Intra-frame Depth Image Compression Based on Anisotropic Partition Scheme and Plane Approximation

Nikolay Ponomarenko
National Aerospace University
17 Chkalova Str.
61070 Kharkov, Ukraine
+380577074841
uagames@mail.ru

Vladimir Lukin
National Aerospace University
17 Chkalova Str.
61070 Kharkov, Ukraine
+380577074841
lukin@ai.kharkov.com

Atanas Gotchev
Tampere University of Technology
Korkeakoulunkatu 1,
33720 Tampere, Finland
+358 3 3115 4349
atanas.gotchev@tut.fi

Karen Egiazarian
Tampere University of Technology
Korkeakoulunkatu 1,
33720 Tampere, Finland
+358 3 3115 4349
karen.egiazarian@tut.fi

ABSTRACT

3D video can be represented by color 2D video sequence accompanied by gray-scale dense depth map (depth image) sequence. In this paper, we describe a novel method for intra-frame compression of the depth modality of such representation. Our method takes into account specific features of depth images, i.e. the presence of large smooth regions delineated by sharp discontinuities (edges). For such images, conventional transform-based coders produce undesirable artifacts which impede the subsequent rendering of virtual views. Instead of block transforms, the proposed method employs horizontal-vertical anisotropic partition scheme which yields a tree-structured decomposition of non-overlapping rectangular blocks adapted to the depth map content. Each block in the decomposition is approximated by plane described by the block corner pixels. The codestream consists of the coded partition scheme and the coded error of prediction of quantized corner pixels. The scheme substantially reduces the amount of artifacts around edges and yields an improvement of several dB in PSNR for typical compression ratios compared with the best transform-based coders. Other advantages of the designed coder are its simplicity and fast decompression, and the possibility to control the rate-distortion performance.

Keywords

Depth image, lossy compression, intra-frame, partition scheme

1. INTRODUCTION

Processing of 3D video is an active area of research with 3DTV and free view-point TV (FTV) being the most appealing applications [12]. Among representation formats of 3D video, the so-called video+depth representation has attracted a special attention. In this representation, each frame of a conventional 2D color video sequence is augmented with per-pixel depth of the scene represented by gray-scale image of certain precision. At the display end, virtual views can be rendered by image warping out of the given video and depth provided the camera parameters are also known (so-called depth-image based rendering). This format has been considered broadcast-friendly as previous studies demonstrated that compressed depth takes a relatively small fraction of the total bit budget.

Depth image sequence can be compressed similarly to other types of video, i.e. by combining intra-frame and inter-frame data compression. Regarding intra-frame compression, depth images are quite different from conventional color (photo) images. Depth maps are commonly represented by gray-scale images comprising smooth segments separated by high-contrast edges [13]. Textures hardly exist. For such image structures, conventional transform-based coders are rather ineffective. In the vicinity of edges, they produce blocking and/or ringing artifacts which degrade the quality of the rendered views [14]. This fact has stimulated the development of alternative compression techniques better adapted to the peculiarities of depth images. In particular, basis systems for representation of smooth regions separated by sharp edges have been explored. Among them, Bandelets [1], Wedgelets [2] and Platelets [3] have got attention. In [4], the platelets system has been combined with quad-three image decomposition to achieve effective local approximation of object surfaces. Four different approximations have been tested at each node of the full quad-tree decomposition and the best one in rate-distortion sense has been assigned. The tree has been pruned in a similar rate-distortion manner. The method yields an improvement of up to 4 dB compared to JPEG for practical bit rates. It provides also fast decoding. A drawback is its complex quantization procedure quad-tree partition scheme optimization.

The method proposed in this paper also addresses peculiarities of depth maps. It utilizes anisotropic partition schemes in tree-structured setting and plane approximation of image blocks. The bitstream includes partition scheme data and coded prediction of quantized parameters of approximating planes.

There are the following distinctions of the proposed method with respect to the technique described in [4]:

1) Instead of quad-tree, we suggest using an anisotropic (horizontal-vertical) partition scheme (PS). It possesses better adaptivity and eliminates the need of fitting different approximations to the given image block.

Immerscom'09, May 27–29, 2009, Berkeley, CA, USA.

2) In contrast to [4], where each plane or gradient is described by three plane coefficients (parameters): a constant and slopes for x and y directions, in our approach planes are described by three corner pixels of the image block. This allows for a better quantization and prediction and subsequently, for effective coding of planes.

3) The PS optimization procedure provides an easy control over rate vs distortion. It allows achieving minimal bit rate for desired quality or maximizing the quality for a given bit rate.

2. DETERMINATION AND CODING PLANE PARAMETERS FOR A GIVEN IMAGE BLOCK

The proposed method is based on partitioning the given depth image into rectangular blocks and approximating each block by a plane. Consider an image block determined by its left upper and right lower corners (x_1, y_1) and (x_2, y_2). The intensities of pixels within the block are approximated by plane approximation as $z(x,y) = k_1 x + k_2 y + k_3 \;\; (x,y) \in [x_1, y_1] \times [x_2, y_2]$. The plane slope parameters k_1, k_2, and k_3 are determined by least-squares fitting minimizing the error:

$$MSE = \frac{1}{N} \sum_{y=y_1}^{y_2} \sum_{x=x_1}^{x_2} (I(x,y) - z(x,y)^2 \;, \qquad (1)$$

where $N = (y_2-y_1+1)(x_2-x_1+1)$, and $I(x,y)$ denotes the original image pixel with coordinates (x,y).

After fitting planes to all image blocks, a straightforward approach for compression is to quantize the plane parameters k_1, k_2, and k_3, and to analyze for some remaining inter-block correlations and to compensate them e.g. by predictive coding. This is the approach suggested in [4]. However, working with plane slope parameters imposes problems related with quantization and predictive coding. The parameters k_1 and k_2 exhibit different behavior than the parameter k_3 and have to be quantized differently. Prediction of these parameters using neighbor blocks is also problematic.

In our approach, we suggest working with block corner pixels instead of plane parameters. Our argument is based on the fact that a plane is uniquely described and can be reconstructed by three distinct plane points. Fig. 1, shows an image block for which a plane with parameters k_1, k_2, and k_3 has been fitted.

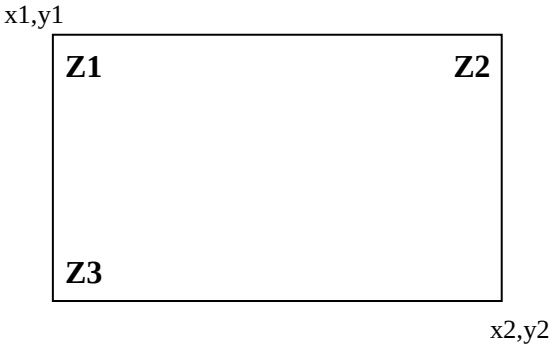

Figure 1. An example of a compressed image block

The values of the three corner pixels Z_1, Z_2, and Z_3 are calculated as:

$$\begin{aligned} Z_1 &= k_1 x_1 + k_2 y_1 + k_3; \\ Z_2 &= k_1 x_2 + k_2 y_1 + k_3; \\ Z_3 &= k_1 x_1 + k_2 y_2 + k_3. \end{aligned} \qquad (2)$$

Having these three pixels at the decoding stage is sufficient to reconstruct the plane. Working with pixels instead of plane parameters facilitates the quantization and prediction as pixels exhibit similar behavior and can be quantized by the same means.

While any three distinct plane pixels would code the plane, the three corner points offer several advantages. (1) Any other selection of distinct points would require transmitting their coordinates. In the case of corners, their coordinates are anyway transmitted by the partition scheme. (2) Corner pixels form the maximum-area triangle within the block. This is beneficial for the quantization, where one would like to have least influence of the quantization step on the reconstruction quality. (3) Corner pixels are easily predicted by pixels from neighboring (and already coded) blocks. In the case of a usual row-wise scanning from the upper left most corner, the values of the corner pixels Z_1, Z_2, and Z_3 shown in Fig. 1 can be predicted by already coded image blocks placed above and left to the block being coded.

In summary: for each image block, values of corner pixels Z_1, Z_2, and Z_3 are calculated from the fitted plane. These values are quantized and then predicted as quantized values of nearest neighbor pixels from already coded and reconstructed image blocks. Prediction errors of Z_1, Z_2, and Z_3 for all image blocks are grouped together and passed to an arithmetic coder for handling any remaining statistical dependency.

3. PARTITION SCHEME OPTIMIZATION

In our approach, we favor the use of an anisotropic horizontal-vertical block-partition scheme (PS), as the one suggested in [6]. In contrast to the quad-tree PS, it divides a given block into only two sub-blocks either vertically or horizontally. Thus, it offers better adaptivity to anisotropic image features. We speculate that by using this PS, there is no need to use four different approximations as in [4]. Instead, we use only plane fitting to a block and rely o the PS to adapt to image edges.

In general, the partitioning could take the entire image as initial block. This, however, will lead to many unnecessary splitting operations and result in lengthily process of getting the final PS. In order to accelerate the procedure, we suggest starting with image division into rather large equal-size square blocks. In our implementation, we use initial blocks of size 128x128 pixels.

The procedure of block partitioning works as follows:

1) For each image block, the optimal values of the corner pixels Z_1, Z_2, and Z_3 are determined and the corresponding reconstruction error E_{ini} is calculated. Then, two possible partitions of this block by horizontal or vertical splitting are checked. For each partition, the values Z_1, Z_2, and Z_3 of the new formed sub-blocks are determined and the corresponding reconstruction errors E_1 and E_2 for these blocks are calculated. The partition providing the minimal value $E_{\min}$ of the sum E_1+E_2 is found.

2) A benefit P due to partitioning a given block into two new sub-blocks is calculated as $P = E_{ini}-E_{min}$. All benefits at the current stage of partitioning are compared and the partitioning continues with division of the block with P_{max}.

Each new division of a chosen block leads to better reconstruction, as it decreases the total reconstruction error. Simultaneously, the size of the coded data also increases due to more nodes added to the expanded tree and more quantized values Z_1, Z_2, and Z_3. The PS optimization can be stopped if either a required (pre-determined) quality of image reconstruction is provided or if a pre-determined bit budget is reached.

The PS data is saved in the following format:

1) A flag showing that a given block is further sub-divided.

2) A flag showing if the sub-divided block is partitioned horizontally or vertically.

3) For divided blocks, one needs $\log_2(y_2-y_1-2)$ bits for horizontal division and $\log_2(x_2-x_1-2)$ bits for vertical division in order to code indices of a block row or column depending on which division has been done.

At each partitioning stage, it is possible to determine the total size occupied by PS data and bits spent for prediction errors of coded values Z_1, Z_2, and Z_3 for the current tree. That is, the total coded image size is known at each stage of partitioning (tree expansion). This allows controlling the bit-budget allocated for intra-frame depth map coding.

4. CODING ALGORITHM AS A WHOLE

The general block-diagram of the proposed compression method is presented in Fig. 2.

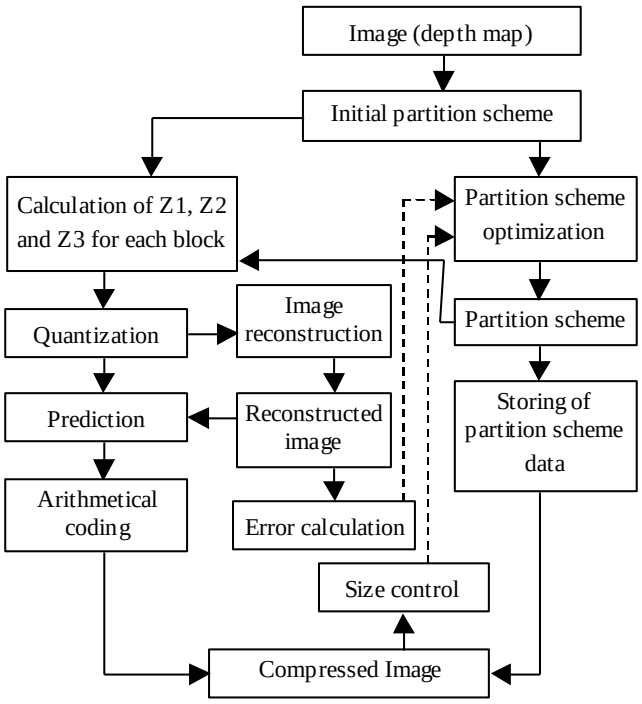

Figure 2. Block-diagram of intra-frame depth map compression method

The compressed file contains quantized and coded block corner pixels originating from fitted planes to the blocks of the PS together with the obtained horizontal-vertical PS data. At each step of PS optimization, new image blocks are reconstructed and taken into account in controlling the decompressed (reconstructed) image quality. The obtained block approximation data and PS data are taken into account in controlling the total size of compressed image. The achieved quality of the compressed image and total bit-size are used for stopping criteria in the PS optimization procedure. This is illustrated by the dashed lines in Fig.2.

PS data is coded first and passed to the total bit-stream. Then plane approximation data is coded (cf. Section 2) and also passed to the bit-stream. The decoding and reconstruction starts with PS data decoding. The decoding stage is quite fast mainly due to two reasons: 1) the arithmetic decoding is fast and 2) linear approximation of surfaces by planes defined by their parameters is very simple.

5. PERFORMANCE TESTING AND CODER PARAMETER SETTING

The performance analysis of the proposed coder has been carried out using the commonly used image sequence Ballet [7] (Figure 3).

In the first experiment, we have addressed the important issue of setting the quantization step QS for the corner pixels. The obtained results in terms of PSNR for different QSs and bits per pixel (bpp) are presented in Table 1. While different QSs are optimal for different bpp, for practically important compression ratios (bpp of about 0.025…0.05) it is expedient to use QS= 2, 3, or 4. Setting a constant QS=4 decreases the PSNR by only 0.08 dB and is a feasible choice.

Table 1. Compression efficiency for the proposed method for different quantization steps of quantizing the values plane reference points Z1, Z2, and Z3, PSNR, dB

Proposed method	Bpp								
	0.005	0.01	0.015	0.025	0.05	0.075	0.1	0.15	0.2
QS=1	29.4	32.5	34.3	37.1	41.5	43.1	**43.5**	**43.8**	**43.8**
QS=2	29.7	32.7	34.7	37.5	**41.9**	**43.1**	43.4	43.6	43.6
QS=3	29.9	33.0	34.9	37.9	41.8	42.9	43.2	43.3	43.3
QS=4	30.1	33.2	35.1	**38.1**	41.8	42.8	43.0	43.1	43.1
QS=8	**30.3**	**33.3**	**35.2**	37.8	40.7	41.0	41.0	41.0	41.0

In the second experiment, the coder performance has been compared with the JPEG [8] and JPEG2000 [9] standards as well as one of the state-of-the-art research coders ADCTC [10] intended for lossy compression of photo images. The comparison has been done in terms of PSNR and bpp with respect to the gray-scale representation of the original data. The results are presented in Table 2. As seen in the table, for typical compression ratios (bpp of about 0.05 or smaller) our method is clearly better. Compared to JPEG2000, the relative improvement by our method is similar to one by the compression method in [4].

Figure 3. The test sequence Ballet, cam3, frame 98. Top: RGB component of the frame; bottom: corresponding depth map.

For smaller compression ratios, i.e. bpp=0.1 or higher, our method is not that effective compared to ADCTC or JPEG2000. The reason is that for such compression ratios, linear approximation is not beneficial for preserving finer details. However, for bpp>0.1 the achieved PSNR is higher than 43 dB and such reconstruction quality is commonly not necessary. Among the compared techniques, JPEG is clearly worst, unable to compress with less than 0.05 bpp.

Table 2. Comparison of JPEG, JPEG2000, ADCTC, and the proposed coder for different bpp, PSNR, dB

Method	Bpp								
	0.005	0.01	0.015	0.025	0.05	0.075	0.1	0.15	0.2
JPEG	-	-	-	-	27.6	34.3	36.2	40.6	43.3
JPEG2000	26.3	30.0	31.7	34.3	38.2	40.7	42.7	45.6	47.5
ADCTC	25.5	29.7	32.8	35.8	39. 8	42.5	**44.7**	**47.7**	**49.6**
Proposed	**30.3**	**33.3**	**35.2**	**38.1**	**41.9**	**43.1**	43.5	43.8	43.8

Figure 4 illustrates the relative compression performance of the proposed method in terms of size (bpp), denoted by S_P, compated to the JPEG2000 and ADCTC compression results, denoted by S_J and S_A, respectively. For high compression ratios (bpp smaller than 0.08), the proposed coder is superior with up to 2.3 times better performance.

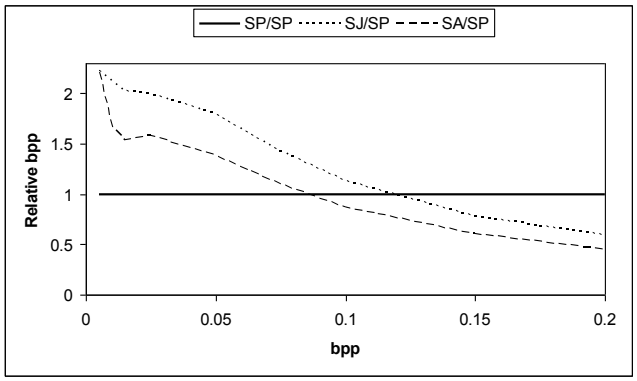

Figure 4. The dependencies of the ratios S_J/S_P and S_A/S_P on bpp under condition of providing the same PSNR (for convenience of comparison, S_P/S_P=1 is shown by horizontal solid line).

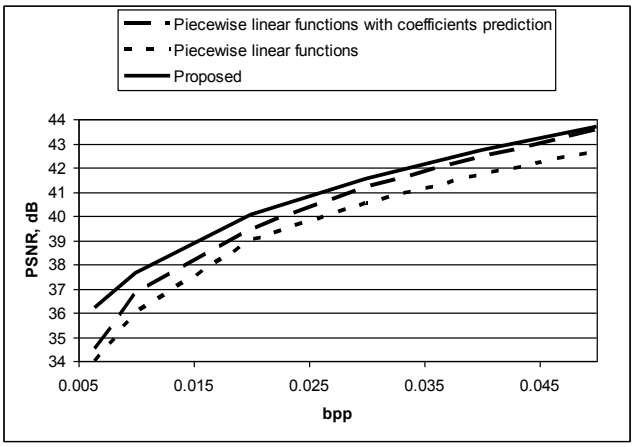

Figure 5. Comparison of the proposed method to the compression methods [4] for the test sequence Breakdancing, cam0, frame 0.

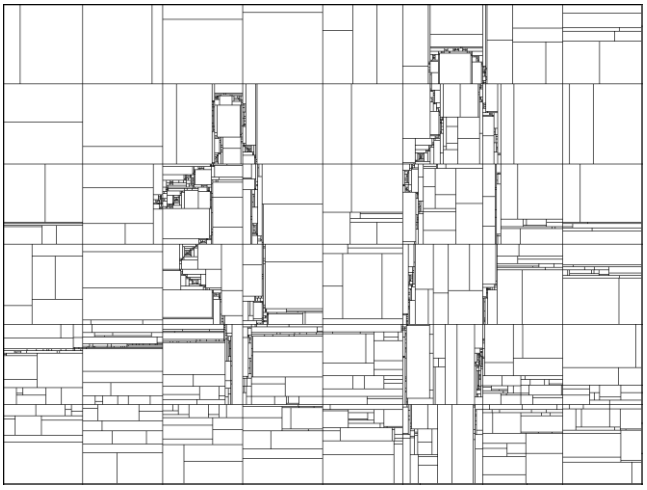

Figure 6. Example of PS obtained for 0.05 bpp (test sequence Ballet). There are 1780 blocks in the scheme.

We have also carried out comparisons for the test depth image Breakdancing. For this image, it is possible to compare our results with those presented in ref. [4] (cf. Fig. 3b in [4]). The rate-distortion curves for our method ('Proposed') and two methods described in [4] ('Piecewise linear functions with coefficient prediction' and 'Piecewise linear functions') are given in Figure 5. As seen, our method performs better, especially for large compression ratios.

An example of PS obtained by the proposed compression method is shown in Fig. 6. As seen in the figure, sharp discontinuities are well localized by small-size blocks with anisotropic configurations adapted to the discontinuity dominant orientation. Due to this, such discontinuities are preserved considerably better than by other coders based on orthogonal transforms [11].

Finally, in Figure 7, we present examples of reconstructed depth maps compressed with fixed bpp=0.05. The image resulting from the proposed method is characterized by better preservation of sharp edges.

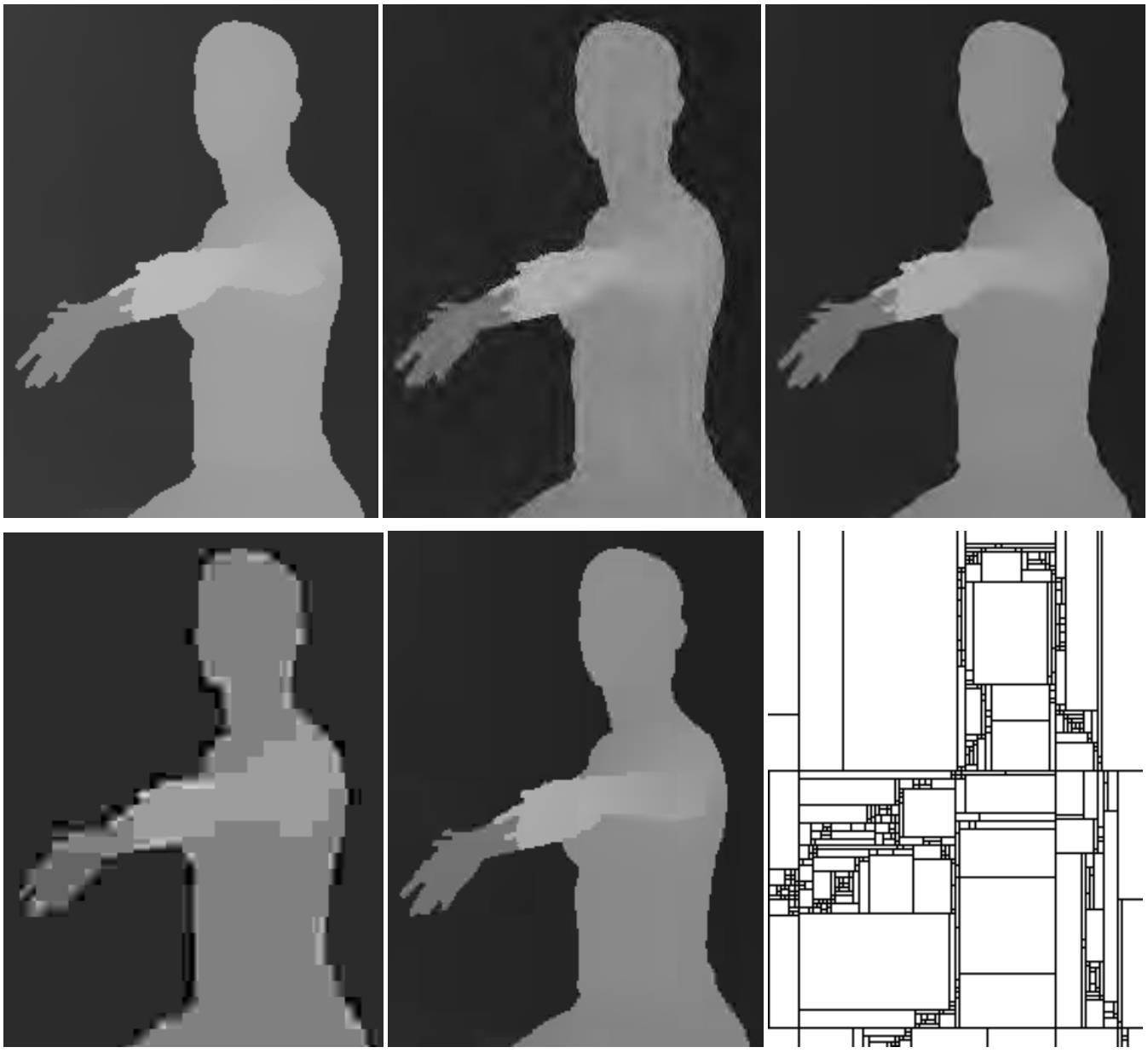

Figure 7. Comparison of compression methods, image compressed with 0.05 bpp. Clockwise from top to bottom: Fragment of depth image in Figure 3; JPEG2000; ADCTC; JPEG; proposed method; partition scheme for the fragment

6. CONCLUSIONS

Images representing depth maps differ from standard images and specific approaches are needed for their compression. The proposed compression method takes into account and exploits specific features of depth maps. It suggests finding anisotropic tree decomposition (PS) and applying simple linear approximation to the blocks at the tree nodes. The designed coder provides high quality of compressed depth maps which is better than for JPEG2000 and ADCTC coders for typical bpp. The results are comparable with the method in [4]. The advantages of the proposed coder consist in providing fast and computationally simple decoding and offering possibility to optionally control either quality or bit-rate.

Future studies might deal with more efficient compression of plane approximation data in blocks and with using non-uniform quantization depending upon block size. It seems possible to find more effective ways for describing object shapes than to apply horizontal-vertical PS. In this paper, no efforts have been spent to compress PS data (tree topology). Typically, it occupies up to 30-35% of the total bit-rate. It is also worth studying how to combine the proposed method with motion compensation for inter-frame coding. We also plan to thoroughly analyze the effect of the proposed depth compression method on view synthesis.

7. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence program (2006 - 2011) and by EC within FP7 (Grant 216503 with the acronym MOBILE3DTV).

8. REFERENCES

[1] G. Peyre and S. Mallat, “Surface compression with geometric bandelets,” in ACM SIGGRAPH, July 2005, vol. 24, pp. 601–608.

[2] D. Donoho, “Wedgelets: nearly minimax estimation of edges,” Annals of Statistics, vol. 27, no. 3, pp. 859–897, March 1999.

[3] R. M.Willett and R. D. Nowak, “Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging,” IEEE Trans. on Medical Imaging, vol. 22, pp. 332–350, March 2003.

[4] Y. Morvan, D. Farin, P. H. N. de With, “Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images”, ICIP 2007, Vol. 5, pp.105-108.

[5] A. E. Jacquin, "Fractal image coding based on a theory of iterated contractive image transformations", Proc. SPIE: Vis. Commun. Image Process., M.Kunt, Ed., Lausanne, Switz., vol.1360, pp.227-239, Oct. 1990.

[6] D. Saupe, R. Hamzaoui, H. Hartenstein, "Fractal image compression - An introductory overview", in: Fractal Models for Image Synthesis, Compression, and Analysis, D. Saupe, J. Hart (eds.), ACM SIGGRAPH'96 Course Notes.

[7] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," ACM SIGGRAPH and ACM Trans. on Graphics, Los Angeles, CA, Aug. 2004, pp. 600-608.

[8] G.K. Wallace, The JPEG Still Picture Compression Standard, Comm. of the ACM, Vol. 34, 1991, pp. 30-44.

[9] D. Taubman, M. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice. Boston: Kluwer, 2002.

[10] N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, ADCTC: A new high quality DCT based coder for lossy image compression, CD ROM Proceedings of LNLA, Switzerland, August 2008, 6 p.

[11] O.V. Tsymbal, V.V. Lukin, N.N. Ponomarenko, A.A. Zelensky, K.O. Egiazarian, J.T. Astola, Three-state locally adaptive texture preserving filter for radar and optical image processing, Eurasip Journal of Applied Signal Processing, issue 8, pp. 1185-1204, 2005.

[12] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, A. Koz, ‘Coding Algorithms for 3DTV—A Survey’, IEEE Transaction on Circuits and Systems for Video Technology, vol. 17, pp. 1606-1621, 2007.

[13] C. Fehn. Depth-Image-Based Rendering (DIBR), Compression, and Transmission for a Flexible Approach on 3DTV, Dissertation, Technical University, Berlin. Mensch & Buch Verlag, Berlin, Germany, 2006.

[14] A. Tikanmäki, A. Gotchev, A. Smolic, K. Müller, “Quality assessment of 3D video in rate allocation experiments”, Proceedings of IEEE International Symposium on Consumer Electronics ISCE 2008, Algarve, Portugal, 14-16 April 2008.