize that range depending on the time offset of the reference frame with respect to t and the depth of the considered occlusion. Clearly, if two frames are far apart in time it is likely that a given occlusion is recovered using larger spatial offset because of the effect of motion. Moreover, a larger search space must be used for occlusion patches that are closer to the camera. As a consequence the spatial search has been constrained to $-\chi \leqslant x,y \leqslant \chi$, where $\chi = \psi \cdot \omega$, being ψ the temporal offset between the current and the reference image, and ω the average disparity of the occlusion under consideration computed according to (1) as follows:

$$\omega = \frac{1}{|\mathrm{O}_{i,j}^t|} \sum_{u,v \in \mathrm{O}_{i,j}^t} |\delta_{m,i}(u,v)|$$

3) Occlusion coding: The last step is the lossless encoding of the offset vectors $\mathbf{w}_{i,j}^t$. This can be done embedding ancillary data in the bitstream corresponding to every coded panorama frame V_{pan}^t, e.g. by exploiting the supplemental enhancement information (SEI) message standardized in AVC/H.264 or other standard means for providing an ancillary compressed bitstream. Since we assume that occlusion patches $\mathrm{O}_{i,j}^t$ are selected and ordered with the same rules by both the encoder and the decoder the pair (i,j) can be recovered implicitly by the decoder and does not require to be coded. Therefore, occlusions can be encoded as a list of signed integer offsets (x,y,k) accompanying every encoded frame V_{pan}^t.

Clearly, the video spatial and temporal correlation can be exploited for entropy coding of the offset, e.g. by using exp-Golomb followed by context based arithmetic coding. Nonetheless, since the number of occlusion patches that need to be coded can be very limited, entropy coding can be omitted without a significant impact on the overall bitrate. As a consequence we let the optimization of offset coding out of the scope of the present paper. In turn, we run all the following experiments using the natural binary representation of the offsets, thus getting an upper-bound on the bitrate. Clearly, the cost of such representation turns to be $2\lceil \log_2(2\chi+1) \rceil + \lceil \log_2(2\theta+1+\omega) \rceil$ bits per each occlusion.

TABLE I
TEST VIDEO SEQUENCES.

ID	Name	Frames	Resolution	Hz	Views
S1	Poznan_Hall2	200	1920 × 1088	25	7,6,5
S2	Poznan_Street	250	1920 × 1088	25	5,4,3
S3	Kendo	300	1024 × 768	30	1,3,5
S4	Balloons	300	1024 × 768	30	1,3,5
S5	Undo_Dancer	250	1920 × 1088	25	1,5,9
S6	GT_Fly	250	1920 × 1088	25	5,3,1

V. EXPERIMENTAL EVALUATION

In this section the performance of the proposed STOP codec will be analyzed and compared with both standard simulcast coding and competing solutions based on coding of occlusions.

The coding tests have been run on a standard set of multiple view plus depth video sequences [55]. The selected sequences include both synthetic and real videos with either static or moving background, and with or without camera motion and zooming. In Tab. I the details of each sequence (sequence name, total number of frames, resolution, frame rate and index of coded views) are summarized; in the following each sequence will be referenced using the label reported in the first column of the table.

[3]This supplementary option has been neglected in Fig. 4 and Fig. 5 to make them more simple and readable.

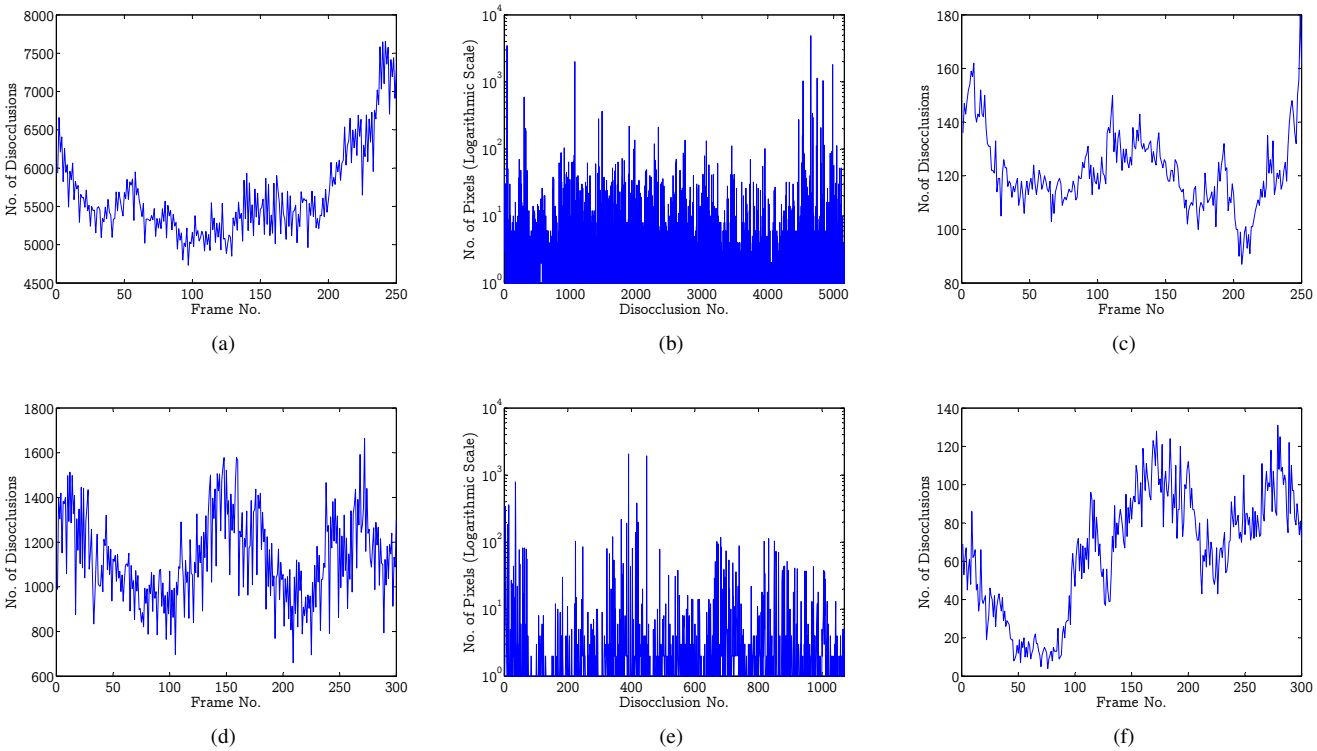


Fig. 6. Number of occlusions per frame (a)(d), occlusion sizes in the frame 100 (b)(e) and number of occlusions selected for coding (c)(f) in S2 when warping view 4 to 5 (top) and S3 when warping view 3 to 1 (bottom).

A. STOP Settings.

In Sect. IV we presented the spatio-temporal estimation and compensation employed in the STOP codec. Such process depends on a number of parameters that can be tuned to achieve different quality/bit-rate trade-offs.

The most important parameter is certainly represented by the threshold τ_S that allows one to select the most significant occlusions for the following matching and compensation. On the other hand the dropped (smaller) occlusions will be recovered by inpainting. Clearly, this choice has a direct impact on visual quality and bit-rate overhead for occlusion coding.

Usually, the warping process generates a very large number of small occlusions and a limited number of large occlusions; these latter are likely to be related to foreground objects and depend on their position in depth and on the baseline of the cameras. Furthermore, complex or highly textured scenes, e.g. tree leaves, crowd of people, etc, can easily yield a large number of very small occlusions. Moreover, other occlusions can be caused by depth estimation errors.

In Fig. 6 we show some experimental data obtained when warping S2 and S3 sequences, respectively. In particular S2 refers to a sequence with highly textured background (buildings and trees) whereas S3 is characterized by a simpler background. Both sequences have few foreground moving characters.

Fig. 6a shows that warping S2 from view 4 to view 5 one gets a large number of occlusions, on average 5639 occlusions per frame. These are mainly due to the complex nature of the background in S2. Fig. 6b shows the size of each occlusion patch in frame 100 of the sequence; it clearly turns out that most occlusions are very small ones, with only 1.38% of the total covering more than 100 pixels. In Fig. 6d, 6e the same results are shown when warping S3 from view 3 to view 1. In this case, 1127 occlusions per frame were found on average, i.e. 5 times less than S2; looking at frame 100 we note that only 15 occlusions contain more than 100 pixels.

If we fix $\tau_S = 36$ to select only large occlusions and we omit cracks by setting $\tau_R = 2$ as defined in Sect. IV-B we got the results shown in Fig. 6c and Fig. 6f. It can be noted that only 121 and 65 occlusions per frame are selected in S2 and S3, respectively.

Previous thresholds have been selected as reasonable trade-off between image quality and occlusion coding overhead and will be employed in all the following experiments.

The remaining STOP coding parameters are reported in Tab. II. In particular we fixed $\tau_L = 20$ to split vertical occlusions as described in Sect. IV-B and HEVC is used to encode the panorama texture and depth videos. The most significant HEVC settings are shown in Tab. II, as well.

B. Coding Experiments and Comparisons

In this section the compression efficiency provided by STOP codec is evaluated experimentally and compared with other coding approaches based on standard coding tools, i.e. video plus depth simulcast using the widespread diffused AVC and the novel HEVC coding standard[4]. When using simulcast 3

[4]The experimental results have been obtained with JMVC-8.3 for AVC and HM-11.0 for HEVC coding, respectively.

TABLE II
STOP CODEC SETTINGS.

Occlusion Settings		HEVC Configuration	
Parameter	value	Parameter	value
τ_S	36	MaxCUWidth	64
τ_R	2	MaxCUHeight	64
τ_L	20	MaxPartitionDepth	4
θ	5	IntraPeriod	32
Ω	2	GOPSize	8

TABLE III
AVC CODEC SETTINGS.

Parameter	value
IntraPeriod	32
GOPSize	8
ReferenceFrames	5
CABAC	enabled

views plus the corresponding depth maps are separately coded using AVC or HEVC, respectively. The HEVC settings are the same as those selected according to Tab.II for the STOP codec, whereas similar parameters have been used in the case of AVC as shown in Tab. III. In all the following experiments, the sequences are encoded at four quality levels by selecting the values for quantization parameter (QP), namely $QP = 26, 30, 34, 38$. The compression performance obtained by simulcast is shown in Tab. IV and will be used as a reference for the following results; the total bitrate in kbps and the average PSNR obtained on the luma component of the 3 views are reported.

Furthermore, a backward HEVC compatible implementation of [27] is used as a benchmark closer to our proposal. In [27] the middle view and depth are encoded with HEVC and the disoccluded regions for the left and the right views are estimated using DIBR on the encoder side. As in STOP, small occlusions are dropped; finally one left occlusion frame (plus corresponding depth) and one right occlusion frame (plus depth) are constructed so as that every 16×16 block that includes at least an occluded pixel is encoded using a modified HEVC encoder. Fig. 7 shows an example of an occlusion frame (sequence S3) created according to [27], where only a subset of blocks carry occlusions that needs to be coded. For the sake of comparison with our approach, a slight variation of [27] has been implemented, where a standard HEVC encoder is used to compress the left and the right occlusion frames, respectively. In the following we term such coding approach as Middle plus Disoccluded Regions (MDR). The major differences between MDR and STOP are that this latter creates a panoramic picture and does not use video coding on occlusions (that are encoded as spatio-temporal offsets).

The compression performance yielded by MDR is shown in Tab. V, where the amount of bitrate taken by the different components is detailed, namely the bitrate spent to encode the middle view V_m, the corresponding depth map D_m and the left and the right occlusion frames (texture and depth). The overall bitrate and the PSNR of the middle view are

TABLE IV
AVC AND HEVC SIMULCAST CODING RESULTS: OVERALL BITRATE [KBPS] AND AVERAGE LUMA PSNR [DB] (AVERAGED ON 3 VIEWS).

Seq.	QP	AVC		HEVC	
		Bitrate	PSNR	Bitrate	PSNR
S1	26	3040.18	41.44	2044.17	41.94
	30	1780.67	40.41	1072.00	41.16
	34	1145.03	39.07	617.19	40.08
	38	772.92	37.41	380.90	38.78
S2	26	6500.99	38.72	5804.78	39.44
	30	3567.56	37.18	2754.69	37.80
	34	2082.57	35.51	1491.99	36.17
	38	1249.60	33.61	868.53	34.50
S3	26	3133.27	42.48	2495.79	43.15
	30	2246.83	40.68	1485.27	41.36
	34	1483.15	38.63	906.83	39.35
	38	997.11	36.28	583.74	37.30
S4	26	3339.07	41.90	2374.16	42.69
	30	2140.09	39.97	1429.03	40.96
	34	1391.17	37.70	877.31	38.87
	38	912.62	35.11	568.85	36.66
S5	26	15498.54	37.50	11986.28	38.21
	30	8882.88	35.30	6224.71	35.91
	34	5204.67	33.25	3359.61	33.84
	38	3082.28	31.24	1876.43	31.95
S6	26	14952.01	39.17	10825.77	39.70
	30	8902.69	37.28	5679.31	37.78
	34	5441.37	35.35	3076.60	35.99
	38	3402.05	33.29	1734.22	34.27

TABLE V
MDR-HEVC COMPRESSION RESULTS: BITRATE [KBPS] AND AVERAGE LUMA PSNR [DB] OF THE MIDDLE VIEW (V_m), BITRATE FOR THE DEPTH MAP D_m, OCCLUSION BITRATE (LEFT AND RIGHT OCCLUSION FRAMES PLUS DEPTH), AND OVERALL MDR BITRATE.

Seq.	QP	V_m		D_m	Occl.	Total
		Bitrate	PSNR	Bitrate	Bitrate	Bitrate
S1	26	618.02	41.94	79.58	630.33	1327.94
	30	323.16	41.13	42.58	344.79	710.54
	34	185.92	40.03	23.53	227.02	436.48
	38	114.97	38.70	14.18	163.45	292.61
S2	26	1770.70	39.47	130.32	814.64	2715.67
	30	830.35	37.85	67.21	468.66	1366.23
	34	451.97	36.23	34.44	277.82	764.25
	38	265.39	34.58	18.85	172.10	456.36
S3	26	596.55	43.18	183.41	607.96	1387.93
	30	363.16	41.40	102.23	395.80	861.21
	34	227.77	39.39	58.08	272.17	558.04
	38	150.90	37.35	34.12	190.62	375.65
S4	26	605.97	42.71	150.13	1059.86	1815.97
	30	377.43	41.01	79.33	640.61	1097.38
	34	239.02	38.95	42.80	415.58	697.40
	38	159.81	36.74	24.09	277.26	461.16
S5	26	3723.94	38.21	269.09	2199.75	6192.77
	30	1902.76	35.91	172.18	1388.81	3463.76
	34	1011.78	33.84	108.69	884.79	2005.27
	38	554.57	31.95	71.25	602.45	1228.27
S6	26	3028.14	39.70	580.39	1885.79	5494.32
	30	1562.91	37.78	330.72	1177.12	3070.75
	34	840.13	35.99	185.65	718.03	1743.81
	38	470.72	34.27	109.26	449.14	1029.12

TABLE VI
STOP COMPRESSION RESULTS (HEVC BASED): BITRATE AND AVERAGE LUMA PSNR [DB] OF PANORAMIC VIEW (V_{pan}), BITRATE FOR THE DEPTH MAP D_{pan}, OCCLUSION BITRATE (SPATIO-TEMPORAL OFFSETS), AND OVERALL STOP BITRATE.

Seq.	QP	V_{pan}		D_{pan}	Occl.	Total
		Bitrate	PSNR	Bitrate	Bitrate	Bitrate
S1	26	687.24	42.24	93.74	52.33	833.32
	30	365.60	41.40	52.44	48.67	466.72
	34	217.27	40.28	30.44	43.44	291.16
	38	138.28	38.92	19.29	39.37	196.95
S2	26	1945.53	39.67	156.26	62.08	2163.87
	30	946.79	38.04	83.23	55.56	1085.60
	34	529.81	36.38	43.68	45.17	618.67
	38	319.92	34.69	24.05	40.72	384.70
S3	26	687.85	43.61	224.77	37.19	949.82
	30	428.58	41.81	131.04	33.19	592.82
	34	275.70	39.75	77.62	30.10	383.42
	38	187.53	37.63	47.34	26.87	261.75
S4	26	758.81	43.05	210.49	49.23	1018.54
	30	482.83	41.26	119.41	46.08	648.33
	34	312.63	39.09	69.08	42.08	423.80
	38	212.79	36.82	40.72	38.19	291.71
S5	26	3918.58	38.32	287.72	102.08	4308.39
	30	2010.77	36.02	189.17	95.95	2295.89
	34	1069.84	33.94	123.48	98.57	1291.89
	38	585.88	32.05	84.14	98.86	768.88
S6	26	3149.16	39.91	635.01	59.39	3843.56
	30	1641.79	37.97	372.32	57.51	2071.61
	34	895.06	36.16	216.77	52.92	1164.74
	38	504.73	34.41	130.32	46.14	681.18

(a) (b)

Fig. 7. MDR coding approach: warped image with dis-occlusions (a) and corresponding occlusions frame (b).

shown as well, to compare the results with simulcast (Tab. IV). In MDR and STOP the quality of the side views depends on both compression and DIBR performance and will be thoroughly discussed in Sect. V-C. The results in Tab. V show that using MDR one significantly reduces the total bitrate by more than 30% over simulcast. However it turns out that almost one half of the MDR bitrate is due to the occlusion coding, that in turn represents a limited amount of visual data (in terms of their spatial occupation of the frame). This is clearly due to the fact that the occlusion frames are not efficiently compressed by a standard video codec; indeed, it is likely that both intra (spatial) and inter (temporal) predictions are dramatically impaired in the presence of blocky pictures carrying only occlusion regions.

The proposed STOP codec clearly aims at overcoming the above issue: in fact, occlusion regions on the frame side (that we termed slits) are merged along the central view to form the panorama view, whereas the remaining occlusions are coded and concealed exploiting the spatio-temporal redundancy. The coding performance achieved by STOP is shown in Tab. VI, where bitrate contributions are partitioned following the same principle of Tab. V so as to make it easy to compare the results; in particular, we provide bitrate and PSNR spent on the panorama view V_{pan}, the bitrate allocated to the panorama depth D_{pan} and that spent on occlusions. Representing the occlusions in terms of spatio-temporal offset for each connected component turns out to be very efficient with dramatic bitrate savings in the order of 80% for occlusion coding. As an example of the overall coding efficiency STOP takes 833 kpbs at $QP = 26$, whereas AVC simulcast, HEVC simulcast and MDR achieves 3 Mbps, 2 Mpbs and 1.3 Mbps, respectively. Fig. 8 shows the average bit rate saving (in percentage) experienced on our dataset. The average bit rate saving of STOP turns out to be 72.85% versus AVC simulcast, 58.34% versus HEVC simulcast and 31.88% versus MDR.

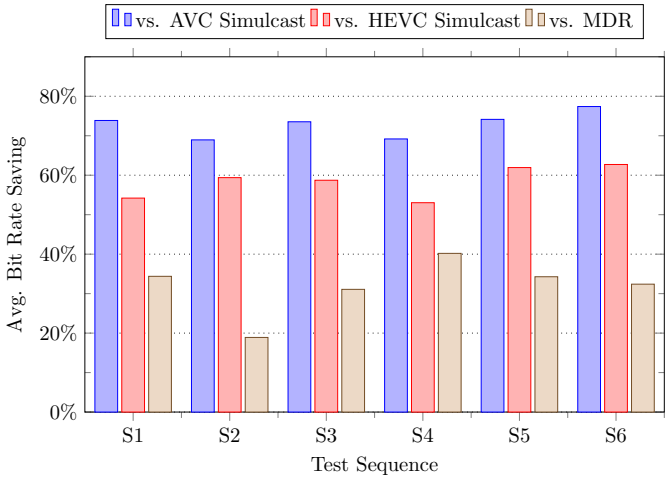


Fig. 8. Average percentage bit rate saving per test sequence.

To better analyze the bitrate distribution of STOP in Fig. 9 the average percentage bitrate allocated to texture, depth and occlusion offsets are reported as a function of QP. It can be noted that texture takes from 75% to 82% of the bitrate, depth from 13% to 14% and occlusion offsets from 12% to 4% as a function of QP. In particular, it is worth noticing that occlusion bitrates does depend on QP; in fact, in the current implementation of STOP we kept fixed the parameters for the selection of the occlusions to be coded as discussed in Sect. V-A, independently on the QP value.

A widespread tool for comparing the compression performance is the Bjøntegaard metric [56]. Tab. VII shows the STOP results in terms of Bjøntegaard ΔPSNR and ΔR (PSNR[5] and bitrate differences) when compared to the reference codecs used in this paper. The table shows that the proposed coding tool saves 78% bitrate compared to AVC

[5]As already mentioned in this section the PSNR of the encoded view (middle view in MDR and panorama view in STOP) is used to represent the rate/distortion curve of the codec.

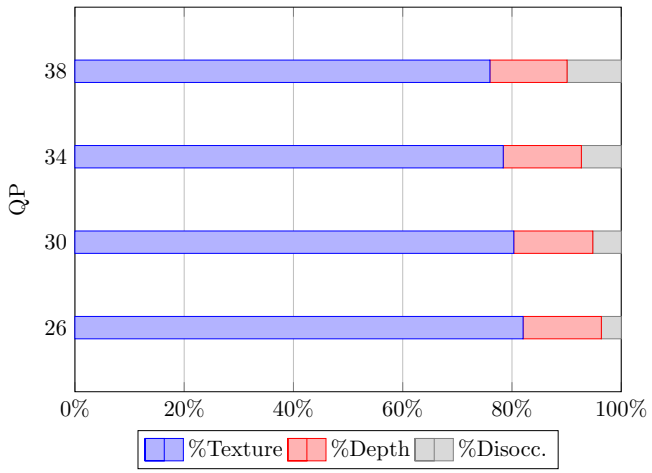


Fig. 9. Average bit rate consumption of texture, depth and occlusions in STOP versus QP.

simulcast, 58% compared to HEVC simulcast and 32% compared to MDR.

TABLE VII
BJØNTEGAARD DELTAS (ΔPSNR [DB] AND ΔBITRATE R [%]) OF STOP VERSUS AVC SIMULCAST, HEVC SIMULCAST AND MDR.

Seq.	vs. AVC simulcast		vs. HEVC simulcast		vs. MDR	
	ΔPSNR	ΔR	ΔPSNR	ΔR	ΔPSNR	ΔR
S1	4.38	-81.37	1.63	-53.91	0.89	-33.94
S2	4.13	-75.77	2.44	-59.47	0.60	-19.30
S3	7.10	-77.59	3.79	-58.73	1.68	-31.15
S4	7.01	-75.54	3.40	-52.92	2.37	-40.02
S5	5.71	-78.39	3.40	-62.37	1.57	-34.50
S6	5.94	-81.00	3.00	-62.84	1.25	-32.58
Avg.	5.71	-78.28	2.95	-58.38	1.40	-31.92

Finally, it is useful to compare the STOP bitrate with that of a single video channel; in the broadcasting scenario, such metric represents an estimate of the bandwidth overhead with respect to the standard 2D service required to enable 3D TV applications. Previous results show that STOP can be used to grant a 3D video service with a bandwidth overhead below 50%.

C. 3D Video Quality

In STOP (and similarly in MDR) only the panoramic picture is encoded whereas all the remaining points of views can be synthesized by DIBR technique. In the following we propose to use the well-known *Structural SIMilarity (SSIM) index* [57] metric as a measure of the obtained image quality. In fact, SSIM has been designed to better match the human visual perception and yields better visual quality assessment with respect to PSNR [57]–[59]. Moreover, PSNR generally estimates values that are not correlated to the visual perception in presence of DIBR: indeed DIBR, can achieve excellent perceived image quality even without guaranteeing a bit faithful reconstruction of the original picture due to resampling issues, color and illumination correction, rectification errors, etc.

In order to rank the overall 3D video quality guaranteed by STOP a set of virtual views has been generated. In particular, a pair of intermediate views is computed in between leftmost and central and central and rightmost views, respectively. Then a supplementary virtual view is created in between in a second iteration, getting a total of 9 views spaced by one quarter of the original baseline. View Synthesis with Inverse Mapping (VSIM) [47] is used for DIBR, employing the Telea fast marching algorithm [31] for inpainting uncoded occlusions.

Fig. 10 shows the visual quality of the decoded views and intermediate virtual views for frame 13 of sequence S2. From top left to bottom right we report the views at position 5 (leftmost), 4.5, 4 and 3.5. Fig. 10c is the central view in the panorama and its quality depends only on HEVC compression efficiency, whereas other images are obtained by warping the central view and by filling the disocclusions with the proposed spatio-temporal approach. While there is a significant drop in PSNR between the coded view 4 and the interpolated views, the visual quality of all images is high as predicted by the reported values of the SSIM.

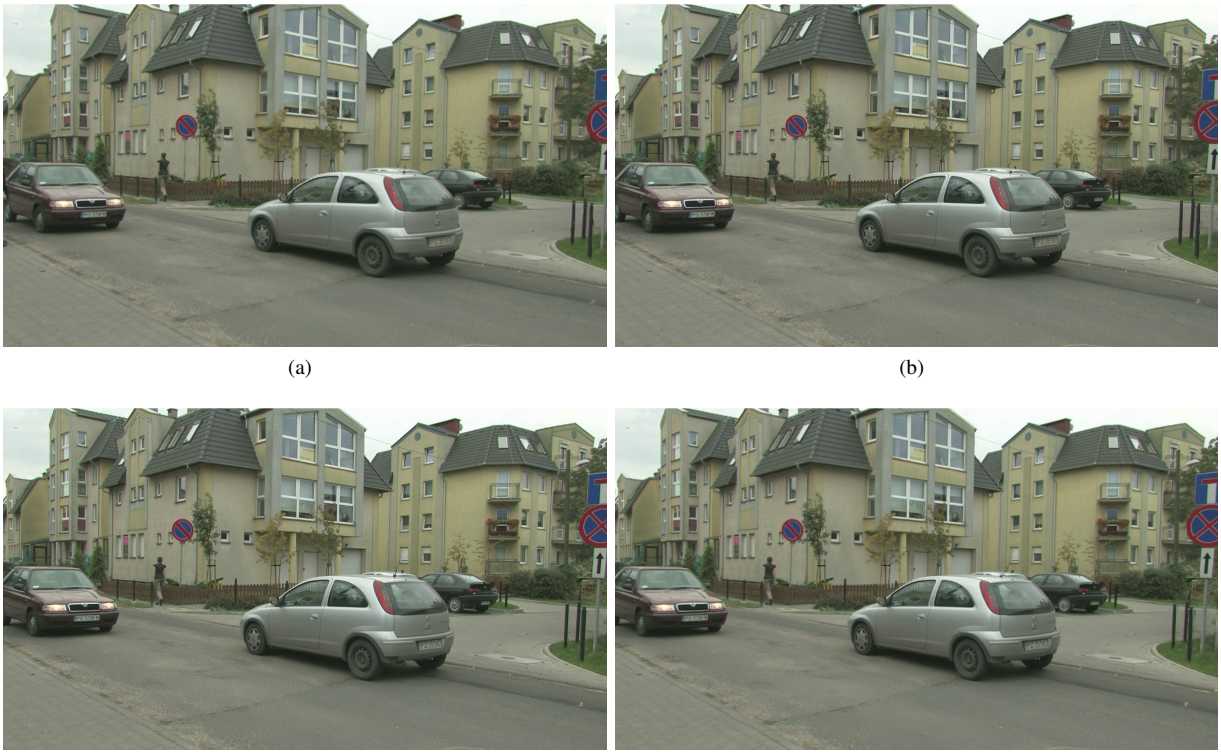
Fig. 11 shows the average SSIM provided by STOP compared with HEVC simulcast and MDR as a function of the view position. The quality of the virtual views created from MDR and the proposed coding tool is almost the same since in both techniques the left and the right views are obtained by warping the middle view. As expected both MDR and STOP exhibit a slight penalty on the quality of the extreme views; in fact, those views are directly coded by HEVC with simulcast whereas in the other cases they are estimated by DIBR plus occlusion compensation. Nonetheless the quality impairment turns out to be negligible, e.g. 0.02, if compared to the bitrate savings discussed in Sect. V-B. Moreover, Fig. 11-(e),(f) show that the quality gap turns to be significantly reduced for synthetic video sequences (S5,S6) where ground truth depth maps are available; the greater depth accuracy in turn improves both STOP coding efficiency (better construction of the panorama picture, occlusion estimation and matching) and quality of the view synthesis on the decoder side. This observation also highlight that STOP performance is expected to improve further exploiting future research and advances in depth acquisition/estimation and DIBR algorithms.

D. Complexity Analysis of STOP

In this section the computational cost of STOP is analyzed and compared against those of standard 2D video codecs, and their forthcoming 3D extensions.

Compared to a standard HEVC encoder, STOP comprises two extra modules: panorama construction, and occlusion matching. Panorama construction involves two warping operations to estimate the slits; either reference view is warped to the outermost views using DIBR or vice versa. The asymptotic cost of DIBR operation is linear in the number of pixels in the image.

Occlusion matching in STOP is a two levels operation. First, their sizes are computed and the smaller occlusions are dropped from further processing as they do not contribute significantly to the decoded image quality; indeed, those can



(a) (b)

(c) (d)

Fig. 10. Examples of decode views of S2: leftmost view 5, SSIM=0.975, PSNR=32.93 dB (a), intermediate views 4.5, SSIM=0.989, PSNR=36.79 dB (b), 4, SSIM=0.992 PSNR=39.46 dB (c) and 3.5, SSIM=0.986, PSNR=35.26 dB (d).

be successfully inpainted on the decoder side. Computing the size of the occlusions is again a linear time operation that is carried out by component labeling algorithm. Second, the occlusion offsets must be estimated using a spatiotemporal search. The best match is found using Sum of Absolute Difference (SAD) in a time window of size Θ. Since a full search would be computationally inefficient, an offset optimization mechanism has been designed to limit the search space. The search space optimization mechanism exploits the average occlusion disparity and the temporal distance (offset) of the target frame as described in details in Sect. IV-B2.

In conclusion, in STOP , the additional computational cost with respect to standard 2D encoding is kept limited by exploiting the pre-processing represented by the panorama construction that is effective in capturing most interview redundancies. Then, a single coding mode is added to process a limited set of occlusions that cannot be handled efficiently in the panorama picture. On the receiver side, the bit stream is decoded with the standard 2D decoder. The base view is obtained by simple cropping of the panorama picture and the occlusions offsets are used to directly copy the target patches for filling the occlusions that show up during DIBR.

The 3D extensions of the most recent HEVC standard follow a different approach based on the design of additional coding modes capable of capturing the inter-view redundancy at the coding unit (CU) level. In 3D-HEVC the dependent views are coded using many additional coding tools: for instance disparity-compensated prediction (DCP) which exploits already coded images of other views, inter-view motion prediction and interview residual prediction. For efficient compression of depth videos 3D-HEVC employs additional coding tools, e.g. depth intra-coding exploiting new depth modeling modes, disparity-compensated prediction and inter-view motion parameter prediction which significantly increases the computational complexity of 3D-HEVC [60], [61]. The new coding modes allows one to reduce significantly the bitrate required to encode multiple views and depths while guaranteeing the same image quality of independent view coding (simulcast). On the other hand, with STOP we exploit DIBR to avoid coding the picture parts that can be estimated by DIBR and we keep additional coding cost as limited as possible. Clearly, as shown in Fig. 11 the STOP picture quality slightly degrades as moving away from the central coded view; nonetheless, if one can tolerate such slight drop (less than about 0.03 in terms of SSIM measured in our experiments) STOP can represent a viable solution for 3D video coding.

VI. Conclusions

In this paper the novel STOP 3D video codec has been designed building around two main ideas, namely the construction of a panorama view and the exploitation of spatio-temporal correlation for filling occlusions caused by DIBR. By exploiting the mentioned concepts STOP efficiently compresses 3D video while using a standard video codec such as AVC and HEVC to encode most of the data. In this paper we showed promising results in terms of bitrate reduction and visual quality. Beside coding efficiency, STOP enables free viewpoint 3D TV while guaranteeing a simple mechanism to

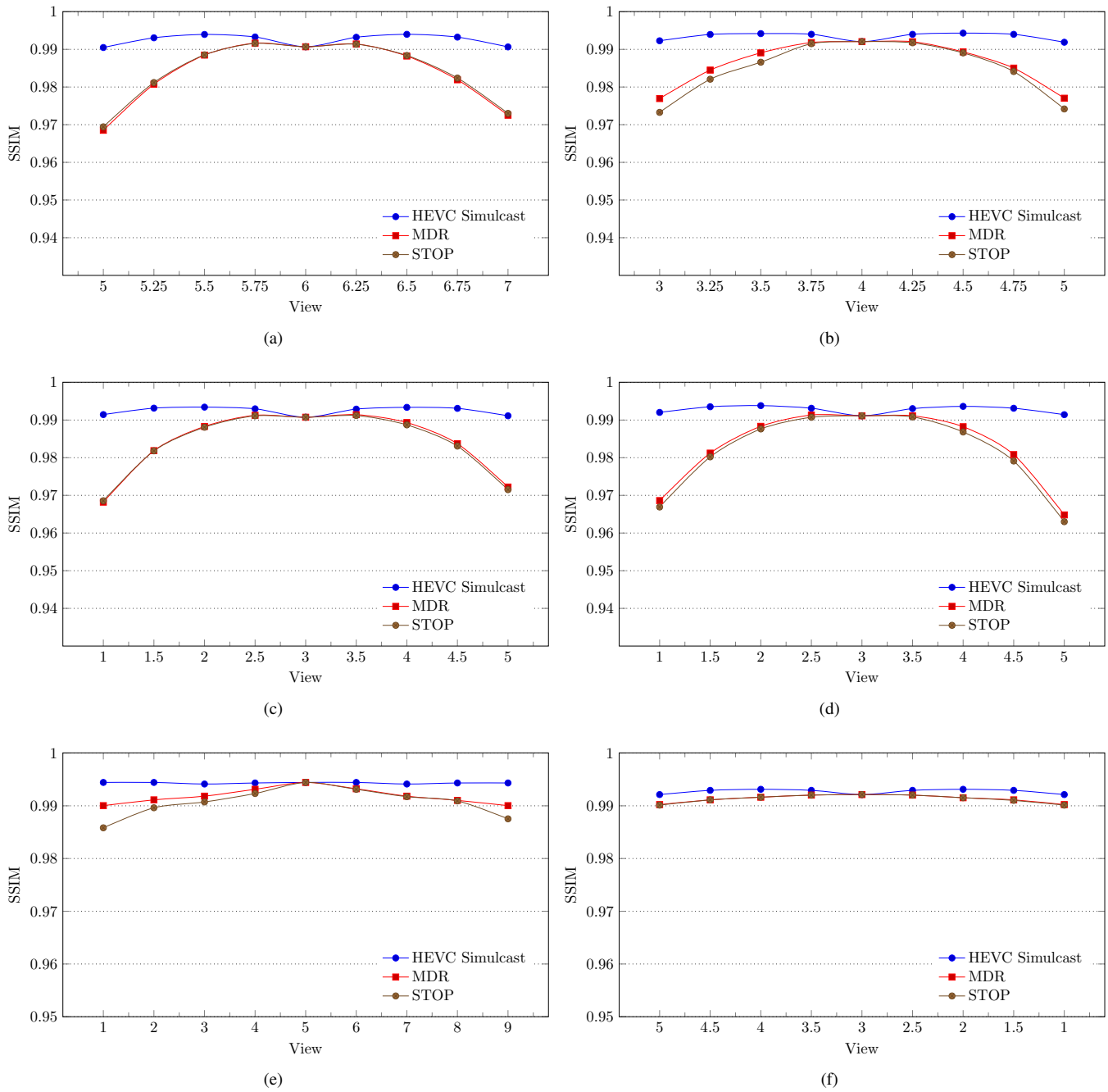


Fig. 11. Average SSIM versus view position coded by STOP with $QP = 26$. Sequence: Poznan_Hall2, views = {6.75, 6.5, 6.25, 5.75, 5.5, 5.25} (a); Sequence: Poznan_Street, views = {4.75, 4.5, 4.25, 3.75, 3.5, 3.25} (b); Sequence: Kendo, views = {1.5, 2, 2.5, 3.5, 4, 4.5} (c); Sequence: Balloons, views = {1.5, 2, 2.5, 3.5, 4, 4.5} (d); Sequence: Undo_Dancer, views = {2, 3, 4, 6, 7, 8} (e); Sequence: GT_Fly, views = {4.5, 4, 3.5, 2.5, 2, 1.5} (f).

allow legacy 2D displays to extract the standard 2D video. Future works include the rate/distortion optimization of the occlusion matching criteria, the design of entropy coding tools for occlusion offsets and the exploitation of the coding ideas proposed in this paper as possible novel coding modes for the future 3D extensions of AVC and HEVC under investigation by the MPEG.

ACKNOWLEDGMENT

The authors would like to thank the Poznan University of Technology for providing 'Poznan_Hall2' and 'Poznan_Street' 3D video sequences, Nagoya University for providing the 'Balloons' and 'Kendo' and Nokia Research for 'Undo_Dancer' and 'GT_Fly' 3D video sequence.

REFERENCES

[1] A. Vetro, A.M. Tourapis, K. Muller, and Tao Chen, "3D-TV content storage and transmission," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 384–394, June 2011.
[2] DVB, "3D-TV Commercial requirements," *DVB Document A151*, 2010.
[3] DVB, "Frame compatible plano-stereoscopic 3DTV," *DVB Document A154*, 2011.

[4] ITU-T ISO/IEC JTC 1, *Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, ISO/IEC, 2010.

[5] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1461–1473, 2007.

[6] A. Smolic, K. Mueller, P. Merkle, P. Kauff, and T. Wiegand, "An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution," in *Picture Coding Symposium, 2009. PCS 2009*, 2009, pp. 1–4.

[7] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski, "Layered depth images," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 231–242.

[8] Steven J Gortler, Li-Wei He, and Michael F Cohen, "Rendering layered depth images," Tech. Rep., Microsoft Research Technical Report MSTR-TR-97-09, 1997.

[9] Aljoscha Smolic, Karsten Mueller, Philipp Merkle, Nicole Atzpadin, Christoph Fehn, Markus Mueller, Oliver Schreer, Ralf Tanger, Peter Kauff, and Thomas Wiegand, *Multi-view video plus depth (MVD) format for advanced 3D video systems*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q 6:21–27, 2007.

[10] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[11] G. Ballocca, P. D'Amato, M. Grangetto, and M. Lucenteforte, "Tile format: A novel frame compatible approach for 3D video broadcasting," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[12] K. Muller, P. Merkle, G. Tech, and T. Wiegand, "3D video formats and coding methods," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 2389–2392.

[13] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH and ACM Transactions on Graphics*, 2004, pp. 600–608.

[14] Liang Z. and W.J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 191 – 199, june 2005.

[15] Ya mei Feng, Dong xiao Li, Kai Luo, and Ming Zhang, "Asymmetric bidirectional view synthesis for free viewpoint and three-dimensional video," *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 4, pp. 2349 –2355, Nov. 2009.

[16] Zefeng Ni, Dong Tian, S. Bhagavathy, J. Llach, and B.S. Manjunath, "Improving the quality of depth image based rendering for 3D video systems," in *16th IEEE International Conference on Image Processing (ICIP)*, nov 2009, pp. 513 –516.

[17] Sehoon Yea and Anthony Vetro, "View synthesis prediction for multiview video coding," *Signal Processing: Image Communication*, vol. 24, no. 12, pp. 89 – 100, 2009, Special issue on advances in three-dimensional television and video.

[18] K. Muller, A. Smolic, K. Dix, P. Merkle, and T. Wiegand, "Coding and intermediate view synthesis of multiview video plus depth," in *16th IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 741 –744.

[19] Woo-Shik Kim, A. Ortega, Jaejoon Lee, and HoCheon Wey, "3D video coding using depth transition data," in *Picture Coding Symposium (PCS), 2010*, Dec. 2010, pp. 178 –181.

[20] ISO/IEC JTC1/SC29/WG11, "Call for proposals on 3D video coding technology," Tech. Rep. N12036, ISO/IEC JTC1/SC29/WG11, mar 2011, ftp://ftp.merl.com/pub/avetro/3dv-cfp/.

[21] "ITU-T/ISO/IEC joint collaborative team on 3D video coding extension development," http://phenix.it-sudparis.eu/jct2/index.php, 2012.

[22] Li Zhang Teruhiko Suzuki Dmytro Rusanovskyy, Fang-Chu Chen, "3D-AVC Test Model 8," Tech. Rep. N13919, ISO/IEC JTC1/SC29/WG11, Nov 2013.

[23] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *Image Processing, IEEE Transactions on*, vol. 22, no. 9, pp. 3366–3378, 2013.

[24] Nam Ling, "High efficiency video coding and its 3D extension: A research perspective," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*. IEEE, 2012, pp. 2150–2155.

[25] G. Van Wallendael, S. Van Leuven, J. De Cock, F. Bruls, and R. Van De Walle, "3D video compression based on high efficiency video coding," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 137–145, 2012.

[26] J. Salvador and J.R. Casas, "Multi-view video representation based on fast monte carlo surface reconstruction," *Image Processing, IEEE Transactions on*, vol. 22, no. 9, pp. 3342–3352, 2013.

[27] Marek Domanski, Jacek Konieczny, Maciej Kurc, Robert Ratajczak, Jakub Siast, Olgierd Stankiewicz, Jakub Stankowski, and Krzysztof Wegner, "3D video compression by coding of disoccluded regions," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 1317–1320.

[28] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2000, SIGGRAPH '00, pp. 417–424, ACM Press/Addison-Wesley Publishing Co.

[29] M. Bertalmio, A.L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Dec. 2001, vol. 1, pp. 355–362.

[30] Manuel M Oliveira Brian Bowen Richard and McKenna Yu-Sung Chang, "Fast digital image inpainting," in *Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain*, 2001, pp. 106–107.

[31] Alexandru Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[32] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[33] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun, "Fragment-based image completion," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 303–312, 2003.

[34] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher, "Simultaneous structure and texture image inpainting," *Image Processing, IEEE Transactions on*, vol. 12, no. 8, pp. 882–889, 2003.

[35] Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," in *Picture Coding Symposium, 2009. PCS 2009*. IEEE, 2009, pp. 1–4.

[36] Alexandre Hervieu, Nicolas Papadakis, Aurélie Bugeau, Pau Gargallo, and Vicent Caselles, "Stereoscopic image inpainting: distinct depth maps and images inpainting," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 4101–4104.

[37] Josselin Gautier, Olivier Le Meur, and Christine Guillemot, "Depth-based image completion for view synthesis," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[38] N. Plath, S. Knorr, L. Goldmann, and T. Sikora, "Adaptive image warping for hole prevention in 3D view synthesis," *Image Processing, IEEE Transactions on*, vol. 22, no. 9, pp. 3420–3432, 2013.

[39] M Koppel, Patrick Ndjiki-Nya, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, K Muller, and Thomas Wiegand, "Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1809–1812.

[40] Patrick Ndjiki-Nya, Martin Koppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, K Muller, and Thomas Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-D video," *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 453–465, 2011.

[41] Martin Koppel, Xi Wang, Dimitar Doshkov, Thomas Wiegand, and Patrick Ndjiki-Nya, "Consistent spatio-temporal filling of disocclusions in the multiview-video-plus-depth format," in *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on*. IEEE, 2012, pp. 25–30.

[42] Martin Koppel, Xi Wang, Dimitar Doshkov, Thomas Wiegand, and Patrick Ndjiki-Nya, "Depth image-based rendering with spatio-temporally consistent texture synthesis for 3-D video with global motion," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2713–2716.

[43] Shu-Jyuan Lin, Chia-Ming Cheng, and Shang-Hong Lai, "Spatio-temporally consistent multi-view video synthesis for autostereoscopic displays," in *Advances in Multimedia Information Processing-PCM 2009*, pp. 532–542. Springer, 2009.

[44] Hsiao-An Hsu, Chen-Kuo Chiang, and Shang-Hong Lai, "Global optimization for spatio-temporally consistent view synthesis," in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*. IEEE, 2012, pp. 1–8.

[45] Ming Xi, Liang-Hao Wang, Qing-Qing Yang, Dong-Xiao Li, and Ming Zhang, "Depth-image-based rendering with spatial and temporal texture synthesis for 3dtv," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 1–18, 2013.
[46] Massimo Camplani and Luis Salgado, "Efficient spatio-temporal hole filling strategy for kinect depth maps," 2012, vol. 8290, pp. 82900E–82900E–10.
[47] Muhammad Shahid Farid, Maurizio Lucenteforte, and Marco Grangetto, "Depth image based rendering with inverse mapping," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 135–140.
[48] Jun-Wei Hsieh, Hong-Yuan Mark Liao, Kuo-Chin Fan, Ming-Tat Ko, and Yi-Ping Hung, "Image registration using a new edge-based approach," *Computer Vision and Image Understanding*, vol. 67, no. 2, pp. 112–130, 1997.
[49] Hui Li, B.S. Manjunath, and S.K. Mitra, "A contour-based approach to multisensor image registration," *Image Processing, IEEE Transactions on*, vol. 4, no. 3, pp. 320–334, 1995.
[50] Barbara Zitova and Jan Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.
[51] Lisa Gottesfeld Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.
[52] Jun Yu and Mandyam D Srinath, "An efficient method for scene cut detection," *Pattern Recognition Letters*, vol. 22, no. 13, pp. 1379–1391, 2001.
[53] T. Vlachos, "Cut detection in video sequences using phase correlation," *Signal Processing Letters, IEEE*, vol. 7, no. 7, pp. 173–175, 2000.
[54] Chung-Lin Huang and Bing-Yao Liao, "A robust scene-change detection method for video segmentation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 12, pp. 1281–1288, 2001.
[55] ISO/IEC JTC1/SC29/WG11, "Call for Proposals on 3D Video Coding Technology," *Doc. N12036, Geneva, Switzerland*, March 2011.
[56] Gisle Bjontegard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, 2001.
[57] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
[58] Zhou Wang and AC. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98–117, Jan 2009.
[59] Weisi Lin and C.-C. Jay Kuo, "Perceptual visual quality metrics: A survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297 – 312, 2011.
[60] Hamid Reza Tohidypour, Mahsa T. Pourazad, and Panos Nasiopoulos, "A low complexity mode decision approach for HEVC-based 3D video coding using a Bayesian method," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 895–899.
[61] Qiuwen Zhang, Nana Li, and Qinggang Wu, "Fast Mode Decision for 3D-HEVC Depth Intracoding," *The Scientific World Journal*, vol. 2014, pp. 1–9, 2014.

Muhammad Shahid Farid (S'13) received his B.S., M.Sc. and M.Phil degrees all in Computer Science with distinction from the University of the Punjab, Lahore, Pakistan in 2004, 2006 and 2009 respectively. He is a Lecturer at Punjab University College of Information Technology, University of the Punjab, Lahore since 2007. Currently, he is pursuing his Ph.D. in 3D television (3DTV) technology at Computer Science Department, University of Torino, Italy. His research interests broadly include image segmentation, image morphing and 3DTV specifically 3D video representation and coding, virtual view synthesis and quality assessment of 3D videos.

Maurizio Lucenteforte M. Lucenteforte (M10) is assistant professor at the Dept. of Computer Science. He previously worked at the National Research Council of Italy (CNR). His current research activity is oriented towards problems of image processing, 3D representation and coding, virtual and augmented reality. In particular, he has been working on topics comprising 3D representation of geographic information, virtual simulation of hostile environments, reversible watermarking applied to medical imaging, video plus depth based approaches for stereo and multi-view television signals.

Marco Grangetto M. Grangetto (S99-M03-SM09) received his Electrical Engineering degree and Ph.D. degree from the Politecnico di Torino, Italy, in 1999 and 2003, respectively. He is currently Associate Professor at the Computer Science Department, University of Torino. His research interests are in the fields of multimedia signal processing and networking. In particular, his expertise includes wavelets, image and video coding, data compression, video error concealment, error resilient video coding unequal error protection, and joint source channel coding. Prof. Grangetto was awarded the Premio Optime by Unione Industriale di Torino in September 2000, and a Fulbright grant in 2001 for a research period with the Department of Electrical and Computer Engineering, University of California at San Diego. He has participated in the ISO standardization activities on Part 11 of the JPEG 2000 standard. He has been a member of the Technical Program Committee for several international conferences, including the IEEE ICME, ICIP, ICASSP, and ISCAS.