Contour-based Segmentation and Coding for Depth Map Compression

Fabian Jäger
Institut für Nachrichtentechnik
RWTH Aachen University
52056 Aachen, Germany
Email: jaeger@ient.rwth-aachen.de

***Abstract*— Emerging video technologies like 3DTV and Free Viewpoint Video require to transmit more information than just 2D color data. To allow rendering of arbitrary viewpoints of a video scene a new data format including 2D color and an accompanying depth map has been proposed. Consequently this new technique requires an efficient coding method not only for color information, but also for depth data. Depth maps are characterized by segments describable by piecewise linear functions bounded by sharp edges. Preserving these depth discontinuities is a crucial requirement for high quality view synthesis. To adapt to these characteristics we propose a novel algorithm for depth map compression explicitly signaling the location of discontinuities. Experimental results show that the proposed method yields up to 9dB PSNR gain compared to a JPEG-2000 encoder in high-bitrate scenarios. Subjective quality assessment of synthesized views using compressed depth maps can prove the superior visual quality originating from less geometric distortions.**

I. Introduction

Recent developments in the field of 3D display technologies allow to incorporate more visual depth cues than just stereopsis. With Free Viewpoint Video the viewer is able to dynamically choose a viewpoint on the current video scene. Possible applications could be head-coupled rendering of the video, which would incorporate motion parallax as another depth cue to the display system. Autostereoscopic display technology requires multiple viewpoints to be displayed at the same time to enable 3D perception without glasses.

To enable the mentioned 3D applications dynamic view synthesis at the receiver becomes inevitable to keep the amount of data to be transmitted low while allowing high quality visual perception for the viewer. Synthesizing arbitrary viewpoints depends on the availability of depth information in addition to the color video. Therefore new compression algorithms for depth data have to be investigated.

Previous work on depth coding regarded depth data as gray colored images that can be encoded with classical transform-based approaches like in H.264/AVC[1]. To consider the importance of preserving depth discontinuities, which is not guaranteed by transform-based algorithms, a region-of-interest based extension to a JPEG-2000 based depth map coding was proposed[3]. But transform based methods inherently result in ringing artifacts along edges, which then lead to incorrectly warped pixel positions during view synthesis and are perceived as geometric distortions by the viewer.

Another category of depth compression algorithms attempts to approximate depth information by dividing the image into triangular meshes[6] or platelets[5] and modeling each segment by a piecewise linear function while the partitioning is described by a corresponding tree structure. Although these approaches better adapt to sharp edges in depth images, they still cannot provide a pixel accurate representation of object boundaries as it would be required by high-quality view synthesis. To achieve very precise information about depth discontinuities with these methods, the image partitioning needs to be done down to pixel level, which then again results in high bitrates for the tree structure.

A third category tries to combine the advantages of the two described approaches by integrating a new coding mode into existing block and transform based algorithms[7]. In one of these methods the encoder can select the sparse dyadic (SD) mode[4] as an alternative to transform-coding the current block. An SD-coded block is partitioned into two areas, which are described by two representative depth values. In a subsequent refinement step edge information from the corresponding block in the color image is taken into account to get a more precise representation of the object boundary. As this approach relies on strong correlation between object boundaries in color and depth images, it can result in incorrect approximations of depth discontinuities if this assumption does not hold.

In this paper, a novel coding method is proposed to preserve depth discontinuities on a pixel accurate basis for high-fidelity view synthesis. Depth information for each segment within the depth image is represented by a piecewise-linear function to allow for surfaces, which are not parallel to the image plane and are characterized by a linear gradient in the depth image. The remainder of this paper is organized as follows. Section II introduces the framework of the proposed depth image coding algorithm. Section III explains the coding process of the data describing the depth map representation after the algorithm is performed. Experimental results are shown in Section IV before Section V concludes the paper with a summary and an outlook for future investigations.

II. Depth Image Segmentation and Modeling

To ensure a pixel accurate representation of depth discontinuities the proposed algorithm starts with computing contours in the depth image. Contours have the advantage, compared

to gradient-based edge information based on image gradients, that they always lead to closed contours and therefore perfectly describe segment boundaries. As the algorithm fills segments with depth information based on a modeling function closed contours are crucial to this approach. After determining an appropriate contour mask the algorithm continues to select suitable modeling functions for each segment. In a last step, depth values of pixels lying on contours are also approximated by adjacent segment depths. To allow for contour pixels with depth values that do not match neighboring segment models, a residual signal can be encoded and transmitted.

A. Contour Selection

We define the contour for a certain depth value λ to be the boundary ∂D_λ of the set $D_\lambda = \{x, y \mid d(x, y) > \lambda\}$ where $d(x, y)$ is the depth value at position (x, y) in the depth map. The computation of contours is implemented as an iterative process. In each iteration the number of candidate contours is increased by using more λ values as candidates, which results in a finer-grained representation of the depth image. Then the algorithm tries to find the best approximation for the resulting segments by applying the model explained in II-B. Segments, which can be approximated by one of the possible modeling functions are marked accordingly and skipped in subsequent iterations. In the next iteration, segments without a suitable modeling function will again be segmented into more depth levels. The iterative process stops as soon as all pixels are either marked as being contour pixels or approximated by a segment's modeling function. The distortion of a segment S with a given modeling function $\hat{d}_S(x, y)$ is the mean squared error (MSE_S) between the original depth values of the segment and their approximation.

To better resemble geometric distortion, alternative distortion measures have to be investigated in further research activities, but for an initial comparison between this novel coding method and other approaches the MSE_S is a reasonable choice.

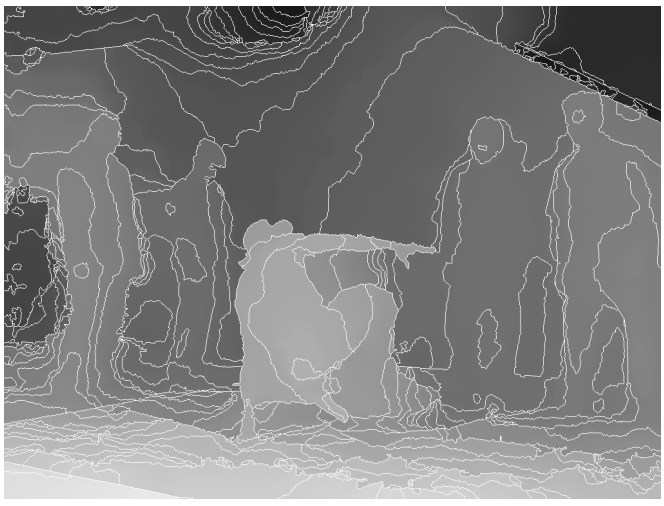


Fig. 1. Original depth map of "Breakdancers"[8] and the superimposed contour map (white).

After the iterative process to find a suitable segmentation of the depth map, a refinement step removes contour lines shorter than a given threshold or which do separate segments with the same model parameters. These segments are merged in the refinement step to reduce the number of contour pixels.

B. Modeling Functions

To approximate a segment S the proposed algorithm allows two possible modeling functions. These were selected after an analysis of the depth maps used in recent exploration experiments for 3D video coding.[1] Those depth maps are characterized by piecewise-linear segments bounded by sharp edges resembling depth discontinuities along object boundaries. Therefore one of the modeling functions to describe a segment's depth has to be a linear approximation, described by equation (1).

$$\hat{d}_1(x, y) = \beta_0 + \beta_1 x + \beta_2 y \qquad (x, y) \in S \tag{1}$$

The parameters for modeling function $\hat{d}_1$ are computed by a least square minimization for the segment's pixels.

Many objects or segments in the investigated depth maps have constant depth values. To allow for an efficient modeling of these segments, a constant modeling function can be signaled.

$$\hat{d}_0(x, y) = \alpha_0 \qquad (x, y) \in S \tag{2}$$

In this case, the model parameter α_0 is the mean value of the segment's pixels.

An optimal choice for a segment's modeling function has to minimize a cost function combining both, required bitrate $R(\hat{d}_i)$ and resulting distortion $MSE(\hat{d}_i)$. Since the rate and distortion are additive over all image segments, an independent optimization for each segment also leads to a general rate-distortion optimization for the whole image. Thus for each segment the algorithm selects the best modeling function $\hat{d}_S$ according to equation (3). The weighting factor μ defines the rate-distortion trade-off.

$$\hat{d}_S = \underset{\hat{d}_i \in \{\hat{d}_0, \hat{d}_1\}}{\arg\min} \left(MSE(\hat{d}_i) + \mu R(\hat{d}_i) \right) \tag{3}$$

In the algorithm's current implementation the rate $R(\hat{d}_i)$ for each modeling function is defined by the number of bits per pixel (bpp). As a quantization of the modeling parameters is still to be implemented, function $\hat{d}_0$ always requires a single 8 bit integer value and function $\hat{d}_1$ requires three 32 bit floating point values, resulting in 96 bits per segment. These values are additionally normalized to bits per pixel (bpp) to generate the final rate $R(\hat{d}_i)$ of a modeling function.

III. Coding Model Parameters and Contours

The final step of the encoding algorithm is to compress the resulting contour map, the model parameters for each segment and an optional residual signal for contour pixels, which cannot be approximated by the modeling functions of the bounded segments. For the segments themselves no residual signal is being transmitted.

A. Coding Contour Lines

The contour map is resembled by a simple bit map with the same dimensions as the input depth map. As the whole algorithm relies on preserving depth discontinuities with pixel accuracy, the contour map has to be encoded losslessly. To achieve this, we use a very efficient algorithm for compression of bi-level images, called JBIG2[2]. This algorithm yields very high compression rates while being perfectly lossless.

In most cases depth values of contour pixels can also be approximated by the bounded segment's modeling function. To allow for contour pixels not matching the modeling function of the bounded segment, a difference signal (residual) for these depth values is encoded using entropy coding.

B. Coding Model Parameters

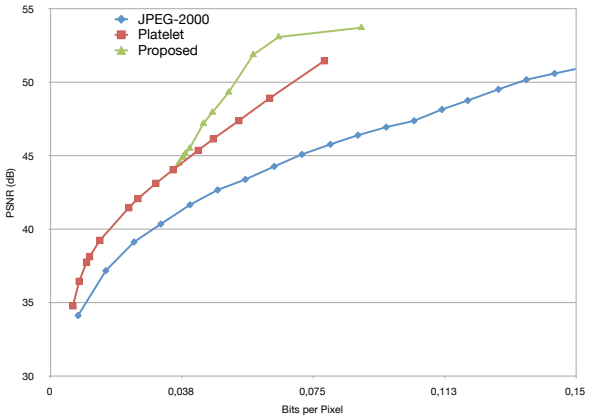
For each segment the proposed algorithm outputs which modeling function to use (1 bit) and the corresponding model parameters. The current implementation only uses linear prediction from the preceding (line-scan order) segment to increase coding efficiency of the Huffman coder, which works pretty well for modeling function $\hat{d}_0(x, y)$. For the floating point values of $\hat{d}_1(x, y)$ a quantization step is not yet implemented, but it is also not expected to change the presented results significantly.

IV. Experimental Results

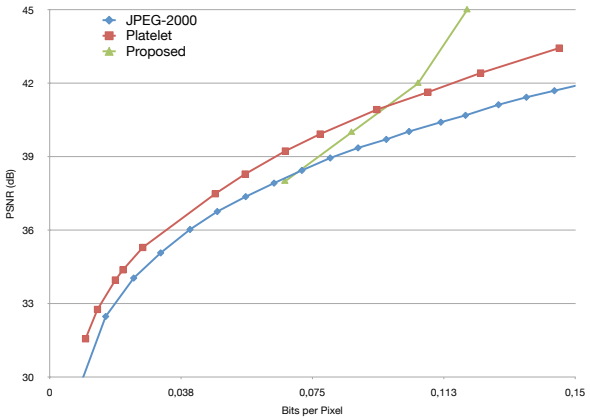
The proposed algorithm to encode depth maps was tested with multiple depth images and compared to classical transform-based algorithms like JPEG-2000 and to the mentioned platelet-based approach[5]. As this algorithm still lacks support for motion compensation, it can only be compared with single-image coding methods. The quality of depth map coding has to be examined in two different ways: First we compare the quality of the decoded depth maps themselves by means of PSNR to give a rough idea of the encoding quality.

As depth maps are typically not displayed and just used to synthesize new virtual views, quality of synthesized images is also to be investigated and play an even more important role than the objective quality of the depth maps themselves. For this investigation the color information is not encoded. For a fixed virtual camera position an image is synthesized using uncoded color and uncoded depth information. This virtual image is then used as the reference image for comparison with synthesized results using compressed depth maps and still uncompressed color information.

Figure 2 shows rate-distortion curves for encoded depth maps without view synthesis. It is obvious that the two model-based approaches outperform the pure transform-based method. This becomes very clear in the results for the "balloons" image. For low bitrates the platelet-based approach performs equally well as the proposed method. At approximately 0.065 bpp, the proposed algorithm is about 5dB better than the platelet method and even 9dB compared to JPEG-2000. For the "book_arrival" image the proposed algorithm performs differently. It is always better than the JPEG-2000 coded depth maps and intersects the platelet-based R-D curve at about 0.1 bpp. At 0.12 bpp it outperforms the platelet-based approach by approximately 3dB in PSNR. While the proposed method yields very good result for medium bitrates, it cannot reach very low bitrates in its current implementation. This can be explained by the JBIG2 coder used for the contour map. Even with sparse contour maps there is a lower bound for the number of bytes resulting from the compression. To become competitive with very low bitrates, the algorithm has to be optimized to also approximate the contour map and potentially use a different coder for it.



(a)

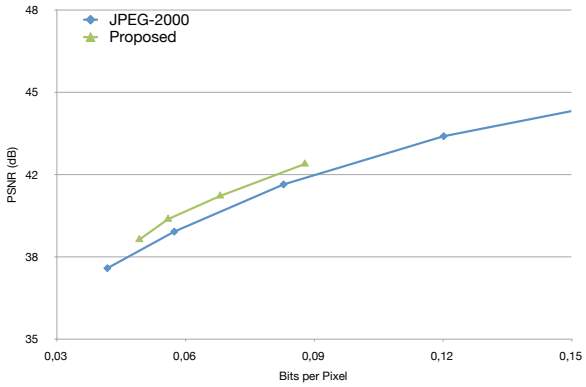


(b)

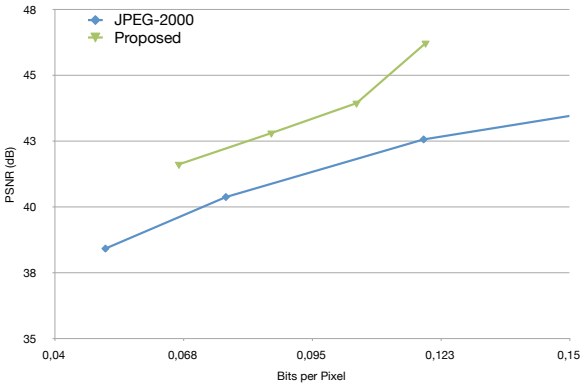
Fig. 2. Rate-distortion curves for the "balloons" (a) and "book_arrival" (b) depth images coded with JPEG-2000, platelet-based approach and also with the presented algorithm.

When analyzing the synthesis results using compressed depth maps, as done in Figure 3, it can be seen that the difference in visual quality measured by the PSNR for the "balloons" image is not as distinct as it was when evaluating only the PSNR of the depth map itself. The reason for this behavior is the relatively simple depth structure of this image. With such scenes view synthesis results are not as prone to depth distortions as with more complex scenes like in "book_arrival". There the higher quality of the coded depth maps lead to much better synthesis results, as shown in Figure 3(b).

As the quality of synthesized images is difficult to evaluate



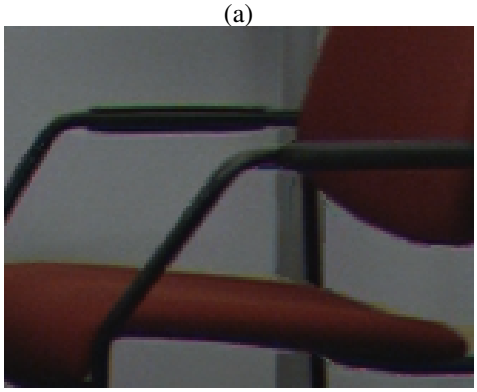
(a)



(b)

Fig. 3. Rate-distortion curves for "balloons" (a) and "book_arrival" (b). The PSNR was computed for the same synthesized virtual viewpoint. Synthesis was based on depth images coded with JPEG-2000 and with the presented algorithm.

(a)



(b)

Fig. 4. Magnified synthesis results for the image "book_arrival". Version (a) was synthesized using a depth map coded with JPEG-2000 at 0.065 bits per pixel and the underlying depth map for version (b) was encoded at approximately the same rate with the proposed algorithm.

with measures like the PSNR, typical artifacts are highlighted in Figure 4. As discussed before, preserving edge information is crucial for high-quality view synthesis and therefore artifacts close to these object boundaries lead to severe artifacts in view synthesis. Figure 4(a) shows the very distorted geometric information like for the armrest, which seems to dissolve for JPEG-2000 coded depth-maps. This effect results from ringing artifacts along edges, which are typical for transform-based algorithms. When depth information is coded on a per-segment basis, as in Figure 4(b), object boundaries are better preserved and result in a more natural visual impression.

V. Conclusions

In this paper a novel compression algorithm for depth data is proposed. It is explained why preserving depth discontinuities is a crucial requirement for depth map coding. By implementing an adaptive computation of suitable contour lines segmenting the depth image into segments, which can be approximated with a piecewise-linear function, a high coding efficiency can be achieved. In comparison to transform-based approaches, synthesized views based on encoded depth maps do not show as severe geometric distortions. Further investigations have to show whether the proposed algorithm can be used as an alternative coding mode for intra coded segments of a video coder like H.264/AVC. Moreover it has to be examined if this algorithm can also be used in a motion compensated scenario.

Acknowledgment

References

[1] C. Fehn, K. Schuur, P. Kauff, and A. Smolic, "Coding results for EE4 in MPEG 3DAV," *ISO/IEC JTC1/SC29/WG11 M*, vol. 9561, 2003.

[2] P. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. Rucklidge, "The emerging JBIG2 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838–848, 2002.

[3] R. Krishnamurthy, B. Chai, H. Tao, and S. Sethuraman, "Compression and transmission of depth maps for image-based rendering," in *Proceedings of International Conference on Image Processing*, vol. 3. IEEE, 2001, pp. 828–831.

[4] S. Liu, P. Lai, D. Tian, C. Gomila, and C. Chen, "Sparse dyadic mode for depth map compression," in *17th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 3421–3424.

[5] Y. Morvan, P. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Proceedings of SPIE, Stereoscopic Displays and Applications*, vol. 6055, 2006, pp. 93–100.

[6] M. Sarkis, W. Zia, and K. Diepold, "Fast depth map compression and meshing with compressed tritree," *Computer Vision–ACCV 2009*, pp. 44–55, 2010.

[7] G. Shen, W. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, pp. 3393–3396.

[8] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 600–608.