A PANORAMIC 3D VIDEO CODING WITH DIRECTIONAL DEPTH AIDED INPAINTING

Muhammad Shahid Farid, Maurizio Lucenteforte, Marco Grangetto

Dipartimento di Informatica, Università degli Studi di Torino
Corso Svizzera 185, 10149 Torino, Italy

ABSTRACT

The success of 3D and free-viewpoint television largely depends on the efficient representation and compression of 3D video in addition to viable rendering methods. This paper presents a novel 3D video coding technique based on the creation of a panorama view to compact the information of a stereoscopic pair. The panorama view represents the information that would be visible to a virtual camera with a larger field of view embracing all the available views. The information in the panorama view is then used to estimate any intermediate view using depth image based rendering. Furthermore, to fill the disocclusions in the reconstructed view a directional depth aided fast marching inpainting technique is presented. The panorama view and corresponding depth map are amenable to standard video compression. In this paper we show that using the novel HEVC standard the proposed 3D video format can be compressed very efficiently.

Index Terms— 3DV coding, depth image based rendering, 3D TV, free viewpoint TV, depth guided inpainting

1. INTRODUCTION

The depth perception in 3D television is provided to the viewer by rendering at least two views captured at slightly different angles. Recently, autostereoscopic displays capable of using several views (up to 64) have been proposed to enable free viewpoint TV, where the viewer can roam around the display by changing his point of view in the scene.

The 3D TV framework clearly requires efficient 3D video representations and coding formats capable of achieving high compression rates. Moreover, these formats shall designed to be backward compatible with legacy 2D devices [1]. To this end, various coding formats have been proposed to efficiently encode multiple views. The multiview extension (MVC) of the widespread AVC standard [2] is considered to be the first efficient mean to encode multiple views. MVC exploits the spatial redundancy among several views and is backward compatible with any AVC decoder. MVC achieved high compression ratio however, few limitations were also observed: the bitrate increases linearly with the number of views [3] and secondly it produces blur effect when used to encode depth maps [4]. Many other 3D video formats emerged, e.g. Depth Enhanced Stereo (DES), Layered Depth Video and Multi View plus Depth (MVD), where pictures are coupled with a per pixel depth map that represents the distance of every pixel from the imaging plane. A good overview of various coding formats may be found in [5].

Video plus depth formats have attracted much attention due to its ability to generate intermediate virtual views through *Depth Image-Based Rendering* (DIBR) techniques [6]. Since 2011 MPEG has issued a call for proposal on 3D video coding technology [7] and in 2012 an ISO/ITU collaborative effort [8] has been started to cope with novel evolutions of the 3D video. These activities are pursuing the standardization of 3D extensions of both AVC and the novel High Efficiency Video Coding (HEVC) [9] aiming at exploiting novel formats and DIBR techniques for compression.

The contribution of this paper is twofold: first, we propose a novel panorama view plus depth (PVD) coding format. The proposed coding tool achieves high compression ratio with good quality and it is backward compatible with existing 2D video decoders. DIBR is exploited to extract any desired intermediate view starting from a single panorama view plus depth picture. Second, a directional depth aided inpainting technique implemented with fast marching method [10, 11] is introduced to fill-in the disocclusions in virtual views.

Recently, many inpainting techniques have been proposed in context of disocclusion filling. Some techniques in the literature aim at filling the holes by maintaining a background sprite. If the background is static (or changing slowly) it can be estimated and exploited to fill the disocclusions [12, 13, 14]. However, the results can be rather poor in presence dynamic and highly textured background. Spatial and temporal information has been exploited to fill the large holes [15], as well. Large holes can be concealed effectively by patch-based inpainting [16]; this class of algorithms can also use depth information to to improve patch selection, e.g. [17, 18, 19]. The drawback of these latter techniques is the high computational cost that may not be compatible with real time display constraint. In this paper we propose a novel inpainting method that at the same time guarantees good visual quality and limits the computational cost exploiting the fast marching method [11].

2. PANORAMA VIEW PLUS DEPTH

Let us consider a standard horizontal multiview setup where all cameras are co-planar with parallel look vectors and the same focal length. Since the distance between adjacent cameras, termed *baseline*, is usually quite limited, the captured scenes are highly redundant. At a very high level one may visualize all the collected information using a single virtual camera with a larger field of view forming what we term the *panorama view*. Unfortunately, because of 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 camera. Nonetheless, in practical settings the panorama view represents a viable approach for representing a set of multiview images compactly. Finally, using DIBR technique one shall be able to re-estimate any intermediate view.

2.1. Building the panorama

To keep the discussion simple and without loss of generality we consider the case of stereoscopic setup with focal length f and baseline b as shown in Fig. 1. Let V_l, V_r be two views of resolution $W \times H$ with corresponding depth maps D_l, D_r respectivly. As graphically represented in Fig. 1, this yields a high overlap among the field of views of the two cameras. While moving from the right to the left view, two types of regions appears. Given the horizontal shift a new portion of the scene s_l shows up on the left hand side of the left view that we refer as *appearing region*. Moreover, dis-occlusions generally appear along the foreground objects that unveil new portions of the background.

We propose to form a panorama view by considering all the acquired pixels (both views) with respect to a single reference view (the right view in the following). Using DIBR we can warp all the pixels of the left view getting an estimate of the right view V_r^l. With horizontal camera setup DIBR amounts at shifting (horizontally) every pixel of the left view by $b \cdot f/z$, being z the pixel's depth; then, proper interpolation is used to map shifted pixels onto integer coordinates of the target right view. Furthermore, during the warping process more than one pixel of V_l may be shifted to the same coordinate (u, v) of V_r^l; in such case only the pixel closest to the camera is copied in the panorama view. In standard DIBR one aims at getting an estimate of the right view starting from the left one. In our case we are only interested in estimating the previously defined appearing region, that is represented by pixels of the left view that are mapped outside the right field of view, i.e. their horizontal coordinate turns to be negative.

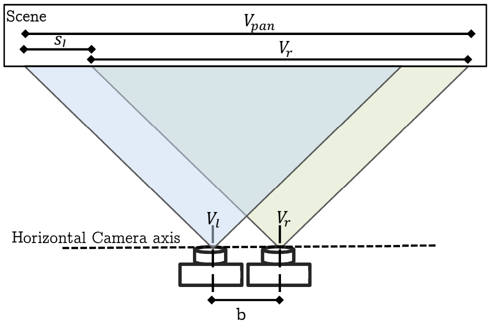

Fig. 1: A two view camera setup with horizontal arrangement.

Since the maximum shift is associated to the pixel with minimum depth, i.e. the one closest to the camera, the appearing region width can be upperbounded by

$$\Delta = \left\lceil \frac{b \times f}{z_n} \right\rceil \tag{1}$$

where z_n is the minimum depth of the left view. Therefore, we define the panorama view as the $W' \times H$ image with $W' = W + \Delta$ computed as:

$$V_{pan}(u, v) = \begin{cases} V_r^l(u, v - \Delta) & \text{if } v < \Delta \\ V_r(u, v - \Delta) & \text{if } \Delta \leq v < W' \end{cases} \tag{2}$$

where $V_{pan}(u, v)$ is a pixel of the panorama view at row $u \in [0, \ldots, H-1]$, and column $v \in [0, W'-1]$, $V_r(u, v)$ is a pixel of the right view and $V_r^l(u, v)$ is an estimated pixel of appearing region (in this case v can be negative as already commented). The first Δ columns of V_{pan} represent the appearing region (outside the field of view of the right camera), whereas the remaining columns simply correspond to the acquired right view. The panorama depth D_{pan} is computed analogously. The panorama construction process is schematically shown in Fig. 2.

The panorama view and depth can be compressed with a standard video encoder. Let us denote as $\hat{V}_{pan}$, $\hat{D}_{pan}$ the decoded view and depth. The right view $\hat{V}_r$ can be obtained from the rightmost W columns of $\hat{V}_{pan}$. Then $\hat{V}_{pan}$ can be warped according to $\hat{D}_{pan}$ to estimate the left view $\hat{V}_l$. The holes in $\hat{V}_l$ are filled through a specialized inpainting algorithm described in the following section. Once, the two views $\hat{V}_r$ and $\hat{V}_l$ and associated depth maps $\hat{D}_r$ and $\hat{D}_l$ are available, one is able to generate any desired intermediate virtual view for autostereoscopic display.

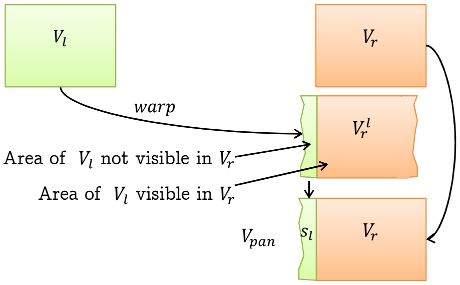
V_l
V_r
warp
V_r^l
Area of V_l not visible in V_r
Area of V_l visible in V_r
V_{pan}
s_l
V_r

Fig. 2: Panorama construction process.

3. DIRECTIONAL DEPTH AIDED INPAINTING

As described in the previous section the left view $\hat{V}_l$ is obtained by warping the panorama view and therefore disocclusions (holes) are likely to appear. In our PVD approach we propose to conceal holes in depth and texture, separately. The depth map holes are recovered with the method proposed in [12] whereas texture is processed with an ad-hoc inpainting algorithm, devised around the following key remarks that hold on in our particular setting:

1. holes are likely to belong to the background uncovered by foreground objects;

2. the right side of a hole it is likely to represent the boundary of a foreground object whereas the left side is likely to lie in the background (this remark is true since we are warping from right to left).

According to the first remark it turns out that it is convenient to fill holes with background information; the second observation tells us that the inpainting result is likely to be more accurate if we employ an interpolation mechanism that propagates the neighboring background moving from left to right. In Fig. 3a the previous concepts are shown: the green region Ω represents a hole of image I (appeared on the left side of the foreground (FG) object), the gray region represents the background (BG), and the red line shows the hole contour that, on the left side, borders the BG.

The proposed inpainting algorithm is inspired by [11], that here we extend taking into account the above observations and the availability of depth information that can be inpainted separately. The authors of [11] have introduced what they termed the fast marching principle that allows one to estimates a missing pixel from the available neighborhood information using a weight function depending on local gradient, level set difference and distance. The hole contour is iteratively propagated into the missing region, i.e. Ω gets inpainted from all directions isotropically. In the context of PVD we propose a novel approach where we put a constraint on the inpainting direction (from left to right) so as to avoid to blur the FG contour into the hole. We further enhance the inpainting performance by substituting the level set difference function with the exploitation of depth information; in particular we prioritize background information in the interpolating function. As shown in Fig. 3b a missing pixel p on the front $\delta\Omega$ is interpolated considering a neighborhood $B_e(p)$ that is a $b \times b$ block including p on its right side. In the following experiments we set $b = 7$. The inpainting mechanism iteratively estimates the missing pixels of the left front $\delta\Omega$. As a consequence, at each iteration, the hole shrinks and the corresponding $\delta\Omega$ moves inward. The pixel p is interpolated as

$$I(p) = \frac{\sum_{q\in B_e(p)} w(p,q)[I(q) + \nabla I(q)\cdot(p-q)]}{\sum_{q\in B_e(p)} w(p,q)} \quad (3)$$

where $\nabla I(q)$ is the gradient of I in q and $w(p,q)$ is a weighting function. Standard vector notation is used where $(p-q)$ is the difference vector and scalar product between two vectors x and y is denoted by $x \cdot y$. The weighting function is computed as

$$w(p,q) = g(p,q)d(p,q)z(q) \quad (4)$$

where $g(\cdot)$ depends on the direction between p and q, $d(p,q)$ is the Euclidean distance between p and q, and $z(q)$ depends on the depth of q. The weighting terms are computes as follows:

$$\begin{aligned} g(p,q) &= \frac{(p-q)\cdot N(p)}{||p-q||}, \\ d(p,q) &= \frac{1}{||p-q||^2}, \\ z(q) &= 1 - \frac{depth(q)}{255} \end{aligned}$$

where $N(p)$ is the normal of the edge in p and $depth(q)$ is the standard representation of the depth map at q, i.e. the inverse of the distance from the imaging plane normalized between 0 and 255. Therefore $z(q) = 0$ when $depth(q) = 255$, i.e. FG and $z(q) = 1$ when $depth(q) = 0$, i.e. BG.

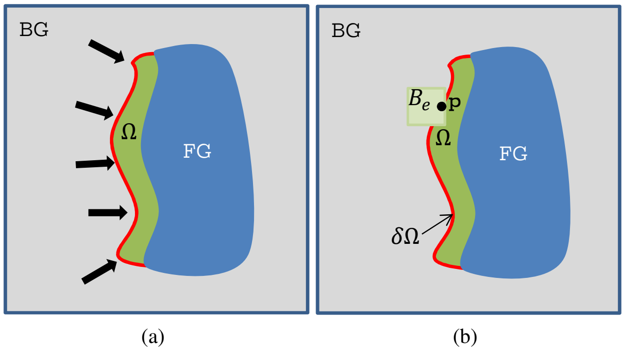

Fig. 3: Hole to be filled (a) and proposed inpainting model (b).

4. CODING EXPERIMENTS AND INPAINTING RESULTS

The proposed PVD format has been evaluated as a possible novel 3D compression friendly approach. To this end the panorama view and depth map have been encoded as separated videos with the novel HEVC standard. The compressed HEVC-PVD bitstream can be received by any standard HEVC decoder. The resulting $\hat{V}_{pan}$ and $\hat{D}_{pan}$ can be post-processed as described in the previous sections to create multiple views that serve as input for the 3D display at hand. Moreover, even legacy 2D service can be provided by simply cropping the rightmost W columns of $\hat{V}_{pan}$.

To measure the HEVC-PVD compression efficiency experiments on standard MPEG MVD sequences have been worked out and compared against HEVC simulcast of video plus depth. Coding results of 2 views and corresponding depth maps are provided using the default random access HEVC settings. Four different quantization parameter values (QP = 26, 30, 34, 38) are used to explore different quality levels. The gain of the proposed HEVC-PVD versus simulcast are evaluated in terms of the well-known Bjøntegaard metric [20] where BD-rate and BD-PSNR measure the bitrate saving and quality difference, respectively (only the PSNR of the decoded right view is taken into account). The obtained results are shown in Tab. 1 and allow to appreciate an average BD-PSNR of 2.12dB and average BD-rate of 47.59%, i.e. only about half the simulcast bitrate is used by HEVC-PVD. The BD-PSNR does not consider the quality of the left view that in the case of PVD must be rendered using DIBR and the proposed novel hole filling technique. It is well known that, because of interpolation and occlusion filling, DIBR does not generally guarantee a pixel-faithful reconstruction; nonetheless, the achieved visual quality can be rather good. As a consequence, to better estimate the visual quality of the reconstructed left view the Structural SIMilarity (SSIM) index, that better correlates with the human visual perception compared to PSNR [21], has been used. In Tab. 1 we also show the average SSIM of the left view obtained with simulcast and PVD-HEVC with $QP = 26$. It can be noted that the penalty of PVD-HEVC is extremely limited, i.e. SSIM loss of 0.02 on average.

To better appreciate the quality of the synthesized left view in Fig. 4 we show both PSNR and SSIM obtained on the first 20 frames of the Poznan_Street sequence. The proposed inpainting method is compared with the results provided by [11]. It can be noted that our modifications to the original approach in [11] are effective in improving the image quality according to both PSNR and SSIM metrics. Moreover the proposed method exhibits a significant advantage also in terms of execution time; in particular, it has turned out that the C implementation of our algorithm is $15 - 20$ times faster than the one of [11]. This significant speedup is mainly due to our usage of a predefined direction for hole filling, compared to the complexity of level set distance computation in [11].

Table 1: BD-PSNR and BD-rate of HEVC-PVD versus simulcast and average SSIM of the decoded left view with $QP = 26$.

Sequence	Frames	Views		SSIM of $\hat{V}_l$		Bjøntegaard	
		L	R	sim.	PVD	BD-PSNR	BD-rate
Poznan_Hall2	200	7	6	0.9906	0.9696	1.14	-45.01
Poznan_Street	250	5	4	0.9919	0.9736	1.77	-49.47
Kendo	300	1	3	0.9914	0.9671	2.76	-48.16
Balloons	300	1	3	0.9920	0.9649	2.70	-46.11
Undo_Dancer	250	1	3	0.9943	0.9832	2.20	-49.22
Average:				**0.9921**	**0.9717**	**2.12**	**-47.59**

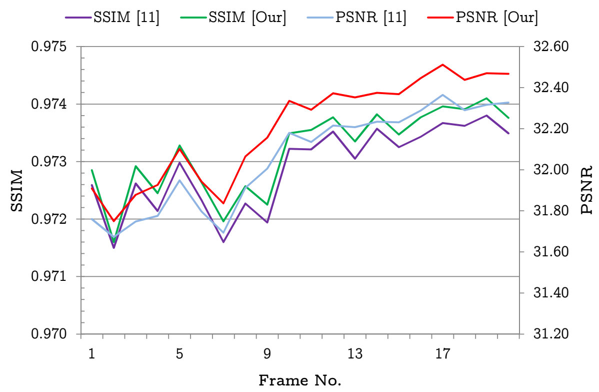

Fig. 4: PSNR and SSIM achieved by proposed inpainting and [11] for the first 20 frames of Poznan_Street sequence ($QP = 26$).

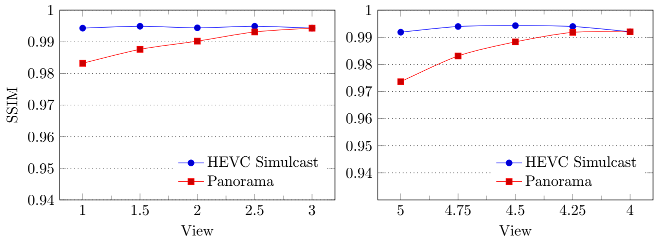

Fig. 5: Average SSIM of coded and interpolated views: Undo_Dancer (left) and Poznan_Street (right) sequence.

Finally, using the left and right views both simulcast and PVD-HEVC have been used to generate 3 intermediate views in order to compare the image quality achieved by the two methods. In Fig. 5 the SSIM of the obtained views is reported as a function of the view index for Poznan_Street and Undo_Dancer coded with $QP = 26$. It can be noted that, as expected, PVD image quality slightly degrades while moving away from the right coded view. Nonetheless, the SSIM penalty is negligible.

5. CONCLUSION

In this paper the novel PVD 3D video format, based on the construction of a panorama view allowing one to compact a stereoscopic pair into a single view, has been proposed. The proposed approach is compression friendly, permit to use DIBR to estimate any desired intermediate views and guarantees backward compatibility with 2D displays by means of simple cropping. Our preliminary coding results obtained using HEVC and a novel disocclusions inpainting technique show that PVD achieves very promising performance.

6. REFERENCES

[1] A. Vetro, A. M. Tourapis, K. Muller, and T. Chen, "3D-TV content storage and transmission," *IEEE Trans. on Broadcasting*, 2011.

[2] 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)*, 2010.

[3] 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.

[4] 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.

[5] 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.

[6] Christoph Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[7] "Call for proposals on 3D video coding technology," Tech. Rep. N12036, ISO/IEC JTC1/SC29/WG11, mar 2011.

[8] "ITU-T/ISO/IEC joint collaborative team on 3D video coding extension development," .

[9] 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.

[10] James A Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

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

[12] 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.

[13] 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.

[14] Chia-Ming Cheng, Shu-Jyuan Lin, and Shang-Hong Lai, "Spatio-temporally consistent novel view synthesis algorithm from video-plus-depth sequences for autostereoscopic displays," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 523–532, 2011.

[15] 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.

[16] 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.

[17] Chia-Ming Cheng, Shu-Jyuan Lin, Shang-Hong Lai, and Jinn-Cherng Yang, "Improved novel view synthesis from depth image with large baseline," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[18] 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.

[19] 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.

[20] Gisle Bjontegard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, 2001.

[21] 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.