REGION-BASED DEPTH MAP CODING USING A 3D SCENE REPRESENTATION

M. Maceira, J.R. Morros, J. Ruiz-Hidalgo

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya (UPC) Barcelona, Spain
{marc.maceira, ramon.morros, j.ruiz}@upc.edu

ABSTRACT

In 3D video, view synthesis is used to process new virtual views between encoded camera views. Errors in the coding of the depth maps introduce geometry inconsistencies in synthesized views. In this paper, a 3D plane representation of the scene is presented which improve the performance of current standard video codecs in the view synthesis domain. Depth maps are segmented into regions without sharp edges and represented with a plane in the 3D world scene coordinates. This 3D representation allows an efficient representation while preserving the 3D characteristics of the scene. Experimental results are provided obtaining gains from 10 to 40 % in bitrate compared to HEVC.

Index Terms— depth map coding, 3dtv, 3D video

1. INTRODUCTION

The increasing development in 3D display devices (3DV) and interactive multimedia systems has led to new applications, such as 3DTV and Free Viewpoint Video (FVV). These techniques have gained interest over the last years [1]. Users can feel real 3D sensations with 3DTV displays or select their desired viewpoint among a certain range in FVV. To deal with the high amount of data required for such 3D systems, new storing formats are required to build practicable systems. Approaches exploiting temporal and inter-view dependencies using solely color images have resulted in the standarization of the Multiview Video Coding (MVC) Extension of H.264 [2].

In order to reduce the number of transmitted views, the texture-plus-depth format has been widely accepted.In this format, color information and depth maps (distance per-pixel samples between the camera and the 3D scene) from several viewpoints are transmitted. Depth maps allow to synthesize virtual views between the encoded viewpoints using depth-image-based rendering (DIBR) techniques [3].

Conventional video compression techniques have been designed to achieve high visual quality. However, in the standard block-based transform, the quantization step creates large artifacts along sharp edges. Depth maps typically consist of homogeneous areas divided by sharp depth discontinuities. Representing depth edges accurately is paramount to achieve a satisfactory view rendering quality. Errors close to a sharp depth transition lead to severe rendering artifacts on the synthesized virtual view, while errors on homogeneous areas may have negligible influence. Moreover, the fact that color images and depth maps are capturing the scene from the same viewpoint leads to a high structural similarity between both images, where the edges in depth are often located in the same location of color discontinuities.

Several approaches for depth map coding have been proposed in the literature (see for e.g. [4, 5]). Some of them approximate the smooth homogeneous areas in depth maps with piecewise-linear functions in a quadtree decomposition [6] while other approaches explicitly encode the position of discontinuities [7, 8]. In [9], an over-segmentation is generated first to find a number of objects in the representation to obtain a progressively-increasing quality in the scene.

The structural similarity between the depth maps and the corresponding color views has been used to better compress depth information, whether to avoid encoding the position of the depth map edges [10, 11, 12] or to jointly code depth and texture [13]. The expected similarities between video and depth map are not always held leading to errors in the rendering process. As depth maps are not directly used for display but to render new images, some authors [4, 14, 15] have proposed to optimize the rate distortion criteria of the synthesized views instead of the depth map directly.

This work proposes a new depth coding technique to compress depth information and prevent coding artifacts along sharp depth discontinuities while encoding efficiently homogeneous areas. For this purpose, the color and depth data are used to build an image partition. This image partition is used to create a 3D plane-based representation of the scene. The idea of representing the geometrical structure of the scene has been used in multiple applications, from object segmentation to scene recognition [16, 17, 18].

Since for 3DV a large number of views are needed, a 3D representation of the scene will allow using a unique representation for the different views. To this goal, the depth map

This work has been developed in the framework of the project BIGGRAPH-TEC2013-43935-R, financed by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF).

values of each region are projected to a set of 3D points. A plane fitting is performed to estimate a plane for each region.

In this work, the goal is to encode a single depth map. Notice that this is a preliminary step to a system where all the depth maps of the multiview sequence will be encoded collaboratively, building a 3D representation capable of representing the different viewpoints thus increasing the potential for further coding gains.

2. PROPOSED CODING SCHEME

The proposed depth map coding technique uses the structural similarities between a color image and the associated depth map. Starting from a 2D segmentation of the depth map, the pixels in each region are projected into the 3D space and encoded using a 3D plane. The plane coefficients are transmitted to the receiver where, given the plane coefficients for each region and the contours of the depth map partition, the 3D planes can be back-projected to the 2D depth map, thus recovering the original signal. As it is not viable to explicitly encode the position of the depth edges due to its high coding cost, the contours obtained from the decoded color image (which we assume has been encoded using the standard *HEVC* and is available at the decoder) are used to approximate the contours of the depth partition.

Assuming that edges in the depth maps are often located in the same location of color discontinuities, a segmentation technique using the contours extracted from the decoded color image can recover the location of most depth edges. While in many cases the assumption is valid, differences between color and depth structure may result in regions that contain depth discontinuities (undersegmentation). These regions can not be properly represented using a 3D plane and will result in coding errors. To solve that, a method which progressively adds depth discontinuities is proposed, providing the depth edges when color and depth discontinuities are inconsistent.

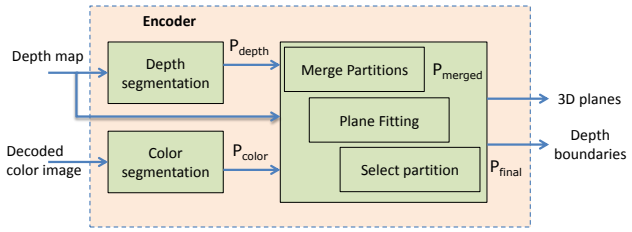


Fig. 1. Encoder scheme. Color image and depth map are used to build two independent partitions. 3D planes are fitted using the color partition and the intersection of both partitions. The depth boundaries that solve the inconsistencies between color and depth segmentations are found in a rate-distortion fashion and sent to the decoder.

The encoder uses both the decoded color image and the depth map to build two independent partitions. The color partition provides most of the depth boundaries and can be built also at the decoder without any extra coding cost. The depth partition contains all the main depth boundaries (including the ones not present at the color partition). As this partition is not available at the decoder, the information of their boundaries have to be sent explicitly. The combined use of color and depth partitions helps recovering all the depth boundaries while reducing the cost of sending the complete depth partition. The encoding process is depicted in figure 1.

Color Segmentation

The color image is used to obtain a base image partition P_{color} (see figure 3 a) with SLICs [19] which gives a initial set of regions from which the 3D planes are estimated. Most of the regions succeed at fitting the points of the region with a plane, however a small subset of them are not able to fit the plane correctly due to segmentation errors in the partition or to inconsistencies between color and depth.

The number of under-segmented regions can be reduced by increasing the number of regions in the color partition. However, the rate needed to encode the planes grows linearly with the number of regions. Moreover, even a highly over-segmented color partition is not able to extract all depth edges. A depth map partition is used to refine the color partition by creating depth edges that are not present in the color partition.

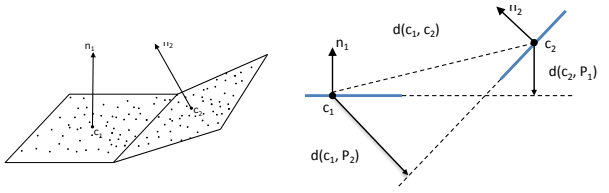
Depth Map segmentation

The depth map partition P_{depth} is obtained with a Binary Partition Tree (BPT) [20] which builds a hierarchical region-based representation by iterative region-merging. Starting from a fine (with a large number of regions) initial partition, the creation of a BPT is based on two major notions: the merging criterion and the region model. The merging criterion is built upon the notion of disimilarity between neighboring regions, thus indicating the order in which regions are to be merged. The region model specifies how regions are represented and how to compute the model of the union of two regions. The BPT algorithm proceeds by merging iteratively pairs of neighboring regions according to the lowest dissimilarity criterion.

Even though P_{depth} is a 2D partition, a 3D planar region model alike to the 3D representation desired for the encoding of the regions is chosen for the BPT. In this model, each region is characterized by the centroid of the 3D points of the region c_i and the normal orientation n_i of the plane, as shown in figure 2 a).

The merging criterion combines two different dissimilarity measures between regions R_1 and R_2: $o_{p(R_1,R_2)}$ and $o_{c(R_1,R_2)}$. The measure $o_{p(R_1,R_2)}$ indicates whether the centroid of one plane is well approximated with the neighboring plane equation.

$$o_{p(R_1,R_2)} = a_1 * d(c_1, P_2) + a_2 * d(c_2, P_1) \qquad (1)$$



a) b)

Fig. 2. a) Region model: plane with normal n_i and centered in c_i. b) Distances in the merging criterion

where a_i is the area of the region in number of pixels, $d(c, P)$ is the euclidean distance between a point c and a plane P.

The measure $o_{c(R_1,R_2)}$ is based on the euclidean distance between centroids and promotes the creation of regions that are closer in the 3D space.

$$o_{c(R_1,R_2)} = \frac{a_1 + a_2}{2} * ||c_1 - c_2|| \quad (2)$$

The total merging criterion is defined as:

$$o_{3d(R_1,R_2)} = o_{p(R_1,R_2)} + o_{c(R_1,R_2)} \quad (3)$$

Figure 2 b) shows a graphical example of the proposed merging criterion.

At each iteration of the merging process, the algorithm selects the pair of regions with the lowest o_{3d}, which correspond to the most similar pair of regions, and merges them into a new region. The BPT algorithm stops when the model representing the new region does not fit properly (the back-projection of the 3D plane into the 2D depth map differs by more than a predefined threshold).

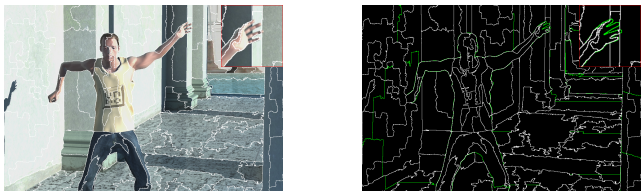
Final partition construction

Both partitions P_{color} and P_{depth} are combined to form a new partition P_{merged}. The P_{merged} partition is built by taking all the P_{color} and P_{depth} boundaries. Figure 3 b) shows the resulting P_{merged}, distinguishing the boundaries from P_{color} and P_{depth}. With the addition of the edges from P_{depth}, the inconsistencies between color and depth map that lead to under-segmentation errors are solved.

The P_{color} and P_{merged} are used to build two 3D scene representations. Each representation is formed by 3D plane fitted for each region using RANSAC [21].

While adding edges from P_{depth} removes under-segmentation and thus, reduces the coding distortion, the opposite problem may arise: the number of regions of P_{merged} may be too large, which will increase coding cost. To avoid that, new regions in P_{merged} are classified according to the distortion reduction that results when adding the corresponding region boundary to P_{color}. In this case, distortion is measured in the 3D space as the mean square distance between the plane and the region points.

Different rate-distortion points are obtained by progressively adding region boundaries to P_{color} until the budget rate for this image is reached, resulting in the final partition P_{final}. These added region boundaries should be also encoded (a lossless Freeman Chain-Code technique [22] is used) and sent to the decoder.



a) b)

Fig. 3. Partitions combination. a) Color partition. b) Merged partition: contours from the color partition depicted in white; depth partition contours depicted in green.

The process at the decoder is as follows: P_{color} is built as done in the encoder. Then, P_{final} can be constructed by decoding the additional boundaries and adding them to the P_{color} partition. The decoded depth map image is obtained by projecting the 3D planes to each corresponding region.

3D plane coding

The planes modeling each region in P_{final} can be represented using a) the plane normal and b) the distance from the plane to the camera. Each of the 3 components of the plane normal are encoded by means of an uniform quantizer using N_{orient} bits. The distance from the plane to the camera is converted to an alternative quantized representation using:

$$C_{dist} = \frac{1.0}{\frac{Dist(pe,c)}{(2^{N_{dist}}-1} * (\frac{1.0}{MinZ} - \frac{1.0}{MaxZ}) + \frac{1.0}{MaxZ}} \quad (4)$$

where $Dist(pl, c)$ is the distance between the region plane and the camera, N_{dist} is the number of bits to be used in the quantization and $MaxZ$ and $MinZ$ are the maximum and minimum depth values of the image.

Figure 4 shows the resulting dequantized planes projected onto the decoded depth map image. In figure 4.a) only the P_{color} partition is used while at 4.b) the P_{final} is used. Using directly P_{color} results in depth discontinuities inside the regions that lead to poorly fitted 3D planes and in high errors in the decoded depth map. However, the P_{final} partition corrects under-segmentation errors that correspond to a more efficient representation in the decoded depth map (see for instance the detail in the hand).

3. EXPERIMENTAL RESULTS

The proposed coding method is evaluated using 10 images of the 3D multiview sequences sets *ballet* and *Undo Dancer*. For

a) b)

Fig. 4. 3D planes coding example

each sequence, three views are used, the left and right views are encoded and the middle one is employed as the location for the virtual view. The color image in the virtual view is available, thus the performance of the depth map coding technique can be compared with the original color image in the virtual view.

To objectively evaluate the method proposed, error measures are taken in the synthesized virtual view. For each frame of the sequence, the virtual view is synthesized using the original depth maps, the depth maps coded with the proposed method and the intra mode of *HEVC*. The color images for the synthesized process are encoded using the same quality for all the experiments.

The average peak signal-to-noise ratio (PSNR) is computed with the image obtained using the original depth maps. The comparison with the virtual image generated using the original depth maps allows to establish the rate distortion performance of the method when using different methods to compress the depth map. Figure 5 shows the comparison with the image synthesized using the original depth maps. The vertical axis shows the average PSNR of the synthesized virtual view, while the horizontal corresponds to the bitrate in bits per pixel to encode the depth maps. The method proposed is able to obtain better rate distortion efficiency than the *HEVC* for low bitrates.

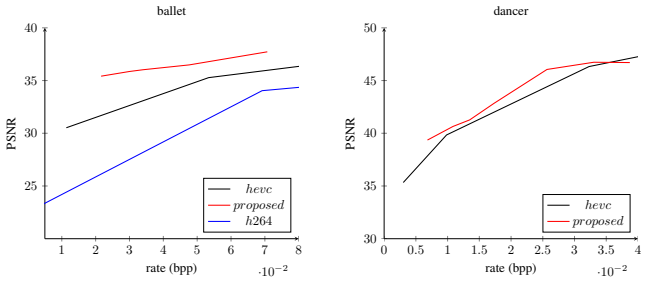


Fig. 5. Rate Distortion over virtual view

Additionally, as the 3D representation proposed is an alternative support to depth maps, rate-distortion curves are provided taking as a reference the original view in the virtual view location 6. the proposed method and the *HEVC*, the PSNR obtained using the original depth maps is displayed. Since the 3D representation is based on the data of the original depth maps, the PSNR using the original depth maps is the upper bound for our method. Comparing the results with the ones in figure 5, it can be seen that the proposed method achieve better performance when measuring PSNR between the computed virtual views and the ground truth image rather than when comparing with the virtual view generated with the original depth map. This means that by using the color segmentation as a base partition for the 3D representation, the planes estimated are able to solve original inconsistencies in the depth map, obtaining a better performance than *HEVC* which is unaware of the color transitions.

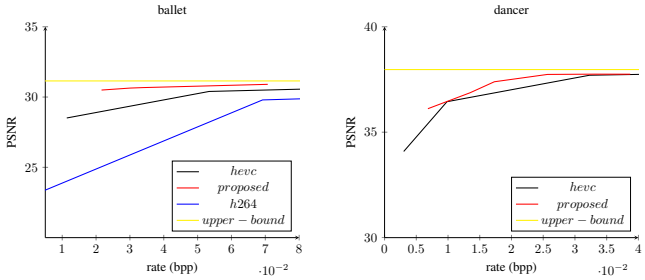


Fig. 6. Rate Distortion over original color view

In table 3 the Bjontegaard metric for the sequences used are shown. The proposed method obtains a rate reduction of 40% in the sequence ballet and 10% for dancer over HEVC for low bitrates. While comparison with other techniques with the same environment can not be provided here due to lack of code available, in [4] average gains between 7 and 22% are achieved over H.264 in the view synthesized image.

	Virtual View		Color original	
	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate
Ballet	1.52	-39.99	0.59	-23.81
Dancer	0.52	-10.03	0.11	-27.35

4. CONCLUSION

This paper presents a new depth map coding method that uses a color images segmentation to build a 3D plane representation of the scene. The two major contributions of this paper are: 1) a depth map segmentation technique that allows to jointly segment the depth map in homogeneous regions and represent those regions with 3D planes and 2) the generation of a 3D model that can recover the structure of the scene while coding the depth map efficiently. Results are provided showing the performance of the method against the standard *HEVC*, obtaining better performance specially at lower bitrates. As a further work, the proposed method could be extended to a multiview scenario, where multiple depth maps can be encoded with a unique 3D representation.

5. REFERENCES

[1] P. Merkle et al., "Multi-view video plus depth representation and coding," in *ICIP*, Oct 2007.

[2] A. Vetro, T. Wiegand, and G.J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, April 2011.

[3] Christoph Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," *Proc. SPIE*, vol. 5291, pp. 93–104, 2004.

[4] Feng Shao, Weisi Lin, Gangyi Jiang, Mei Yu, and Qionghai Dai, "Depth map coding for view synthesis based on distortion analyses," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 4, no. 1, pp. 106–117, March 2014.

[5] Ismael Daribo, Gene Cheung, and Dinei Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 30 2012-oct. 3 2012, pp. 1541 –1544.

[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P.H.N. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *3DTV Conference*, May 2008, pp. 245–248.

[7] F. Jager, "Contour-based segmentation and coding for depth map compression," in *VCIP 2011*, Nov. 2011, pp. 1–4.

[8] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, Jaejoon Lee, and Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding," in *PCS, 2010*, Dec 2010, pp. 566–569.

[9] S. Milani and G. Calvagno, "A depth image coder based on progressive silhouettes," *Signal Processing Letters, IEEE*, vol. 17, no. 8, pp. 711–714, Aug 2010.

[10] M. Maceira, J. Ruiz-Hidalgo, and J.R. Morros, "Depth map coding based on a optimal hierarchical region representation," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, oct. 2012, pp. 1 –4.

[11] Huiping Deng, Li Yu, Jinbo Qiu, and Juntao Zhang, "A joint texture/depth edge-directed up-sampling algorithm for depth map coding," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, july 2012, pp. 646 –650.

[12] Shujie Liu, PoLin Lai, Dong Tian, and Chang Wen Chen, "New depth coding techniques with utilization of corresponding video," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 551–561, June 2011.

[13] Jun Zhang, M.M. Hannuksela, and Houqiang Li, "Joint multiview video plus depth coding," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 2865–2868.

[14] G. Tech, H. Schwarz, K. Muller, and T. Wiegand, "3d video coding using the synthesized view distortion change," in *Picture Coding Symposium (PCS), 2012*, May 2012, pp. 25–28.

[15] Yun Zhang, Sam Kwong, Long Xu, Sudeng Hu, Gangyi Jiang, and C.-C.J. Kuo, "Regional bit allocation and rate distortion optimization for multiview depth video coding with view synthesis distortion model," *Image Processing, IEEE Transactions on*, vol. 22, no. 9, pp. 3497–3512, Sept 2013.

[16] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *CVPR, 2010 IEEE Conference on*, June 2010, pp. 1418–1425.

[17] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from rgb-d images," in *CVPR, 2013 IEEE Conference on*, June 2013, pp. 564–571.

[18] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and T. Garaas, "Tracking an rgb-d camera using points and planes," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, Dec 2013, pp. 51–58.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274 –2282, nov. 2012.

[20] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Trans. on IP*, vol. 9, no. 4, pp. 561–576, Apr. 2000.

[21] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[22] H. Freeman, "On the coding of arbitrary geometric configurations," *IRE Trans. Electronic, Comp.*, p. EC(10):260268, June 1961.