2012 Eighth International Conference on Signal Image Technology and Internet Based Systems

EFFICIENT DEPTH IMAGE COMPRESSION USING ACCURATE DEPTH DISCONTINUITY DETECTION AND PREDICTION

Reuben A. Farrugia

Department of Communications and Computer Engineering
University of Malta
Msida, Malta

ABSTRACT

This paper presents a novel depth image compression algorithm for both 3D Television (3DTV) and Free Viewpoint Television (FVTV) services. The proposed scheme adopts the K-means clustering algorithm to segment the depth image into K segments. The resulting segmented image is losslessly compressed and transmitted to the decoder. The depth image is then compressed using a bi-modal block encoder, where the smooth blocks are predicted using direct spatial prediction. On the other hand, blocks containing depth discontinuities are approximated using a novel depth discontinuity predictor. The residual information is then compressed using a lossy compression strategy and transmitted to the receiver. Simulation results indicate that the proposed scheme outperforms the state of the art spatial video coding systems available today such as JPEG and H.264/AVC Intra. Moreover, the proposed scheme manages to outperform specialized depth image compression algorithms such as the one proposed by Zanuttigh and Cortelazzo.

Index Terms— 3D video compression; depth-image compression; predictive coding; segmentation

1. INTRODUCTION

The standardization of Multiview Video Coding (MVC) [1] and recent advances in multimedia technology have enabled new applications such as Free Viewpoint Television (FVTV) and 3D Television (3DTV). FVTV allows the user to interactively select the viewpoint of interest while 3DTV offers a three dimensional immersive experience [2]. These applications are currently being introduced to home users through different channels including Blu-ray discs, cable TV, satellite TV, and Internet streaming services [3].

These applications have pushed the need for video plus depth representations, where the video signal and the corresponding per pixel depth images are made available to the user [4]. Stereo images and virtual views can be generated at the decoder using both video and corresponding depth information. However, given the huge amount of storage space and bandwidth demanded by these services, both video and depth signals need to be efficiently compressed. This has inspired MPEG to issue a Call for Proposals (CfP) to identify methodologies to improve both compression and the quality of the reconstructed views [5].

A naive approach is to consider depth images as monochromatic video signals and compress them using state of the art video coding systems. The depth information can be efficiently compressed at 10-20% of the bit rate needed to compress color video [6]. However, significantly larger storage space and bandwidth might be required when compressing more complicated depth information [4]. Moreover, it has been observed that the human vision system (HVS) is more sensitive to edge discontinuities than to smooth regions [7]. This suggests that novel approaches need to be investigated in order to improve the quality of both FVTV and 3DTV services.

Various depth image compression algorithms can be found in literature. One of the first approaches presented in [8] employed region of interest (ROI) coding with adaptive dynamic range allocation. This method demonstrated that higher quality can be perceived when employing compression strategies which preserve depth boundaries. The authors in [9, 10] have modeled edge discontinuities using piecewise-linear approximations while triangular meshes were adopted instead in [11]. More recently, the authors in [12] have presented an adaptive geometric based Intra predictor. Nonetheless, these approximations do not adequately model depth discontinuities and produce flattened surfaces which cause annoying rendering artifacts. Jager [13] has employed contour based segmentation to model depth object boundaries. However, this method fails to outperform state of the art codecs at low rates. Adaptive up-sampling filters were adopted in [14, 15]. These methods were found to perform well at low rates but their performance drops at high rates. Compressive sensing was employed in [16] where although obtaining lower Peak Signal-to-Noise Ratio (PSNR) relative to JPEG and JPEG2000, it better preserves depth discontinuities.

The correlation between video and depth information was exploited through color segmentation in [17, 18]. However, the performance of the method presented in [17] significantly

978-0-7695-4911-8/12 $26.00 © 2012 IEEE
DOI 10.1109/SITIS.2012.15

drops at high rates while the method published in [18] manages to outperform H.264/AVC Intra at all rates. However, the edge discontinuities present in depth and color images are generally not perfectly aligned, which might cause inconsistencies when employing color segmentation. Another segmentation based algorithm was presented by [19], where this time the segmentation is performed directly on the depth image. This method manages to preserve the depth discontinuities and outperforms state of the art codecs at all rates.

This paper presents a depth map compression algorithm which extends the capabilities of the Zanuttigh-Cortelazzo [19] approach. The proposed mechanism manages to preserve the depth discontinuities resulting in higher quality reconstructed views. The proposed method manages to outperform the Zanuttigh-Cortelazzo approach where it can achieve significant lower rates at the same rendering quality. Moreover, the proposed system manages to achieve PSNR gains of up to 4dB and 0.5dB relative to JPEG and H.264/AVC Intra respectively. Subjective results indicate that, unlike traditional transform based coding schemes, the proposed method is able to generate higher quality intermediate views, even at low rates.

This paper is structured as follows: The proposed system is presented in Section 2. The simulation environment considered is presented in Section 3 together with the results obtained. The final comments and concluding remarks are delivered in Section 4.

2. SYSTEM OVERVIEW

The depth map compression algorithm presented in this paper is illustrated in Fig. 1. The depth image $\mathbf{D}$ is first segmented using the K-means clustering algorithms [20], where K represents the number of segments. The resulting segmented image $\mathbf{S}$ and the depth image $\mathbf{D}$ are divided into non-overlapping blocks $\mathbf{S}_{m,n}$ and $\mathbf{D}_{m,n}$ respectively, of size $N \times N$ pixels, where m, n represent the row and column block indices respectively.

The segmented image $\mathbf{S}$ is compressed using a lossless compression algorithm and transmitted to the receiver. This information is required by both encoder and decoder in order to select the prediction mode and to compute the prediction block $\mathbf{P}_{m,n}$. $\mathbf{P}_{m,n}$ is predicted using the *Spatial Prediction* mode if the depth block $\mathbf{D}_{m,n}$ is smooth while the *Depth Discontinuity Prediction* function is used if the block contains depth discontinuities. The *Mode Selection* function adopts the content of the block $\mathbf{S}_{m,n}$ to determine the prediction mode to be used.

The residual $\mathbf{R}_{m,n}$, computed as the difference between $\mathbf{D}_{m,n}$ and the predicted block $\mathbf{P}_{m,n}$, is compressed using a lossy compression strategy, where the encoded bitstream is transmitted on the channel. The *Lossy Decoder* is then used to derive the residual block $\hat{\mathbf{R}}_{m,n}$ which, assuming normal transmission, will be available at the decoder. The decoded depth block $\hat{\mathbf{D}}_{m,n}$ is derived by adding blocks $\hat{\mathbf{R}}_{m,n}$ and $\mathbf{P}_{m,n}$, and will be used for prediction purposes. This feedback loop was included to ensure perfect synchronization between the encoder and decoder.

2.1. Segmentation Module

The depth compression algorithm presented in this paper can employ any segmentation strategy found in literature. For simplicity, and to minimize the computational resources, the K-means clustering algorithm was adopted in this work. The K-means clustering algorithm defines a vector $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$ of intensity values, where μ_k represents the mean intensity value of cluster k. Consider

$$\mathbf{D} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,X} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,X} \\ \vdots & \vdots & \ddots & \vdots \\ d_{Y,1} & d_{Y,2} & \cdots & d_{Y,X} \end{pmatrix}$$

The K-means clustering algorithm assigns each value $d_{i,j}$ to the cluster which minimizes the Euclidean distance measure $\|d_{i,j} - \mu_k\|$. Therefore, each pixel is assigned to a segment $s_{i,j}$ using

$$s_{i,j} = \underset{k}{\operatorname{argmin}} \|d_{i,j} - \mu_k\| \quad (1)$$

forming the segmented image

$$\mathbf{S} = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,X} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,X} \\ \vdots & \vdots & \ddots & \vdots \\ s_{Y,1} & s_{Y,2} & \cdots & s_{Y,X} \end{pmatrix}$$

The mean vector μ is updated with the mean intensity value of each cluster, such that

$$\mu_k = \frac{1}{|S_k|} \sum_{i,j \in S_k} d_{i,j} \quad (2)$$

where S_k is the set of all pixels assigned to cluster k ($s_{i,j} = k$) and the operator $| \bullet |$ is the cardinality of a set. The algorithm iterates the assignment and updating steps until μ no longer changes.

2.2. Lossless Compression

The segmented image $\mathbf{S}$ can be represented using K binary masks, which can be individually compressed using lossless compression schemes such as JBIG. However, the authors in [19] have demonstrated that higher compression efficiency can be achieved if the bitplanes of $\mathbf{S}$ are encoded instead. In essence, consider a set of $\acute{K} = log_2 K$ masks M_j, where $j \in 1, \ldots, \acute{K}$ represent the bitplane indices. Therefore, instead of encoding K binary images, the Zanuttigh-Cortelazzo

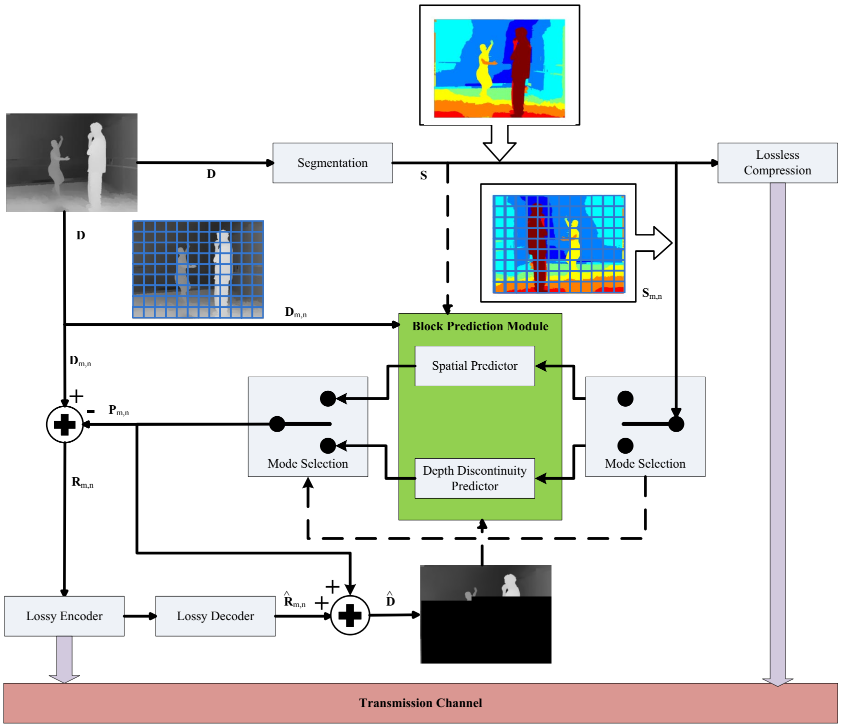

Fig. 1. Schematic diagram of the proposed depth map compression algorithm.

approach only needs to encode $\acute{K}$ binary images using JBIG. This approach reduces the number of masks that need to be compressed and the size of the corresponding compressed data. It is important to notice that no additional side information is required to compute mode decision since the discrimintation between smooth and non-smooth blocks is done based on the corresponding segmented block $\mathbf{S}_{m,n}$.

An alternative approach was inspired by the observation that the segmented block is only needed at the decoder to compute the *Depth Discontinuity Prediction* and the *Mode Decision* functions. Driven by this observation, this work considers the possibility to only transmit the blocks needed by the *Depth Discontinuity Prediction* function while signaling the prediction mode used by each block through the transmission channel. Therefore, this alternative approach converts the segmented blocks needed by the *Depth Discontinuity Prediction* module to a string of characters, and compress it using universal lossless data compression algorithms such as the Lempel-Ziv-Welch [21] and the Deflate algorithm [22].

2.3. Mode Selection

The *Mode Selection* function needs to discriminate between depth blocks $\mathbf{D}_{m,n}$ which are smooth and those containing depth discontinuities. This can be done by exploiting the information contained within the segmented block $\mathbf{S}_{m,n}$, which will also be available at the decoder.

Consider the segmented block

$$\mathbf{S}_{m,n} = \begin{pmatrix} s_{1,1}^{m,n} & s_{1,2}^{m,n} & \cdots & s_{1,N}^{m,n} \\ s_{2,1}^{m,n} & s_{2,2}^{m,n} & \cdots & s_{2,N}^{m,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1}^{m,n} & s_{N,2}^{m,n} & \cdots & s_{N,N}^{m,n} \end{pmatrix}$$

Let $N_s = |\mathbf{S}_{m,n}|$ denote the number of distinct indices within $\mathbf{S}_{m,n}$ and k_p denote the distinct segment indices. In order to reduce the effect of small segments, a segment index k_p is considered a dominant segment index k_ϕ if

$$\frac{\sum_{p=1}^{N_s} n_{k_p}}{N \times N} \geq T_s \quad (3)$$

where n_{k_p} represents the frequency of cluster k_p within block $\mathbf{S}_{m,n}$ and $T_s \in [0,1]$ is a heuristic threshold. Now consider N_ϕ to represent the number of dominant segments within the considered $N \times N$ block. A depth block $\mathbf{D}_{m,n}$ is considered to be smooth if $N_\phi = 1$ and therefore the *Spatial Prediction* function is employed to generate the prediction block $\mathbf{P}_{m,n}$. The *Depth Discontinuity Predictor* function is only selected if $N_\phi > 1$. $T_s = 0.2$ was found to be adequate after several simulations.

2.4. Block Prediction Module

2.4.1. Spatial Predictor

The *Spatial Block Predictor* predicts the depth block $\mathbf{D}_{m,n}$ using the spatially neighboring blocks $\hat{\mathbf{D}}_{m-1,n}$ and $\hat{\mathbf{D}}_{m,n-1}$ available at decoding time. This work considers three simple spatial predictors, and the one which minimizes the Sum of Absolute Difference (SAD) with respect to the original depth block $\mathbf{D}_{m,n}$ is chosen to represent the prediction block $\mathbf{P}_{m,n}$. Consider the following notation to describe pixels within a block

$$\mathbf{Z}_{m,n} = \begin{pmatrix} z_{1,1}^{m,n} & z_{1,2}^{m,n} & \cdots & z_{1,N}^{m,n} \\ z_{2,1}^{m,n} & z_{2,2}^{m,n} & \cdots & z_{2,N}^{m,n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N,1}^{m,n} & z_{N,2}^{m,n} & \cdots & z_{N,N}^{m,n} \end{pmatrix}$$

The *vertical predictor* replicates the last row of the block above the one being encoded. That is, each row of $\mathbf{P_{m,n}}$ is copied from the last row of $\hat{\mathbf{D}}_{m-1,n}$, or $p_{i,j}^{m,n} = \hat{d}_{N,j}^{m-1,n}$ for $j = 1, \dots, N$. Similarly, the *horizontal predictor* replicates the rightmost column of the block to the left of the current one. That is, each column of $\mathbf{P_{m,n}}$ is copied from the last column of $\hat{\mathbf{D}}_{m,n-1}$, or $p_{i,j}^{m,n} = \hat{d}_{i,N}^{m,n-1}$ for $i = 1, \dots, N$. Finally, the *DC predictor* computes the average of the last row of $\hat{\mathbf{D}}_{m-1,n}$ and the last column of $\hat{\mathbf{D}}_{m,n-1}$ and the resulting value is replicated in every row and column of $\mathbf{P}_{m,n}$.

2.4.2. Depth Discontinuity Predictor

Like the *Spatial Predictor* mentioned above, the *Depth Discontinuity Predictor* adopts the spatially neighboring blocks to predict $\mathbf{P}_{m,n}$. However, since the block is employed to predict edge boundaries, the spatial predictors mentioned above will not be able to accurately predict the depth block $\mathbf{D}_{m,n}$. Transmitting the residual error at a high enough fidelity would severly reduce the compression efficiency of the overall system.

The neighbouring segmented blocks $\mathbf{S}_{m-1,n}$,$\mathbf{S}_{m-1,n-1}$, $\mathbf{S}_{m-1,n}$ and $\mathbf{S}_{m-1,n+1}$ are grouped in a set $\mathbf{S}_{\nu}$, while the corresponding depth blocks $\hat{\mathbf{D}}_{m-1,n}$,$\hat{\mathbf{D}}_{m-1,n-1}$, $\hat{\mathbf{D}}_{m-1,n}$ and $\hat{\mathbf{D}}_{m-1,n+1}$ are grouped in a set $\hat{\mathbf{D}}_{\nu}$. The cluster mean of every segment $k \in K$ is computed using

$$\bar{\mu}_k = \frac{1}{|\Gamma_k|} \sum_{i \in \Gamma_k} \hat{\mathbf{D}}_{\nu}(i) \tag{4}$$

where Γ_k represents the set of indices where $\mathbf{S}_{\nu} = k$, and $| \bullet |$ represents the cardinality of a set. The predicted block is obtained using the information contained within $\mathbf{S}_{m,n}$, which is available at the decoder, and adopts the cluster mean to approximate the depth values at each pixel position. This is formally represented using

$$p_{i,j}^{m,n} = \bar{\mu}_k, \text{ where } k = s_{i,j}^{m,n} \tag{5}$$

The prediction block $\mathbf{P}_{m,n}$ was found to be better predicted using the local cluster means mentioned above rather then using the global cluster means μ_k. However, if $k \notin \mathbf{S}_{\nu}$ the pixels $p_{i,j}^{m,n}$ associated to cluster k are predicted using the corresponding global cluster mean.

2.5. Lossy Encoder and Decoder

The residual block $\mathbf{R}_{m,n}$, resulting from the difference between the original depth block $\mathbf{D}_{m,n}$ and the corresponding prediction block $\mathbf{P}_{m,n}$, needs to be compressed and delivered to the receiver. The information contained within the depth block $\mathbf{D}_{m,n}$ is accurately predicted using the methods discussed in sub-section 2.4, even at depth boundaries. Therefore, the residual block $\mathbf{R}_{m,n}$ is expected to be relatively smooth with small pixel intensities, even for blocks containing depth discontinuities. Therefore, the residual block $\mathbf{R}_{m,n}$ can be compressed using any lossy compression algorithm without significantly affecting the quality of the rendered image.

The depth compression algorithm presented in this paper is an image compression scheme which does not exploit the temporal correlation within the depth video. Therefore, both JPEG and H.264/AVC Intra were considered to suppress the spatial redundancies present within the residual block $\mathbf{R}_{m,n}$. Future work will involve the extension of the proposed system to exploit the temporal correlation within depth videos.

3. SIMULATION RESULTS

The first issue considered in the design of the proposed system is to identify the lossless compression algorithm to be used to compress the segmented image $\mathbf{S}$. Table 1 summarizes the average number of bytes per frame needed to compress $\mathbf{S}$ for 100-frames from the Ballet sequence [23]. The JBIG and Zanuttigh-Cortelazzo approach compress the whole segmented image $\mathbf{S}$ while the Deflate and Lempel-Ziv-Welch algorithms were used to compress only the blocks needed by the *Depth Discontinuity Prediction* as described in sub-section 2.2. The ZLib library [24] was adopted to compute the Deflate algorithm while the JBIG-Kit [25] was used to compute JBIG compression. These results demonstrate that the Zanuttigh-Cortelazzo approach managed to outperform the other lossless compression strategies, even if more blocks need to be compressed. Therefore, the Zanuttigh-Cortelazzo approach will be used as part of the proposed system in the simulation results that follow.

The proposed algorithm and Zanuttigh-Cortelazzo [19] approach were implemented using MATLAB and were both configured such that $K = 8$ and $N = 8$. The Zanuttigh-Cortelazzo was used to compare the performance of the proposed system to a boundary edge preserving algorithm found

Table 1. Performance of different lossless coding schemes

Method	Nr Bytes
Deflate algorithm	8528.24
Lempel-Ziv-Welch - 12-bit	7597.54
Zanuttigh-Cortelazzo Approach	6163.39
JBIG	8815.01

in literature. In addition, JPEG and H.264/AVC Intra were used to compare the performance of the proposed system to that of state of the art transform coding schemes. For both JPEG and H.264/AVC Intra, the depth information was considered as a grayscale image. The H.264/AVC Intra was computed using the JM software [26] using all the spatial prediction modes available and encoded using CAVLC, while the rate-distortion optimization feature was disabled.

Results are given for 100-frames from the Ballet and Breakdancers sequences [23], which have a resolution of 1024×768. The performance of the depth compression algorithms considered in this paper were evaluated using a methodology similar to the one adopted in [7]. The original color video from view 1 was used as a reference while the rendered view was computed using the compressed depth images for view 0 and view 2 and their corresponding original color sequences. The PSNR was then computed using the original view 1 as reference. In this work, the rendering algorithm presented in [23] was used.

Fig. 2 summarizes the performance of the methods considered in this paper. It can be immediately noticed that both the proposed method and the Zanuttigh-Cortelazzo manage to outperform the standard JPEG compression, with the proposed system using JPEG to compress $\mathbf{R}_{m,n}$ performing better. It is important to mention that the proposed system manages to achieve lower rates because of the more accurate prediction strategies employed. The performance of the proposed system using H.264/AVC Intra to compress the residual information has registered the best overall performance, outperforming the standard H.264/AVC Intra especially at lower rates. The proposed method using H.264/AVC Intra for lossy compression has also managed to mantain high rendering quality (higher than 28.5dB) even at low rates. However, the proposed scheme cannot achieve rates lower than 0.13 bpp, which is the lower bound present due to the lossless compression of the segmented block $\mathbf{S}$. Nonetheless, this is significantly lower than the lower bound of the Zanuttigh-Cortelazzo as it needs to losslessly compress both segmented block and a subsampled version of the original depth image.

A subjective evaluation can be seen in Fig. 3, which shows the rendered view 1 of the first frame using the compressed depth maps of view 0 and view 2, which were individually encoded at around 0.15 bpp. Significant artifacts, especially at object boundaries, are noticeable when using H.264/AVC Intra to compress the depth view. The proposed

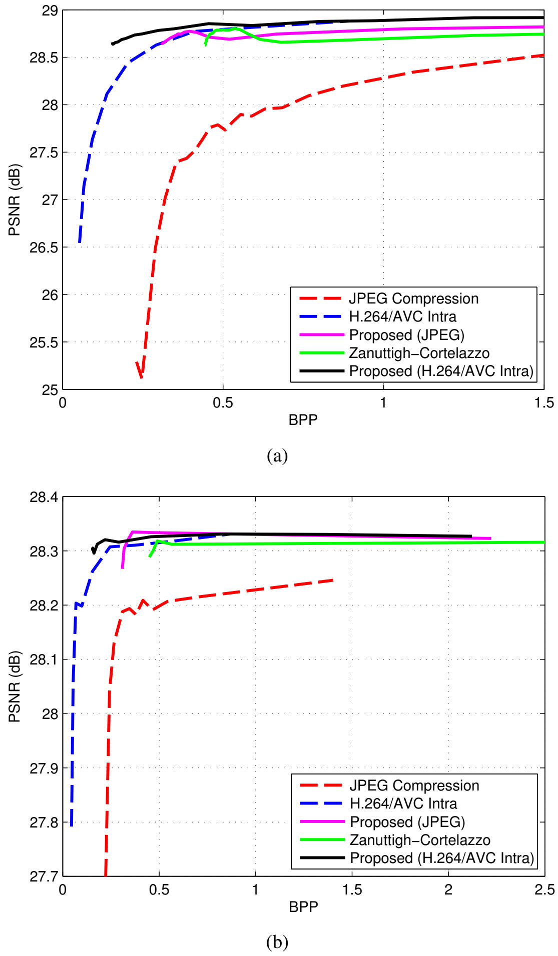

Fig. 2. Rate-distortion curve of the rendered view 1 for (a) Ballet and (b) Breakdancers sequence.

scheme manages to provide higher quality intermediate views at the same rates, thus achieving the best overall performance relative to the methods considered in this work.

4. COMMENTS AND CONCLUSION

This paper presents a novel block-based depth image compression scheme with a bi-modal block encoder. The number of distinct segments within a block was adopted to discriminate between smooth and non-smooth blocks. The smooth blocks were predicted using spatial predictors while non-smooth blocks were approximated using the novel depth-discontinuity predictor. Simulation results have indicated that the lossless scheme adopted by Zanuttigh-Cortelazzo [19] to compress the whole segmented image $\mathbf{S}$ manages to outperform all the other lossless schemes considered in this work.

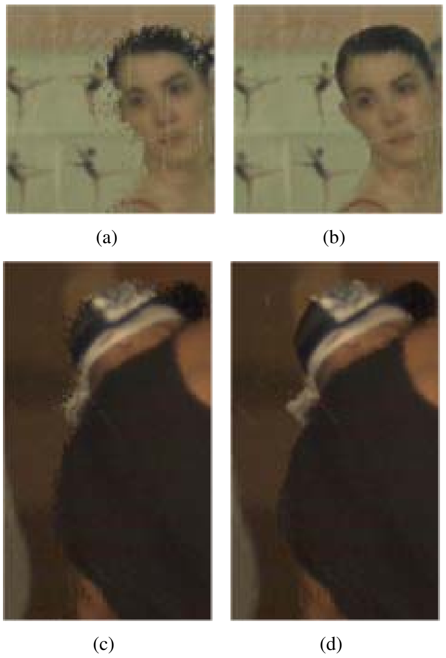

Fig. 3. Subjective evaluation when the Ballet depth image is comressed using (a) H.264/AVC Intra and (b) Proposed method with H.264/AVC Intra to compress the $\mathbf{R}_{m,n}$, and when the Breakdancers depth image is compressed using (c) H.264/AVC Intra and (d) Proposed method with H.264/AVC Intra to compress the $\mathbf{R}_{m,n}$.

Moreover, it was shown that any lossy compression algorithms can be used to compress the residual information.

Simulation results have further indicated that the proposed scheme can achive the same quality of the rendered view at much lower bitrates compared to all other schemes. Moreover, the proposed scheme adopting H.264/AVC Intra to compress the residual information managed to outperform the transforn coding schemes both objectively and subjectively. Future work will consider the extension of the proposed algorithm to exploit temporal correlation in depth video.

5. REFERENCES

[1] ITU-T and ISO/IEC JTC 1, "H.264: Advanced video coding for generic audiovisual services," *Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2010.

[2] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3d video and free viewpoint video - technologies, applications and mpeg standards," in *Multimedia and Expo, 2006 IEEE International Conference on*, july 2006, pp. 2161 –2164.

[3] A. Vetro, T. Wiegand, and G.J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626 –642, april 2011.

[4] A. Smolic, K. Mueller, P. Merkle, P. Kauff, and T. Wiegand, "An overview of available and emerging 3d video formats and depth enhanced stereo as efficient generic solution," in *Picture Coding Symposium, 2009. PCS 2009*, may 2009, pp. 1 –4.

[5] ISO/IEC JTC1/SC29/WG11 N12036, "Call for proposals on 3d video coding technology," 2011.

[6] C. Fehn, P. Kauff, M. op de Beeck, F.M.L. Ernst, W.A. IJsselsteijn, M. Pollefeys, L. van Gool, E. Ofek, and I. Sexton, "An evolutionary and optimised approach on 3d-tv," in *Proceedings IBC'02, 13-17 September 2002, Amsterdam, The Netherlands*, Amsterdam, NL, 2002 2002, IBC, pp. 357 – 365, IBC.

[7] 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, Jan. 2009.

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

[9] Y. Morvan, D. Farin, and P.H.N. de With, "Depth-image compression based on an r-d optimized quadtree decomposition for the transmission of multiview images," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 16 2007-oct. 19 2007, vol. 5, pp. V –105 –V –108.

[10] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 16 2007-oct. 19 2007, vol. 1, pp. I –201 –I –204.

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

[12] Min-Koo Kang and Yo-Sung Ho, "Depth video coding using adaptive geometry based intra prediction for 3-d video systems," *Multimedia, IEEE Transactions on*, vol. 14, no. 1, pp. 121 –128, feb. 2012.

[13] F. Jager, "Contour-based segmentation and coding for depth map compyression," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, nov. 2011, pp. 1 –4.

[14] E. Ekmekcioglu, M. Mrak, S. Worrall, and A. Kondoz, "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, nov. 2009, pp. 733 –736.

[15] Yanjie Li and Lifeng Sun, "A novel upsampling scheme for depth map compression in 3dtv system," in *Picture Coding Symposium (PCS), 2010*, dec. 2010, pp. 186 – 189.

[16] M. Sarkis and K. Diepold, "Depth map compression via compressed sensing," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, nov. 2009, pp. 737 –740.

[17] C. De Raffaele, K. Camilleri, C.J. Debono, and R.A. Farrugia, "Efficient multiview depth representation based on image segmentation," *Picture Coding Symposium (PCS), 2012 IEEE*, May 2012.

[18] Simone Milani, Pietro Zanuttigh, Marco Zamarin, and Soren Forchhammer, "Efficient depth map compression exploiting segmented color data," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, july 2011, pp. 1 –6.

[19] P. Zanuttigh and G.M. Cortelazzo, "Compression of depth information for 3d rendering," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, may 2009, pp. 1 –4.

[20] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[21] T.A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8 –19, june 1984.

[22] P. Deutsch, "Deflate compressed data format specification version 1.3," 1996.

[23] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004.

[24] "Zlib general purpose compression library," Version 1.2.5.

[25] M. Kuhn, "Jbig-kit," Version 2.0.

[26] "H.264/AVC JM Reference Software," Aug. 2011.