IMPROVING THE VISUAL QUALITY OF AVC/H.264 BY COMBINING IT WITH CONTENT ADAPTIVE DEPTH MAP COMPRESSION

Christian Keimel, Klaus Diepold and Michel Sarkis

Technische Universität München
Institute for Data Processing
Arcisstr. 21, 80333 München, Germany
christian.keimel@tum.de, kldi@tum.de, michel.sarkis@sony.de

ABSTRACT

The future of video coding for 3DTV lies in the combination of depth maps and corresponding textures. Most current video coding standards, however, are only optimized for visual quality and are not able to efficiently compress depth maps. We present in this work a content adaptive depth map meshing with tritree and entropy encoding for 3D videos. We show that this approach outperforms the intra frame prediction of AVC/H.264 for the coding of depth maps of still images. We also demonstrate by combining AVC/H.264 with our algorithm that we are able to increase the visual quality of the encoded texture on average by 6 dB. This work is currently limited to still images but an extension to intra coding of 3D video is straightforward.

Index Terms— AVC/H.264, 3D scene analysis, 3DTV, depth map compression, content adaptive meshing.

1. INTRODUCTION

Three dimensional television (3DTV) is on its way to play an important role not only in the consumer business, but also in broadcasting since companies have started their broadcasting services in 3DTV. So far, the two different views needed for 3DTV have been mostly coded separately and simulcasted with AVC/H.264. In the near future, its extension for multiview coding (MVC, AVC/H.264 - Annex H) will enable us to exploit the interdependencies between the two different views for a more efficient coding. Still, we will see in the long term a shift from the coding of two separate views to a combination of one view representing the texture of the scene and an accompanying depth map describing the disparities between the two original views [1]. The two views needed for 3D perception can then be rendered from the depth map and the texture. Therefore, there is a need for methods and algorithms to efficiently compress depth maps.

Current coding technology for video and image compression e.g. AVC/H.264 and JPEG2000, was designed to optimize the perceived visual quality of 2D video using the mean squared error (MSE) or the peak signal to noise ratio (PSNR) as a quality metric. These metrics, however, describe the overall pixel errors of a compressed disparity image represented by a depth map insufficiently, see [2, 3]. Take for example the case where each pixel of an image has an absolute difference of 2 between the original and the reconstructed depth map. The overall MSE is 4 but the pixel error rate (PERR) is 100%. In addition, MSE does not take the discontinuities of depth maps into account, see [4]. This will result in a lot of errors especially at high compression ratios which will lead to a significant amount of errors in the 3D reconstruction of the scene [2].

To overcome such limitations, we propose in this paper a more efficient depth map compression scheme based on content adaptive meshing with tritree and entropy encoding. Our method is able to achieve similar depth error rates but at a much higher compression ratio than with AVC/H.264. The results we obtained show that we can increase the visual quality of the AVC/H.264 coded texture while maintaining the same overall bit rate. Our algorithm is an extension of our previous work in [2]. The contribution of this work can be described in two points. Firstly, we extend the comparisons and test our scheme versus AVC/H.264. Secondly, we propose an efficient 3D video coding scheme by integrating tritree with AVC/H.264.

This paper is organized as follows: we will present in Section 2 our some related work and then introduce depth map compression scheme. In Section 3, we compare the performance of our scheme to AVC/H.264. In Section 4, we introduce a new a approach to increase the overall visual quality by combining tritree with AVC/H.264. In the end, we present some concluding remarks in Section 5.

2. DEPTH MAP COMPRESSION

Our approach for depth map compression is composed of two parts. Firstly, the depth map is approximated with a content adaptive mesh. This is done by detecting the non-uniform samples of the depth map in such a way that the error is minimal. The depth map will be partitioned into an adaptive mesh, where the size of the triangles is small near depth discontinuities, i.e. edges in the depth map, and large elsewhere. This provides a higher sampling rate in critical areas and a lower sampling rate in the smooth areas. The non-uniform samples or nodes are able to interpolate all the other pixels of the depth map using the mesh up to predefined error. Secondly, we compress the mesh and the (sparse) disparity values by entropy encoding. The proposed approach will be explained is the sequel. But, we will first start with a brief review on the related work.

2.1. Related Work

In [5], the method is based on the generation of adaptive meshes. The mesh, however, does only deal with small depth details and no depth discontinuities. This was later improved in [6] by detecting discontinuities, modeling them and using a constrained triangulation. Our approach already takes discontinuities during the meshing into account by increasing the sampling rate at these locations. In [7] a JPEG2000 encoder is modified to consider region of interest coding and reshaping of the dynamic range of depth maps. In [8] depth

This work was conducted when M. Sarkis was at the Institute of Data Processing, in Technische Universität München. M. Sarkis is currently with Sensing Systems Laboratory (SSL) at the Sony Deutschland GmbH.

© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Digital Object Identifier: 10.1109/PCS.2010.5702545

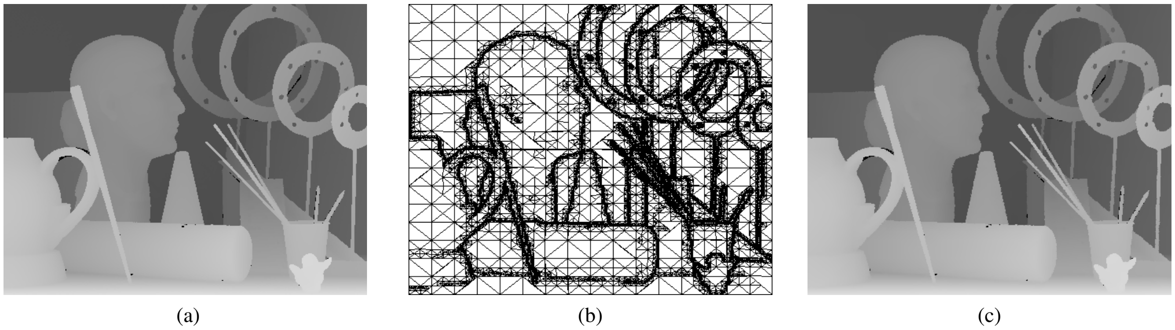


Fig. 1: The original disparity map (a), the corresponding adaptive mesh (b) and the reconstructed disparity map (c) for *Art*.

maps are hierarchically decomposed into four regions depending on the location of edges, merged and then encoded using AVC/H.264.

2.2. Content Adaptive Meshing with Tritree

The pixels of a disparity map form a 3D space represented by the 2D coordinates of the pixels in the depth map and the corresponding disparity value. Each triangle T of the mesh is defined by three vertices $\mathbf{v}_i(x_i, y_i, d_i)$, with $i = 1, 2, 3$. Thus a plane Π is described by T using the normal equation

$$\vec{\eta} \cdot \mathbf{p}_n + k = 0, \tag{1}$$

where $\mathbf{p}_n(x_n, y_n, d_n)$ denotes a pixel with coordinates (x_n, y_n) and depth value d_n on Π, $\vec{\eta}$ is the normal vector on Π and k a constant satisfying $k = -\vec{\eta} \cdot \mathbf{v}_i$. The vector $\vec{\eta} = (\eta_1, \eta_2, \eta_3)$ can be computed as the cross product of any two edges of the triangle. To recover the disparity value $\hat{d}_n$ of a pixel inside T, we can rearrange (1):

$$\hat{d}_n = -(\eta_1 x_n + \eta_2 y_n + k)/\eta_3. \tag{2}$$

In order to determine the content adaptive mesh of a disparity map, it is necessary to find all triangles T and their vertices in the map that can reconstruct the depth information of all pixels within each triangle T. To determine the quality of the reconstructed depth values within each triangle, the percentage of the PERR is used. The goal is to minimize the PERR over each triangle of the mesh.

Using binary space partioning, we first divide the disparity map along one of the diagonals into two triangles. Then we check if each triangle satifies a certain PERR threshold ϵ. If not, the triangle will be recursively subdivided into further triangles until the condition is met or no further subdivisions are possible. This leads to a triangular tree or *Tritree*, which we will also use to refer to our scheme. Also we gained implicitly a binary tree representation of the mesh. Another advantage is that the described algorithm can easily be parallelized to increase the computational performance. In Fig. 1 an example using the image *Art* from the Middelbury stereo data set [9] is shown. For more details about this scheme, please refer to [10].

2.3. Entropy Encoding

The mesh generated in Section 2.2 is then further compressed by employing entropy coding as described in [2]. As the depth map itself, but also the sparse depth map are highly correlated since it is a piecewise smooth surface [4], its histogram is rather uniform and thus not well suited for entropy coding. If we, however, use differential coding by predicting the next disparity from the preceding disparity, we achieve a more peak shaped distribution, better suited to entropy coding. We used two of the most popular entropy coding techniques: Huffman coding and arithmetic coding.

3. COMPARISION WITH AVC/H.264

In this section we will compare our depth map coding approach with AVC/H.264. Although AVC/H.264 was designed to primarily encode video and its high encoding performance is largely due to its sophisticated methods for motion compensation in the inter frame prediction, it has been shown that its encoding performance for still images using only intra frame prediction is superior to dedicated still image encoding standards as JPEG2000 [11]. Therefore we decided to use the images and depth maps from the Middlebury stereo data sets [9, 12] to get a first impression of the performance of our approach compared to AVC/H.264. We used both Huffman and arithmetic coding for our approach. For completeness, we also include JPEG2000 in the comparison. We used the AVC/H.264 reference software [13] version 15.2 which was tuned to use only intra frame coding with activated rate distortion optimization.

The results for the images *Teddy*, *Cones*, *Art* and *Moebius* are presented in Fig. 3. In this first two columns, we present the pixel error rate (PERR) and the MSE versus the compression factor, which is nothing but the inverse of the compression ratio. In the third column, we plot the PERR versus the number of bits required to represent each pixel or Bits per Pixel (BPP). We can clearly see that our approach compresses the depth maps with a much lower PERR or depth pixel error rate than both AVC/H.264 and JPEG2000. Also we achieve a much better coding efficiency, as we need far fewer BPP for a comparable PERR in the depth map. The difference between entropy coding with Huffman or arithmetic coding ist negligible. Not surprisingly, both AVC/H.264 and JPEG2000 deliver a much better MSE than our approach, as both are using MSE in the form of PSNR to minimize the prediction error during the encoding. We can confirm the superior encoding performance of AVC/H.264 compared to JPEG2000 also for depth maps with regard to the MSE and PERR. It should be noted, however, that at least in our case JPEG2000 seemingly encodes more efficiently with regards to the BPP than AVC/H.264.

4. COMBINING AVC/H.264 AND TRITREE FOR 3DTV

We can see in Table 1 that for the same depth bit error rate of 1% we save at least 0.72 bits per pixel and on average 0.95 bits per pixel with our approach compared to AVC/H.264.

Assuming we have a fixed overall bit rate or *bit budget* available and a fixed target PERR, we can invest the bits saved by using

Table 1: BPP required by each scheme for a depth error of 1%

Image	AVC/H.264	Tritree[(a)]	Improvement
Teddy	1.54	0.53	1,01
Cones	1.85	0.67	1.18
Art	1.69	0.79	0.90
Moebius	1.34	0.62	0.72

[(a)] for Huffman coding, arithmetic coding similar

Table 2: Increase in visual quality of the texture for the combination AVC/H.264 & Tritree compared to AVC/H.264

Image	AVC/H.264 PSNR [dB]	AVC/H.264 & Tritree PSNR [dB]
Teddy	38.48	43.79
Cones	45.10	56.37
Art	41.59	46.75
Moebius	41.18	45.03

Tritree to compress the depth map into the encoding of the texture with AVC/H.264 and achieve an overall better visual quality. Therefore we propose to combine both by using AVC/H.264 to compress the texture and Tritree to compress the depth map.

Our proposed combined system is shown in Fig.2. As current 3DTV systems in most cases need to utilize existing infrastructure and can therefore not increase the overall bit rate of the signal, the texture and the depth map are usually mixed into one frame: typical arrangements are *checkerboard*, *interleaved*, *side-by-side* and *up-down*. Hence we first split our input signal into the texture and depth map component. Then we proceed to encode the texture with AVC/H.264, while encoding the depth map with Tritree. After decoding texture and depth map separately, we once again mix both components into the necessary output format.

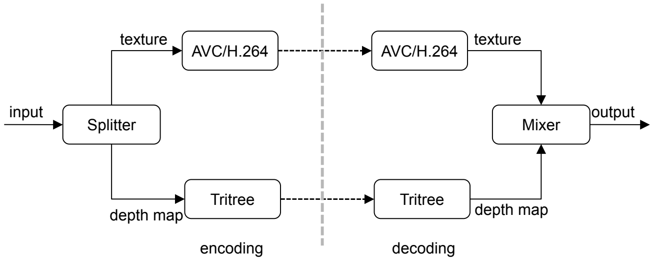


Fig. 2: Proposed system combining AVC/H.264 and Tritree

To demonstrate the gain in visual quality that can be achieved by our proposed combination of AVC/H.264 and Tritree, we use once again *Teddy*, *Cones*, *Art* and *Moebius* from [9, 12]. As a starting point, we assume 1.00 BPP for the encoded texture of all images and selected the closest available rate point. Then we consider the bits per pixel saved by Tritree as shown in Table 1. These saved bits per pixel from the depth map are then added to the bit budget for the texture coding with AVC/H.264 e.g. for *Teddy* $1.03 + 1.01 = 2.04$ BPP were allocated for texture coding. We should keep in mind that the overall bit rate for both texture and depth map together remained constant. We just redistributed the bits between both parts. In Table 2 we can see that the PSNR for the selected images is increased noticeably and on average we gain about 6 dB. While PSNR may not be best suited to describe the perceived visual quality of depth maps, it is a good indicator for the improvement we achieve with the quality of the texture images.

But we should also keep in mind that the definition of overall visual quality for 3D i.e. visual quality of the texture and depth perception is still an open research area and neither objective metrics nor subjective methods are available, yet [14].

5. CONCLUSION

We introduced an efficient depth map compression for 3DTV based on content adaptive meshing with tritree and entropy encoding, called *Tritree*. Our comparison with AVC/H.264 showed that our approach compresses depth maps more efficiently thus saving bits spent on coding the depth map. Based on these results, we proposed a system combining the Tritree approach for depth map coding and AVC/H.264 for texture coding. By reallocating the bits saved by Tritree, we gained on average an improvement in visual quality for the texture of 6 dB while keeping the overall bit rate constant.

Even tough we only considered still images and the intra frame prediction capabilities of AVC/H.264, we believe that our approach can also be extended to video, especially for pure intra frame prediction. Future work will therefore include the application of Tritree on 3D video, but also the implementation of our proposed combination in the framework of the AVC/H.264 reference software, in order to exploit the full possibilities of inter frame prediction.

6. REFERENCES

[1] B. Coll, K. Dean, F. Ishtiaq, and K. O'Connell, "3D TV at home: Status, challenges and solutions for delivering a high quality experience," in *Int. Workshop Video Processing and Quality Metrics for Consumer Electronics*, Jan. 2010.

[2] M. Sarkis, W. Zia, and K. Diepold, "Fast depth map compression and meshing with compressed tritree," in *Asian Conf. Computer Vision (ACCV)*, Sep. 2009.

[3] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. D. With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Processing: Image Communication*, vol. 24, no. 1–2, pp. 73–88, Jan. 2009.

[4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

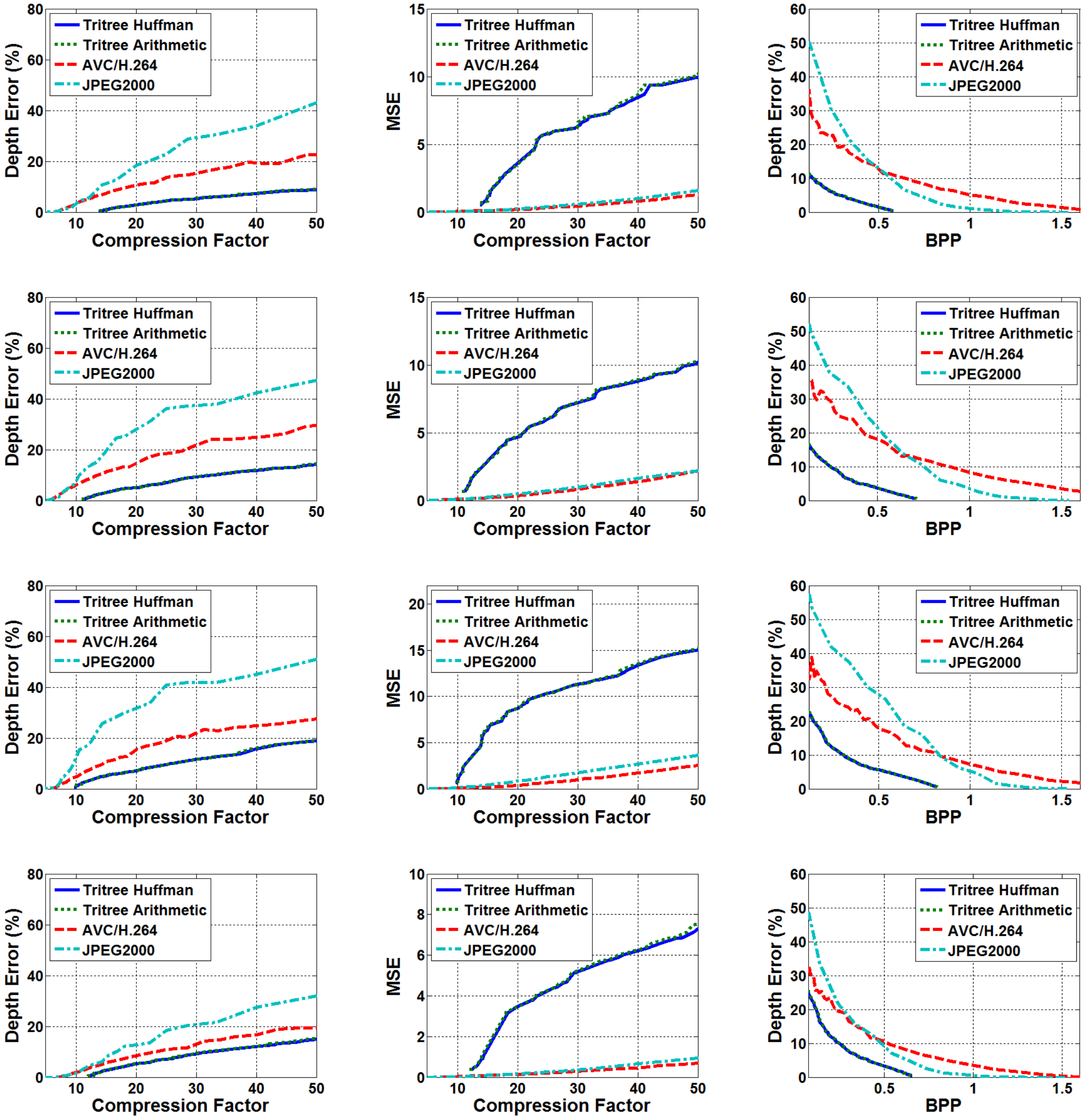
[5] B.-B. Chai, S. Sethuraman, H. S. Sawhney, and P. Hatrack, "Depth map compression for real-time view-based rendering," *Pattern Recognition Let.*, vol. 25, no. 7, pp. 755–766, 2004.

[6] D. Farin, R. Peerlings, and P. de With, "Depth-image representation employing meshes for intermediate-view rendering and coding," in *3DTV Conf.*, May 2007, pp. 1–4.

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

[8] S.-Y. Kim and Y.-S. Ho, "Mesh-based depth coding for 3d video using hierarchical decomposition of depth maps," in *Int. Conf. Image Processing*, vol. 5, Oct. 2007, pp. V –117–V –120.

[9] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.



(a) (b) (c)

Fig. 3: Comparison of the proposed algorithm with AVC/H.264 and JPEG2000. From top to down: *Teddy*, *Cones*, *Art* and *Moebius*. Depth pixel error rate in % in the first column (a), MSE in the second column (b) and the rate distortion curve in the third column (c).

[10] M. Sarkis and K. Diepold, “Content adaptive mesh representation of images using binary space partitions,” *IEEE T. Image Processing*, vol. 18, no. 5, pp. 1069–1079, May 2009.

[11] A. Al, B. Rao, S. Kudva, S. Babu, D. Sumam, and A. Rao, “Quality and complexity comparison of H.264 intra mode with JPEG2000 and JPEG,” in *Int. Conf. Image Processing*, vol. 1, Oct. 2004, pp. 525–528.

[12] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003, pp. I–195–I–202.

[13] K. Sühring. (2010) H.264/AVC software coordination. [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[14] W. Chen, J. Fournier, M. Barkowsky, and P. L. Callet, “New requirements of subjective video quality assessment methodologies for 3DTV,” in *Int. Workshop Video Processing and Quality Metrics for Consumer Electronics*, Jan. 2010.