

Technical University of Denmark



Edge-preserving Intra Depth Coding based on Context-coding and H.264/AVC

Zamarin, Marco; Salmistraro, Matteo; Forchhammer, Søren; Ortega, Antonio

Published in:

2013 IEEE International Conference on Multimedia and Expo (ICME)

Link to article, DOI:

[10.1109/ICME.2013.6607516](https://doi.org/10.1109/ICME.2013.6607516)

Publication date:

2013

[Link back to DTU Orbit](#)

Citation (APA):

Zamarin, M., Salmistraro, M., Forchhammer, S., & Ortega, A. (2013). Edge-preserving Intra Depth Coding based on Context-coding and H.264/AVC. In 2013 IEEE International Conference on Multimedia and Expo (ICME) IEEE. DOI: 10.1109/ICME.2013.6607516

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

EDGE-PRESERVING INTRA DEPTH CODING BASED ON CONTEXT-CODING AND H.264/AVC

Marco Zamarin¹, Matteo Salmistraro¹, Søren Forchhammer¹, Antonio Ortega²

¹ Dept. of Photonics Eng., Technical University of Denmark, Denmark

² Ming Hsieh Dept. of Electrical Eng., University of Southern California, CA (USA)
{mzam, matsl, sofo}@fotonik.dtu.dk, ortega@sipi.usc.edu

ABSTRACT

Depth map coding plays a crucial role in 3D Video communication systems based on the “Multi-view Video plus Depth” representation as view synthesis performance is strongly affected by the accuracy of depth information, especially at edges in the depth map image. In this paper an efficient algorithm for edge-preserving intra depth compression based on H.264/AVC is presented. The proposed method introduces a new Intra mode specifically targeted to depth macroblocks with arbitrarily shaped edges, which are typically not efficiently represented by DCT. Edge macroblocks are partitioned into two regions each approximated by a flat surface. Edge information is encoded by means of context-coding with an adaptive template. As a novel element, the proposed method allows exploiting the edge structure of previously encoded edge macroblocks during the context-coding step to further increase compression performance. Experiments show that the proposed Intra mode can improve view synthesis performance: average Bjøntegaard bit rate savings of 25% have been reported over a standard H.264/AVC Intra coder.

Index Terms— Block-based depth compression, context-coding, edge-based depth representation, video-plus-depth, depth-image-based-rendering.

1. INTRODUCTION

In the recent years the interest in three-dimensional (3D) video technologies has grown considerably in both the academic and industrial worlds. A number of 3D-capable solutions and products are becoming available on the consumer market. One of the key challenges in the implementation of a 3D Video communication system is the decoupling of the capture and transmission format from the display format in order to allow a multitude of acquisition, transmission, and display devices to work together seamlessly [1]. One representation that enables such decoupling is the so-called “Multi-view Video plus Depth” (MVD), in which depth or disparity information is provided together with typically 2 or 3 views. By using depth information together with the input views, the

desired output views can be synthesized at the decoder side thus allowing different 3D display devices to operate properly. Efficient coding solutions based on the MVD format are currently being developed by the 3DV group of MPEG [2]. Solutions compatible with both the current H.264/AVC and the upcoming High Efficiency Video Coding (HEVC) standards are being investigated.

As view synthesis algorithms based on Depth-Image-Bases-Rendering (DIBR) typically show a high sensitivity to depth inaccuracies [3], depth coding plays a crucial role in the development of an effective 3D Video system based on the MVD format. Specifically, depth edges should be preserved in order to avoid the appearance of annoying unnatural artifacts in the synthesized views. Due to the fact that standard DCT-based approaches fail to efficiently represent sharp edges, a number of specialized algorithms have been proposed in the literature to cope with this problem, e.g. based on edge-adaptive transforms and transform domain sparsification [4] or local explicit coding of edge information in blocks with sharp discontinuities [5, 6]. An overview of some recent works is provided in the next section.

In this paper a novel algorithm for efficient edge-aware intra depth coding is presented. The proposed scheme is based on the H.264/AVC Intra framework and operates at a MacroBlock (MB) level. While DCT fails to efficiently represent blocks with arbitrarily shaped edges, it provides very compact representations in the case of blocks with (nearly-)uniform values or smooth gradients. Therefore, we propose to modify the reference encoder by adding one Intra mode specifically targeted to edge MBs. When the new mode is selected, the MB is partitioned into two regions and a constant value is assigned to each region and encoded. Partitioning information also needs to be transmitted in order to allow for a proper reconstruction. This is done by encoding a per-pixel binary mask by means of context-coding. The proposed algorithm allows exploiting previously-encoded edge MBs in order to predict the constant values of the current MB being encoded. Moreover, the binary masks of adjacent edge MBs can be combined and jointly context-coded to allow further bit rate savings.

The remainder of this paper is organized as follows. In Section 2 related works on edge-based depth compression are briefly discussed. Section 3 describes the proposed coding method highlighting the novelties it introduces. Section 4 discusses the coding performance of the proposed framework on a number of 3D video test sequences.

2. RELATED WORK

As mentioned in Sec. 1, a number of edge-preserving depth coding algorithms have recently been proposed in the literature. Most of them are motivated by the fact that edge information is critical for the achievement of high view synthesis performance and therefore should be preserved in the coding process.

Shen *et al.* [7] introduced a set of edge-adaptive transforms for block-based depth coding. Edge detection was used to identify depth discontinuities and define a local graph structure to be transformed. Even though some view synthesis improvements were achieved, the explicit calculation of the eigenvectors of the graph Laplacian made the algorithm not suitable for fast implementations. Cheung *et al.* [8, 9] introduced the concept of *don't care region*, exploited to obtain sparse depth representations and improve the coding performance of a JPEG coder. The concept was then combined with the edge-adaptive transforms previously mentioned to achieve further improvements [4]. In [10] Kim *et al.* proposed to use graph based transforms as an alternative to DCT for 4×4 edge blocks within the H.264/AVC framework. Significant bit rate savings are reported, but again the calculation of the graph Laplacian caused a considerable increase of the overall algorithm complexity. Morvan *et al.* [11] proposed to extend the wedgelets - which locally approximate a signal with two constant functions separated by a straight line - with piecewise-linear functions (platelets) defined using a quadtree decomposition, reporting good results in terms of depth compression and edge preservation.

Block-based depth coding exploiting co-located texture edge information has been proposed by Merkle *et al.* in [12] for the case of HEVC. When texture is exploited to bi-partite a depth edge block, a constant depth value is assigned to each partition. As the same depth partitioning can be reproduced at the decoder side, only the two constant values need to be encoded. Depth image coding exploiting texture edge information has also been proposed by Milani *et al.* in [13]. In this case, segmentation of reconstructed texture data is exploited to predict shapes in the depth image. Within each segment the depth signal is approximated by a linear function and only the corresponding coefficients are transmitted, thus avoiding the explicit encoding of edge information.

Shimizu *et al.* [5] proposed a depth coding scheme based on H.264/AVC similar to the one proposed in this paper: edge macroblocks are approximated by a palette with two entries and a (binary) object shape map, both predictively

encoded exploiting intra/inter correlation. Shape maps are generated by minimizing a prediction error function and encoded by means of context-adaptive binary arithmetic coding as in shape coding in MPEG-4 Part 2. Even though intra/inter neighboring blocks are exploited in order to properly define the context at any pixel position, the edge structure of previously encoded edge macroblocks is not directly exploited to improve the coding efficiency of context coding. If the edge structure correlation of neighboring edge macroblocks is taken into account as in the method proposed in this paper, lower bit rates can be achieved for the coding of shape information, which can be of benefit especially at low bit rates.

A different approach based on a similar framework has been proposed for the case of HEVC by Lan *et al.* in [6]. In this case a wider range of block sizes are exploited, namely from 32×32 down to 4×4 . Blocks are partitioned in up to 8 regions each approximated by a flat surface. Arithmetic coding is then employed to encode the region map and surface values. Significant gains are reported for the cases of both estimated and acquired depth sequences. However, higher performance are expected if the region map overhead is reduced. This can be achieved by exploiting the correlation between the edge structures of the current and previously encoded region maps during the encoding process as proposed in this work.

3. PROPOSED METHOD

As mentioned in Sec. 1, a new Intra mode for H.264/AVC - referred to as "EDGE" in the following - able to efficiently represent arbitrarily-shaped edges is proposed. Differently from other approaches, the proposed mode aims at combining the encoding of depth discontinuities that span multiple adjacent MBs in order to increase the coding efficiency of edge information. The method operates at a MB level as part of the Rate-Distortion (RD) optimization strategy: the EDGE mode is tested together with the standard intra modes and the mode with the lowest RD cost is selected.

The EDGE mode partitions the whole depth MB into two regions and associates one Constant Value (CV) to each of them. A per-pixel binary map M is used to identify pixels belonging to the two regions. In order to allow a perfect reconstruction at the decoder side, the mask M is losslessly encoded by means of context-coding together with the two CVs. A detailed description of the EDGE algorithm for a depth MB is provided below (see Fig. 1).

1. The input depth MB is partitioned into two regions. A 3D-point is associated to each pixel from its spatial coordinates and depth value; pixels are then grouped with neighbors if corresponding 3D-points are closer than a distance d , initially set to 1. If more than two partitions are formed, d is increased and the procedure iterated. A 16×16 binary mask M is defined.

2. One CV is associated to each region of M by selecting the median value of the corresponding pixels in the input depth MB.¹
3. The mask M is losslessly encoded by means of context-based arithmetic coding (see Subsection 3.1 for more details). Let R_M be the number of bits spent to encode M , R_{CV} the number of bits spent to encode the two CVs, and $R = R_M + R_{CV}$ the total number of bits.
4. The distortion D is computed as MSE between the input depth MB and the binary EDGE MB.
5. The EDGE RD cost is computed as follows: $RD_{EDGE}^{intra} = D + \lambda \cdot R$, λ being the same Lagrangian multiplier used in the RD cost calculation of the standard Intra modes.
6. If the left-neighbor MB is available and encoded as EDGE MB:
 - (a) Steps 3-5 are repeated using the same CVs used in the left-neighbor MB. In this way no bits are spent to encode the CVs for the current MB. Let RD_{EDGE}^{left1} be the corresponding RD cost.
 - (b) Steps 3-5 are repeated using the same CVs and coding parameters for the mask M used in the left-neighbor MB (see Subsection 3.1). Let RD_{EDGE}^{left2} be the corresponding RD cost.
7. As Step 6 but using the top-neighbor MB. If it is available and encoded as EDGE MB, the costs RD_{EDGE}^{top1} and RD_{EDGE}^{top2} are defined.
8. The RD_{EDGE} cost is defined as the lowest cost among RD_{EDGE}^{left1} , RD_{EDGE}^{left2} , RD_{EDGE}^{top1} , RD_{EDGE}^{top2} (if available), and RD_{EDGE}^{intra} .

If RD_{EDGE} is lower than the RD cost obtained with standard H.264/AVC Intra coding, the EDGE mode is selected and the corresponding data (mask M , CVs and coding parameters) are included in the bit stream. At the decoder side, if the usage of the EDGE mode is detected, the binary mask is decoded by means of context-based decoding, CVs are decoded from the bit stream and the output EDGE MB is produced. If the EDGE mode is not used, standard Intra decoding is performed.

3.1. Binary mask encoding

The encoding of a binary mask M is done by means of context-based arithmetic coding. The encoding can be done in two different ways: *intra* (i.e. without exploiting previously-encoded EDGE MBs) and *inter* (i.e. exploiting previously-encoded EDGE MBs in the same slice). The two methods are described here in detail.

¹The choice of the median over the mean - even though not necessarily optimal from a MSE point of view - reduces the influence of outliers, i.e. noisy pixels in the MB partition.

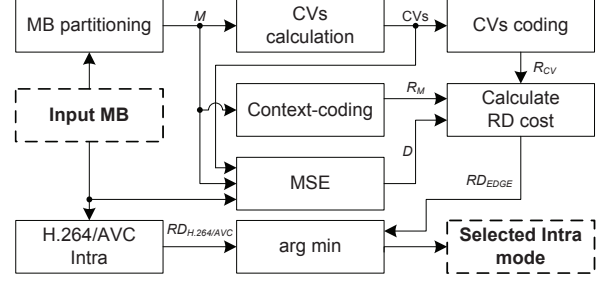


Fig. 1: Encoder block scheme

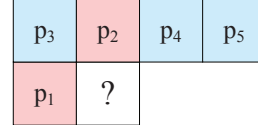


Fig. 2: Template selection: pixels p_1 and p_2 are fixed; one additional pixel among p_3 , p_4 , and p_5 can be included in the template. “?” indicates the current pixel being encoded.

3.1.1. Intra EDGE coding

In order to make the context prediction efficient, it is important to select the most relevant template pixels for the edge structure of the mask M being encoded. For this reason, an adaptive template inspired by [14] is used. The template is defined among a set of 5 candidate pixels (see Fig. 2) and has thus a smaller size than the one used in [14] and in MPEG-4 Part 2 shape coding. As shown in Fig. 2, the template includes the pixels on the left and on top of the one being encoded. A third pixel chosen among a set of three candidates can be included if of benefit in terms of code length: the binary mask is encoded 4 times - with the 2-pixel template and with the three 3-pixel templates - and the template providing the lower rate is selected.

In order to allow the decoder to correctly decode the mask M , the selected template is explicitly signalled in the bit stream. This is done by encoding two parameters: the number of pixels in the template and, in case of a 3-pixel template, the third pixel index. The bits needed to encode these parameters (referred to as $R_{\#}$ and R_p , respectively) are also considered in the template selection process.

For the *intra* case, the EDGE rate R introduced in Section 3, Step 3 is therefore given by

$$R^{intra} = R_{\#} + R_p + R_M + R_{CV} + R_{flag}^{i/p}, \quad (1)$$

where $R_{flag}^{i/p}$ is the rate of a flag encoded only when at least one between the left and top neighboring MBs is available and encoded as EDGE MB, to signal whether *intra* or *inter* EDGE coding is used.

3.1.2. Inter EDGE coding

As specified in Section 3, Steps 6 and 7, if the left or top neighboring MB is available and encoded as EDGE MB, the

current MB can be inter encoded. This can be done in two ways referred to as *partial inter coding* and *full inter coding*, corresponding to Steps 6a-6b, and 7a-7b, respectively.

In the first case the encoding is done as in the intra case with the only difference that the CVs are not calculated and encoded but simply copied from the predicting MB, under the assumption that adjacent parts of the same edge refer to the same background and foreground objects. In this case the EDGE rate is given by

$$R^{p.inter} = R_{\#} + R_p + R_M + R_{flag}^{i/p} + R_{flag}^{p/f} + R_{flag}^{l/t}, \quad (2)$$

where $R_{flag}^{p/f}$ is the rate of a flag specifying whether *partial* or *full* inter coding is used, and $R_{flag}^{l/t}$ is the rate of a flag indicating which of the two neighboring MBs is used for prediction when both of them are available and encoded as EDGE MBs.

When *full inter coding* is used, together with the CVs also the template pixels are copied from the predicting MB. Moreover, since the context-coding will be based on the same template, it is possible to use the edge statistics of the predicting MB - which are stored for each EDGE MB after the encoding of the mask M - as a starting point for the context-coding, thus allowing to exploit the statistics of the edge in the predicting MB. When *full inter coding* is used, adjacent blocks are therefore encoded as part of a unique binary mask. Note that edge statistics do not need to be explicitly transmitted as they can be generated during the decoding process. Finally, in order to properly initialize the boundaries of the binary mask M for the context coding step, the encoded binary mask of the predicting MB is used (the same is done for the *partial inter coding* too). In case of *full inter coding* the EDGE rate is given by

$$R^{f.inter} = R_M + R_{flag}^{i/p} + R_{flag}^{p/f} + R_{flag}^{l/t}. \quad (3)$$

The availability of these two inter modes allows for efficient representation of depth MBs that separate between the same background and foreground of the previous MB (i.e. they share the same CVs) and present different edge direction/structure (*partial inter coding*) or similar structure (*full inter coding*).

4. EXPERIMENTAL RESULTS

The proposed method has been implemented on the H.264/AVC reference software JM 17.1 and evaluated on the following 3D video sequences: *Ballet* and *Breakdancers* [15], *Book Arrival*, *Lovebird1*, *Dancer*, *Cafe*. The first 15 frames of each sequence have been encoded. Two views have been selected for each sequence (reported in Table 1) and the corresponding depth images have been encoded with both H.264/AVC Intra and the proposed method. Depth maps have been encoded with every second QP in the interval (24, 48). All the standard Intra modes of H.264/AVC have been

Table 1: Sequence resolutions and input left (L), virtual (V) and right (R) views.

Sequence	Resolution	L	V	R
Ballet	1024 × 768	5	4	3
Breakdancers	1024 × 768	5	4	3
Book Arrival	1024 × 768	8	7	6
Lovebird1	1024 × 768	6	7	8
Dancer	1920 × 1088	2	3	5
Cafe	1920 × 1088	2	3	4

enabled and RD-based mode decision was selected for the Intra 16 × 16 mode. In order to avoid introducing blurring artifacts, the deblocking filter has been disabled. In order to evaluate the effectiveness of the proposed algorithm, not only the RD performance on the depth signals have been evaluated (Figs. 3a and 3b for the sequences *Ballet* and *Breakdancers*), but also the view synthesis performance using reconstructed depth data and uncompressed texture as done in [16] (Figs. 3c and 3d) have been considered. In this case, PSNR values are measured against reference virtual views synthesized from uncompressed depth and texture data. Virtual views have been synthesized using the MPEG VSRS 3.5.

Table 2 reports the Bjøntegaard bit rate savings [17] between the proposed method and H.264/AVC for both the depth signals and the synthesized views. As it can be noticed, when comparing the view synthesis performance the proposed method improves depth compression efficiency for all the sequences, confirming that it is of benefit in a MVD scenario. However, the improvement depends on the depth accuracy of the particular test sequence and on its resolution. Major bit rate savings in terms of both depth coding and view synthesis performance are reported in the case of *Ballet* and *Breakdancers* due to the sharp and clean edges that characterize depth images of these two sequences. Experimental results show that the proposed *inter EDGE coding* is often selected: the binary masks of adjacent edge macroblocks are therefore jointly encoded as part of a single binary image allowing additional bit rate savings (see Fig. 4b for an example). *Book Arrival* shows a comparable bit rate reduction in terms of view synthesis, but a smaller gain in terms of depth compression due to the presence of less sharp object discontinuities in the depth images which favor DCT coding. In the case of *Lovebird1*, a smaller gain in terms of synthesis performance is observed, together with a minor increase of the depth bit rate. This sequence shows that even though the proposed algorithm might not always be more efficient than H.264/AVC in terms of depth MSE, it does provide a benefit when reconstructed depth data are used for view synthesis.

The performance on the two high definition sequences are also satisfactory but lower gains are noticed, even for the *Dancer* sequence which features high quality synthetic depth data. One reason can be found in the fact that edge mac-

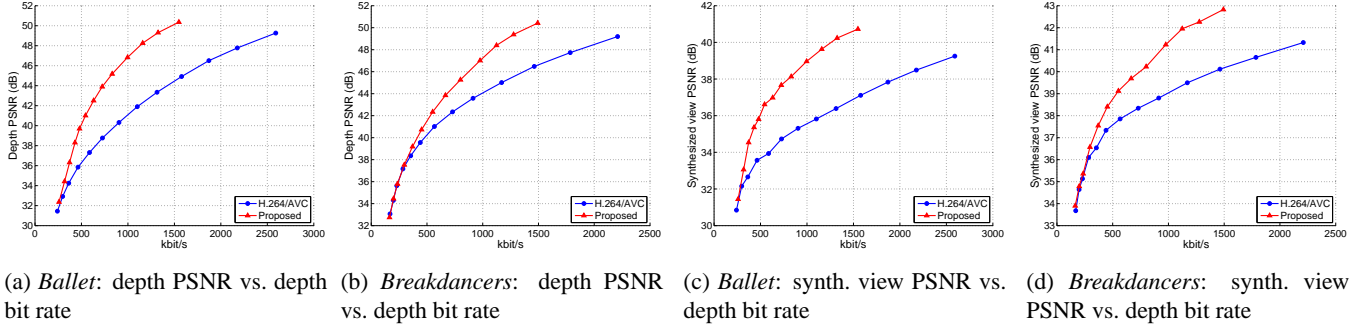


Fig. 3: Rate-Distortion performance comparisons

Table 2: Bjøntegaard bit rate savings (BDBR) between the proposed method and H.264/AVC for depth signals and corresponding synthesized views. Note that the minus sign means bit rate reduction.

Sequence	Depth BDBR (%)	Synth. BDBR (%)
Ballet	-38.33	-46.82
Breakdancers	-21.94	-34.54
Book Arrival	-7.91	-34.39
Lovebird1	1.45	-10.78
Dancer	-3.60	-20.84
Cafe	0.45	-4.21
Average	-11.65	-25.26

roblocks in high resolution depth images present on average less arbitrarily shaped discontinuities. Therefore, the standard H.264/AVC Intra modes can predict well the edge structures most of the times. For high resolution data, bigger block sizes - such as those defined in the upcoming HEVC standard - would hence be beneficial for the proposed algorithm especially at very low bit rates, where the overhead due to DCT/EDGE signalling can significantly affect the RD performance (as in the *Dancer*, *Cafe*, *Book Arrival*, and *Lovebird1* sequences in which H.264/AVC slightly outperforms the proposed method at very low bit rates).

Figures 4 and 5 provide a visual quality comparison between H.264/AVC Intra and the proposed method for both depth and synthesized data at similar bit rates, highlighting the advantages of the latter over the former.

Coding performance considering total bit rate (texture plus depth) versus synthesized view PSNR have also been evaluated, as done in [18]. Even though a direct comparison with the plane segmentation based coding method in [18] is not possible due to different reference settings and test conditions, the proposed method shows similar Bjøntegaard bit rate savings over the reference encoder for the common sequences, ranging between -11.50% and -3.98%.

Finally, Fig. 6 compares the RD performance of the proposed approach in terms of depth quality against H.264/AVC,

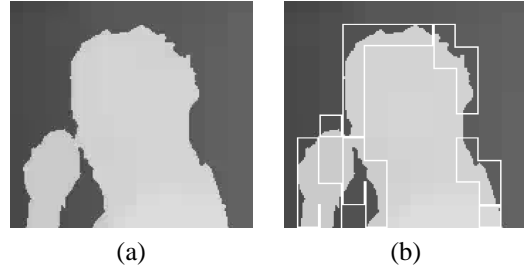


Fig. 4: Reconstructed depth comparison: detail of the Ballet sequence, left view, frame 0. (a) H.264/AVC Intra, 38.85 dB @ 361 kbit/s, (b) Proposed, 43.83 dB @ 366 kbit/s. White squares indicate MBs that have been encoded as EDGE MBs. EDGE MBs encoded by means of EDGE prediction are connected to the corresponding prediction MBs.

JPEG2000, the method proposed by Milani *et al.* in [13], and the Platelet-based algorithm proposed by Morvan *et al.* in [11]. The comparison is made on the *Teddy* depth image from the stereo sets at [19]. As it can be noticed, the proposed method is able to outperform H.264/AVC Intra, the planar fitting method proposed in [13], which exploits reconstructed texture data in order to predict depth shapes, and the Platelet-based scheme. JPEG2000 appears as the least efficient still image coder in the case of depth images due to the heavy ringing artifacts it introduces.

Compared with Shimizu *et al.* [5] in which the number of bits needed to encode a shape/binary mask is around 50 per MB, the proposed method exhibits a more flexible behavior thanks to the *intra/inter EDGE coding* methods. The total number of bits needed to encode an EDGE MB (including binary mask, constant values, and flags) ranges from an average of 51 at high rates to an average of 45 at low rates, where *inter EDGE coding* is often selected.

5. CONCLUSIONS AND FUTURE WORK

In this paper a novel edge-preserving H.264/AVC Intra mode for efficient depth coding has been presented. Edge macroblocks are partitioned in two regions each approximated

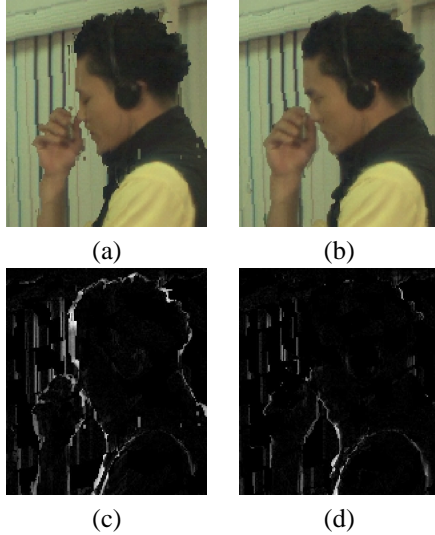


Fig. 5: Visual quality comparison versus total depth bit rate: detail of the synthesized view from uncompressed texture, Ballet sequence, frame 0. H.264/AVC Intra, 34.73 dB @ 726 kbit/s (a), Proposed, 37.67 dB @ 723 kbit/s (b), absolute errors on the luminance components (with same scale) between synthesized views obtained with uncompressed depth data and H.264/AVC Intra (c) and Proposed (d).

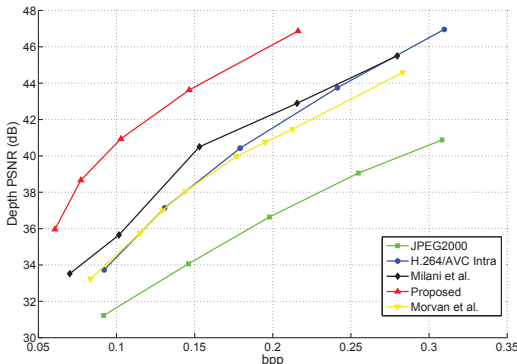


Fig. 6: Teddy depth image: RD performance comparison

by a flat surface. A binary mask identifying the two regions is defined and encoded by means of adaptive context-coding exploiting previously encoded edge macroblocks to increase the compression efficiency. Experimental results show that adjacent edge macroblocks can be jointly encoded in an efficient manner. Compared with a standard H.264/AVC Intra coder, the proposed algorithm achieved average Bjøntegaard bit rate savings of about 12% in terms of depth compression, and about 25% in terms of synthesized view quality versus depth bit rate. The complexities of the proposed and standard Intra modes are comparable, thus the overall complexity is not affected significantly. Future developments include binary mask pre-filtering for efficient low bit rate coding, prediction from non-EDGE macroblocks and quantization of flat surface values, and extension to depth video coding.

6. REFERENCES

- [1] L. Onural, *3D Video Technologies: An Overview of Research Trends*, Society of Photo Optical, 2010.
- [2] ISO/IEC JTC1/SC29/WG11, "Applications and Requirements on 3D Video Coding," Doc. N12035, Geneva (CH), Mar. 2011.
- [3] K. Müller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," *Proc. of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.
- [4] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth Map Coding using Graph based Transform and Transform Domain Sparsification," in *IEEE MMSP 2011*, Oct. 2011, pp. 1–6.
- [5] S. Shimizu, H. Kimata, S. Sugimoto, and N. Matsuura, "Block-adaptive Palette-based Prediction for Depth Map Coding," in *IEEE ICIP 2011*, Sep. 2011, pp. 117–120.
- [6] C. Lan, J. Xu, and F. Wu, "Improving Depth Compression in HEVC by Pre/Post Processing," in *2012 Int'l Work. on Hot Topics in 3D Multim. (Hot3D 2012)*, July 2012, pp. 611–616.
- [7] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive Transforms for Efficient Depth Map Coding," in *PCS 2010*, Dec. 2010, pp. 566–569.
- [8] G. Cheung, A. Kubota, and A. Ortega, "Sparse Representation of Depth Maps for Efficient Transform Coding," in *Picture Coding Symposium (PCS) 2010*, Dec. 2010, pp. 298–301.
- [9] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform Domain Sparsification of Depth Maps using Iterative Quadratic Programming," in *IEEE ICIP 2011*, Sep. 2011, pp. 129–132.
- [10] W.-S. Kim, S.K. Narang, and A. Ortega, "Graph based Transforms for Depth Video Coding," in *IEEE ICASSP 2012*, Mar. 2012, pp. 813–816.
- [11] Y. Morvan, D. Farin, and P. de With, "Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images," in *Proc. of IEEE ICIP 2007*, 2007.
- [12] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Wiegand, "3D Video: Depth Coding based on Inter-component Prediction of Block Partitions," in *Picture Coding Symposium (PCS) 2012*, May 2012, pp. 149–152.
- [13] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient Depth Map Compression Exploiting Segmented Color Data," in *IEEE ICME 2011*, July 2011, pp. 1–6.
- [14] M. Zamarin and S. Forchhammer, "Lossless Compression of Stereo Disparity Maps for 3D," in *2012 Int'l Work. on Hot Topics in 3D Multim. (Hot3D 2012)*, July 2012, pp. 617–622.
- [15] L.C. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-Quality Video View Interpolation using a Layered Representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.
- [16] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand, "The Effects of Multiview Depth Video Compression on Multiview Rendering," *Image Commun.*, vol. 24, no. 1-2, pp. 73–88, 2009.
- [17] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-Curves," VCEG-M33, Apr. 2001.
- [18] B.T. Oh, H.-C. Wey, and D.-S. Park, "Plane segmentation based intra prediction for depth map coding," in *PCS 2012*, May 2012, pp. 41–44.
- [19] "Repository vision.middlebury.edu: Stereo datasets," <http://vision.middlebury.edu/stereo>.