Depth Coding using Depth Discontinuity Prediction and in-loop Boundary Reconstruction Filtering

Reuben A. Farrugia [#1], Maverick Hili [*2]
[#] Department of CCE, University of Malta
[1] reuben.farrugia@um.edu.mt
[2] mhil0002@gmail.com

***Abstract*—This paper presents a depth coding strategy that employs K-means clustering to segment the sequence of depth images into K clusters. The resulting clusters are losslessly compressed and transmitted as supplemental enhancement information to aid the decoder in predicting macroblocks containing depth discontinuities. This method further employs an in-loop boundary reconstruction filter to reduce distortions at the edges. The proposed algorithm was integrated within both H.264/AVC and H.264/MVC video coding standards. Simulation results demonstrate that the proposed scheme outperforms the state of the art depth coding schemes, where rendered Peak Signal to Noise Ratio (PSNR) gains between 0.1 dB and 0.5 dB were observed.**

***Index Terms*—Boundary Reconstruction Filter, Depth coding, Depth Discontinuity Prediction, H.264/AVC, H.264/MVC**

I. Introduction

Recent advancements in technology have permitted 3DTV to be combined with free viewpoint video which allows the user to choose a viewpoint angle whilst still experiencing the depth impression. The multi-view video plus depth (MVD) [1] format enables the generation of an infinite set of viewpoints using a finite set of color and depth videos. Intermediate views between the real views are then rendered at the receiver to provide the whole range of viewpoints to the viewer. However, traditional image and video compression standards produce distortions along the depth boundaries, which significantly reduce the quality of the rendered views [2].

The authors in [3], [4], [5] have modelled edge discontinuities to reduce the above mentioned distortions. On the other hand, mesh-based depth map compression was considered in [6]. Nonetheless, these methods are not sufficiently robust and still provide distortions at the edges. Directional transforms were adopted in [7], [8] which are highly computational intensive. Depth map prediction schemes were proposed in [9], [10] which were found to provide high quality depth maps which significantly improved the rendered video quality. However, these methods are not compatible with existing video coding standards. An in-loop boundary reconstruction filter was proposed in [11] to enhance the performance of the standard H.264/MVC to compress depth-maps. More recently, the authors in [12], [13] exploit the similarity between depth and texture video to improve the rate distortion performance.

This work presents a novel depth map compression algorithm which employs a depth discontinuity prediction strategy and an in-loop filter which were integrated within the H.264/AVC and H.264/MVC video coding standards. Simulation results demonstrate a significant gain in performance relative to H.264/AVC, H.264/MVC and the method presented in [11]. The encoding complexity was increased by around 23% relative to traditional video coding schemes and 10% relative to the method adopted in [11].

This paper is organized as follows. In Section II, an overview of the proposed depth coding scheme is presented. Section III explains how the depth image is segmented and the resulting sequence of segmented images is compressed and transmitted. The following two sections describe the extended intra depth map prediction and the in-loop boundary reconstruction filter. Experiments and comparisons with state-of-the-art methods are performed in section VI while the final comments and concluding remarks are drawn in Section VII.

II. System Overview

Fig. 1 shows the proposed depth map coding schemes where the extended modules are marked in red. The *Segmentation Module* is used to segment the depth image $\mathbf{D}$ into K segments. The resulting segmented image $\mathbf{S}$ is then losslessly compressed using the *Segmented Image Compression* algorithm, which is a lossless compression scheme. The resulting compressed bitstream $\mathbf{s}$ is then transmitted as side information.

The image $\mathbf{S}$ is used to aid the prediction of blocks containing depth discontinuities and is available at both encoder and decoder. The proposed depth-discontinuity predictor is then included as an additional intra prediction mode which together form the *Intra Depth Prediction* module. The *Boundary Reconstruction Filter* is then included within the loop to remove additional distortions at edge discontinuities present within every macroblock. Rate distortion optimization is used to identify the optimal prediction modes to be used. This information is signalled using the standard H.264 syntax which was modified to enable the signalling of the additional prediction mode.

The decoder reverses this process and recovers the segmented image $\mathbf{S}$ by decoding the supplemental enhancement

978-1-4799-6139-9/14/$31.00

information $\mathbf{s}$. The segmented image $\mathbf{S}$ is then used to derive the predictor $\mathbf{P}$ which is then used to reconstruct the depth sequence. The *Boundary Reconstruction Filter* is then used to suppress residual noise present at the boundaries.

III. Depth Image Segmentation and Compression

The depth-map compression algorithm presented in this work can employ any segmentation strategy. The K-Means clustering algorithm was adopted to minimize the computational complexity. The goal of the K-means clustering algorithm is to find the mean vector $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_K\}$ which clusters the depth image $\mathbf{D}$ into K segments. Each depth pixel $d_{x,y}$ at pixel coordinates (x, y) is therefore clustered within the segment $s_{x,y} = k$, such that

$$s_{x,y} = \arg\min_{k} \|d_{x,y} - \mu_k\| \tag{1}$$

where $k \in \{1, 2, \ldots, K\}$. The mean vector $\boldsymbol{\mu}$ is updated with the mean intensity value of each cluster, such that

$$\mu_k = \frac{1}{|\mathbf{S}_k|} \sum_{i,j \in \mathbf{S}_k} d_{i,j} \tag{2}$$

where $\mathbf{S}_k$ represents the set of all depth pixels assigned to cluster k and $|\bullet|$ is the cardinality of the set. This algorithm iterates the assignment and update step until the mean vector $\boldsymbol{\mu}$ no longer changes.

The segmented image $\mathbf{S}$ is then compressed using a lossless binary compression scheme similar to the one adopted in [9]. However, this work transforms the segmented image $\mathbf{S}$ using gray coding to maximize the uniform regions within every bitplane. The resulting gray coded segmented image $\mathbf{S}_g$ is then dissected in $N = \log_2(K)$ bitplanes which are compressed using JBIG. This process was found to improve the compression efficiency of the *Segment Image Compression* scheme. The compressed bitstream $\mathbf{s}$ is then transmitted as supplemental enhancement information.

IV. Intra Depth Prediction

The segmented image $\mathbf{S}$ and the depth image $\mathbf{D}$ are both 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 macroblock index. The *Intra Depth Prediction* module extends the intra prediction mode adopted by the standard by including an additional prediction mode, which we refer to as the *Depth Discontinuity Predictor*. Rate distortion optimization is used to derive the optimal mode.

The *Depth Discontinuity Predictor*, which is similar to the one adopted in [10], groups the neighbouring segment blocks $\mathbf{S}_{m-1,n}, \mathbf{S}_{m-1,n-1}, \mathbf{S}_{m,n-1}, \mathbf{S}_{m+1,n-1}$ in a set $\mathbf{S}_\zeta$ while the corresponding depth blocks $\tilde{\mathbf{D}}_{m-1,n}, \tilde{\mathbf{D}}_{m-1,n-1}, \tilde{\mathbf{D}}_{m,n-1}, \tilde{\mathbf{D}}_{m+1,n-1}$ are grouped within a set $\tilde{\mathbf{D}}_\zeta$. The local cluster mean of every segment $k \in K$ is computed using

$$\hat{\mu}_k = \frac{1}{|\Gamma_k|} \sum_{i \in \Gamma_k} \tilde{\mathbf{D}}_\zeta(i) \tag{3}$$

where Γ_k represents the set of indices where $\mathbf{S}_\zeta = k$. Consider $s^{m,n}_{i,j}$ to represent the pixel at coordinates (i, j) within segment block at coordinates (m, n). The corresponding depth pixel $p^{m,n}_{i,j}$ is then predicted using

$$p^{m,n}_{i,j} = \begin{cases} \hat{\mu}_k & \text{if } k \in \mathbf{S}_\zeta \\ \mu_k & \text{Otherwise} \end{cases} \tag{4}$$

The *Depth Discontinuity Prediction* process can be extended for H.264/MVC by simply including the segmented depth map and decoded neighbouring blocks of the neighbouring views within the lists $\mathbf{S}_\zeta$ and $\tilde{\mathbf{D}}_\zeta$ respectively.

Fig. 2 shows an illustrative example of how the local means are predicted. The segmentation blocks (Fig. 2 (a)) and the decoded depth blocks (Fig. 2 (b)) are available at both encoder and decoder and will be used to derive the prediction block $\mathbf{P}_{m,n}$ (Fig. 2 (d)). The local mean of segment $k = 1$ is computed by computing the mean of the coefficients $\tilde{d}^{m,n}_{i,j}$ where $s^{m,n}_{i,j} = 1$. The local and global cluster means are shown in Fig. 2 (c). The local mean of segment $k = 2$ is computed using a similar approach. However, there is no neighbouring decoded pixel which was clustered in segment 3. Therefore, this value is predicted using the global cluster mean.

V. Boundary Reconstruction Filtering

The boundary reconstruction filter adopted in this work is similar to the one presented in [11]. It is a non-linear filter which is computed using

$$J_T(k) = J_F(\tilde{d}^{\{2\}}_{i,j}) + J_S(\tilde{d}^{\{2\}}_{i,j}) + J_C(\tilde{d}^{\{2\}}_{i,j}) \tag{5}$$

where $\tilde{d}^{\{2\}}_{i,j}$ represents the depth intensity value after the deblocking filter and $J_F(\tilde{d}^{\{2\}}_{i,j})$, $J_S(\tilde{d}^{\{2\}}_{i,j})$ and $J_C(\tilde{d}^{\{2\}}_{i,j})$ represent the sub-cost due to occurrence frequency, similarity and closeness respectively. The overall cost $J_T(k)$ is computed using the neighbouring pixels and derives the best intensity $\tilde{d}_{i,j}$ which maximizes this cost. The individual sub-costs are computed using

$$\begin{aligned} J_F(\tilde{d}^{\{2\}}_{i,j}) &= \text{normalize}(N_{oc}) \\ J_S(\tilde{d}^{\{2\}}_{i,j}) &= \text{normalize}(S) \\ J_C(\tilde{d}^{\{2\}}_{i,j}) &= \text{normalize}(C) \end{aligned} \tag{6}$$

where

$$\text{normalize}(X) = \frac{X(\max) - X(\tilde{d}^{\{2\}}_{i,j})}{X(\max) - X(\min)} \tag{7}$$

where $X(\min)$ and $X(\max)$ represent the minimum and maximum value of X. N_{oc} represents number of occurrence of the depth value $\tilde{d}^{\{2\}}_{i,j}$ within an $M \times M$ window $W_{M\times M}$ and is defined using

$$N_{oc}(\beta) = \sum_{i=0}^{M\times M-1} \delta[\beta, W_{M\times M}(i)] \tag{8}$$

where

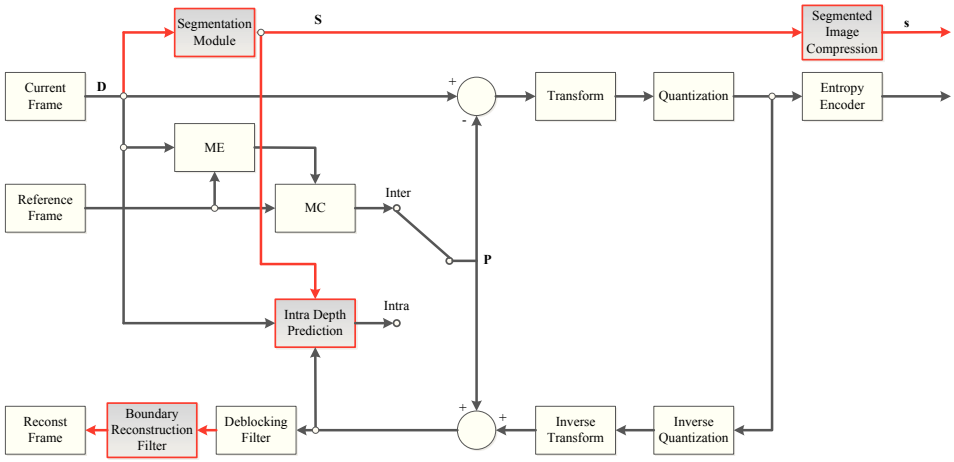
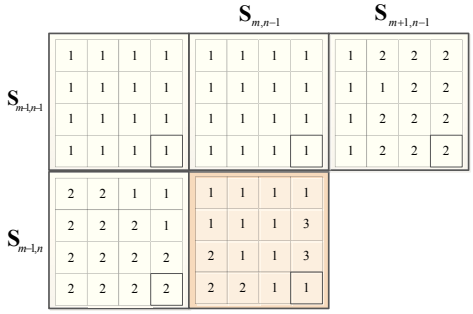
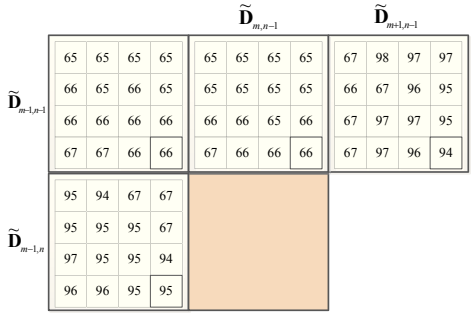


Fig. 1: Proposed depth map Encoding scheme.



(a) Neighbouring Segment blocks



(b) Neighbouring Decoded depth blocks

k	1	2	3	4	5	6	7	8
$\hat{\mu}_k$	66	96	-	-	-	-	-	-
μ_k	50	80	120	150	160	180	190	210

(c) Global and Local means1

$P_{m,n}$

66	66	66	66
66	66	66	120
96	66	66	120
96	66	66	66

(d) Predicted block

Fig. 2: Example of the *Depth Discontinuity Predictor*.

$$\delta[a,b] = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

and $W_{M\times M}(i)$ represents the i^{th} pixel in window $W_{M\times M}$. The similarity score $S(\beta) = |\beta - d_{x,y}|$ where $d_{x,y}$ correspond to the current depth pixel value. The closeness score $C(\beta)$ between the current pixel coordinates (x, y) and the neighbouring pixel coordinates (x_k, y_k) is computed using

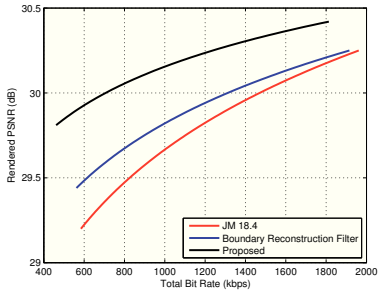
$$C(\beta) = \frac{1}{N_{oc}(\beta)} \sum_{i=0}^{N\times N-1} \sqrt{(x-x_k)^2 + (y-y_k)^2} \quad (10)$$

The remaining noise is removed using a Gaussian bilateral filter adopted in [14].

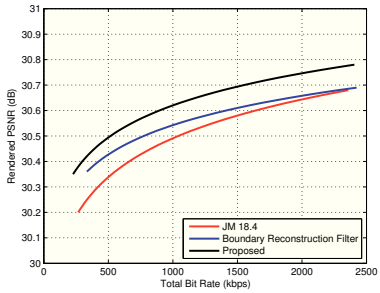
VI. Simulation Results

The proposed method was tested using the Ballet and Breakdancer sequences which have a resolution of 1024×768. The perfrmance of the depth compression algorithms considered in this work were evaluated using a method similar to the one adopted in [2]. The original color sequence from a particular view is used as reference while the same view is rendered using the decoded depth maps using the rendering procedure presented in [15]. The proposed method was integrated within the JM 18.4 for single view depth compression and JMVC 3.01 for multiview depth coding. The High profile is used with a GOP size of 15. Through these simulations a window size $M = 3$ and $K = 8$ clusters are used since they were found to provide the best compromise between performance and complexity. Gray-coding was used prior to lossless compression of the segmented image $\mathbf{S}$ which increases the compression efficiency by around 14.94%.

Fig. 3 shows the performance of the proposed method in relation to the standard H.264/AVC and the method employed in [11], where the total bit-rate includes the bandwidth needed to transmit the side information. The proposed method achieved an average PSNR gain of 0.47dB for the Ballet sequence and 0.1 dB for the Breakdancers sequence. This is achieved at an

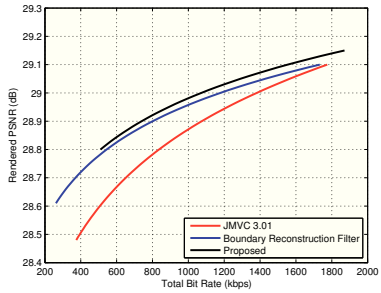


(a) Ballet sequence

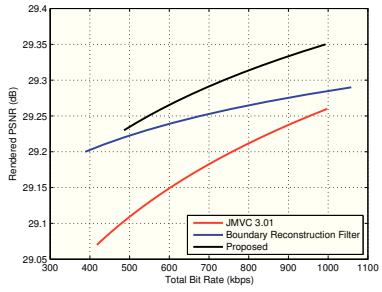


(b) Breaakdancers sequence

Fig. 3: Rate-distortion curve of rendered view (single view).



(a) Ballet sequence



(b) Breaakdancers sequence

Fig. 4: Rate-distortion curve of rendered view (multi-view).

increase in complexity of around 22%, which is just 10% more complicated that the method in [11].

To measure the effectiveness of the proposed method when encoding using multi-view video coding, all views are encoded and the average quality is plotted against the total bit rate. Fig. 4 shows that the proposed method outperforms both standard and the method presented in [11] achieving average PSNR gains between 0.1 dB and 0.2 dB relative to the standard.

VII. Comments and Conclusion

This work presents a depth coding technique which preserves the depth discontinuities of the compressed depth maps. The proposed method derives a segmented image and transmits it as supplemental information which is then used to improve the prediction of depth regions containing depth discontinuities. An in-loop boundary reconstruction filter is then used to improve the quality depth video. The proposed method was integrated within both H.264/AVC and H.264/MVC and managed to outperform both standard codecs and a recent state of the art method. This method has increase the computational complexity by around 22% relative to the standards.

References

[1] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 2448–2451.

[2] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, May 2008, pp. 245–248.

[3] 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 *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5, Sept 2007, pp. V – 105–V – 108.

[4] 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.

[5] M.-K. Kang and Y.-S. 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.

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

[7] M. Maitre and M. 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.

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

[9] P. Zanuttigh and G. 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.

[10] R. Farrugia, "Efficient depth image compression using accurate depth discontinuity detection and prediction," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, Nov 2012, pp. 29–35.

[11] K.-J. Oh, A. Vetro, and Y.-S. Ho, "Depth coding using a boundary reconstruction filter for 3-d video systems," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 3, pp. 350–359, March 2011.

[12] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-based inter prediction in coded texture assisted depth coding," *Signal Processing Letters, IEEE*, vol. 21, no. 1, pp. 74–78, Jan 2014.

[13] L. Shen, P. An, Z. Liu, and Z. Zhang, "Low complexity depth coding assisted by coding information from color video," *Broadcasting, IEEE Transactions on*, vol. 60, no. 1, pp. 128–133, March 2014.

[14] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, Jan 1998, pp. 839–846.

[15] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 600–608. [Online]. Available: http://doi.acm.org/10.1145/1186562.1015766