

# INHERENTLY EDGE-PRESERVING DEPTH-MAP CODING WITHOUT EXPLICIT EDGE DETECTION AND APPROXIMATION

*anonymous*

Shampa Shahriyar, Mortuza Ali, Manzur Murshed, and Manoranjan Paul

## ABSTRACT

In emerging 3D video coding, depth has significant importance in view synthesis, scene analysis, and 3D object reconstruction. Depth images can be characterized by sharp edges and smooth large regions. Most of the existing depth coding techniques use intra-coding mode and try to preserve edges explicitly with approximated edge modelling. However, edges can be implicitly preserved as long as transformation is avoided. In this paper, we have demonstrated that inherent edge preserving encoding of inter-coded block residuals, uniformly quantized at pixel domain using motion data from associated texture components, is more efficient than explicitly edge preserving intra-coding techniques. Experimental results show that the proposed technique has achieved superior image quality of synthesized views against the new 3D-HEVC standard for depth-map bit-rate 0.025 bpp or higher. Lossless applications of the proposed technique has achieved on average 66% and 23% bit-rate savings against 3D-HEVC with negligible quantization and perceptually unnoticeable view synthesis, respectively.

**Index Terms**— depth-map coding, residual coding, multiview plus depth, pixel-based scalar quantization.

## 1. INTRODUCTION

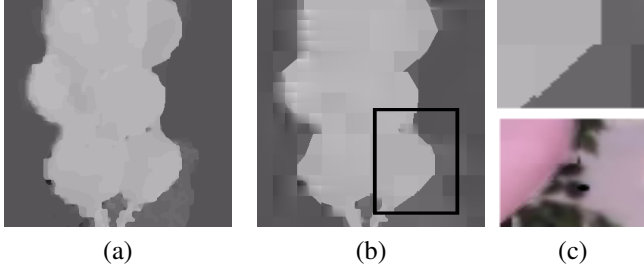
Multiview video technologies, encompassing 3DTV and *free viewpoint video* (FVV), have gained popularity for the added interactivity and user choice in selecting desired viewing angle. Recently, multiview video plus depth [1] architecture has been adopted by the latest multiview video coding standard 3D-HEVC [2, 3] where scene geometry is captured by depth-maps. A depth-map is usually an 8-bit grey-scale image containing distance information of 3D points in the scene from the camera view-plane. Depth-maps can be effectively used to synthesize intermediate views using *depth image based rendering* (DIBR) [4] to exploit inter-view correlations using *view synthesis prediction* as well as to support FVV. Depth coding has much wider applications besides view synthesis e.g., object tracking, gesture recognition, augmented reality, scene analysis, 3D object reconstruction, interactive gaming and many more.

Unlike multispectral texture images, depth images are

more homogeneous or smooth except for the boundary regions that preserve the shape of the objects. Therefore, depth value discontinuity occurs mostly on the edges. Consequently, edges in depth images are more sensitive to coding errors that affect perceptual video quality due to misinterpretation of foreground and background pixels. Therefore, recent depth coding algorithms are motivated to preserve edge. Morvan *et al.* [5] proposed an edge aware representation of depth-maps and platelet based depth coding scheme by assuming piecewise planar depth-maps, straight-line edges, and regions having constant and gradient depth values. Kang *et al.* [6] used line-partitioned intra prediction to model block edges with fewer coefficients. Merkle *et al.* [7] proposed an intra-coding based depth modelling using wedglet segmentation, which also approximates edges using straight-lines. All these techniques are based on piecewise planar assumption, which does not hold for depth-maps in general, and their handling of edges by approximation distort the shape of edges.

Shen *et al.* [8] introduced additional mode to predict blocks with edge graph information, which is later used to develop edge adaptive transformation [9]. This technique, however, requires explicit calculation of eigenvectors which is computationally expensive. As transform-domain residual data of blocks with edges consume more bits in high frequency coefficients, Lee and Ho [10] avoided transformation altogether by using segmentation of intra-coded block residuals and encoding the segment map along with the quantized mean values. While adverse effect of transformation on edges has been established, all of these edge-preserving depth coding techniques used intra-coding that cannot take the advantage of high temporal correlations in successive depth images and spatial correlations of inter-coded residuals.

In this paper, we propose an efficient depth coding technique where inter-coded block residuals are quantized at pixel domain and the quantized residual frame is encoded with *lossless JPEG* (JPEG-LS) [11]. By avoiding traditional transformation and any explicit edge detection and followed by approximation, the proposed technique has achieved superior image quality of synthesized views against 3D-HEVC for depth-map bit-rate 0.025 *bits per pixel* (bpp) or higher. As capturing high quality depth-maps is difficult, lossy compression of depth-maps brings additional geometrical distortion to view synthesis [12] and 3D object reconstruction. Lossless



**Fig. 1:** *Balloons* sequence: (a) original depth-map; (b) decoded depth-map by 3D-HEVC with  $QP$  40; and (c) flawed balloon shape in synthesized view generated using the “Renderer version 7.0” incorporated in 3D-HEVC Test Model (3D-HTM)

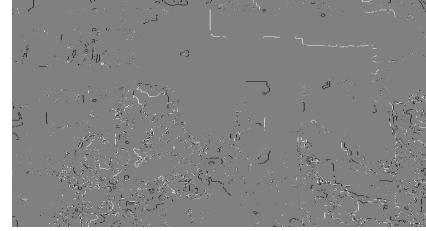
application of the proposed technique without any quantization has achieved on average 66% and 23% bit-rate savings against 3D-HEVC with negligible quantization ( $QP = 1$ ) and perceptually unnoticeable ( $\Delta PSNR \approx 0.5$  dB) view synthesis, respectively. The rest of the paper is organized as follows: Section 2 identifies the problems of explicitly approximated edge preservation techniques for depth-map coding. Section 3 presents the proposed inherently edge preserving depth coding technique followed by the experimental results, analysis, and comparison in Section 4. Section 5 concludes the paper.

## 2. PROBLEMS IN 3D-HEVC DEPTH CODING

The latest 3D-HEVC standard encodes depth-maps using both intra-coding and motion-compensated inter-coding modes similar to texture coding. To handle the smoothness and sharp-edge properties of depth-maps, four additional intra-coding modes are used with explicit edge detection and approximation at block level. In the quadtree based block decomposition, a depth block containing edges is partitioned into two non-rectangular regions that are represented by constant depth values and signalled as difference from predicted values that are also available at the decoder. Partitioning is carried out (i) on the depth block; (ii) from the partitions of the already encoded neighbouring depth blocks; or (iii) on the co-located texture block, which has already been encoded. In all three cases, partitioning is modelled with a wedgelet, i.e., a straight-line dissection of the block and explicit wedglet signalling is needed only for the first case. For the third partitioning case, contour partitioning model is also used as the decoder can estimate the same contour without any explicit signalling. These intra-coding modes suffer from the following problems:

**Problem 1:** Approximation of edges using wedgelets can distort the shape of objects, especially when applied in larger blocks, resulting in rendering flaws in view synthesis as shown in Figure 1.

**Problem 2:** It is not always true that an edge in texture



**Fig. 2:** *Newspaper* sequence: eliminated (black) and false (white) edges due to high quantization of transform coefficients by 3D-HEVC. Edges were detected in the depth-map using Canny edge detector with minimum and maximum threshold automatically set as 0.0125 and 0.03125, respectively.

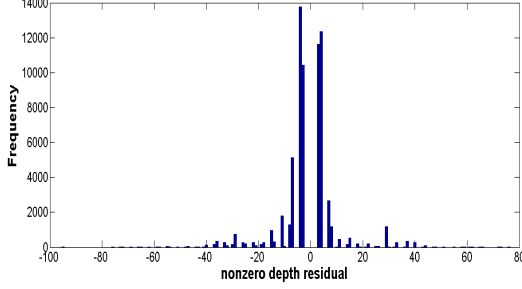
image is reflected on the corresponding depth-map. If an object in the texture image has textured patterns, abrupt discontinuity of colour values may appear inside as well as at the boundary. The corresponding depth-map, however, exhibits discontinuity of depth values only at the boundary as the inner part of the object is represented by a constant depth value.

When motion-compensated inter-coding modes are used, 3D-HEVC directly uses the block partitioning and motion data from the associated video component and encodes the residual with transformation and quantization similar to texture coding.

**Problem 3:** Transformation with high quantization can severely distort edge information in blocks. Traditional transformation such as *discrete cosine transform* (DCT) is used to decorrelate pixel-domain information into frequency domain so that quantization level can be increased for higher frequency coefficients without significantly impacting the image quality. However, discontinuity of depth values at edges is represented mainly by the high frequency coefficients. Consequently, high quantization of high frequency coefficients can potentially eliminate some edges. Moreover, high quantization in general can also introduce false edges at block boundaries as consecutive two blocks of same depth can experience different levels of quantization errors. Figure 2 demonstrates the adverse impact of high quantization in transform coefficients on edges.

## 3. PROPOSED DEPTH CODING

Temporal prediction can be used for exploiting the correlation between temporally adjacent frames of the same depth sequence. Difference of two successive frames in depth-map are mostly zero, except in the motion area where variance of depth can be observed. Thus encoding of residual data generated by inter prediction requires fewer bits compared to original depth-map coding. When the residual is encoded using lossless compression without any quantization, edges are preserved exactly in the decoded depth image.



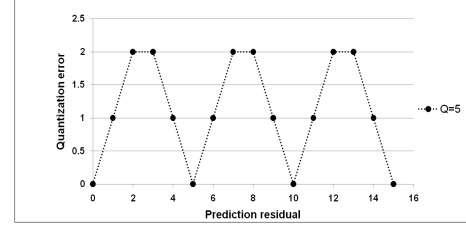
**Fig. 3:** Histogram of non-zero temporal depth residual values for *Newspaper* sequence

If scalar quantization is introduced at pixel-level, unlike the conventional block-based transformation and vector quantization, quantization errors are evenly distributed among the pixels. By restricting quantization error to near-lossless region, edge preservation can be guaranteed within reasonably small uncertainty. In fact, using only small scalar quantization is sufficient for depth coding. We have analysed the histogram of inter-coded depth residuals and observed that the frequency of non-zero values diminishes quickly beyond the range  $[-10,10]$  as shown in Figure 3 where the highly dominant (above 90%) zero residual value is excluded to show the frequencies of other values more prominently. So, using high quantization step will simply increase quantization errors without any compression benefit. Moreover, if the residual is encoded at frame-level without considering the underlying blocks, impact of blocking artefacts can also be minimised.

It is also observed that although pixel values in texture and depth sequences are different, boundaries of objects in the scene coincide, their movements are very similar, and the structure of objects are also preserved in depth-maps. Consequently, motion vectors of corresponding texture and depth sequences are similar enough to use texture motion information in depth coding, especially for larger block size. Since motion estimation in depth-map sequence is skipped and motion information is taken from texture video the entire encoding cost is significantly reduced.

### 3.1. Encoding and Decoding

**Residual estimation:** We assume that corresponding texture frame has already been encoded using H.264 inter-coded mode with block size  $8 \times 8$  pixels. Depth residual is calculated using the motion vectors of corresponding texture blocks. Sub-blocking at  $4 \times 4$  pixels is considered if the *mean absolute deviation* (MAD) of an  $8 \times 8$  pixel block is above a threshold (0.3). Local motion search is performed for each sub-block on the depth reference frame and only the non-zero sub motion vectors are identified using quad-tree based decomposition.



**Fig. 4:** Magnitude of quantization error vs prediction residual of the proposed quantization scheme.

**Residual quantization:** Each pixel-level residual value  $\epsilon = x - \hat{x}$  is quantized using the same scalar quantization step  $Q$  as follows:

$$\epsilon_Q = \text{round}\left(\frac{\epsilon}{Q}\right) \quad (1)$$

where  $x$  is the actual depth value of pixel and  $\hat{x}$  is its temporal prediction. The corresponding de-quantization is obtained as follows:

$$\epsilon' = \epsilon_Q Q \quad (2)$$

with quantization error  $\eta = \epsilon - \epsilon'$ , magnitude of which is bounded by  $|\eta| \leq \lfloor Q/2 \rfloor$ . As the maximum quantization error for an even quantization step is same as that for the immediate next odd quantization step,  $Q$  is drawn from odd integers to achieve better bit-rate for the same image distortion compared to using the preceding even integer. From our empirical observation in Figure 3 that the depth temporal residuals rarely fall outside the range  $[-10,10]$ , we recommend restricting  $Q \in \{1, 3, 5, 7\}$  where  $Q = 1$  represents lossless depth coding.

For reasonably good prediction modelling such as motion search, residuals in general fits best with the Laplacian probability distribution where zero residual has the highest probability and the probability of residuals decreases exponentially with respect to their absolute distance from zero residual. It is, therefore, highly desirable that any quantization scheme introduces no quantization error for zero residuals and the magnitude of quantization errors for non-zero residuals are monotonically increasing with respect to their absolute distance from zero residual so that distortion due to quantization is minimised. The proposed quantization scheme exhibits this property for  $|\epsilon| \leq \lfloor Q/2 \rfloor$  as proved in the following lemma.

**Lemma 3.1.** *Quantization error  $\eta$  of the proposed scheme is equal to the prediction residual  $\epsilon$ , when  $|\epsilon| \leq \lfloor Q/2 \rfloor$ .*

*Proof.*  $|\epsilon| \leq \lfloor Q/2 \rfloor \Rightarrow \epsilon_Q = 0 \Rightarrow \epsilon' = 0 \Rightarrow \eta = \epsilon - 0 = \epsilon$ .  $\square$

As the magnitude of quantization error is bounded in the range  $[0, \lfloor Q/2 \rfloor]$ , the proposed quantization scheme exhibits monotonically increasing and then decreasing quantization

error magnitudes in cycles exhaustively within this range as shown in Figure 4.

**Entropy coding:** Sub-blocking decomposition information and associated non-zero sub motion vectors are encoded using adaptive Huffman variable length coding [13]. We plan to use JPEG-LS coder to encode the quantized residuals at frame-level by taking advantage of long runs of same values. JPEG-LS’s in built quantization scheme, where residual values may be changed within the quantization error limit to maximise coding efficiency by constructing longer runs of same values, is avoided as it can potentially introduce quantization errors in zero residuals.

**Negative value handling by offset:** JPEG-LS cannot handle negative values as it takes input in the form of a grey-scale image. For  $Q \geq 2$ , an 8-bit grey-scale image can be generated from the quantized residual frame by adding the offset  $\phi_Q = \max(-\min_{\forall pixel} \epsilon_Q, 0)$  to all  $\epsilon_Q$ ’s. The offset  $\phi_Q$ , which is the minimum negative quantized residual or zero if no negative value exists, is encoded separately for all frames. Adding the same offset uniformly does not affect the compression efficiency of JPEG-LS as spatial correlations among the residuals, such as length of runs, are not disturbed.

**Negative value handling by folding:** For  $Q = 1$ , adding uniform offset cannot guarantee that all quantized residuals are mapped in the range  $[0, 255]$ , so that an 8-bit grey-scale image can be used as input with JPEG-LS. Although JPEG-LS can handle 16-bit grey-scale image as input, the compression efficiency degrades if the same information can be provided using an 8-bit image. The following folding scheme can be used to generate 8-bit grey-scale input:

$$\epsilon_F = \begin{cases} \epsilon, & \epsilon \geq 0; \\ R - |\epsilon|, & \epsilon < 0 \end{cases} \quad (3)$$

where  $R = 256$  is the range of 8-bit unsigned integer. Despite two residual values, one positive and one negative, folded onto each value in the range  $[1, R - 1]$  by this folding scheme, unambiguous decoding is guaranteed by the following lemma.

**Lemma 3.2.** *If residual  $\epsilon = x - \hat{x}$  is encoded with the folding scheme in 3, decoder can recover  $x$  correctly from its prediction  $\hat{x}$  for all  $0 \leq x < R$ .*

*Proof.* If  $\epsilon = 0$ , the folding scheme maps  $\epsilon$  uniquely onto 0 and hence, decoding cannot have any ambiguity. For each value  $y \in [1, R - 1]$ , positive residual value  $\epsilon_+ = y$  and negative residual value  $\epsilon_- = -(R - y)$  are mapped onto  $y$  by the folding scheme in 3. Hence, the decoder can deduce  $x \in \{x_+, x_-\}$  where  $x_+ = \hat{x} + y$  and  $x_- = \hat{x} + y - R = x_+ - R$ . However, both  $x_+$  and  $x_-$  cannot satisfy the condition that  $0 \leq x < R$ . Hence, decoder can recover  $x$  without any ambiguity.  $\square$

**Decoding:** The decoding process is straightforward in the inverse order of encoding. As the temporal prediction is carried out on the decoded reference frame for both texture and

depth coding, the decoded depth-map will only suffer from quantization errors. The main strength of the proposed pixel-based scalar quantization is that when the residual error is zero, no quantization error is introduced in the corresponding pixels of the decoded depth-map.

## 4. EXPERIMENTAL RESULTS

The proposed method has been tested and compared with HEVC-based Test Model (3D-HTM) version 7.0 [2], reference software of 3D-HEVC and evaluated on the 3D video sequence *Newspaper*, *Balloons*, *Kendo* and *Pozan-street*. Two views have been selected for each sequence (reported in Table 1), and the corresponding depth images have been encoded using the technique described in Section 3. For proposed technique, motion vectors are derived from corresponding compressed texture data using  $QP$  25 of H.264. GOP size 8 is used and I and P frames are encoded for depth-maps. I frames are lossless coded using JPEG-LS. Performance are compared with 3D-HTM (using fixed  $QP$  values 1-30 with 5 difference). 3D-HTM uses I, P and B frames and GOP size and period of I frame is set as 8 similar to proposed technique. All the standard depth modeling modes of 3D-HTM have been enabled. For creating same initial condition 3D-HTM’s I frames are coded using their  $QP = 1$  for almost lossless compression. In order to evaluate the effectiveness of proposed algorithm, view synthesis performance using reconstructed depth data and compressed texture data (Figure 5(a) and 5(b)) have been measured for the input sequences. In this case,  $PSNR$  values are measured against reference virtual views synthesized from uncompressed depth and texture data. Virtual views have been synthesized using ‘‘Renderer version

**Table 1:** Test sequences

Sequence	Resolution	Views	Synthetic views
<i>Newspaper</i>	1024×768	2, 4	2.5, 3, 3.5
<i>Balloons</i>	1024×768	1, 3	1.5, 2, 2.5
<i>Kendo</i>	1024×768	1, 5	2, 3, 4
<i>Pozan_Street</i>	1920×1088	3,5	3.5, 4, 4.5

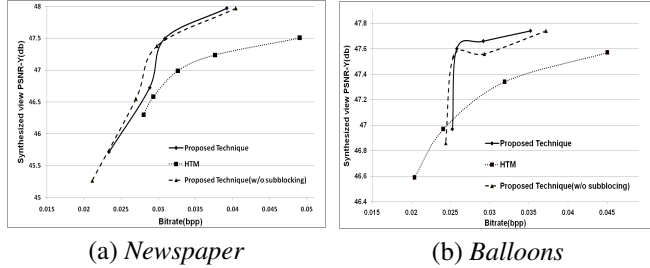
7.0’’ incorporated in 3D-HTM.

Our scalar quantization step is denoted with  $Q$  and HTM’s vector quantization parameter is denoted with  $QP$ . Bit-rate gain is calculated using  $\Delta BR = \frac{\text{proposed-HTM}}{\text{HTM}}$  and synthesized view quality gain is calculated by  $\Delta PSNR = \text{proposed-HTM}$ . Bit-rate is calculated in bit per pixel (bpp) and synthesized view  $PSNR$ -Y is calculated in db.

As the proposed technique don’t lose any information in lossless scheme ( $QP = 1$ ) so depth  $PSNR$  are infinite for it and synthesized view  $PSNR$  are very high. Table 2 reports the bit-rate savings between the proposed method and 3D-HEVC for depth-map signals. In Table 2 *Proposed-*

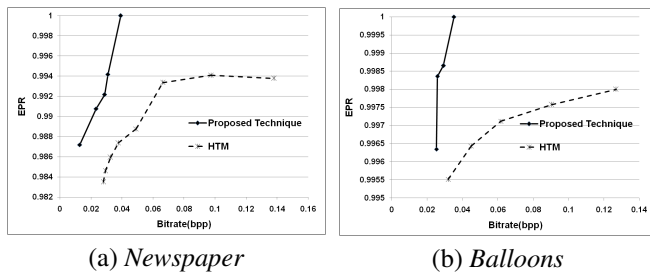
**Table 2:** Depth coding and synthesis view performance comparison between 3D-HEVC and proposed technique( $Q = 1$ ). Note that the minus sign means bit rate reduction.

Sequence	Proposed-QP1(folding)	Proposed-QP1(offset)	HTM-QP1	HTM- $\Delta PSNR$	$\Delta BR(LS)$	$\Delta BR(\Delta PSNR)$
<i>Newspaper</i>	0.039 (47.97)	0.039 (47.97)	0.138 (47.96)	0.049 (47.50)	-72%	-20%
<i>Balloons</i>	0.035 (47.74)	0.037 (47.74)	0.127 (47.74)	0.030 (47.24)	-72%	17%
Kendo	0.056 (47.94)	0.058 (47.94)	0.138 (47.77)	0.099 (47.46)	-59%	-43%
Pozan_Street	0.059 (45.01)	0.061 (45.01)	0.150 (44.82)	0.109 (44.49)	-61%	-46%
Average					-66%	-23%



**Fig. 5:** Synthesized View  $PSNR$  vs Bit-rate. (a) *Newspaper*:  $Q$  [1,3,5,7],  $QP$  [15,20,25,30,35]; (b) *Balloons*:  $Q$  [1,3,5,7],  $QP$  [15,20,25,30]

$QP1(offset)$  are showing the bit-rate without our proposed folding technique using offset. It is higher than our proposed technique  $Proposed-QP1(folding)$  and indicates the significance of our novel folding technique. Proposed technique without any quantization ( $QP = 1$ ) has achieved on average 66% and 23% bit-rate savings against 3D-HEVC with negligible quantization ( $QP = 1$ ) and perceptually unnoticeable ( $\Delta PSNR \approx 0.5$  dB) view synthesis, respectively. The proposed technique has achieved superior image quality of synthesized views against 3D-HEVC for depth-map bit-rate 0.025 bpp or higher (Figure 5) for  $Q > 1$ . The performance of our proposed technique without-sub-blocking has also been reported. Although depth-map quality is better with sub-

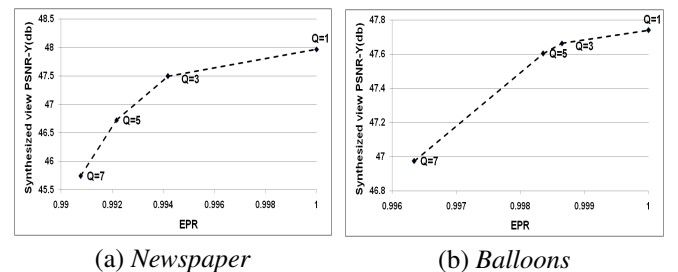


**Fig. 6:** EPR vs Bit-rate (a) *Newspaper*:  $Q$  [1,3,5,7,9]  $QP$  [1,5,10,15,20,25,30,35]; (b) *Balloons*:  $Q$  [1,3,5,7],  $QP$  [1,5,10,15,20]

blocking approach, but in some cases without-sub-blocking scheme outperforms sub-blocking approach in bit-rate reduction due to overhead of sub-blocking. For scalar  $Q > 5$ , like  $Q = 7$  produces maximum pixel level quantization error in range of [+3 -3], therefore synthesized view quality starts to degrade for that.

Our proposed technique outperforms 3D-HEVC up-to 0.025 bpp and this range is sufficient enough in respect of bit-rate, compared to other results presented and compared in [10]. In [10] using compressed texture  $QP 25$  by H.264 intra and intra-residual depth coding, synthesized view  $PSNR$  42 db and 44 db for *Newspaper* and *Balloons* sequences respectively is achieved with bit-rate 0.22 bpp and 0.14 bpp. Their results are better compared to H.264/AVC and JPEG2000. But, in our case, with only bit-rate 0.029 bpp and 0.026 bpp we can achieve 46.72 db and 47.6 db synthesized view  $PSNR$  for *Newspaper* and *Balloons* sequences respectively.

As our proposed technique inherently preserves edge without any approximation and explicit edge handling, therefore edges are preserved with their original shape. Due to quantization error, in decoded depth-map false edge pixels can be introduced and some edge pixels can be eliminated in our proposed technique and 3D-HEVC. To compute the edge preserving capability of proposed technique and 3D-HEVC, we use our own *edge preserving rate* (EPR) measure which represents how effectively coding scheme preserve the edges in depth-map and how efficiently it don't introduce any false edges. Edge is detected using Canny edge detector with minimum and maximum threshold automatically set as 0.0125 and 0.03125, respectively in depth frame. The original depth-map



**Fig. 7:** Effect of edge preservation on synthesized view

is used as reference and edge testing is done on both original and decoded frame. For this test we specify some variables like, *true positive* (TP) *true negative* (TN), *false positive* (FP) and *false negative* (FN). TP means edge was in original frame and is also preserved in decoded frame. TN means no edge was in original frame and also no false edge is introduced in decoded frame. FP means no edge was in original frame but false edge is introduced in decoded frame. FN means edge was in original frame and is not preserved in decoded frame. *EPR* can be calculated as follows:

$$EPR = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

where all are indicating total number of that particular type pixels. Higher value indicates high edge preserving quality of depth-map. If number of false edges introduced and number of edges eliminated are higher then *EPR* will be low. Figure 6 indicates that proposed technique is preserving edge with low bit-rate and better than HTM as false edge due to blocking artefacts is minimized in proposed scheme due to pixel based compression. Figure 7 is showing the effect of edge preservation on synthesized view quality for proposed technique. False edge addition or real edge elimination both hampers synthesized view quality.

## 5. CONCLUSIONS

In this paper, an inherently edge preserving depth-map coding technique has been proposed for applications requiring high quality depth-maps. It exploits temporal correlations in consecutive depth-maps and spatial correlations of inter-coded residuals using uniform pixel-based scalar quantization that avoid any use of transformation. It does not require any edge detection as edges are inherently preserved for the recommended low scalar quantization steps  $Q \in \{1, 3, 5, 7\}$ . As a result it can avoid potential distortion due to edge approximation used by other techniques and conserve significant computational time. On standard 3D video sequences, the proposed depth coding has achieved superior image quality of synthesized views against the new 3D-HEVC standard for depth-map bit-rate 0.025 bpp or higher. The proposed technique with no quantization has achieved on average 66% and 23% bit-rate savings against 3D-HEVC with negligible quantization and perceptually unnoticeable view synthesis, respectively.

## 6. REFERENCES

- [1] ISO/IEC JTC 1/SC 29/WG 11, *Call for Proposals on 3D Video Coding Technology, Switzerland*, vol. Doc. N12036, Geneva (CH), March 2011.
- [2] Joint Collaborative Team on 3D Video Coding Extension Development(JCT-3V) of ITUT VCEG and ISO/IEC MPEG, “3d-hevc test model,” July 2012.
- [3] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken, and T. Wiegand, “3d high-efficiency video coding for multi-view video and depth data,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, Sept 2013.
- [4] C. Fehn, “Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv,” in *Stereoscopic Displays and Virtual Reality Systems XI*, may 2004, vol. 5291 of *SPIE Conference Series*, pp. 93–104.
- [5] Y. Morvan, P. H. N. de With, and D. Farin, “Platelet-based coding of depth maps for the transmission of multiview images,” in *Proceedings of SPIE, Stereoscopic Displays and Applications*, 2006, pp. 93–100.
- [6] M.-K. Kang, J. Lee, J.-Y. Lee, and Y.-S. Ho, “Geometry-based block partitioning for efficient intra prediction in depth video coding,” in *Proceedings of SPIE*, 2010, vol. 7543, pp. 75430A–75430A–11.
- [7] P. Merkle, K. Muller, and T. Wiegand, “Coding of depth signals for 3d video using wedgelet block segmentation with residual adaptation,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.
- [8] G. Shen, W.-S., A. Ortega, J. Lee, and H. Wey, “Edge-aware intra prediction for depth-map coding,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 3393–3396.
- [9] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, “Edge-adaptive transforms for efficient depth map coding,” in *Proceedings of Picture Coding Symposium (PCS)*, 2010, pp. 566–569.
- [10] C. Lee and Y.-S. Ho, “Depth map coding using residual segmentation for 3d video system,” *3D Research*, vol. 4, no. 2, pp. 1–9, 2013.
- [11] M.J. Weinberger, G. Seroussi, and G. Sapiro, “From logo-i to the jpeg-ls standard,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 1999, vol. 4, pp. 68–72.
- [12] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, “Depth map distortion analysis for view rendering and depth coding,” in *16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 721–724.
- [13] D.A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sept 1952.