EuroCon 2013 • 1-4 July 2013 • Zagreb, Croatia

Exploiting Color-Depth Image Correlation to improve Depth Map Compression

Reuben A. Farrugia [1], Isabel Gambin [2]

Department of Communications and Computer Engineering
University of Malta
Msida, Malta
[1] reuben.farrugia@um.edu.mt
[2] igam0001@um.edu.mt

***Abstract*—The multimedia signal processing community has recently identified the need to design depth map compression algorithms which preserve depth discontinuities in order to improve the rendering quality of virtual views for Free Viewpoint Video (FVV) services. This paper adopts contour detection with surround suppression on the color video to approximate the foreground edges present in the depth image. Displacement estimation and compensation is then used to improve this prediction and reduce the amount of side information required by the decoder. Simulation results indicate that the proposed method manages to accurately predict around 64% of the blocks. Moreover, the proposed scheme achieves a Peak Signal-to-Noise Ratio (PSNR) gain of around 4.9 - 6.6 dB relative to the JPEG standard and manages to outperform other state of the art depth map compression algorithms found in literature.**

***Index Terms*—3D television, depth map compression, displacement estimation and compensation, edge detection, surround suppression**

I. Introduction

Advances in technology have contributed to the development of new applications such as 3D Television (3DTV) and Free Viewpoint Video (FVV), which have recently started to attract interest within the marketplace. The multi-view video plus depth (MVD) [1] format enables the generation of an infinite set of videos using a finite set of color and corresponding depth videos [2]. The MVD format significantly reduces the amount of information needed to deploy these services. Moreover, in order to make these applications more viable, both color and depth videos need to be compressed. However, the quality of the rendered views is significantly dependent on the quality of the depth video.

Traditional image and video compression standards produce blurring artifacts along depth discontinuities, which negatively affect the Quality of Experience (QoE) of the rendered virtual views [3], [4]. The authors in [5] have demonstrated that higher rendering quality can be perceived when employing depth map compression strategies which preserve edge boundaries. Following this observation the same authors have proposed to reshape the dynamic range and use Region of Interest (RoI) coding to improve the performance of traditional image compression standards. Directional transforms were considered in [3], [6] which are highly computational intensive. The authors in [7], [8] have modeled the edge discontinuities within depth maps using Piecewise linear approximations, while geometric Intra predictors were adopted in [9]. However, these approximations do not manage to adequately model the edge discontinuities which contribute to additional artifacts in the rendered views. Contour based segmentation of depth maps was presented in [10] which fails to outperform traditional compression schemes at low rates.

The authors in [11] has presented a depth segmentation based algorithm which was later on improved by Farrugia in [12]. These depth map compression algorithms manage to preserve the edge discontinuities to achieve high quality virtual views. The correlation between color and depth images was exploited in [13], [14]. However, the performance of the former approach significantly drops at high rates while that of the latter suffers from the fact that the discontinuities present in depth and color images are generally not perfectly aligned.

This paper exploits the inherent correlation that exists between the color and corresponding depth images to improve the compression efficiency. This method employs contour detection with surround suppression to approximate the edge map using the color view, which is available at the decoder. The performance is further improved using displacement prediction and compensation, which tries to alleviate the misalignment issue between edges in the color and depth images. This method manages to significantly reduce the amount of information that needs to be transmitted achieving a Peak Signal-to-Noise Ratio (PSNR) gain of up to 6.6 dB relative to JPEG and reduce the lower bound on the data rate by around 0.38 bpp. Moreover, the proposed system manages to outperform the Zanuttigh-Cortelazzo method [11] and has a slight edge over the method presented in [12].

The paper is structured as follows: A high level description of the proposed system is delivered in Section II while the major novelty components of this work is delivered in Section III. The simulation results and discussion of the results are delivered in Section IV while the final comments and concluding remarks are drawn in the last section.

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

II. System Overview

The depth map compression algorithm presented in this paper extends the method presented by the same author in [12]. Fig. 1 shows a schematic diagram of the proposed system where the novelty components proposed in this paper are highlighted in red.

The K-means clustering algorithms [15] is used to segment the depth image $\mathbf{D}$ into K segments. The segmented image $\mathbf{S}$ is then passed through a Canny edge detector [16] to compute the edge map $\mathbf{E}$ which identifies the locations of the depth discontinuities. The depth image $\mathbf{D}$, segmented image $\mathbf{S}$ and the edge map $\mathbf{E}$ are then divided into non-overlapping $N \times N$ pixel blocks where each block (m, n) is denoted by $\mathbf{D}_{m,n}$, $\mathbf{S}_{m,n}$ and $\mathbf{E}_{m,n}$ respectively.

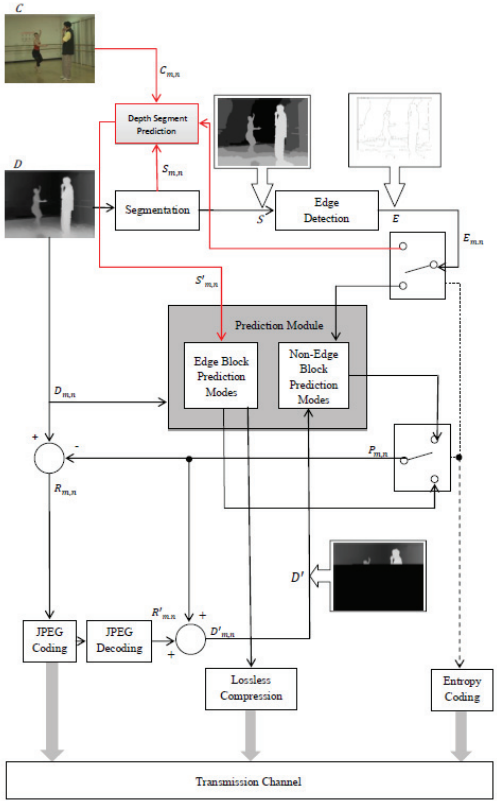


Fig. 1: Schematic diagram of the proposed depth map compression algorithm.

The *Mode Decision* module processes the edge blocks $\mathbf{E}_{m,n}$ sequentially, and depending on its content determines a suitable prediction mode. If the edge block $\mathbf{E}_{m,n}$ contains no edge pixels, spatial prediction is employed to derive the prediction block $\mathbf{P}_{m,n}$. The spatial prediction modes considered in this work are similar to those employed by the standard H.264/AVC video codec and are summarized in Fig. 2. Alternatively, an edge block prediction mode is selected which is

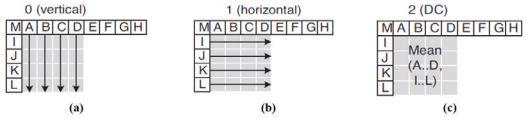


Fig. 2: Schematic diagram of the spatial prediction modes considered in this work (a) vertical predictor, (b) horizontal predictor and (c) DC predictor.

optimized to improve the prediction of blocks containing edge discontinuities.

The edge block predictor employs the neighbouring segmented blocks $\mathbf{S}_{m-1,n}$, $\mathbf{S}_{m-1,n-1}$, $\mathbf{S}_{m-1,n}$ and $\mathbf{S}_{m-1,n+1}$ which are also available at the decoder. These neighbouring segment blocks are grouped to form a set $\mathbf{S}_v$. Similarly, the depth blocks $\bar{\mathbf{D}}_{m-1,n}$, $\bar{\mathbf{D}}_{m-1,n-1}$, $\bar{\mathbf{D}}_{m-1,n}$ and $\bar{\mathbf{D}}_{m-1,n+1}$ are grouped in a set $\bar{\mathbf{D}}_v$. The cluster mean of every segment $k \in K$ is then computed using

$$\bar{\mu}_k = \frac{1}{|\mathbf{\Gamma}_k|} \sum_{v \in \Gamma_k} \bar{\mathbf{D}}_v \tag{1}$$

where $\mathbf{\Gamma}_k$ represents the set of indices where $\mathbf{S}_v = k$, and $|\cdot|$ is 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

$$p_{i,j}^{m,n} = \bar{\mu}_k, \text{ where } k = s_{i,j}^{m,n} \tag{2}$$

where $p_{i,j}^{m,n}$ represent the pixels within block (m, n) at coordinates (i, j) within the prediction block $\mathbf{P}_{m,n}$. Similarly, $s_{i,j}^{m,n}$ represent the segment index within block (m, n) at coordinates (i, j) within the segmented block $\mathbf{S}_{m,n}$.

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

The segmented blocks which are needed by the decoder to compute edge block prediction $\mathbf{S}_{m,n}^{e}$ are a subset of the set of all segmented blocks $\mathbf{S}_{m,n}$. Therefore, only the subset $\mathbf{S}_{m,n}^{e}$ of segmented blocks needs to be coded and transmitted, which usually corresponds to around 10% of the total number of segmented blocks. The *Depth Segment Predictor* module was adopted in this work to exploit the inherent correlation between the color and depth images. This process receives the color block $\mathbf{C}_{m,n}$ and computes an approximation of the segmented block $\mathbf{S}_{m,n}^{e\prime}$. Therefore, only the residual information $\mathbf{S}_{m,n}^{e(R)} = \mathbf{S}_{m,n}^{e} - \mathbf{S}_{m,n}^{e\prime}$ needs to be transmitted, thus reducing the amount of information that needs to be transmitted. More information about this process is provided in section III.

The residual block $\mathbf{R}_{m,n} = \mathbf{D}_{m,n} - \mathbf{P}_{m,n}$ is then encoded using the standard JPEG compression algorithm and transmitted. The JPEG decoder is used to recover the lossy residual

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

block $\mathbf{R}'_{m,n}$ that is used to recover the depth block $\mathbf{D}'_{m,n}$. The recovered depth blocks $\mathbf{D}'_{m,n}$ are used as reference in the prediction process. This feedback look is essential to maintain synchronization between encoder and decoder. The residual blocks $\mathbf{S}^{e(R)}_{m,n}$ are losslessly encoded while other information needed by the decoder is transmitted as side information.

III. Depth Segment Prediction

The major contribution of this paper relies in the exploitation of the redundant information present between texture and depth images to achieve higher compression efficiency. This is because the edge map of the depth image can be accurately predicted using the color image, and therefore less information needs to be transmitted to the decoder.

A straightforward approach is to compute the edge detection on the color image to approximate the locations of the edges in the depth map. However, regular edge detectors mark all local luminance variations as edges and therefore provide a huge number of false positives. Moreover, as shown in Fig. 3, the edges derived using edge detectors on color and depth images are not perfectly aligned. This phenomenon is generally due to the smooth transition between two adjacent colors in the color image, which may result in shifted edges with respect to the original depth discontinuities. The former problem can be alleviated using the surround suppression strategy presented in [17], [18] while the latter can be alleviated by using displacement estimation and compensation. These two methods will be described in the following subsections while the actual segment prediction and the encoding mechanism of the side information are provided in sections III-C and III-D respectively.

Fig. 3: Comparison of color view and depth map (a) Edge in color view (b) Edge in depth map.

A. Contour Detection with Surround Suppression

Contour detection with surround suppression [17], [18] is an image processing algorithm devised to enhance the detection of object contours and region boundaries from natural scenes. The concept behind surround suppression is biologically inspired by the human visual system (HVS). which is able to distinguish amongst isolated edges (object contours, region boundaries) and texture regions.

The contour detection with surround suppression strategy first convolves the image $C(x,y)$ with the x and y derivatives of a Gaussian kernel to derive the scale dependent gradient $M_\sigma(x,y)$ using

$$M_\sigma(x,y) = \sqrt{[\nabla_x F_\sigma(x,y)]^2 + [\nabla_y F_\sigma(x,y)]^2} \tag{3}$$

where

$$\nabla_x F_\sigma(x,y) = \left\{ \mathbf{C} * \frac{\delta g_\sigma}{\delta x} \right\}(x,y)$$
$$\nabla_y F_\sigma(x,y) = \left\{ \mathbf{C} * \frac{\delta g_\sigma}{\delta y} \right\}(x,y)$$
$$g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{x^2+y^2}{2\sigma^2} \right)$$

where $*$ corresponds to the convolution function. To enhance the performance of the standard edge detector, a surround suppression operator is included in the next stage. The surround suppression operator is based on the observation that if edges are in close proximity to one another then they most likely constitute a textured region. Alternatively, isolated edges denote true object contours and region boundaries. Thereby, a local average of the gradient intensity within a ring area surrounding each pixel of the image $C(x,y)$ is computed. This is accomplished using the difference of two Gaussian functions $DoG_\sigma(x,y)$ which defines a ring area around image pixels at coordinates (x,y) using

$$DoG_\sigma(x,y) = \left| \frac{1}{2\pi(4\sigma)^2} \exp\left(-\frac{x^2+y^2}{2(4\sigma)^2} \right) - \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2} \right) \right|^+ \tag{4}$$

where $|\cdot|^+$ symbolizes half-wave rectification. The suppression term $t_\sigma(x,y)$ is then determined which is a weighted sum of gradient values within this ring area surrounding the coordinates (x,y). The distance amongst the pixel coordinate (x,y) and a point in its annular neighborhood is taken into consideration by the weighting function $w_\sigma(x,y)$ which is computed using

$$w_\sigma(x,y) = \frac{H(DoG_\sigma(x,y))}{\|H(DoG_\sigma)\|_1} \tag{5}$$

where

$$H(z) = \begin{cases} 0 & \text{when } x < 0 \\ z & \text{Otherwise} \end{cases}$$

and $\|\cdot\|_1$ is the L_1 norm. The suppression term is then defined as

$$t_\sigma(x,y) = \{\mathbf{M}_\sigma * \mathbf{w}_\sigma\}(x,y) \tag{6}$$

The suppression term $t_\sigma(x,y)$ is then employed to suppress edges stemming from textured regions using the following relation

$$c_\sigma(x,y) = |M_\sigma(x,y) - \alpha t_\sigma(x,y)|^+ \tag{7}$$

where α regulates the strength of the surround suppression operator upon the surroundings of the image pixel (x, y). The final edge map is extracted using non-maxima suppression and hysteresis thresholding.

The resulting edge map tends to produce fragmented edges which is caused by low variance in the pixel intensity distribution. This can be alleviated using an edge linking strategy which fine tunes the initial edge map by linking breaks between edges. The edge linking algorithm presented in [19] was used for this purpose since it is a fast solution that analysis local information in the proximity of edge terminations to perform edge linking.

The performance of different edge detection methods can be analyzed in Fig. 4. It can be immediately noticed that the surround suppression step is crucial since performing the Canny edge detection on the color view produces several false positives which are caused by texture. The surround suppression method is capable to get rid of most texture edges and the resulting edge predictor $\mathbf{E}'$ is comparable to the one produced using Canny edge detection on the depth image, especially at object boundaries in the foreground. The edges detected in the background do not have a significant effect on the performance of the proposed system since the *Mode Selection* process basis its decisions on an edge map $\mathbf{E}$ derived from the segmented depth map $\mathbf{S}$. If this area corresponds to a homogenous depth region the *Non-edge prediction modes* are used and therefore no segmented blocks need to be transmitted to the decoder.

B. Displacement Estimation and Compensation

A limitation of the abovementioned prediction strategy is attributed to the fact that edges in color images are not as sharp as those in depth images. As a consequence, the edge maps derived using the color and depth images are similar in shape, but are not perfectly aligned (see Fig. 3).

As a counter measure, *Displacement Estimation and Compensation* is performed. This process modifies the edge map $\mathbf{E}'_{m,n}$ using different combinations of shifting and extrapolation operations. The shift operator moves the pixels in a row/column in the vertical/horizontal direction, while the vacant pixels are extrapolated using the neighbouring pixels. Fig. 5 demonstrates a subset of the combinations considered. Each set of combinations will provide a different edge map and the one which minimizes the mean absolute error (MAE) with respect to the edge map $\mathbf{E}_{m,n}$ is considered as the predicted edge map $\mathbf{E}'_{m,n}$.

The offset in the x and y directions, which correspond to the horizontal and vertical shifting operations needed to compute the predicted edge map $\mathbf{E}'_{m,n}$, are denoted as *displacement vectors*. The *displacement vectors* are required to align the edges present in the color and depth images at the decoder.

C. Segment Prediction

The optimal edge block $\mathbf{E}'_{m,n}$, derived using the combined contour detection with surround suppression followed by *Displacement Estimation and Compensation*, is then used

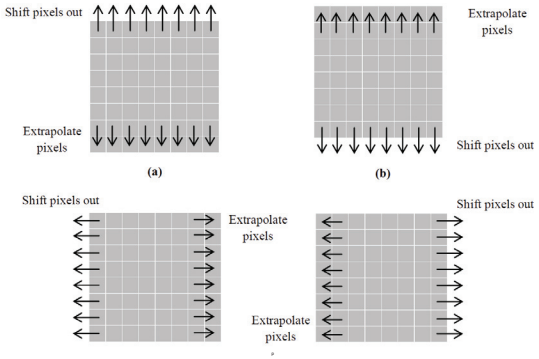


Fig. 5: The four modes of Displacement Estimation (a) top, (b) bottom, (c) left and (d) right.

to predict the segment block $\mathbf{S}^{e\prime}_{m,n}$. This prediction strategy employed in this work can be easily explained using the example illustrated in Fig. 6. From this example it is clear that there is an edge within the segment $\mathbf{S}^{e}_{m,n}$ with segment indices 4 and 2. It can also be seen that the edge map $\mathbf{E}'_{m,n}$ has accurately predicted the edge within $\mathbf{S}^{e}_{m,n}$. Therefore, the predicted segment $\mathbf{S}^{e\prime}_{m,n}$ can be dissected into three disjoint sets; the two non edge regions on either side of the edge and the edge itself. In order to minimize the amount of information to be transmitted, the encoder needs only to transmit the sequence $\mathbf{k} = \{4, 4, 2\}$ which correspond to the index of each disjoint set. This results in a significant reduction in the information being transmitted. Given that the indices are available at the decoder, it can use the edge block prediction described in section II to derive the predicted depth block $\mathbf{P}_{m,n}$.

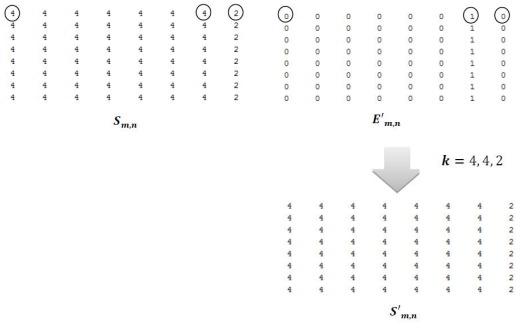


Fig. 6: An example using the Texture Suppression Mode to approximate the segmented depth image block $S^{e}_{m,n}$. The approximated value is denoted by $S^{e\prime}_{m,n}$ while $E'_{m,n}$ is the edge block derived using texture suppression.

In the case where the edge map $\mathbf{E}_{m,n}$ does not manage to accurately represent the edges within the segment block $\mathbf{S}_{m,n}$, the spatial prediction modes described in Section II and motion estimation and compensation are used to compute a set of predictors $\mathbf{S}'^{e}_{p,m,n}$. The mode which minimizes the MAD

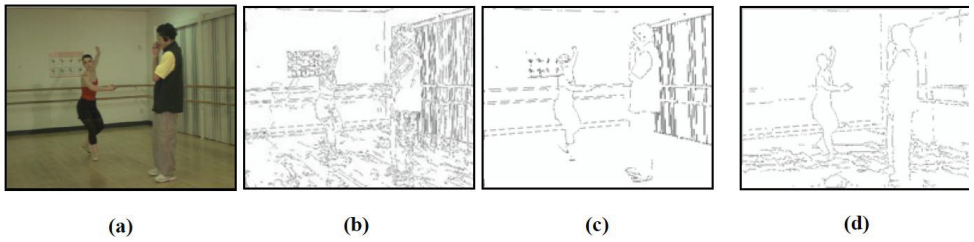


Fig. 4: Comparison of edge maps derived from the color view and depth map (a) Color view (b) Canny edge detection on color view, (c) Contour detection using surround suppression (d) Canny edge detection on depth image.

is then used to derive the residual segmented block $\mathbf{R}^{(S)}_{m,n} = \mathbf{S}_{m,n} - \mathbf{S}'^{,e}_{p,m,n}$ which is entropy coded and transmitted.

D. Side Information Encoding

The segment block prediction strategy described in the previous subsections transmits some side information which is needed by the decoder to recover the compressed depth map. The stream of values $\mathbf{k}$, which is used to indicate the unique index values of the segments contained in $\mathbf{S}_{m,n}$, are encoded using the Deflate algorithm [20]. The Deflate algorithm is known to provide high compression efficiency while it also prevents data expansion even in the worst possible scenarios.

Both the *displacement vectors* and the motion vectors were found to follow a Laplacian distribution and were thus optimally compressed using Huffman codes. On the other hand, the residual information $\mathbf{R}^{(S)}_{m,n}$ is characterized by large streams of zeros. For this purpose, the residual information is encoded using run-level coding. Given that both run and level values follow a zero-mean Laplacian distributed, they were Huffman encoded.

IV. Simulation Results

Testing of the proposed depth map compression algorithm was conducted using $K = 8$ for the K-means clustering and a block size $N = 8$. Results are given for 100-frames from the Ballet and Breakdancers sequences [21], which have a resolution of 1024×768. The performance of the depth compression algorithms considered in this work were evaluated using a methodology similar to the one presented in [4]. 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. View 1 was rendered using the algorithm presented in [21].

The main objective of this work was to compress the side information needed by the decoder to reconstruct high quality depth maps. Table I shows the average number of bytes needed by the proposed algorithm, which includes the *displacement vectors*, the sequence $\mathbf{k}$ and the residual information $\mathbf{R}^{(S)}_{m,n}$. The performance of the proposed system was compared to the JBIG compression, which compresses each individual bitplane, and the Deflate algorithm. The ZLib library [22] was used to compute the Deflate algorithm while te JBIG-Kit [23] was used to compute JBIG compression. These results show that the proposed method requires fewer bytes per frame relative to both JBIG and Deflate algorithms.

TABLE I: Performance of different lossless compression algorithms

Compression Algorithm	**Average Number of bytes per frame**	
	Ballet	**Breakdancer**
JBIG	7732	7851
Deflate	5984	6778
Proposed	**7806**	**6433**

The performance of the *Depth Segment Prediction Modes* presented in this paper was evaluated using both Ballet and Breakdancers sequences. The mode which minimizes the MAE was selected and the frequency of occurrence of each mode was stored for further analysis. The results are summarized in Table II, which show the average probability of occurrence of each mode per frame. It can be noted that the *Texture Suppression Mode* is used on average around $13 - 22\%$ of the times. This mode manages to accurately locate the original edge block $\mathbf{E}_{m,n}$ and therefore the color and depth edges are perfectly aligned. In this case, the encoder needs only to transmit the sequence of indices $\mathbf{k}$.

TABLE II: Performance of the different prediction modes

Mode of Operation	**Probability of Occurrence per frame (%)**	
	Ballet	**Breakdancer**
Texture-suppression (only)	22.22	12.17
Displacement prediction	3.40	4.37
Spatial segment prediction	23.50	25.58
Temporal segment prediction	8.55	4.88
Segment residual plus displacement prediction	37.97	48.63
Segment residual	4.35	4.37

A limitation of the former mode is attributed to the fact that edges in color views are not as sharp as those in depth images. This has a negative impact on the performance of the contour detection. The *Displacement Prediction Mode* was meant to make up for this limitation. At a slight increase in side information caused by the transmission of *displacement*

vectors, it increases the number of perfectly predicted edge blocks by $3-4\%$. However, when the latter scheme does not manage to accurately recover the original segment, the residual information $\mathbf{R}_{m,n}^{(S)}$ together with the *displacement vectors* need to be transmitted. This mode is chosen around $35-45\%$ of the times.

In case where the texture suppression and displacement strategy performs poorly, the segment is predicted using spatial and temporal prediction. Around $24-26\%$ of the blocks can be spatially predicted from the segmented approximation of neighbouring blocks at the decoder, while $5-9\%$ are temporally predicted. These results demonstrate that the segmented blocks are more correlated spatially rather than temporally. The remaining blocks which could not be predicted to an acceptable level of accuracy using these methods, were compressed using the Deflate algorithm. However, it can be seen that the predictors fail less than 5%.

The RD performance in terms of depth distortion can be visualised in Fig. 7 while the rendered quality against total bitrate is shown in Fig. 8. These results clearly demonstrate that the proposed scheme manages to outperform both the standard JPEG compression and the state of the art approach presented by Zanuttigh and Cortelazzo in [11]. These results also demonstrate that the proposed method manages to reduce the amount of side information that needs to be transmitted and contributes to a slight improvement over the baseline method presented in [12].

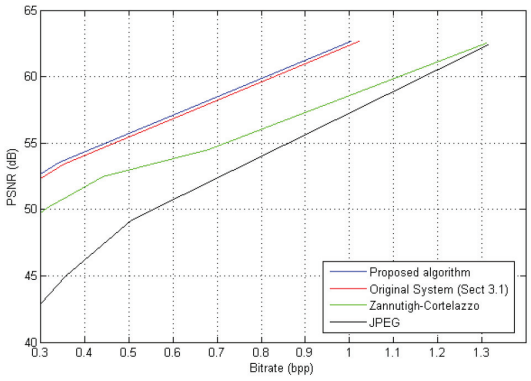
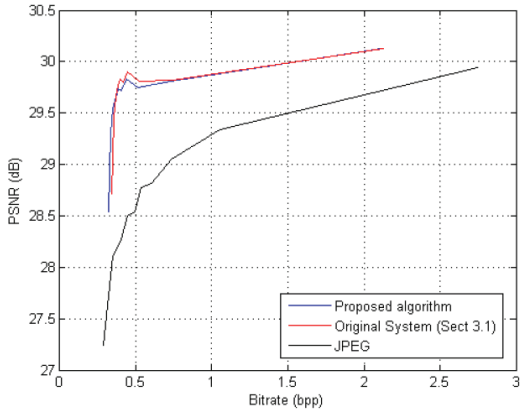
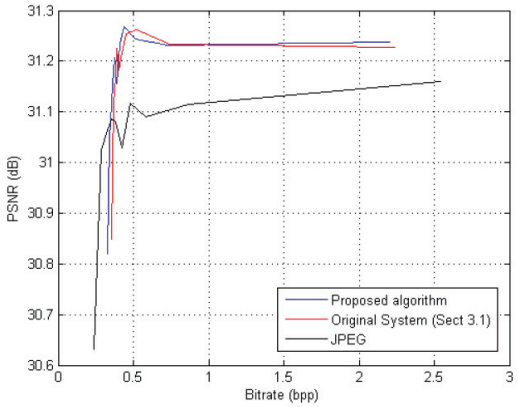


Fig. 7: Rate-Depth Distortion performance for the Ballet test sequence

These results demonstrate that the information present in the texture images can be used to predict the shape of the foreground objects in the depth view. In fact, more than 64% of the edge blocks can be predicted quite accurately using the proposed contour detection with surround suppression and the displacement estimation and compensation strategies. These results also demonstrate that they can be integrated to predict the segment blocks that need to be transmitted providing additional compression without compromising the quality of the rendered views. This can be seen subjectively in Fig. 9.



(a) Ballert



(b) Breakdancers

Fig. 8: Rate-Distortion curves for the (a) Ballet and (b) Breakdancers sequences

V. Conclusion

This paper presents a depth map compression algorithm which exploits the inherent correlation that exists between color and the corresponding depth images. This work demonstrates that contour detection with surround suppression can be used to approximately locate the depth discontinuities present in the depth image using the color image. The edges derived using this method may provide edges which are not aligned. Therefore, a displacement estimation and compensation mechanism was proposed to align the edges derived using contour detection with surround suppression and the depth discontinuities present in the depth image.

Simulation results indicate that more than 64% of the depth discontinuities can be detected with an acceptable level of confidence. Furthermore, the proposed prediction strategy can be used to predict the segmented blocks that need to be transmitted, thus reducing the amount of information that

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

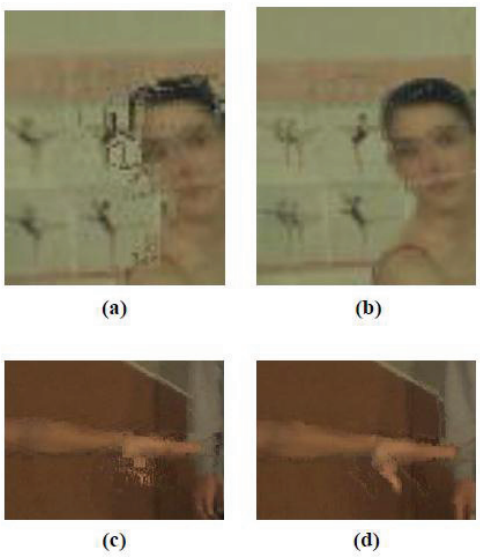


Fig. 9: Subjective evaluation of the depth image compressed with different coding schemes (left) JPEG compressed depth map (right) depth map compressed using proposed method.

needs to be transmitted to the decoder. The proposed system clearly ouperforms both JPEG and the Zanuttigh-Cortelazzo method presented in [11]. Moreover, the proposed system has a small edge over the method previously published in [12].

REFERENCES

[1] A. Smolic, K. Mueller, Merkle P., N. Atzpadin, C. Fen, M. Mueller, M. Schreer, R. Tanger, P. Kauff, T. Wiegand, T. Balogh, Z. Megyesi, and A. Barsi, "Multi-view video plus depth (mvd) format for advanced 3d video systems," Tech. Rep. JVT-W100, ISO/IEC JTC1/SC29/WG11 and ITU-T SG 16 Q.6, San Jose, 2007.

[2] Aljoscha Smolic, Karsten Mueller, Philipp Merkle, Peter Kauff, and Thomas Wiegand, "An overview of available and emerging 3d video formats and depth enhanced stereo as efficient generic solution," in *Proceedings of the 27th conference on Picture Coding Symposium*, Piscataway, NJ, USA, 2009, PCS'09, pp. 389–392, IEEE Press.

[3] M. Maitre and M.N. Do, "Joint encoding of the depth image based representation using shape-adaptive wavelets," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, oct. 2008, pp. 1768 –1771.

[4] 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.

[5] 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.

[6] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, Jaejoon Lee, and Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS), 2010*, dec. 2010, pp. 566 –569.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] R. Farrugia, "Efficient depth image compression using accurate depth discontinuity detection and prediction," in *International Conference on Signal Image Technology and Internet Systems*, nov 2012.

[13] C. De Raffaele, K.P. Camilleri, C.J. Debono, and R.A. Farrugia, "Efficient multiview depth representation based on image segmentation," in *Picture Coding Symposium (PCS), 2012*, may 2012, pp. 65 –68.

[14] 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.

[15] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 1st ed. 2006. corr. 2nd printing edition, Oct. 2006.

[16] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[17] C. Grigorescu, N. Petkov, and M. A. Westenberg, "Contour and boundary detection improved by surround suppression of texture edges," *Journal of Image and Vision Computing*, vol. 22, no. 8, pp. 583–679, 2004.

[18] G. Papari, P. Campisi, N. Petkov, and A. Neri, "A multiscale approach to contour detection by texture suppression," in *Image Processing: Algorithms and Systems, Neural Network, and Machine Learning; Proc. SPIE-IS&T Electronic Imaging 2006, San Jose, CA, USA, January 16-18, 2006*, E. R. Dougherty, J. T. Astola, K. O. Egiazarian, N. M. Nasrabadi, and S. A. Rizvi, Eds. 2006, vol. 6064, pp. 60640D–1 – 60640D–12, SPIE, Bellingham, Washington; IS&T, Springfield, Virginia.

[19] Ovidiu Ghita and Paul F. Whelan, "Computational approach for edge linking," *Journal of Electronic Imaging*, vol. 11, no. 4, pp. 479–485, 2002.

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

[21] 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.

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

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

SP-04: Audio Signal Processing and Analysis 2

978-1-4673-2232-4/13/$31.00 ©2013 IEEE