Noname manuscript No.
(will be inserted by the editor)

Depth map compression via 3D region-based representation

Marc Maceira · Josep-Ramon Morros · Javier Ruiz-Hidalgo

Received: date / Accepted: date

Abstract In 3D video, view synthesis is used to create 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 new 3D plane representation of the scene is presented which improves the performance of current standard video codecs in the view synthesis domain. Two image segmentation algorithms are proposed for generating a color and depth segmentation. Using both partitions, depth maps are segmented into regions without sharp discontinuities without having to explicitly signal all depth edges. The resulting regions are represented using a planar model in the 3D world scene. This 3D representation allows an efficient encoding while preserving the 3D characteristics of the scene. The 3D planes open up the possibility to code multiview images with a unique representation.

Keywords Depth map coding · 3D representation · image segmentation · data compression

Marc Maceira Duch
EDIFICI D5 DESPATX 120 C. JORDI GIRONA, 1-3 BARCELONA SPAIN
Tel.: +34-93-4011627
E-mail: marc.maceira@upc.edu

Josep-Ramon Morros
EDIFICI D5 DESPATX 008 C. JORDI GIRONA, 1-3 BARCELONA SPAIN
Tel.: +34-93-4015765
E-mail: ramon.morros@upc.edu

Javier Ruiz-Hidalgo
EDIFICI D5 DESPATX 008 C. JORDI GIRONA, 1-3 BARCELONA SPAIN
Tel.: +34-93-4015765
E-mail: j.ruiz@upc.edu

1 Introduction

The extension of current visual displays and systems to the third dimension (3D) aims to convey depth perception to the viewer. Many applications exploiting 3D video have arisen over the last years, such as 3D video games, 3D films (IMAX cinemas) or medical imaging (SPECT). With the increasing development of 3D display devices and interactive multimedia systems, 3D video has gained interest in the last decades for acquisition, display and compression purposes.

There are two different methodologies for bringing 3D video to television devices: 3D Television (3DTV) [6] and Free Viewpoint Video (FVV) [38]. 3DTV produces depth perception creating 3D scenes in movement. The viewer perceives the 3D video from a single static position in the space. On the other hand, FVV allows the user to interactively control the viewpoint by generating different views of a dynamic scene from any position in the 3D space. By selecting at any moment the position from which the scene is displayed, the viewer participates of the creative process by focusing more in one subset of the available content or another.

The information required for 3D video applications results in a massive amount of data that has to be stored and transmitted. Therefore, an efficient compression method is needed to design feasible systems. Different 3D scene representation formats have been proposed for compression purposes [38] such as multiview video [41], depth information related to each view [22] and 3D meshes [30].

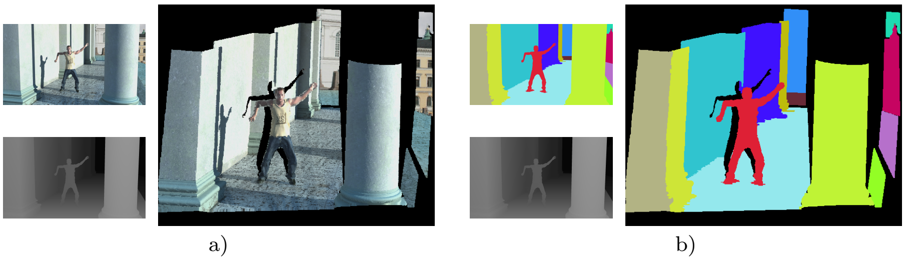


Fig. 1 a) The point cloud associated with the depth map can be recovered using the camera parameters. In the figure, each point is represented with the corresponding color of the color image. b) An image partition is used to build a 3D model for all the points in the region.

Different 3D video systems can be classified depending on the number of cameras used. The most widespread technique is the stereographic camera configuration. Classical stereo video is built from two viewpoints located at the distance of human eyes creating the perception of depth. New configurations can include up to 15 cameras.

The 3D geometry of the scene jointly with the camera parameters allow to relate the different viewpoints through inter-view prediction, similarly as the forecast of two consecutive images in a video. As images obtained with multiview sets are very similar to one another, one can easily predict one of the views from another. A coding gain can be achieved in comparison to the coding of a single view depending on the degree of common content shared by a subset of the cameras. Approaches exploiting temporal and inter-view resemblances resulted in the standardization of the Multiview Video Coding (MVC) Extension of H.264 [41]. MVC provides up to 40% bit rate reduction for multiview data in comparison to single-view H.264 coding [21] while providing the user good subjective 3D perception. However, the bit rate resulting from MVC is linearly proportional to the number of coded views [26].

In order to reduce the number of transmitted views, the format Multiview Plus Depth (MVD) has become popular in the last few years [21]. MVD coding format is created by sending a per-pixel depth map associated to each viewpoint as can be seen in figure 1. The depth map consists of a grayscale image where the depth can take values between the maximum and the minimum distance to the camera position. The value of the depth is quantized with 8 bits with the points closer to the camera having values near 255 and the furthest near 0. Depth map information can be back-projected to the 3D world enabling encoders to establish relations between views. Using color images and depth maps, the decoder is capable to synthesize virtual views through Depth-Image-Based Rendering (DIBR) techniques (see for e.g. [8]).

The main contribution of this work is the formulation of a new depth map coding technique for compression purposes. The goal is to prevent coding artifacts along sharp depth discontinuities while efficiently encoding homogeneous areas. The considered compression scheme relies on the choice of an adequate image segmentation method. This work proposes two algorithms to independently segment the color image and the depth data. The color data is available in the decoder and can be used to obtain a partition without any extra cost. This color partition contains the depth edges when color and depth edges are located in the same position. The color and depth partitions are then combined to obtain a final partition that properly segments the depth map. From this partition, a new 3D plane-based representation is introduced to store the scene structure. Figure 1 illustrates the generation of the 3D representation using the camera parameters of the viewpoint and an image partition.

This paper is organized as follows. Section 2 gives an overview of the existing depth map coding techniques. The proposed depth coding algorithm is described in Section 3. Section 4 is devoted to experimental results. Conclusions and perspectives are drawn in Section 5.

2 Related work

Depth data present a set of characteristics that diverge from color images. Typically, they are composed of large smooth regions separated by sharp depth transitions. Classical video compression techniques for color images have been designed to achieve high visual quality. For this purpose, the image is divided in blocks and each block is coded using a transform and a quantization step [28]. However, the direct application of these techniques on depth maps leads to coding artifacts in the edges due to quantization. An accurate representation of the sharp edges is capital for the DIBR process. Since it generates virtual views, the depth transitions allow to separate regions of the image with different inter-view motion vectors. Quantization-based artifacts near the edges introduce severe rendering artifacts in synthesized virtual views [23].

Many techniques aimed at preserving the depth map edges while reducing the cost of coding the smooth regions have been presented in the depth map coding literature. Some approaches explicitly encode the position of the most significant depth edges (see for e.g. [15]). The main depth contours are signaled and the in-between texture is encoded with piecewise-linear functions. In [5], two different modes are proposed to signal the depth edges depending of their complexity over one unified framework. Then, the simplest transform is chosen for each block. Instead of explicitly representing the depth maps boundaries, [35] proposes encoding the residual prediction errors with quantization at pixel domain rather than in the transform domain. The coding of prediction errors in the spatial domain is also used in [24]. In addition, new intra-picture prediction modes based on geometric primitives are described. They allow the prediction of depth lying in the same plane than the previous blocks.

Since depth maps are not directly displayed but used to render new images, the usual rate-distortion criteria over the depth map may not give a proper measure of the quality of the representation. To this end, it is preferable to optimize the rate-distortion criteria over the synthesized views rather than over the depth map directly [16]. The main advantage of modeling the coding error on the synthesized view instead of calculating it on the depth map is that the impact of the coding errors can be determined in the generated virtual view. Moreover, in [40], a new distortion metric is proposed to measure the influence of depth errors in the synthesized virtual view.

The fact that color images and depth maps capture the scene from the same viewpoint leads to a high structural similarity between both images. The edges in depth are often located in the same location of color discontinuities. In order to avoid signaling the explicit location of depth edges, this similarity between depth maps and the corresponding texture image can be used. This strategy is used in [18], where skip-coding mode and motion vectors in the coded texture are used in the depth map.

A color image segmentation is proposed in [25] to predict the shape of the different surfaces in the depth map. Then, each region is approximated either by a parameterized plane or by the standard H.264/AVC Intra coder. The color segmentation in [25] is used to extract the flat areas of the image, while inaccuracies between color and depth, and the problems in the segmentation are encoded with the H.264 coder. Our paper deals also with the segmentation of the color images to create an approximation of the depth map segmentation. However, instead of using a different coder in the problematic areas, we propose to improve the obtained color partition with a partition obtained from the depth image.

Moreover, in [36], the inter-view redundancy is removed by an analysis performed over the occluded areas. In the second view, only new areas which were occluded in the first view are coded. This inter-view redundancy can be removed by extracting the geometrical structure of the scene. 3D representations have been widely used in multiple applications, from object segmentation to scene recognition [3, 11, 12]. In [17], a multiview object co-segmentation method is proposed that estimates a depth map with a set of 3D planar surfaces.

In depth map coding, new proposals have explored the reduction of the redundancy in the multiview image sets. In [20], a geometrical representation is introduced to describe the multiview information with a graph. Starting from an initial view, inter-view redundancy is avoided by adding new graph nodes only if new information appears in the subsequent views. Similarly, a novel 3D video coding technique based on the creation of a panorama view is detailed in [7]. This view represents most of the visual information acquired from multiple views using a single virtual view characterized by a larger field of view.

Interestingly, some approaches take advantage of the correlation between color view and depth, to jointly encode both signals. The High Efficiency Video Coding (*HEVC*) standard [39] has an extension (*3D-HEVC*) to encode multiple views and associated depth map [27]. In this extension, in addition to the inter-view motion and residual predictions, new intra coding modes are included to handle the depth edges. *HEVC* systems increase the computational complexity of previous standards. To mitigate that, recent implementations of *HEVC* are able to reduce the encoding time while maintaining the coding efficiency, parallelizing among multiple processors [43, 44] or skipping some modes [37].

A 3D planar representation of stereo MVD was proposed in [29]. Using a rate-distortion like procedure, a co-segmentation between the two views and a 3D planar approximation for each region of the image is found. Two different compression techniques are explored, the first one follows the two-view structure while the second fuses the data in a single-reference MVD format. Following a similar strategy, our work deals with a 3D representation of the scene, which presents a much larger number of segments than the one proposed in [29]. Results in the same dataset are presented and discussed in the experimental section.

3 Depth map coding proposed technique

The proposed depth map coding technique uses an image partition to fit a 3D plane for each region. Starting from a depth map segmentation, the pixels belonging to each region are projected into the 3D space and then, encoded using a 3D plane. These 3D planes are able to represent the smooth texture of depth maps with few coefficients. Furthermore, encoding the region in the 3D domain opens the possibility to use this representation for multiple views in a future work. Given the 3D plane coefficients for each region and the final coding partition, the decoder can project the 3D planes back to the 2D depth map, recovering the original signal. Besides the color image, the camera model of the viewpoint is required to recover the 3D points of the scene. Both color images and camera model are used in the DIBR process. Thus, no extra information needs to be sent to the decoder.

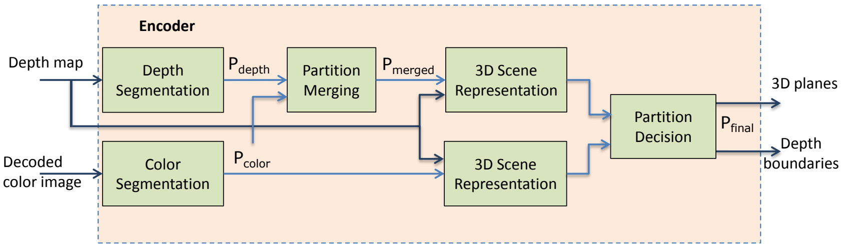


Fig. 2 Encoder scheme. Color image and depth map are used to build two independent partitions. 3D planes are fitted using the color partition and the intersection of both partitions. The depth boundaries that solve the inconsistencies between color and depth segmentations are found in a rate-distortion fashion and sent to the decoder.

Assuming that most depth edges in the depth maps are located in the same position as color discontinuities, a segmentation technique using the decoded color image allows to recover most of these depth edges. While this assumption is generally valid, differences between color and depth structure may result in regions that contain depth discontinuities (under-segmentation). These regions can not be properly represented using a 3D plane and will lead to coding errors. In order to prevent these errors, a method that progressively adds depth discontinuities is proposed. The location of depth edges, which have to be encoded and sent to the decoder, are included when color and depth discontinuities are inconsistent. Combining the color partition and the depth edges, the decoder can obtain a new partition reproducing the structure of the depth map without explicitly encoding the position of all the depth edges.

The encoding process is depicted in figure 2. The encoder uses both the decoded color image and the original unencoded depth map to build two partitions with the methods explained in sections 3.1 and 3.2. 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 has 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.

3.1 Color image segmentation

The objective of the color segmentation is to obtain a P_{color} partition similar to superpixel partitions in the literature [1, 4]. Superpixels provide a useful representation of the image with a reduced number of entities with respect to the pixel representation. The P_{color} segmentation will be used as a base partition for obtaining the 3D planes representation. Thus, it will determine the minimum rate needed to encode the set of planes. To be competitive with *HEVC* encoders, experimental results have shown that P_{color} has to contain a few hundred regions. At that number of regions, superpixels techniques fail at retrieving accurately the discontinuities in the image. To overcome that, a superpixel inspired technique is proposed, which obtains higher boundary retrieval.

Superpixels techniques do a clustering process using the color similarities and spatial proximity. We propose here to use these two characteristics in a region merging algorithm. The color segmentation technique proposed in this work is based on the Binary Partition Tree (BPT) described in [32]: Starting from an initial partition with an arbitrary number of regions, the algorithm proceeds iteratively by merging two neighboring regions according to a similarity measure (merging criterion) as depicted in figure 3. The steps in each merging step are the following:

- computing a similarity measure for each pair of neighbor regions
- selecting the most similar pair of regions and merging them into a new region
- updating the neighborhood and the similarity measures. The algorithm iterates until the desired number of regions is obtained.

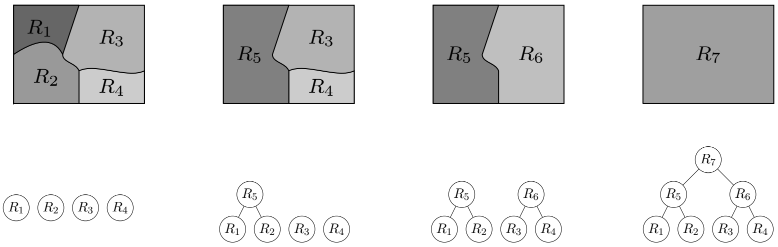


Fig. 3 From left to right, the two most-similar neighboring regions are merged at each step. The hierarchical representation is depicted as a tree, where the region formed by merging two segments is represented as the parent of the respective nodes

The similarity measure proposed is derived from the *bpt_nwmc* in [42]. In *bpt_nwmc*, the region model defined is constant for all the pixels of the region and the model M_R is obtained by averaging the values of the pixels $p \in R$, in the YCbCr color space:

$$M_R = \frac{1}{N_R} \sum_{p \in R} I(p) \tag{1}$$

where N_R is the number of pixels of region R.

The *bpt_nwmc* criterion consists of two terms: The first one, based on color similarity, is the Weighted Euclidean Distance between Models (*wedm*) which compares the models of the original regions, R_1 and R_2, with the model of the region obtained after the merging $R_1 \cup R_2$:

$$O_{wedm}(R_1, R_2) = N_{R_1}||M_{R_1} - M_{R_1 \cup R_2}||_2 + N_{R_2}||M_{R_2} - M_{R_1 \cup R_2}||_2 \quad (2)$$

The second term is related to the contour complexity of the merged regions. The measure computes the increase in perimeter $\mathit{\Delta} P(R_1, R_2)$ of the new region with respect to the largest of the two merged regions: $\mathit{\Delta} P(R_1, R_2)$. The term that measures contour complexity is:

$$O_{Cont}(R_1, R_2) = max(0, \mathit{\Delta} P(R_1, R_2)) \quad (3)$$

The contour term promotes the creation of smooth contours between regions. Since most objects are regular and compact (that is, tend to have simple contours), the analysis of shape complexity can provide additional information for the mergings.

Color and contour similarity measures are linearly combined to form the *bpt_nwmc* criterion:

$$O_{bpt\_nwmc}(R_1, R_2) = \alpha\, O_{wedm}(R_1, R_2) + (1 - \alpha)\, O_{Cont}(R_1, R_2) \quad (4)$$

The *bpt_nwmc* criterion creates color homogeneous regions with smooth contours, but tends to create elongated regions. As the regions will be used for fitting a 3D plane, more compact regions are desirable. To this end, a new term which measures the spatial proximity is added based on the distance between region centroids. The centroid of a region is defined as:

$$Cent(R_1) = \frac{1}{N_{R_1}} \sum_{p \in R_1} Coord(p) \quad (5)$$

where *Coord* are the coordinates of the pixels p in the region R_1. The O_{Cent} is defined as the euclidean distance d between centroids:

$$O_{Cent}(R_1, R_2) = d\,(\,Cent(R_1),\, Cent(R_2)) \quad (6)$$

The proposed O_{Cent} criterion is combined with the O_{bpt_nwmc} to form the O_{bpt_spx} superpixel criterion:

$$O_{bpt\_spx}(R_1, R_2) = \beta\; O_{Cent}(R_1, R_2) + (1 - \beta)\, O_{bpt\_nwmc}(R_1, R_2) \quad (7)$$

After testing different weights for α and β, we found that the weight factors in equations (4) and (7) do not affect severely the coding performance. For simplicity, the three terms are combined with an addition, creating the superpixels criterion (8):

$$O_{bpt\_spx}(R_1, R_2) = O_{Cent}(R_1, R_2) + O_{wedm}(R_1, R_2) + O_{Cont}(R_1, R_2) \quad (8)$$

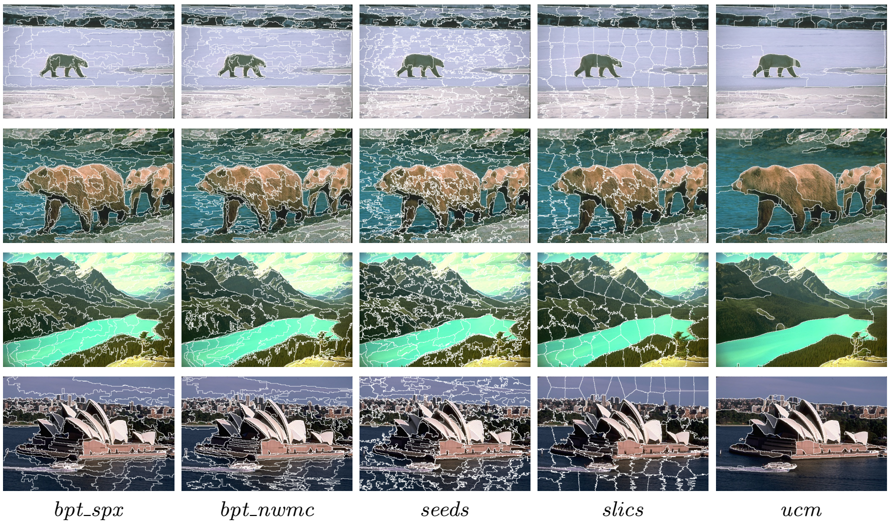


Fig. 4 Visual comparison between color segmentation techniques.

As stated before, the number of regions in the color partition N_{color}^{regs} fixes the minimum rate for the depth coding method as is the minimum number of planes encoded. Thus, N_{color}^{regs} is set as a fraction of the maximum rate, reserving the remainder for the creation of new regions with the contours from the depth segmentation. The O_{bpt_spx} criterion builds the hierarchy until N_{color}^{regs} regions are obtained. This number is sent to the decoder which is able to replicate the same hierarchy as done in the encoder. Visual results for the different methods are shown in figure 4.

3.2 Depth map segmentation

As a complement to the color segmentation, a depth map segmentation P_{depth} is needed to provide the main depth edges that are missing in the color segmentation. The objective here is to find a depth partition able to represent the depth map image with the lower number of regions while extracting the maximum number of depth edges.

The depth map partition is created in two steps. The first is done using the region growing algorithm in [31]. Depth values are projected to the 3D space to form a point cloud and then, an automatic segmentation [31] obtains smoothly connected areas. It computes the local surface normals and uses the point connectivity to join 3D points that have the same orientation.

Depth maps used in multiview scenarios are often noisy and quantized into 8 bits. To avoid discontinuities in the surface normals which stops the [31] algorithm, an initial superpixel over-segmentation of the depth map is performed as an input for the region growing. This severely over-segmented partition (∼10000 regions) allows removing the noisy and quantization errors while capturing the scene with a sufficient density to apply the region growing algorithm. Using that initial over-segmentation, the 3D point cloud is generated by computing the centroid of all the points in the region and projecting them to the 3D world using the mean value of all the pixels in the region.

The partition obtained after the region growing step is able to recover the structure of the scene but still has some small-sized regions which have not been merged as shown in figure 6.a). To find the final partition, the bpt algorithm [42] is used creating a new region model and merging criterion.

A 3D planar region model alike to the 3D representation desired for the encoding of the regions is chosen as a model for the merging process. 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 5.a).

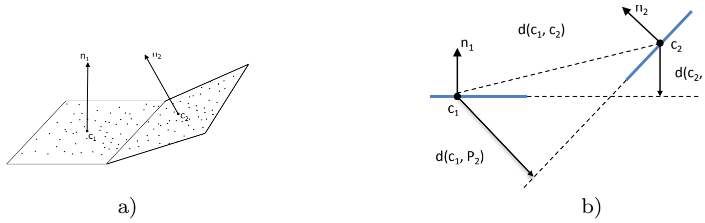


Fig. 5 a) Region model: plane with normal n$_i$ and centered in c$_i$. b) Distances in the merging criterion

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.

$$o_p(R_1, R_2) = a_1 * d(c_1, P_2) + a_2 * d(c_2, P_1) \tag{9}$$

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.

$$o_c(R_1, R_2) = \frac{a_1 + a_2}{2} * d(c_1, c_2) \tag{10}$$

The final merging criterion is defined as:

$$o_{3d-bpt}(R_1, R_2) = o_p(R_1, R_2) + o_c(R_1, R_2) \tag{11}$$

Figure 5.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-bpt}, 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). An example of final partition obtained with the algorithm is depicted in figure 6.b).

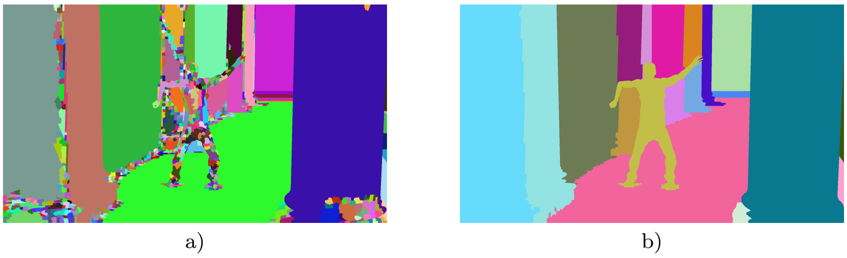


Fig. 6 Depth Map Partition process. a) Result of applying the region growing algorithm to the superpixels partition. b) Final coding partition after the *3d_bpt* algorithm.

3.3 Partition merging

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. To ensure the creation of meaningful regions, only new regions that are larger than a certain size are created. Figure 7.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. Notice that some non-meaningful boundaries are added in this step which will be removed in the *partition decision* step.

3.4 3D scene representation

In order to obtain the 3D plane coefficients in the bitstream, each region in P_{color} and P_{merged} is represented by fitting a 3D plane using RANSAC [9]. Each 3D plane is represented using the distance from the plane to the camera and the plane orientation. The distance from the plane to the camera is converted to an alternative quantized representation using the distance to depth map conversion:

$$C_{dist} = \frac{1.0}{\frac{d(pl,c)}{(2^{N_{dist}}-1)} * (\frac{1.0}{MinZ} - \frac{1.0}{MaxZ}) + \frac{1.0}{MaxZ}} \tag{12}$$

where $d(pl,c)$ is the euclidean 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.

The plane orientation is stored in spherical coordinates with their 3D angles θ and ϕ:

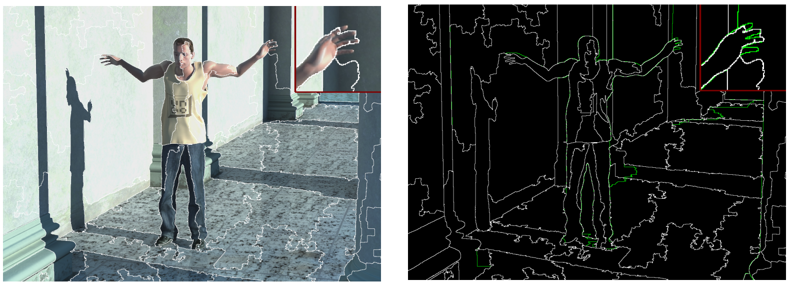
$$\theta = arccos\left(\frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}}\right) \tag{13}$$

$$\phi = arctan\left(\frac{n_y}{n_x}\right) \tag{14}$$

The n_z component is pointed towards $z > 0$. The resulting angles have the following dynamic range: $0 \leq \theta \leq \frac{\pi}{2}$ and $0 \leq \phi \leq 2\pi$. Each angle is encoded with equal precision with a uniform quantizer.

3.5 Partition decision

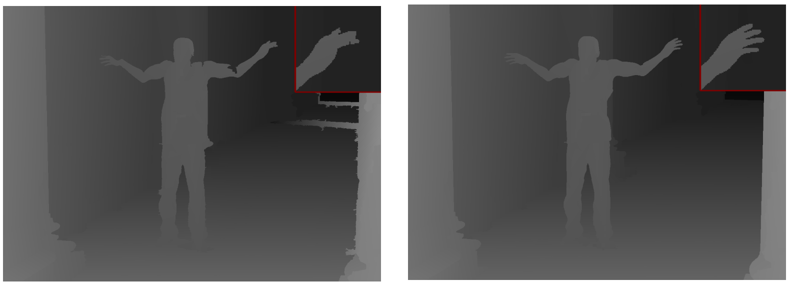
While adding edges from P_{depth} removes under-segmentation and thus, reduces the coding distortion, the opposite problem may arise: the number of regions of P_{merged} may be too large, which will increase coding cost. New contours added in the P_{color} have to be sent to the decoder, which rapidly surpasses the cost of coding the texture. To achieve the budget rate, the method controls how the new boundaries are added, prioritizing the boundaries that have a larger impact to the coding of the depth map. 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.



a) P_{color} b) P_{merged}

Fig. 7 Partition merging. Contours from the color partition depicted in white; depth partition contours depicted in green.

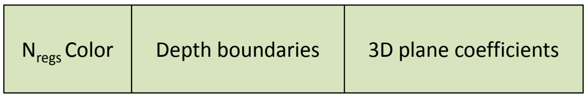
Figure 8 shows the resulting reconstructed planes projected onto the decoded depth map image. In figure 8.a) only the P_{color} partition is used while at 8.b) the P_{final} is used. Using directly P_{color} results in depth discontinuities inside the regions that lead to poorly fitted 3D planes and 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).



a) P_{color} b) P_{final}

Fig. 8 Partition decisision coding example. 3D planes coding example using P_{color} and P_{final}

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 [10] is used) and sent to the decoder. The bitstream sent to the decoder is depicted in figure 9.



N_{regs} Color	Depth boundaries	3D plane coefficients

Fig. 9 Bitstream containing the number of regions for the color image, the depth boundaries coded with chain-code and the 3D plane coefficients

3.6 Decoder scheme

The bitstream in figure 9 is decoded with the scheme shown in figure 10. Using the number of regions for the color partition, the decoder is able to build P_{color} as done in the encoder without any added cost. Then, P_{final} can be recovered 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.

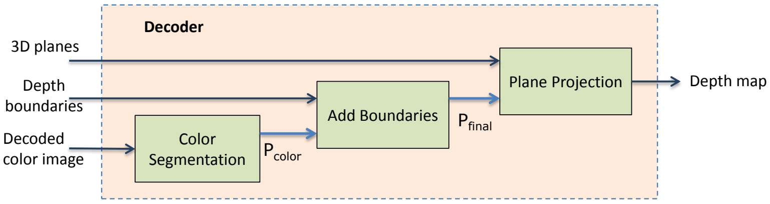


Fig. 10 Decoder scheme. By adding the transmitted depth boundaries to the color partition the decoder obtains the final coding partition.

4 Experimental results

The different stages of the depth coding scheme proposed are evaluated in this section. Firstly, the segmentation methods proposed for the color and depth partitions are evaluated separately in sections 4.1.1 and 4.1.2. Secondly, the different design parameters are discussed in section 4.2. Finally, the complete coding scheme is compared against *H.264*, *HEVC*, *3D-HEVC* and *MV-HEVC* and the similar state of the art method in [29].

4.1 Segmentation evaluation

4.1.1 Color Image Segmentation results

The color image segmentation evaluation is performed against other superpixels segmentations to validate the use of the new centroid distance. The benchmark for this work consist of the BSDS500 dataset [19] which contains 500 images with human-marked-boundaries as ground-truth. The proposed *bpt_spx* method is compared with the bpt merging criterion *bpt_nwmc* [42], with two state of the art super-pixel methods *seeds* [4] and *slics* [1] and with the ultrametric contour map (*ucm*) [2] that gives a hierarchical structure similar as the one derived with the BPT are provided.

Figure 4 shows visual results of the methods compared in this work. The regions generated with the proposed method present smoother contours than the *bpt_nwmc* method due to the centroid term. This term promotes compactness in the first stages of the hierarchical segmentation which leads to smoother contours. *Slics* segmentation recovers even simpler contours achieving segmentations with less false boundaries at the cost of losing also some meaningful contours. The objective of the proposed method differs from the one of *ucm* since, on the one hand, the *ucm* partition aims to obtain a partition that represent the objects of the scene with minimal regions while, on the other hand, our intention is to achieve a superpixel representation of the scene suitable for coding proposes.

Figure 11 shows numerical results in terms of precision, recall and F-measure for boundaries between segmentation and ground-truth. In the pro-

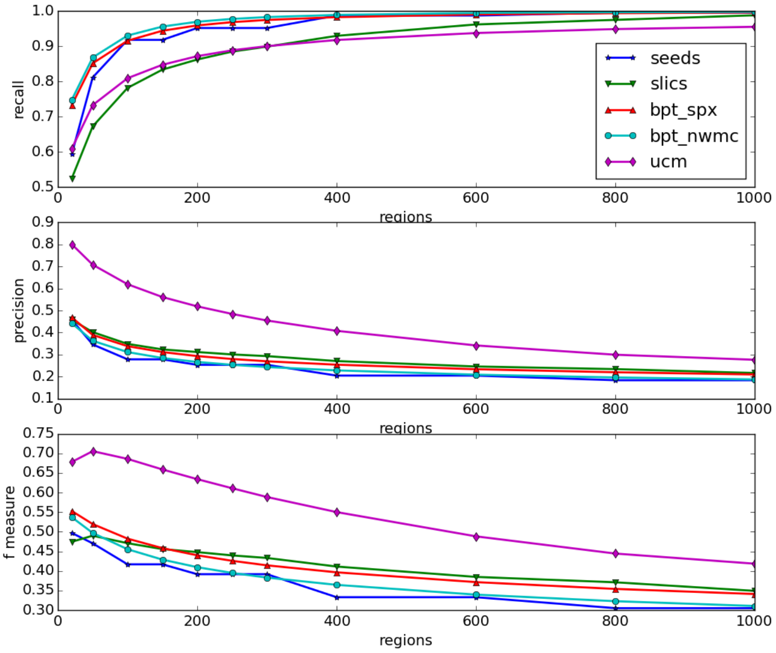


Fig. 11 Evaluation of different methods for the color segmentation. Precision, recall and F-measure depending on the number of regions

posed *bpt_spx* method, the use of the centroid slightly decreases the recall with respect to the *bpt_nwmc* criteria but the precision is improved, obtaining a result similar to the segmentation obtained with *slics*. Globally the usage of the centroid criteria achieves a better trade-off between superpixel compactness and boundary adherence than *bpt_nwmc*. The loss of precision in the boundaries is compensated in the depth coding technique proposed in this work by the depth map segmentation.

The F-measure of the proposed *bpt_spx* results are comparable to *slics*, but as the main objective of the color segmentation is to procure the maximum number of contours, a higher recall is desired. Since the *ucm* representation promotes a representation where each region is meaningful, the number of boundaries that are not from the object is lower, obtaining a much higher precision figure. On the other hand, the *ucm* obtains smooth contours which occasionally are slightly displaced from the ground-truth, leading to lower recall measure. Moreover, the high computational requirements of the *ucm* make their use costly for a video coding scheme.

4.1.2 Depth Map Segmentation results

The depth map segmentation proposed is evaluated with 25 depth maps from the MVD sequences *dancer*, *balloons*, *kendo*, *breakdancers* and *ballet*. Results generated with the proposed scheme are compared against the segmentation produced with *rgbd-ucm* [12] and with *ucm* using only the depth image. For *rgbd-ucm*, a hierarchical segmentation is generated using color and depth clues. Also, the proposed 3D merging process is computed with (*3d-bpt-rgrow*) and without (*3d-bpt*), the region growing stage presented in 3.2.

Figure 12 shows the partitions obtained with the different methods. By using the color image in addition of the depth image, the *rgbd-ucm* generally is able to represent the foreground objects with more regions. Despite that, when the depth map is noisy the *rgbd-ucm* fails at obtaining the main depth edges as can be seen in the balloons image in figure 12. Using the standard *ucm* on the depth image this over-segmentation is reduced.

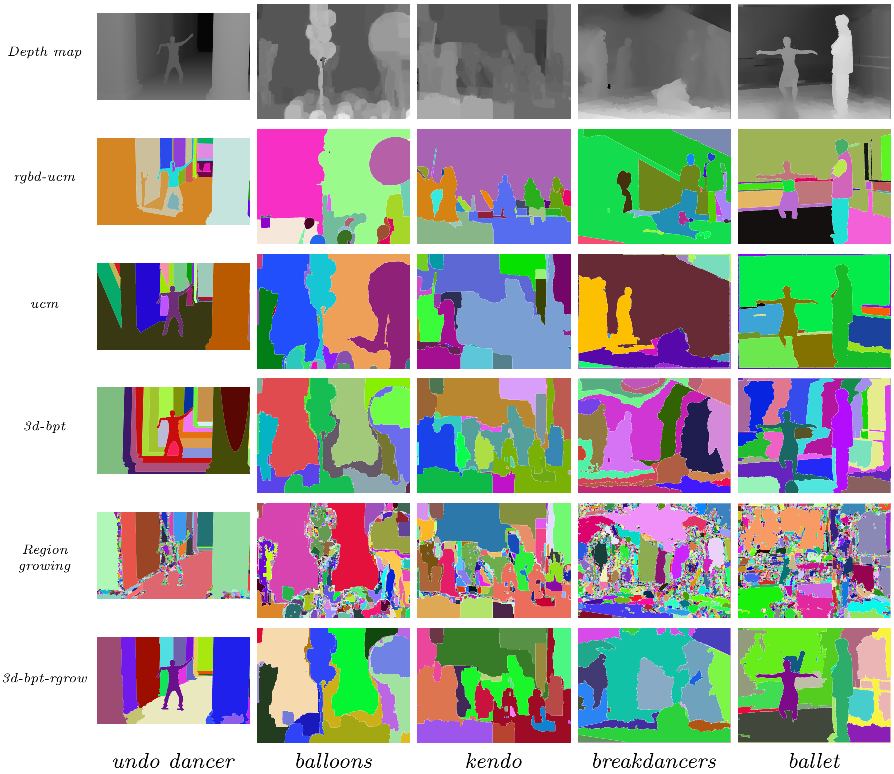


Fig. 12 Visual comparison between the depth segmentation techniques. In row ascendent order: Depth map to segment, *rgbd-ucm*, *ucm*, *3d-bpt*, Output of Region growing segmentation stage and *3d-bpt-rgrow*

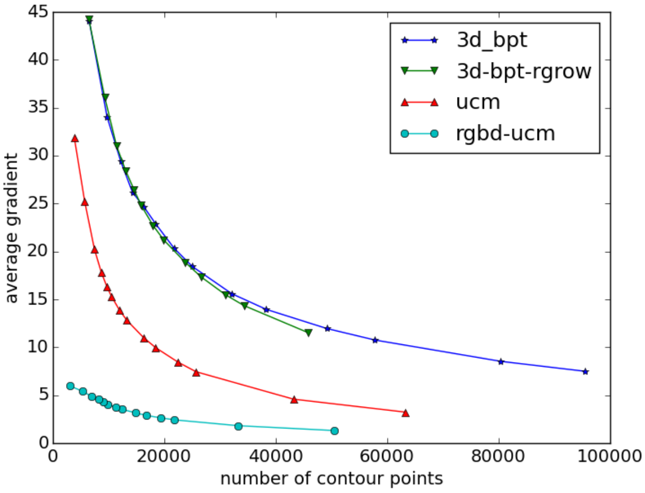


Fig. 13 Gradient measure in function of the number of contour points in the segmentation

The *3d-bpt* obtains a representation of the scene where objects in the same depth are correctly separated but has problems at recovering the overall structure of the scene as can be seen in the *undo dancer* image in figure 12 where regions are created at increasing depths values, joining the walls and the floor. The region growing stage creates an initial segmentation where the flat areas of the scene are joined in a unique region. From the region growing segmentation, the o_{3d-bpt} criterion merges the regions in non-smooth areas with the best 3D plane model.

The evaluation of the different depth segmentation techniques is computed using a gradient measure in the contour points. The purpose of the depth segmentation is to provide the main depth edges of the depth map with the minimum contour points. The maximum directional gradient (horizontal or vertical) is computed for each contour point and then averaged. This measure is computed at different cuts of the hierarchy. Notice that this metric is helpful to determine if areas at different depth distances are in different regions. However it cannot measure the areas where there is not a depth edge but a change of the orientation, as the joints between floor and walls.

In figure 13, the average results for the different sequences are shown. The *3d-bpt* and the *3d-bpt-rgrow* obtain better results than the *rgbd-ucm* and *ucm*. In the two *ucm* options the contours obtained are smooth, losing some meaningful depth boundaries. The *rgbd-ucm* obtains even lower gradient measure since it uses also the color image to generate the hierarchical partition.

While the results are similar for the two bpt options, the *3d-bpt-rgrow* is selected by its better 3D scene reconstruction. The separation of walls and floor is desirable from a conceptual point of view more than the edges that are in the middle of them.

From the previous results, we conclude that the proposed segmentation techniques *bpt_spx* for color and *3d-bpt-rgrow* for depth are good choices for a coding framework.

In the following sections the full depth coding technique is compared with state of the art methods using the segmentation methods proposed.

4.2 Configuration

The proposed method uses the color partition to avoid sending all the depth contours. Figure 14 shows the averaged cost of sending the texture and the contour information for the sequences *dancer*, *balloons*, *kendo*, *breakdancers* and *ballet*. The starting point corresponds to use only the P_{color}, thus the rate for the partition is 0. The number of regions in P_{color} is determined according to the budget rate. Adding an increasing number of contours increments the rate for both the contour and the texture, since new regions are created which result into new 3D planes. The contour cost grows at a higher pace. Notice that, in the last rate-distortion point, the rate employed for texture has increased by a half of the starting rate while the contour cost is more than 5 times larger.

Since the cost of adding new boundaries rapidly surpasses the initial texture cost, it is compulsory to add only the boundaries that improve greatly the distortion figure. This behavior has motivated the proposed method, which only adds new boundaries in regions where the distortion is reduced heavily. With this methodology, the increased rate is employed solely in regions where the P_{color} has problems representing the depth map.

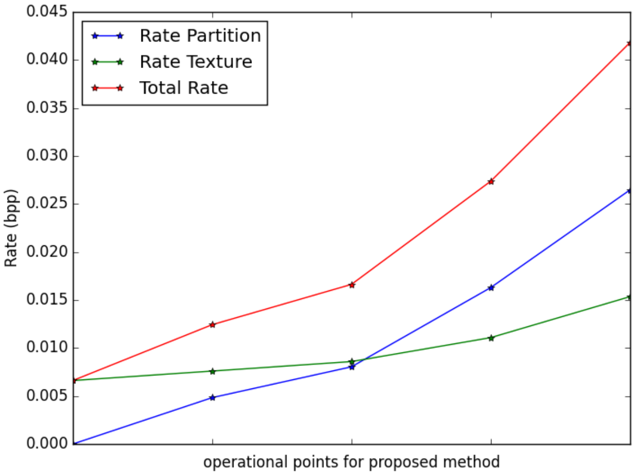


Fig. 14 Comparative between the rate employed for coding the texture and the contour for different rate distortion points obtained with the proposed method

4.3 Coding performance

4.3.1 Multiview sequences

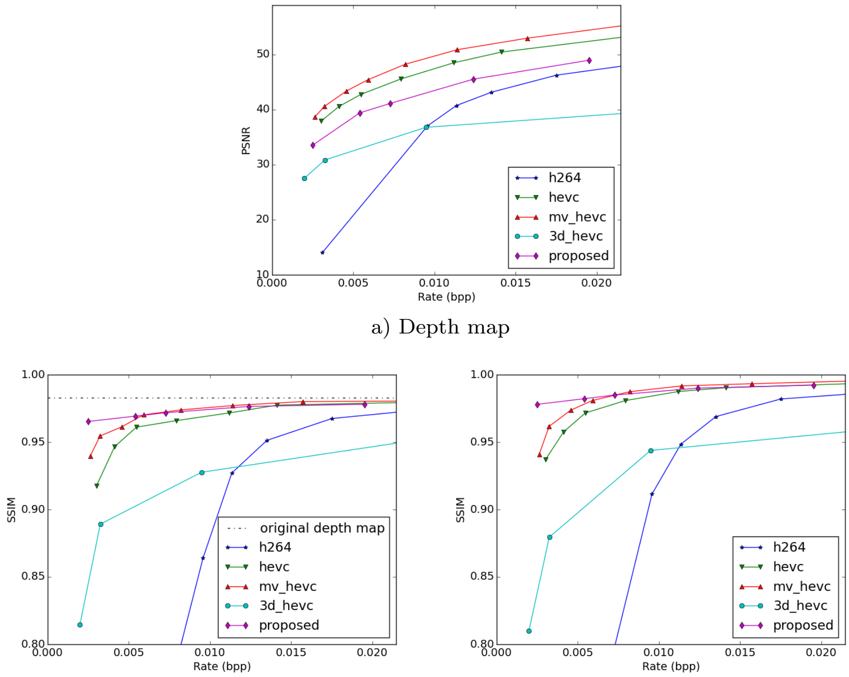
The proposed coding method is evaluated using 10 frames of the 3D multiview sequence sets *undo dancer*, *ballet*, *kendo*, *breakdancers* and *balloons*. For each sequence, three views are used, the left and right views are encoded and the middle one is employed as the location for the virtual view. The color image at the position of 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. As the proposed method does not have temporal prediction, only intra modes for the different methods are taken.

To objectively evaluate the proposed method, error measures are taken both in the depth map and in the synthesized virtual view. The valuable measure is in the virtual view but measuring directly on the depth map gives an overview of how good can the original depth map be represented with planes. The PSNR measure is taken to evaluate the error in the depth map directly and the results for the different sequences are shown in the top row of figures 15 16 17 18 19. In that comparison, the *3D-HEVC* performs worse than the other methods of the literature. Since color and depth map for two views are encoded altogether in *3D-HEVC*, the view synthesis optimization maximize the quality in the virtual view and not directly in the depth map. Our method also is in that category and the results in depth map are below the other methods.

To compare the results in the virtual view, for each frame of the sequence, the virtual view is synthesized using the original depth maps and the decoded depth maps (using the proposed method and the intra mode of *H.264*, *HEVC*, *3D-HEVC* and *MV-HEVC*). The color images for the synthesized process are encoded using the same quality for all the experiments. Notice that *3D-HEVC* encodes color and depth altogether but, since we are comparing solely the depth coding part, only the coded depth are used in the comparison.

The average structural similarity (SSIM) index is the measure used in the virtual view domain since it correlates better with subjective tests [13]. The comparison is performed using the encoded depth maps against the synthesized color image obtained using the original depth maps (*Rendered view*) and against the original color image of the sequence (*Original view*). 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.

Figures 15 16 17 18 19 show the rate distortion results for the sequences evaluated. The vertical axis shows the average SSIM of the synthesized virtual view, while the horizontal axis corresponds to the bitrate needed to encode the depth maps. The proposed method is able to obtain better rate distortion efficiency than the *HEVC* for low bitrates in the sequences *undo dancer*, *ballet*, *breakdancers*. For *kendo*, and *balloons* the result obtained is not as good as their depth maps are noisy and the planarity assumption does not hold. In

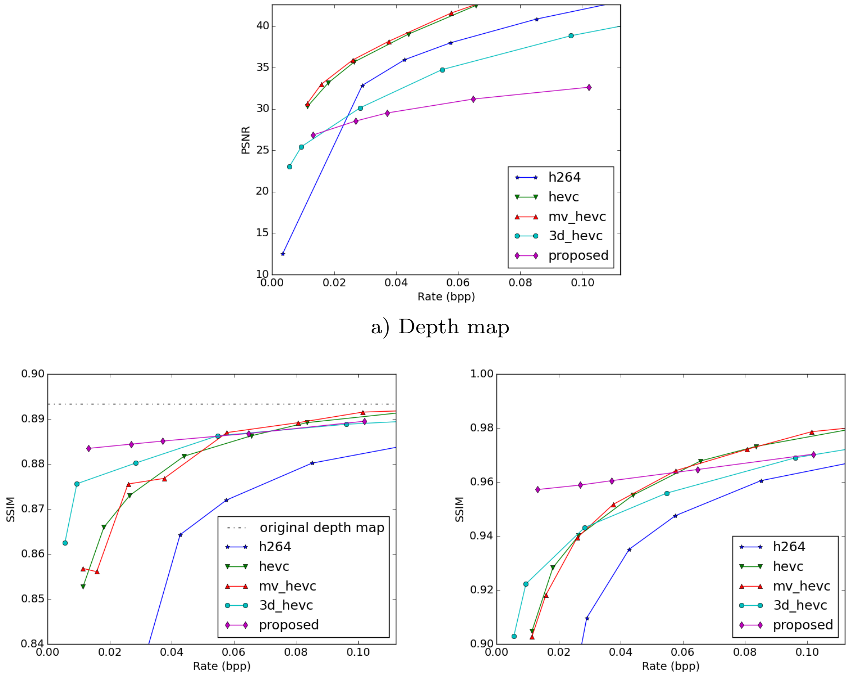


a) Depth map

b) Original view

c) Rendered view

Fig. 15 Rate distortion results for *undo dancer*.

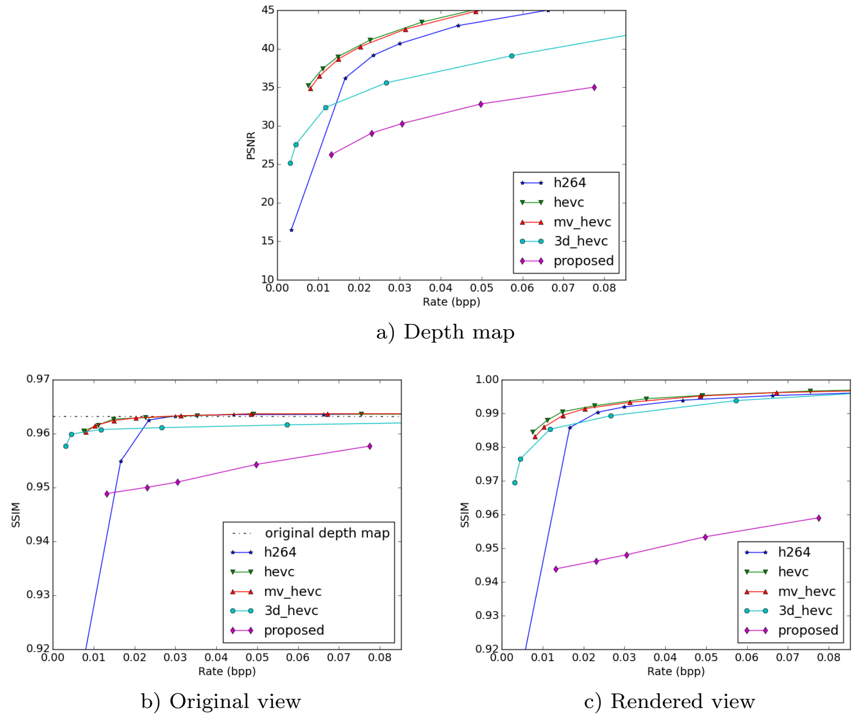


a) Depth map

b) Original view

c) Rendered view

Fig. 16 Rate distortion results for *ballet*.

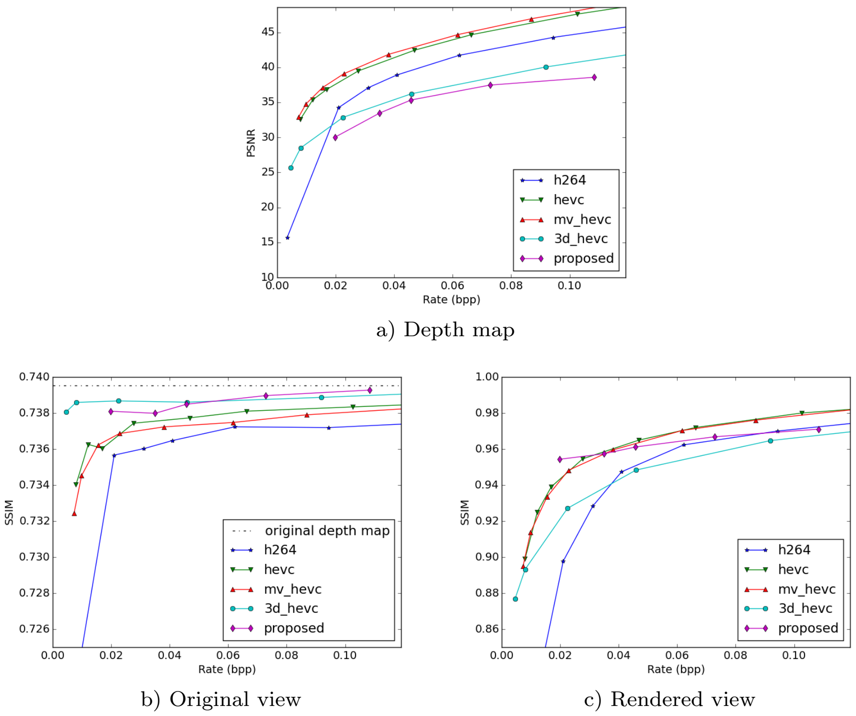


a) Depth map

b) Original view

c) Rendered view

Fig. 17 Rate distortion results for *kendo*.

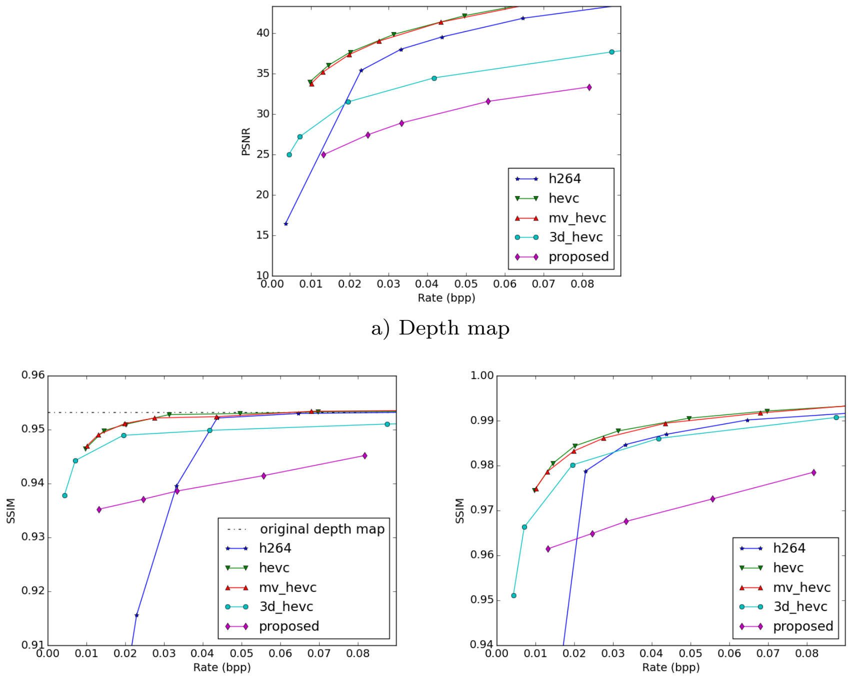


a) Depth map

b) Original view

c) Rendered view

Fig. 18 Rate distortion results for *breakdancers*.



a) Depth map

b) Original view

c) Rendered view

Fig. 19 Rate distortion results for *balloons*.

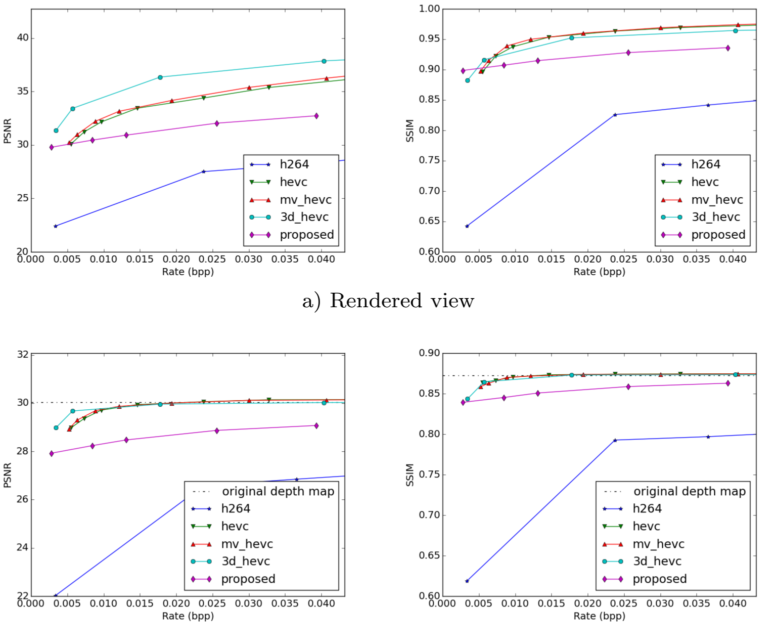
those sequences, the depth boundaries are not well defined and often are not in correspondence with color edges. Fitting planes in those areas results in a poorly 3D estimated representations.

Furthermore, it can be seen that the proposed method achieves better performance when measuring SSIM in the original view rather than in the virtual view. This means that, by using the color segmentation as a base partition for the 3D representation, the estimated planes are able to solve original inconsistencies in the depth map, obtaining a better performance than *HEVC* which is unaware of color transitions.

4.3.2 Results in Middlebury dataset

In order to compare against a state of the art method [29], results on the Middlebury Dataset were generated. The Middlebury Stereo Dataset [34] consists of many stereo images with ground-truth disparities between several viewpoints. It is widely used as a database to evaluate different methods of computing disparities. The datasets chosen from the website are the 2003 [34], 2005 [33] and 2006 [14] which were the ones used in [29]. Each sequence contains color information from several viewpoints and disparity for two of them.

These ground-truth disparities have some unknown values which have been filled using the same in-painting method than [29] to have a fair comparison between the methods. The images are cropped to a multiple of 8 in order to be able to be encoded with the *H.264* and *HEVC* encoders.



a) Rendered view

b) Original view

Fig. 20 Results obtained for the middlebury dataset

The results obtained in the full dataset at maximum resolution are shown in figure 20. In the first row, the rendered image obtained with the depth maps coding using different methods is compared with the rendered image using the original depth map. The proposed method obtains PSNR and SSIM values comparable to *HEVC* for low bitrates. In the original view comparison, the values for the different methods saturate at 30 dB which is the maximum quality achievable with the given in-painted depth maps. Is worth noting that for the two comparisons the proposed method achieves better results evaluated with SSIM rather than PSNR. Also, the *3D-HEVC* method is clearly the best in PSNR over rendered views, since it uses the encoder to find the best rate-distortion points for each block in terms of PSNR. Despite that, when comparing with SSIM, the proposed method obtains similar results than other *HEVC* configurations.

Similar to [29], the performance of the proposed method varies a lot depending on the characteristics of the depth map. In [29] the optimization process is done using the two depth images and their performance depends on the sharpness of the depth contours. Here we rely also in the color segmentation to perform the depth coding. For the most heavily textured color images, the color segmentation process has problems of under-segmentation. Adding depth boundaries progressively, the main under-segmentation problems in the color image are solved but increasing the coding cost. On the other hand, for easier to segment color sequences, the performance of our method overcomes the *HEVC* standards. A direct comparison with [29] is difficult because they provide results just on 6 single selected images. We do prefer to provide averaged results over the full dataset. While not shown in figure 20, our results obtained in single views are comparable to their proposed MVD method. In [29], they also propose a unique representation combining both views, which results in increased performance, this encourages us to work towards obtaining a multiview representation for our method, able to increase the coding efficiency when coding multiple views altogether.

5 Conclusion

In this work we have presented a new depth map coding technique based on segmentation techniques. The two main contributions are two new segmentations algorithms and a planar 3D scene representation. Two image segmentation algorithms have been proposed for generating the color and depth partitions independently. Comparing with different state of the art segmentation methods, we show the benefits of using the proposed methods for depth map compression. The proposed depth map coding technique combines the color partition and the depth map partition to obtain the final coding partition that properly segments the depth map without having to encode all depth edges. A new 3D planar representation that models the scene structure is introduced. The proposed coding method shows competitive results against current standards and state of the art encoders. By representing the views in the 3D space, we open the possibility to use a common, single 3D representation for all the views, which may result in further coding gains. This will be explored in future work.

Acknowledgements 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).

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11):2274–2282, DOI 10.1109/TPAMI.2012.120
2. Arbelaez P, Maire M, Fowlkes C, Malik J (2009) From contours to regions: An empirical evaluation. In: IEEE Conference on Computer

Vision and Pattern Recognition, Miami, USA, pp 2294–2301, DOI 10.1109/CVPR.2009.5206707

3. Ataer-Cansizoglu E, Taguchi Y, Ramalingam S, Garaas T (2013) Tracking an RGB-D camera using points and planes. In: IEEE International Conference on Computer Vision Workshops, Sydney, Australia, pp 51–58, DOI 10.1109/ICCVW.2013.14
4. Bergh M, Boix X, Roig G, Capitani B, Gool L (2012) SEEDS: Superpixels extracted via energy-driven sampling. In: European Conference on Computer Vision, Lecture Notes in Computer Science, vol 7578, pp 13–26, DOI 10.1007/978-3-642-33786-4_2
5. Cheung G, Kim WS, Ortega A, Ishida J, Kubota A (2011) Depth map coding using graph based transform and transform domain sparsification. In: International Workshop on Multimedia Signal Processing, pp 1–6, DOI 10.1109/MMSP.2011.6093810
6. Dodgson N (2005) Autostereoscopic 3D displays. Computer 38(8):31–36, DOI 10.1109/MC.2005.252
7. Farid M, Lucenteforte M, Grangetto M (2015) Panorama view with spatiotemporal occlusion compensation for 3D video coding. IEEE Transactions on Image Processing 24(1):205–219, DOI 10.1109/TIP.2014.2374533
8. Fehn C (2004) Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In: Proc. SPIE 5291, Stereoscopic Displays and Virtual Reality Systems, pp 93–104, DOI 10.1117/12.524762
9. Fischler M, Bolles R (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6):381–395, DOI 10.1145/358669.358692
10. Freeman H (1961) On the coding of arbitrary geometric configurations. IRE Transactions on Electronic Computers EC-10:260–268, DOI 10.1109/TEC.1961.5219197
11. Gallup D, Frahm JM, Pollefeys M (2010) Piecewise planar and non-planar stereo for urban scene reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, USA, pp 1418–1425, DOI 10.1109/CVPR.2010.5539804
12. Gupta S, Arbelaez P, Malik J (2013) Perceptual organization and recognition of indoor scenes from RGB-D images. In: IEEE Conference on Computer Vision and Pattern Recognition, Portland, USA, pp 564–571, DOI 10.1109/CVPR.2013.79
13. Hanhart P, Ebrahimi T (2012) Quality assessment of a stereo pair formed from decoded and synthesized views using objective metrics. In: 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp 1–4, DOI 10.1109/3DTV.2012.6365478
14. Hirschmuller H, Scharstein D (2007) Evaluation of cost functions for stereo matching. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8, DOI 10.1109/CVPR.2007.383248

15. Jager F (2011) Contour-based segmentation and coding for depth map compression. In: Visual Communications and Image Processing, pp 1–4, DOI 10.1109/VCIP.2011.6115989
16. Kim WS, Ortega A, Lai P, Tian D (2015) Depth map coding optimization using rendered view distortion for 3D video coding. IEEE Transactions on Image Processing 24(11):3534–3545, DOI 10.1109/TIP.2015.2447737
17. Kowdle A, Sinha S, Szeliski R (2012) Multiple view object cosegmentation using appearance and stereo cues. In: European Conference on Computer Vision, Firenze, Italy, pp 789–803, DOI 10.1007/978-3-642-33715-4_57
18. Lei J, Li S, Zhu C, Sun M, Hou C (2015) Depth coding based on depth-texture motion and structure similarities. IEEE Transactions on Circuits and Systems for Video Technology 25(2):275–286, DOI 10.1109/TCSVT.2014.2335471
19. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: International Conference on Computer Vision, Vancouver, Canada, vol 2, pp 416–423, DOI 10.1109/ICCV.2001.937655
20. Maugey T, Ortega A, Frossard P (2015) Graph-based representation for multiview image geometry. IEEE Transactions on Image Processing 24(5):1573–1586, DOI 10.1109/TIP.2015.2400817
21. Merkle P, Smolic A, Muller K, Wiegand T (2007) Efficient prediction structures for multiview video coding. IEEE Transactions on Circuits and Systems for Video Technology 17(11):1461–1473, DOI 10.1109/TCSVT.2007.903665
22. Merkle P, Smolic A, Muller K, Wiegand T (2007) Multi-view video plus depth representation and coding. In: IEEE International Conference on Image Processing, San Antonio, USA, vol 1, pp 201–204, DOI 10.1109/ICIP.2007.4378926
23. Merkle P, Morvan Y, Smolic A, Farin D, Muller K, de With P, Wiegand T (2008) The effect of depth compression on multiview rendering quality. In: 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp 245–248, DOI 10.1109/3DTV.2008.4547854
24. Merkle P, Muller K, Marpe D, Wiegand T (2015) Depth intra coding for 3D video based on geometric primitives. IEEE Transactions on Circuits and Systems for Video Technology PP(99):1–1, DOI 10.1109/TCSVT.2015.2407791
25. Milani S, Zanuttigh P, Zamarin M, Forchhammer S (2011) Efficient depth map compression exploiting segmented color data. In: IEEE International Conference on Multimedia and Expo, pp 1–6, DOI 10.1109/ICME.2011.6011969
26. Muller K, Merkle P, Wiegand T (2011) 3-D video representation using depth maps. Proceedings of the IEEE 99(4):643–656, DOI 10.1109/JPROC.2010.2091090
27. Muller K, Schwarz H, Marpe D, Bartnik C, Bosse S, Brust H, Hinz T, Lakshman H, Merkle P, Rhee F, Tech G, Winken M, Wiegand T

(2013) 3D high-efficiency video coding for multi-view video and depth data. IEEE Transactions on Image Processing 22(9):3366–3378, DOI 10.1109/TIP.2013.2264820

28. Ostermann J, Bormans J, List P, Marpe D, Narroschke M, Pereira F, Stockhammer T, Wedi T (2004) Video coding with H.264/AVC: tools, performance, and complexity. IEEE Circuits and Systems Magazine 4(1):7–28, DOI 10.1109/MCAS.2004.1286980
29. Ozkalayci B, Alatan A (2014) 3D planar representation of stereo depth images for 3DTV applications. IEEE Transactions on Image Processing 23(12):5222–5232, DOI 10.1109/TIP.2014.2360452
30. Peng J, Kim CS, Jay Kuo CC (2005) Technologies for 3D mesh compression: A survey. Journal of Visual Communication and Image Representation 16(6):688–733, DOI 10.1016/j.jvcir.2005.03.001
31. Rabbani T, van den Heuvel FA, Vosselman G (2006) Segmentation of point clouds using smoothness constraint. In: ISPRS Commission V Symposium 'Image Engineering and Vision Metrology', pp 248–253
32. Salembier P, Garrido L (2000) Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing 9(4):561–576, DOI 10.1109/83.841934
33. Scharstein D, Pal C (2007) Learning conditional random fields for stereo. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8, DOI 10.1109/CVPR.2007.383191
34. Scharstein D, Szeliski R (2003) High-accuracy stereo depth maps using structured light. In: IEEE Conference on Computer Vision and Pattern Recognition, vol 1, pp 195–202, DOI 10.1109/CVPR.2003.1211354
35. Shahriyar S, Murshed M, Ali M, Paul M (2014) Efficient coding of depth map by exploiting temporal correlation. In: International Conference on Digital lmage Computing: Techniques and Applications, pp 1–8, DOI 10.1109/DICTA.2014.7008105
36. Shao F, Lin W, Jiang G, Yu M, Dai Q (2014) Depth map coding for view synthesis based on distortion analyses. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 4(1):106–117, DOI 10.1109/JETCAS.2014.2298314
37. Shen L, Liu Z, Zhang X, Zhao W, Zhang Z (2013) An effective CU size decision method for HEVC encoders. IEEE Transactions on Multimedia 15(2):465–470, DOI 10.1109/TMM.2012.2231060
38. Smolic A, Mueller K, Merkle P, Fehn C, Kauff P, Eisert P, Wiegand T (2006) 3D video and free viewpoint video - technologies, applications and MPEG standards. In: IEEE International Conference on Multimedia and Expo, Toronto, Canada, pp 2161–2164, DOI 10.1109/ICME.2006.262683
39. Sullivan G, Ohm J, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology 22(12):1649–1668, DOI 10.1109/TCSVT.2012.2221191

40. Tech G, Schwarz H, Muller K, Wiegand T (2012) 3D video coding using the synthesized view distortion change. In: Picture Coding Symposium, pp 25–28, DOI 10.1109/PCS.2012.6213277
41. Vetro A, Wiegand T, Sullivan G (2011) Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard. Proceedings of the IEEE 99(4):626–642, DOI 10.1109/JPROC.2010.2098830
42. Vilaplana V, Marqués F, Salembier P (2008) Binary partition trees for object detection. IEEE Transactions on Image Processing 17(11):2201–2216, DOI 10.1109/TIP.2008.2002841
43. Yan C, Zhang Y, Xu J, Dai F, Li L, Dai Q, Wu F (2014) A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Processing Letters 21(5):573–576, DOI 10.1109/LSP.2014.2310494
44. Yan C, Zhang Y, Xu J, Dai F, Zhang J, Dai Q, Wu F (2014) Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Transactions on Circuits and Systems for Video Technology 24(12):2077–2089, DOI 10.1109/TCSVT.2014.2335852