Signal Processing: *Image Communication* ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Adaptive depth truncation filter for MVC based compressed depth image

Xuyuan Xu [a,*], Lai-Man Po [a], Terence Chun-Ho Cheung [b], Kwok-Wai Cheung [c], Litong Feng [a], Chi-Wang Ting [a], Ka-Ho Ng [a]

[a] *Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China*
[b] *Department of Information Systems, City University of Hong Kong, Kowloon, Hong Kong SAR, China*
[c] *Department of Computer Science, Chu Hai College of Higher Education, Hong Kong SAR, China*

ARTICLE INFO

Article history:
Received 2 April 2013
Received in revised form
19 December 2013
Accepted 20 December 2013

Keywords:
Multiview video plus depth (MVD)
Depth image based rendering (DIBR)
3DV-ATM
Multiview video coding (MVC)
Depth image compression
Depth image filter

ABSTRACT

In multiview video plus depth (MVD) format, virtual views are generated from decoded texture videos with corresponding decoded depth images through depth image based rendering (DIBR). 3DV-ATM is a reference model for the H.264/AVC based multiview video coding (MVC) and aims at achieving high coding efficiency for 3D video in MVD format. Depth images are first downsampled then coded by 3DV-ATM. However, sharp object boundary characteristic of depth images does not well match with the transform coding based nature of H.264/AVC in 3DV-ATM. Depth boundaries are often blurred with ringing artifacts in the decoded depth images that result in noticeable artifacts in synthesized virtual views. This paper presents a low complexity adaptive depth truncation filter to recover the sharp object boundaries of the depth images using adaptive block repositioning and expansion for increasing the depth values refinement accuracy. This new approach is very efficient and can avoid false depth boundary refinement when block boundaries lie around the depth edge regions and ensure sufficient information within the processing block for depth layers classification. Experimental results demonstrate that the sharp depth edges can be recovered using the proposed filter and boundary artifacts in the synthesized views can be removed. The proposed method can provide improvement up to 3.25 dB in the depth map enhancement and bitrate reduction of 3.06% in the synthesized views.

1. Introduction

In the last two decades, 3D video (3DV) technology advances significantly. It evolves from the simple stereoscopic system to more realistic multiview video systems, such as autostereoscopic 3DV displays and free-view TV [1,2]. However, realization of multiview systems using large number of texture video views is not efficient. Depth-enhanced video format such as multiview video plus depth (MVD) format has attracted much attention due to its high efficiency and compatibility with conventional multiview video systems. Instead of coding numerous views, two or three texture views with their corresponding depth images in MVD format are used in the recent development of multiview video coding (MVC) standard. Virtual views can be generated by specifying the preference viewing angles through the depth image based rendering (DIBR) [3]. There are two reference test models for 3DV codecs, 3DV-ATM [4] and 3DV-HTM [5], which are based on H.264/AVC and H.265/HEVC, respectively.

In 3DV-ATM, depth images are first downsampled by half in both horizontal and vertical directions using the

[*] Corresponding author.
E-mail address: xuyuanxu2@gmail.com (X. Xu).

0923-5965/$ - see front matter © 2014 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.image.2013.12.005

MPEG 13-tap downsampling filter. After that the texture videos with depth videos at reduced resolution are encoded using 3DV-ATM coding scheme. In decoding, the bilinear filter is used to upsample the decoded depth images into their original resolution. Coding techniques in 3DV-ATM not only include conventional intra-view prediction and inter-view prediction for both texture and depth videos, but also contain new coding techniques favorable for a higher compression ratio, for example, view synthesis prediction (VSP), in-loop joint inter-view depth filter, depth range based weighted prediction for depth coding, etc. The characteristic of depth image, however, is different from that of texture image. Depth image is a piecewise smooth image that has large homogeneous regions within objects and sharp depth changes at object boundaries, as shown in Fig. 1(a). H.264/AVC based coding method in 3DV-ATM, however, does not handle well at sharp object boundaries. These boundaries are always blurred by the subsampling process or transform coding even with ringing artifacts (depth inconsistencies around boundaries) in the decoded depth images as shown in Fig. 1(b). This results in annoying artifacts in the synthesized views, as shown in Fig. 1(d).

To tackle this problem, many depth map enhancement filters have been proposed. They are generally classified as pixel-based or block-based depth map filters. In pixel-based depth map filters, bilateral filter [6] and clustering of depth layer [7] are the most popular approaches. In [6], a depth adaptive joint bilateral filter was proposed. It takes the advantages of smoothing regions inside object boundaries and preserving sharp edges. But the proper selection of filter parameters has a great impact on sharp edge recovery. In the depth clustering filter and depth reconstruction filter [7], sharp edge can be retained through the depth layer classification. However, pixel-based depth map filter may create inconsistent depth values around the

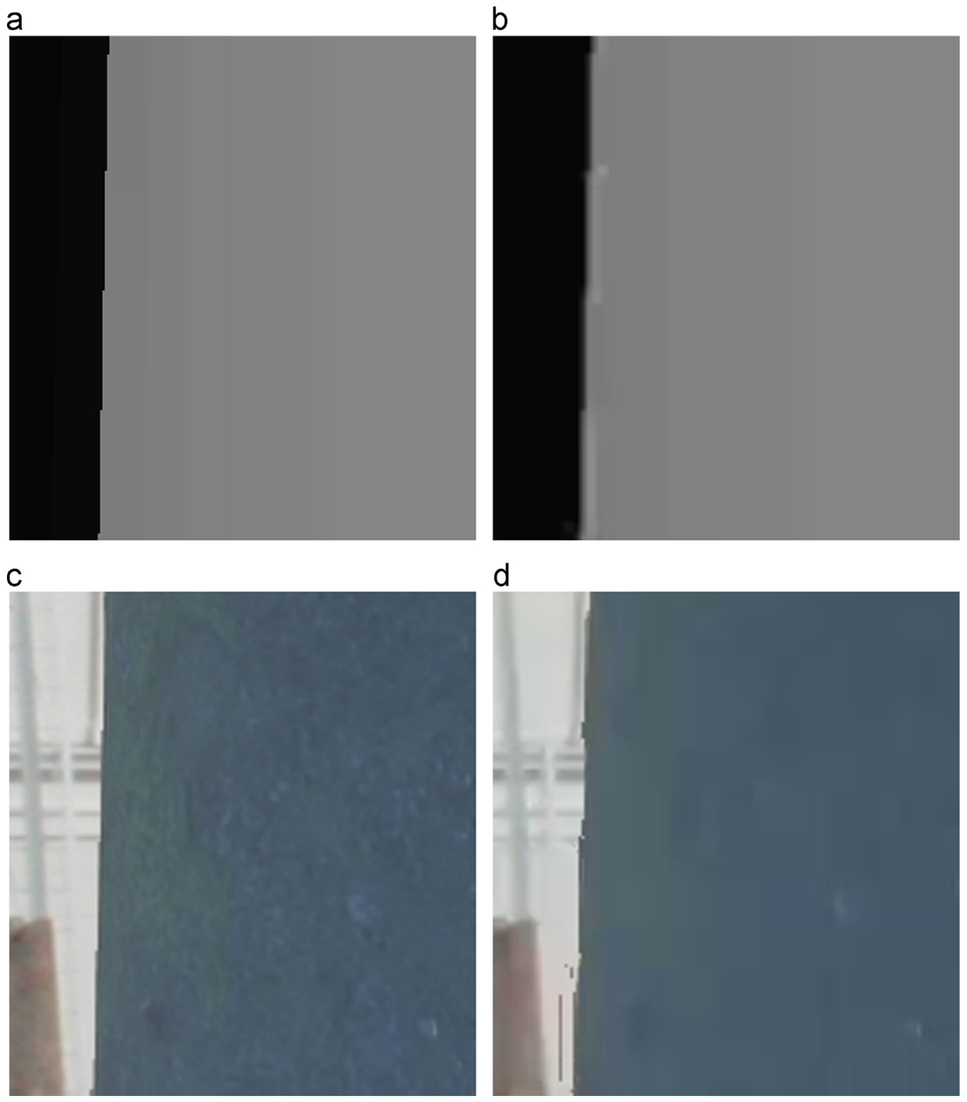

Fig. 1. (a) Original depth image, (b) compressed depth image with $QP=31$, (c) synthesized view from uncompressed texture and depth, (d) synthesized view from compressed texture and depth.

depth edge regions and generate ringing artifacts. In addition, this depth inconsistency will sometimes result in texture distortion in synthesized views. In general, the block-based depth map filter can avoid the inconsistent depth values and remove ringing artifacts. In [8,9], a depth map post-processing filter is designed to minimize the compression artifacts by analyzing the histogram inside each block. Unfortunately, the layer classification fails in the situation that the boundaries of the blocks lying around the depth edge regions due to the lack of information of all the depth layers.

In addition, the conventional block-based depth map filters divide the depth image into non-overlapped blocks. Only blocks containing large depth discontinuity edges, like blocks with white boundaries as shown in Fig. 2(a), are filtered. However, some of these block boundaries located closer to edge regions are similar to the block A shown in Fig. 2(b). In this case, sharp edges are difficult to be recovered due to insufficient depth layer information within the block. For example, the block A of Fig. 2(b) contains foreground information and small part of the smooth edges. It is because the smooth edge regions only take a very small percentage of the block A. They will be treated as noise and refined as the foreground regions. The lack of information of background regions makes it difficult to recover the original sharp edge. Sometimes it even

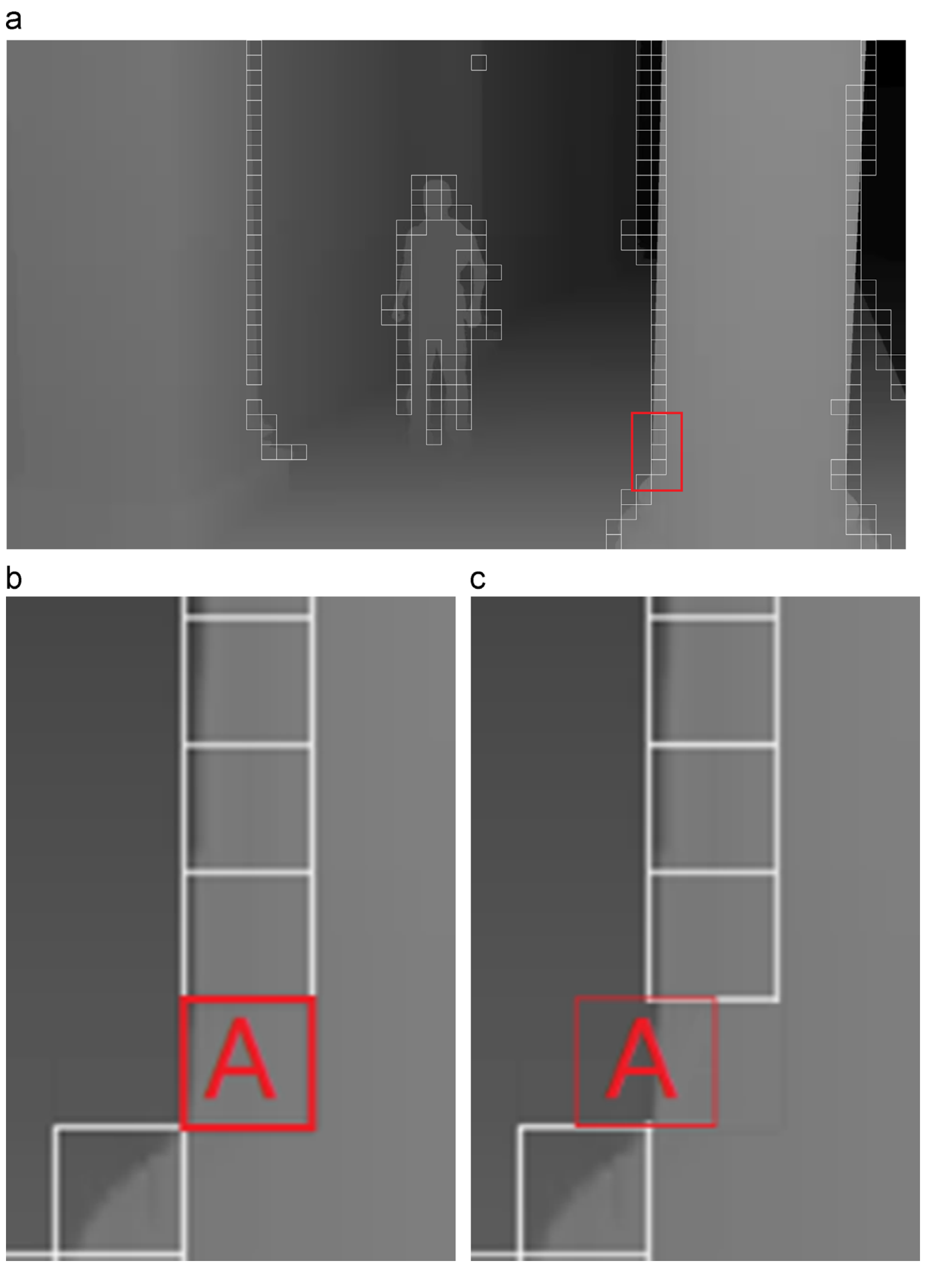

Fig. 2. (a) Decoded depth with edge blocks, (b) cropped rectangular region of the decoded depth, (c) the refined block A position.

degrades the quality of synthesized views in both objective and subjective evaluations that cause the overall performance to degrade after applying the block filter. If these blocks can be placed in the correct position, such as the center of block A is located at the vertical edge regions as shown in Fig. 2(c), the refinement of depth values has reliable reference since this center positioning provides sufficient depth values for each of the depth layers. Smooth edge regions of block A can be easily treated as noise and refined using the correct background or foreground depth layer information to recover its original depth values. Thus the refinement based on clustering can become very effective.

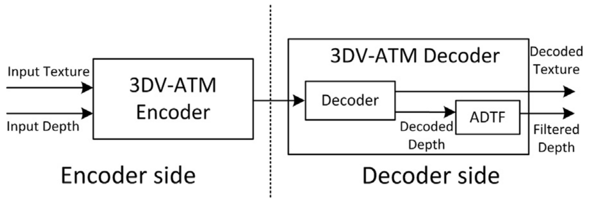

Fig. 3. Adaptive depth truncation filter (ADTF) for 3DV-ATM.

In this paper, a block-based filter with adaptive block positioning and expansion is proposed to sharpen the smooth depth edges of 3DV-ATM compressed depth image. Details of adaptive depth truncation filter are presented in Section 2. Experimental results are shown in Section 3. Finally, a conclusion is drawn in Section 4.

2. Adaptive depth truncation filter

Depth image exhibits a high degree of spatial correlation except at object boundaries [10,11]. Object boundaries

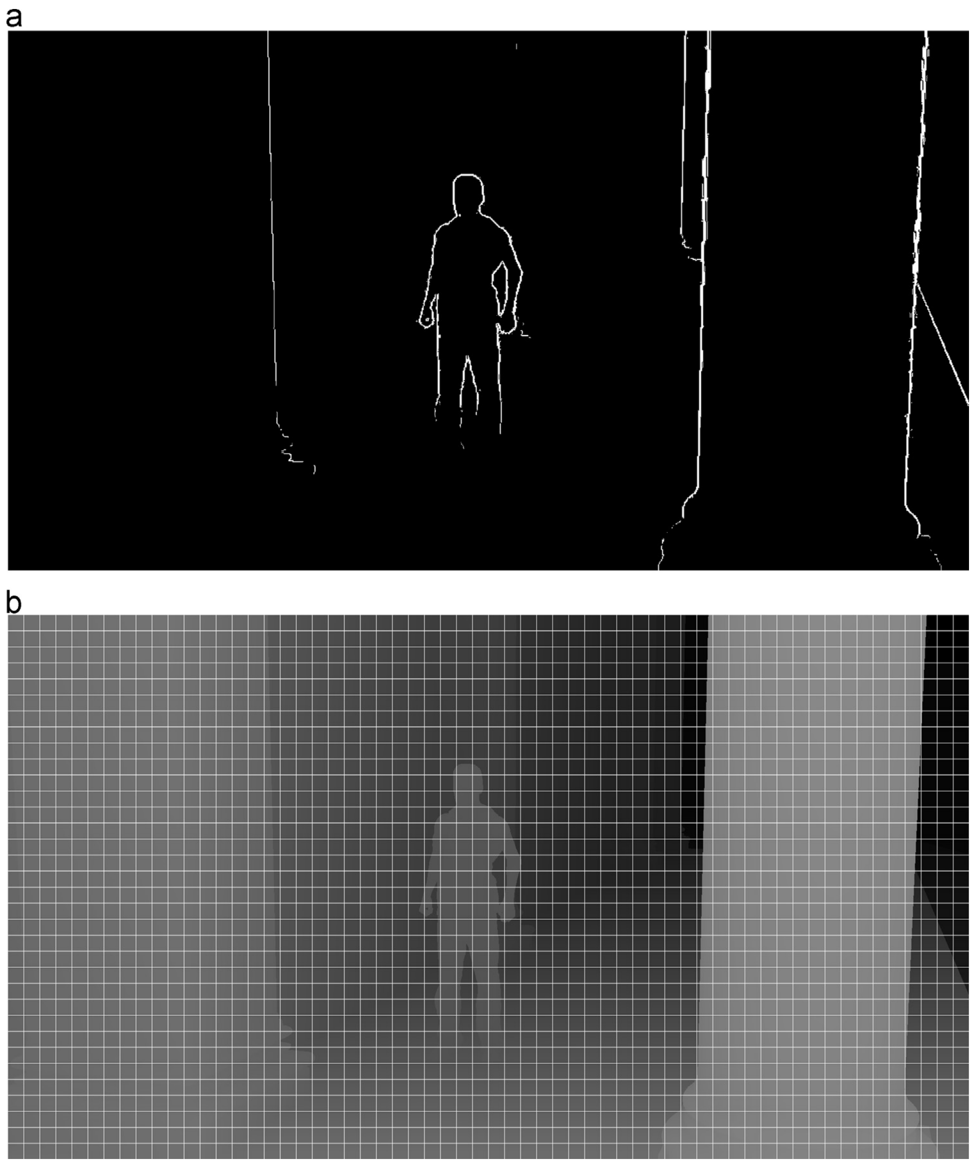

Fig. 4. (a) Example of edge pixels that detected from a decoded depth image, (b) a depth image is divided into non-overlapped blocks.

are more sensitive to coding errors compared to smooth and flat regions [11]. The proposed adaptive depth truncation filter (ADTF) aims to enhance the depth edge regions and to recover the original sharp edges from blurred edges caused by depth image subsampling or compression processes, which is implemented in the decoder of 3DV-ATM 5.0 as a post filter to the decoded depth image as Fig. 3. The ADTF includes three steps: (1) edge pixel detection, (2) edge block repositioning and expansion, and (3) depth refinement and layer based Smoothing. Sharp depth edge pixels are first detected with horizontal and vertical gradients. Depth image is then divided into non-overlapped blocks for edge block detection. Only those blocks consisting of detected depth edge pixels are considered as edge blocks for upcoming processing. To avoid the edge block boundaries being too close to the depth edge regions, each edge block is adaptively repositioned and then expanded in order to locate the edge regions at the center of the block. The adaption in block position and size could increase the correctness of depth

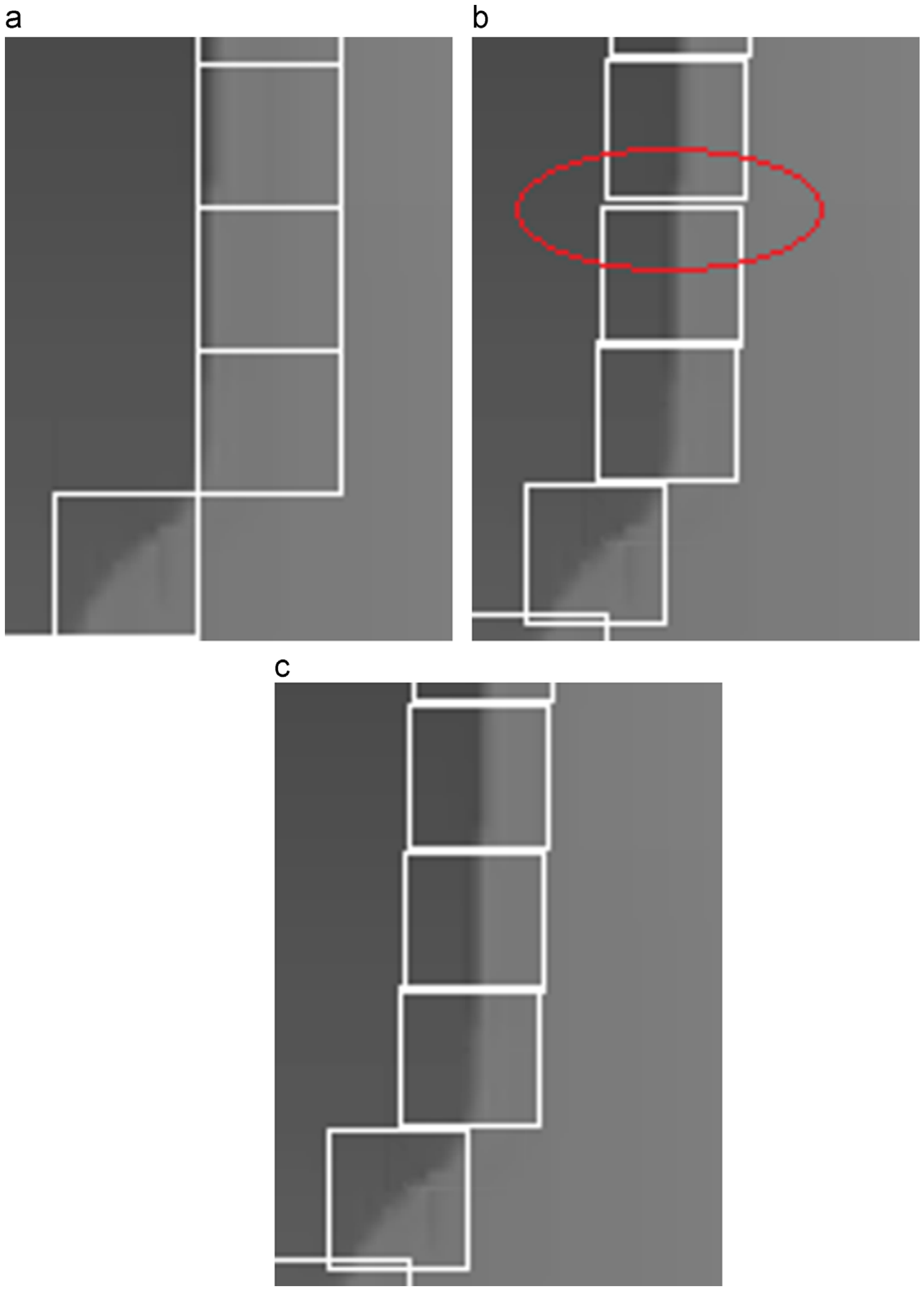

Fig. 5. (a) Detected depth edge blocks, (b) depth edge blocks after repositioning, (c) depth edge blocks after repositioning and expansion.

layer classification in depth refinement. Finally, a modified block truncation technique is used to refine the depth values within the edge blocks. The details of each step are described in the following subsections.

2.1. Edge pixel detection

Sharp depth edge pixels are detected by the horizontal gradients $\nabla f_x$and vertical gradients $\nabla f_y$of depth image defined as below:

$$\nabla f_x = \frac{\partial f}{\partial x} \tag{1}$$

$$\nabla f_y = \frac{\partial f}{\partial y} \tag{2}$$

If horizontal or vertical gradients are larger than a predefined threshold (D_T) (i.e. $|\nabla f_x| > D_T$ or $|\nabla f_y| > D_T$), the depth pixel at position (x, y) is treated as a sharp edge pixel, i.e. the detected edge pixels in white color as shown in Fig. 4(a). Since the view synthesis of ATM only has the horizontal disparity, in our experiment, a 1-D kernel, $(-1, 1)$, is used to calculate both horizontal and vertical gradients and D_T is set adaptively based on the hole's size created by the depth value difference between two horizontal adjacent pixels to the virtual view. The location of the virtual view is the nearer neighborhood of the input view. The threshold D_T is calculated as:

$$D_T = \frac{h}{t_c f} \frac{1}{\left(\frac{1}{255 z_n} - \frac{1}{255 z_f} \right)} \tag{3}$$

where t_c and f are the baseline distance and the focal length, respectively. z_n and z_f represent the nearest distance and the farthest distance in the scene. h is the hole size created by the depth difference of adjacent pixels and set to 2. The camera parameters are encoded and transmitted to the decoder in 3DV-ATM. Thus, using the adaptive threshold D_T, no extra data is needed to be transferred to decoder. Also, no modification is required for the encoder.

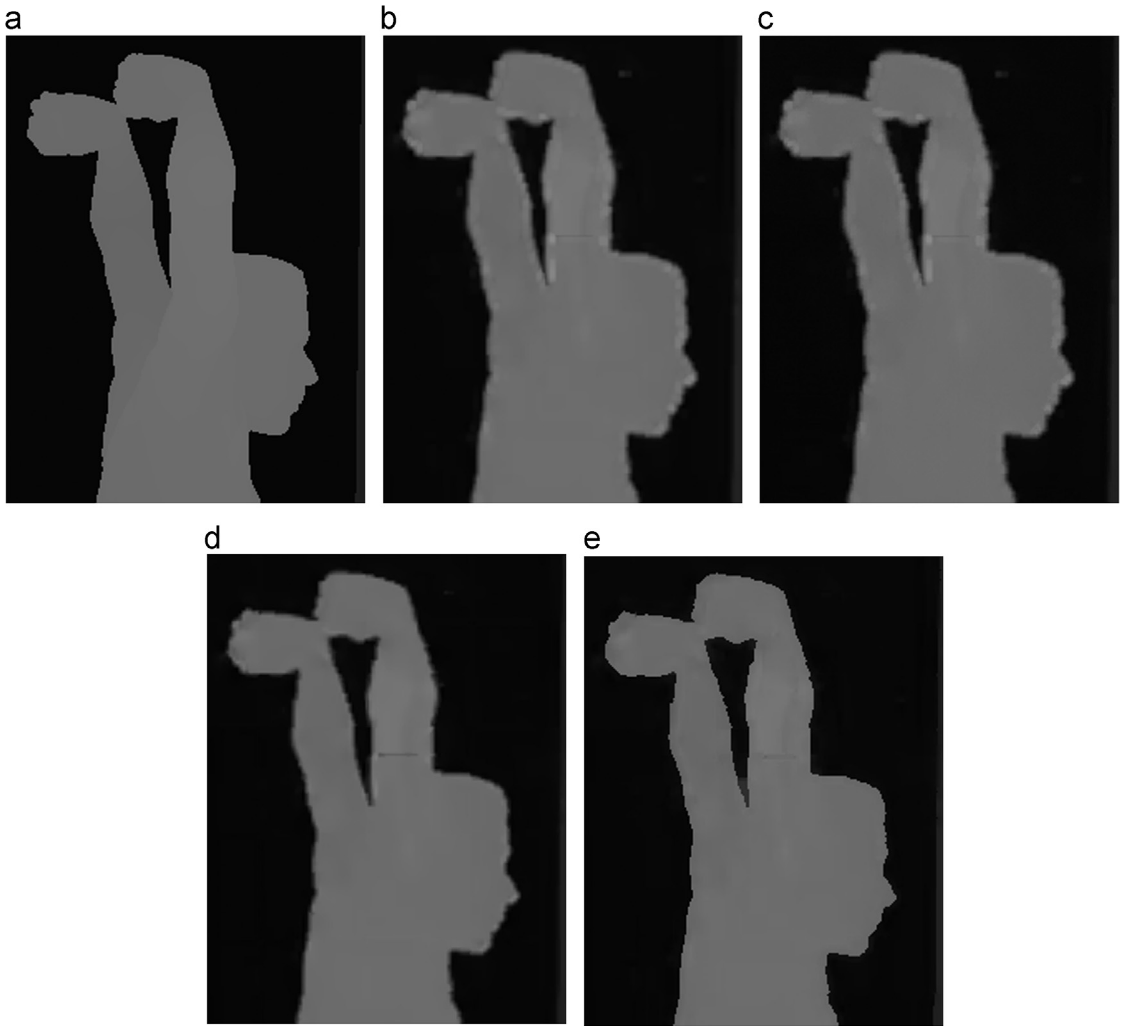

Fig. 6. (a) Original depth image of 148th frame view 5 of *Undo_dancer*, (b) compressed depth image (QP=31), (c) depth image with AJBF, (d) depth image with ASF and (e) depth image with ADTF.

2.2. Adaptive block positioning and expansion

After depth edge pixel detection, a depth image is divided into $M \times M$ non-overlapped blocks as shown in Fig. 4(b). Block size is set adaptively based on the image resolution as

$$M = 2^{\log_2 round\left(\frac{W}{a}\right)} \tag{4}$$

where W is the width of the depth image, a is a constant ($a = 125$) and the *round* $(\cdot)$ function rounds the input value to its nearest integer value. The non-overlapped blocks that contain edge pixels are considered as edge blocks as shown in Fig. 2(a) and only these edge blocks will be processed. In order to avoid the boundaries of these edge blocks lying too close to the smooth depth edge regions as the problem described in the introduction section, the positions and sizes of these edge blocks are adjusted before depth refinement. Basically, the block repositioning aims at locating most of the depth edge pixels at the block′s center. This repositioning can be efficiently realized by using the average of edge pixels′ positions within the processing block as the block′s center. Then, the top-left

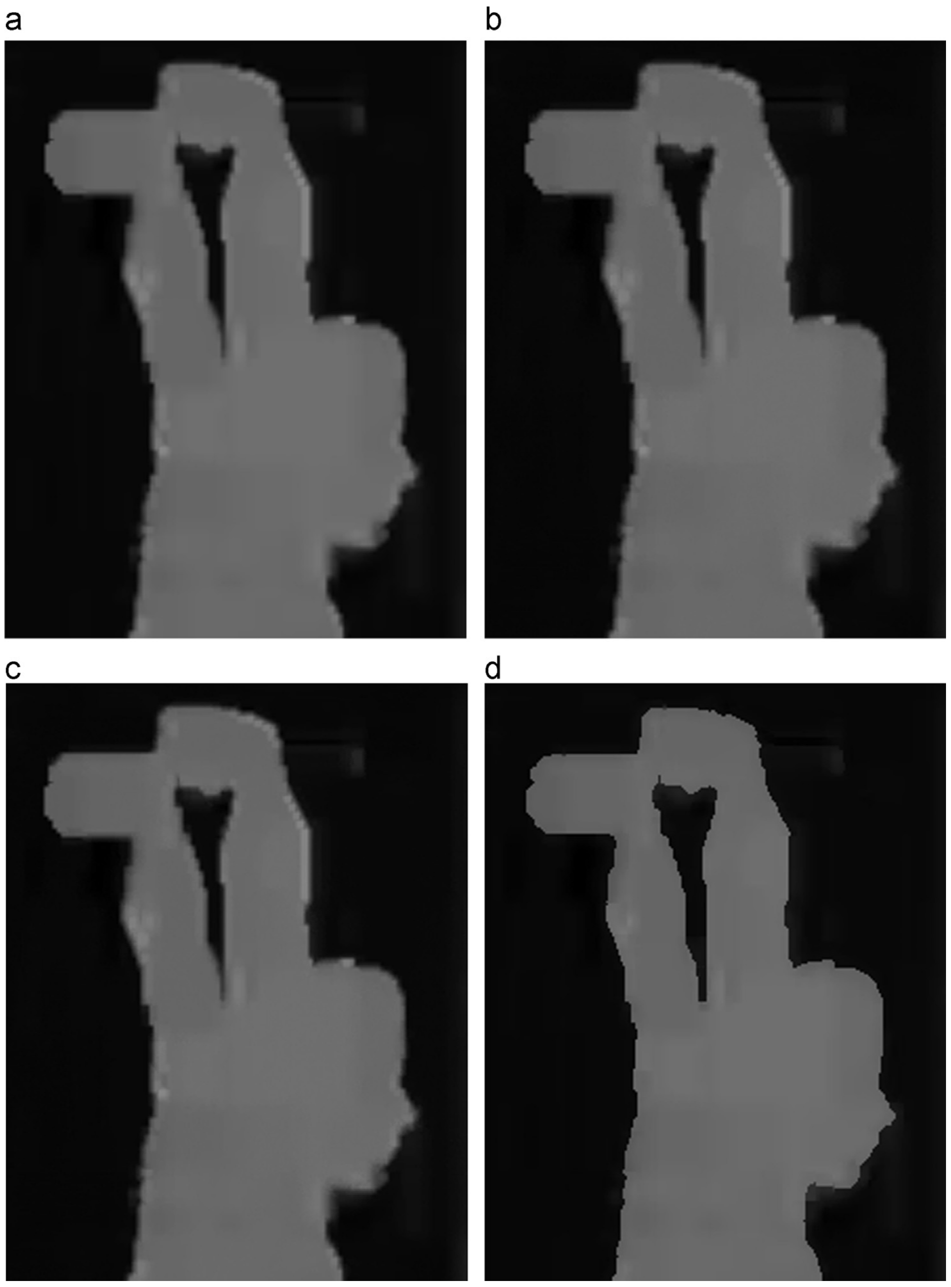

Fig. 7. (a) Compressed depth image ($QP = 41$) of 148th frame view 5 of *Undo_dancer*, (b) depth image with AJBF, (c) depth image with ASF and (d) depth image with ADTF.

position of the repositioned edge block can be determined by

$$B=(x_B,y_B)=round\left(\frac{\sum_{P_i\in\varphi}P_i}{N}-Q\right) \tag{5}$$

where P_i is depth pixel's position (x_i, y_i), φ represents the set of depth edge pixels inside the processing block, N is the total number of depth edge pixels inside the $M\times M$ block, and $Q=(M/2, M/2)$. In Eq. (5), the subtraction of Q is used to get the xy-coordinate of the top-left position of the block. Fig. 5 shows examples of edge block locations before and after repositioning. Centers of the edge blocks can be located to vertical depth edges after this simple process as shown in Fig. 5(b). However, this repositioning may cause some depth edge pixels that lie inside the edge block before reposition to outside region of the block as indicated in the circle region in Fig. 5(b). In order to tackle this problem, the size of the edge block is also needed to adaptively expand for covering all these missing depth edge pixels. This block expansion process can be realized by using the minima of horizontal and vertical coordinates between the edge pixels inside the block before repositioning and the repositioned coordinates B as the new top-left position B' of the expanded block. Thus, the top-left position of the expanded edge block can be expressed as

$$B'=(x'_B,y'_B)=(\min(x_B,\min_{P_i\in\varphi}\{x_i\}),\ \min(y_B,\min_{P_i\in\varphi}\{y_i\})) \tag{6}$$

where the function min (a, b) is used to find the minimum value between a and b. The expanded block of size $M'\times N'$ can be respectively determined by the maxima of horizontal and vertical distances from B' to its lower-right block coordinate (x_B+M, y_B+M) of the repositioned block and from B' to maxima edge pixel coordinates inside the block before repositioning. Thus, M' and N' can be expressed as

$$M'=\max(x_B+M-x'_B,\max_{P_i\in\varphi}\{x_i\}-x'_B) \tag{7}$$

$$N'=\max(y_B+M-y'_B,\max_{P_i\in\varphi}\{y_i\}-y'_B) \tag{8}$$

where the function max (a, b) is used to find the maximum value between a and b. Fig. 5(c) shows the edge blocks after the adaptive block expansion. By using these repositioned and expanded edge blocks for depth values refinement, the refinement accuracy could be significantly improved.

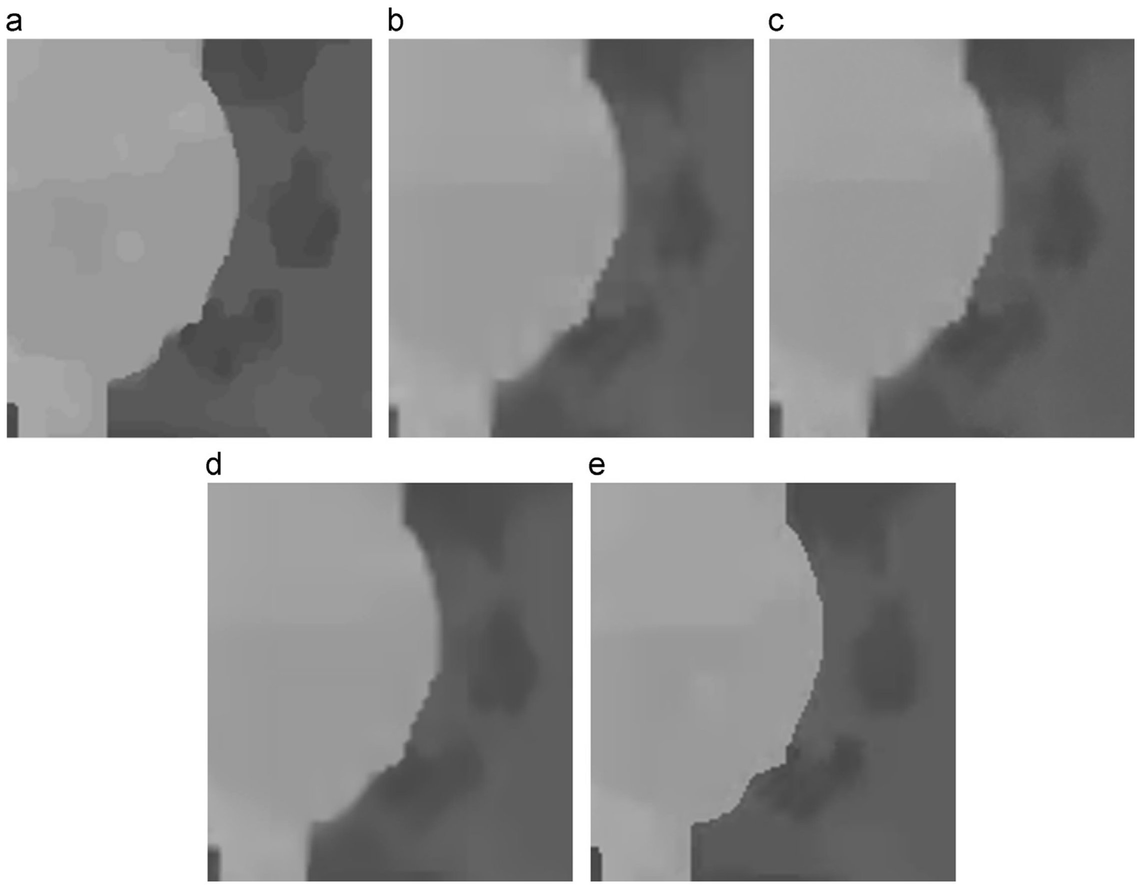

Fig. 8. (a) Original depth of 169th frame view 3 of *Balloon* (b) (b) compressed depth image ($QP=31$), (c) depth image with AJBF, (d) depth image with ASF and (e) depth image with ADTF.

2.3. Depth refinement and layer based smoothing

After the edge block repositioning and expansion, a modified block truncation method is used to refine the depth values. It first classifies the processing block into two layers, foreground region (R_F) and background region (R_B) by comparing depth values with the mean depth value (D_m) of the block through the following equations:

$$R_F = \{(x,y)|D(x,y) \geq D_m\}, \tag{9}$$

$$R_B = \{(x,y)|D(x,y) < D_m\} \tag{10}$$

where $D(x,y)$ is the depth value at the location (x, y). After the layer classification, mean values of the foreground region and the background region are calculated and denoted as m_F and m_B, respectively. Depth values of detected edge pixels are refined by selecting the nearer mean values of foreground and background regions as follow:

$$D'(x,y) = \begin{cases} m_F, & |D(x,y)-m_F| \leq |D(x,y)-m_B| \\ m_B, & \text{otherwise} \end{cases} \tag{11}$$

where $D'(x,y)$is the refined depth value. With this depth value refinement of the edge blocks, the blurred depth edges can be recovered. Since block size is set to a relative small value (e.g.: 8×8 for image with resolution of 1024×768) and the refined block is the edge regions, it has a very high chance of having only two layers and makes the block truncation very efficient.

To further remove the ringing artifacts, a layer based smooth filter is used. The 3×3 average filter is independently applied to the foreground region (R_F) and background region (R_B). Sharp edges can be maintained and the noise around the edges regions due to compression can be reduced.

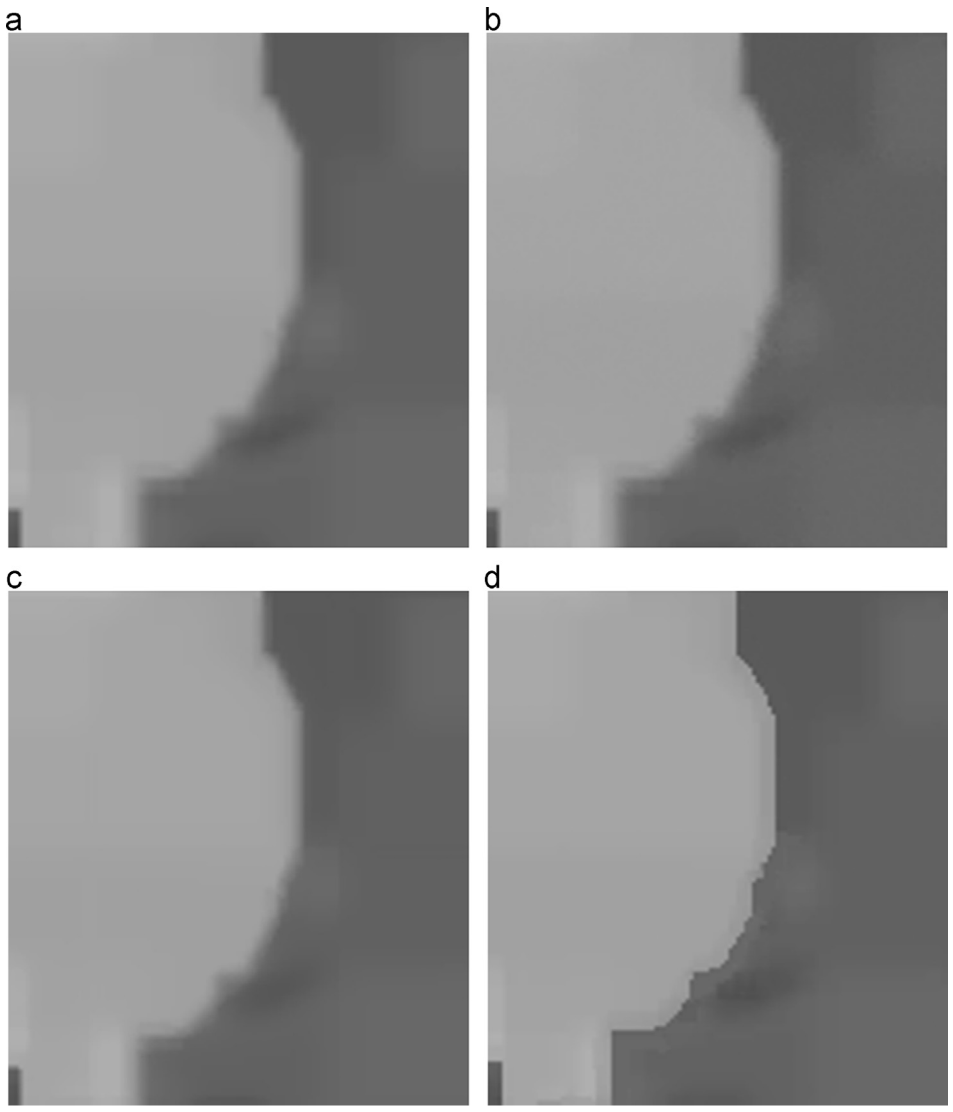

Fig. 9. (a) Original depth of 169th frame view 3 of *Balloon* (b) (b) compressed depth image (QP=41), (c) depth image with AJBF, (d) depth image with ASF and (e) depth image with ADTF.

3. Experimental results

In order to evaluate the performance, the proposed method is used as a post-processing depth enhancement filter of the decoded depth image in 3DV-ATM. The experiment is conducted according to the codec configuration (EHP profile) of the common testing conditions [12] based on 3DV-ATM 5.0. The proposed adaptive depth truncation filter (ADTF) is implemented in the decoder of 3DV-ATM 5.0 and applied to the compressed depth images without dilation filter since our purpose is to recover the depth images from compression. The dilation filter with size 3×3 is integrated in 3DV-ATM 5.0 and applied after depth image is upsampled into its original resolution. The dilation filter aims at extending the foreground to cover the transitional color edges in order to remove the boundaries artifacts of synthesized virtual views and improve the objective performance in terms of Peak signal-to-noise ratio (PSNR). The 3DV-ATM default setting for anchor is with dilation filter enable and view synthesis optimization enable [13]. The results from the proposed ADTF are compared with a pixel-based filter using adaptive joint bilateral filter (AJBF) [6] and a block-based filter using adaptive sharpening filter (ASF) [8]. Three texture views plus three depth views in MVD format are encoded and decoded by the 3DV-ATM 5.0. The decoded depth images are enhanced by AJBF, ASF and ADTF and compared with the original depth videos without downsampling by PSNR. The decoded texture videos with the enhanced depth videos are used to generate the virtual views by the 3DV-HTM 4.0 rendering reference software according to the common test conditions [12]. Six virtual views between three cameras (evenly distributed) are synthesized for evaluation. The virtual views generated from original texture videos with original depth videos without downsampling are used as the reference views for objective performance evaluation by using PSNR. Seven well-known test sequences, *Poznan_hall2*, *Poznan_street*, *Undo_dancer*, *GT_Fly*, *Newspaper*, *Kendo* and *Balloon* are used in the experiment.

The cropped original, compressed with different *QP* values, and enhanced depth images of the 148th frame

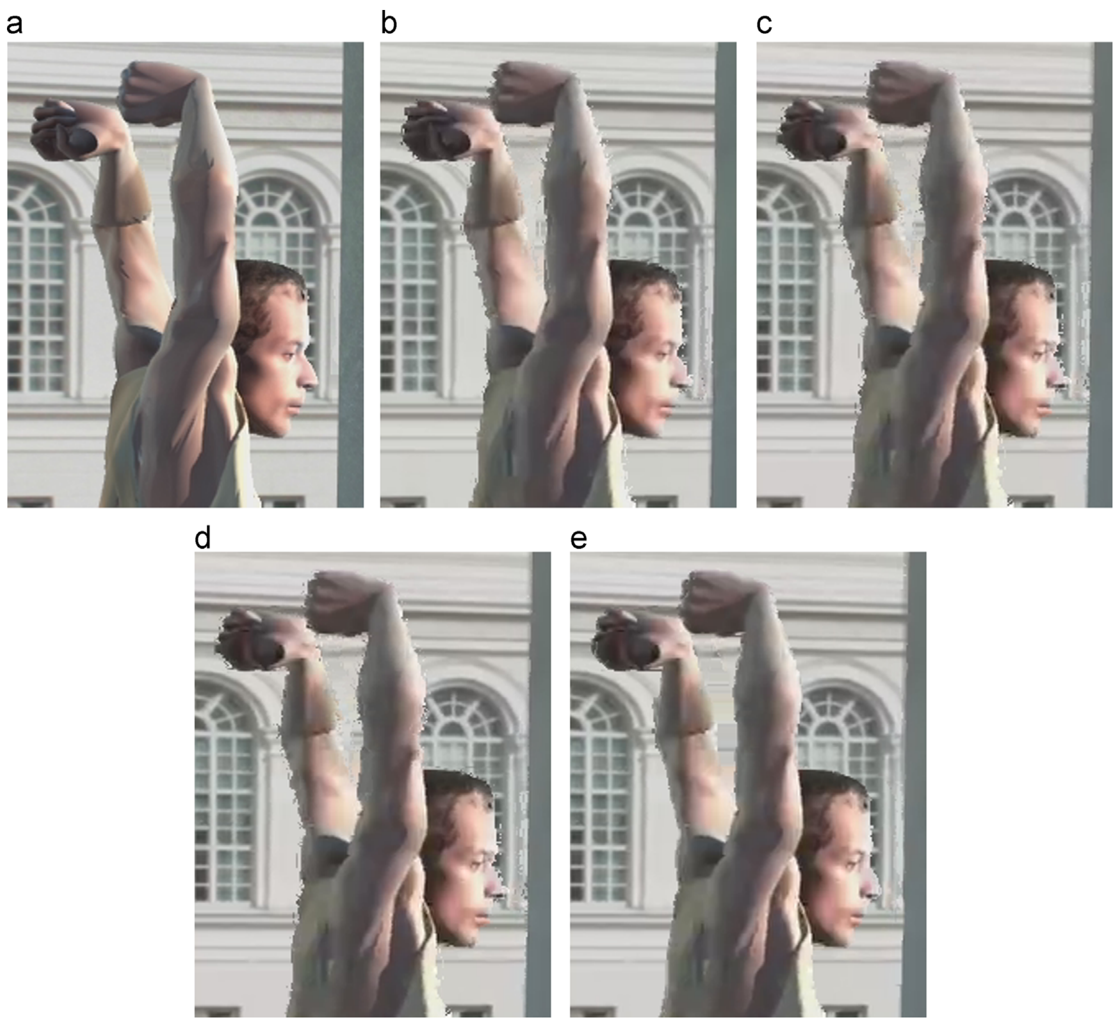

Fig. 10. Synthesized view of 148th frame view 3 of *Undo_dancer* based on (a) original texture and depth, (b) compressed texture and depth ($QP=31$), (c) compressed texture and depth using AJBF, (d) compressed texture and depth using ASF, (e) compressed texture and depth using ADTF.

view 5 of *Undo_dancer* sequence and 169th frame view 3 of *Balloon* sequence are shown in Figs. 6, 7, 8 and 9, respectively. Their corresponding synthesized views are shown in Figs. 10, 11, 12 and 13, respectively. The large distortion around the object boundaries is easily observed in the reconstructed depth images that compressed by 3DV-ATM as shown in Figs. 6 and 8(b), which is mainly due to information loss in subsampling and transform coding. Depth images enhanced by AJBF and ASF still have blurred edges around the depth transitional regions as in Figs. 6 and 8(c) and (d). In view synthesis, the blurred edges result in the annoying boundary artifacts as in Fig. 10 (c)–(d) and Fig. 12(c)–(d). Both AJBF and ASF are based on the bilateral filter. They try to recover the depth image after being compressed. Depth images enhanced by ASF are better than AJBF, since the ASF optimizes the parameter for each of the blocks adaptively while AJBF uses a fixed parameter setting. In addition, ringing artifacts are reduced in ASF since it is a block-based depth map filter. From Figs. 6 and 8(e), sharp depth edges can be recovered

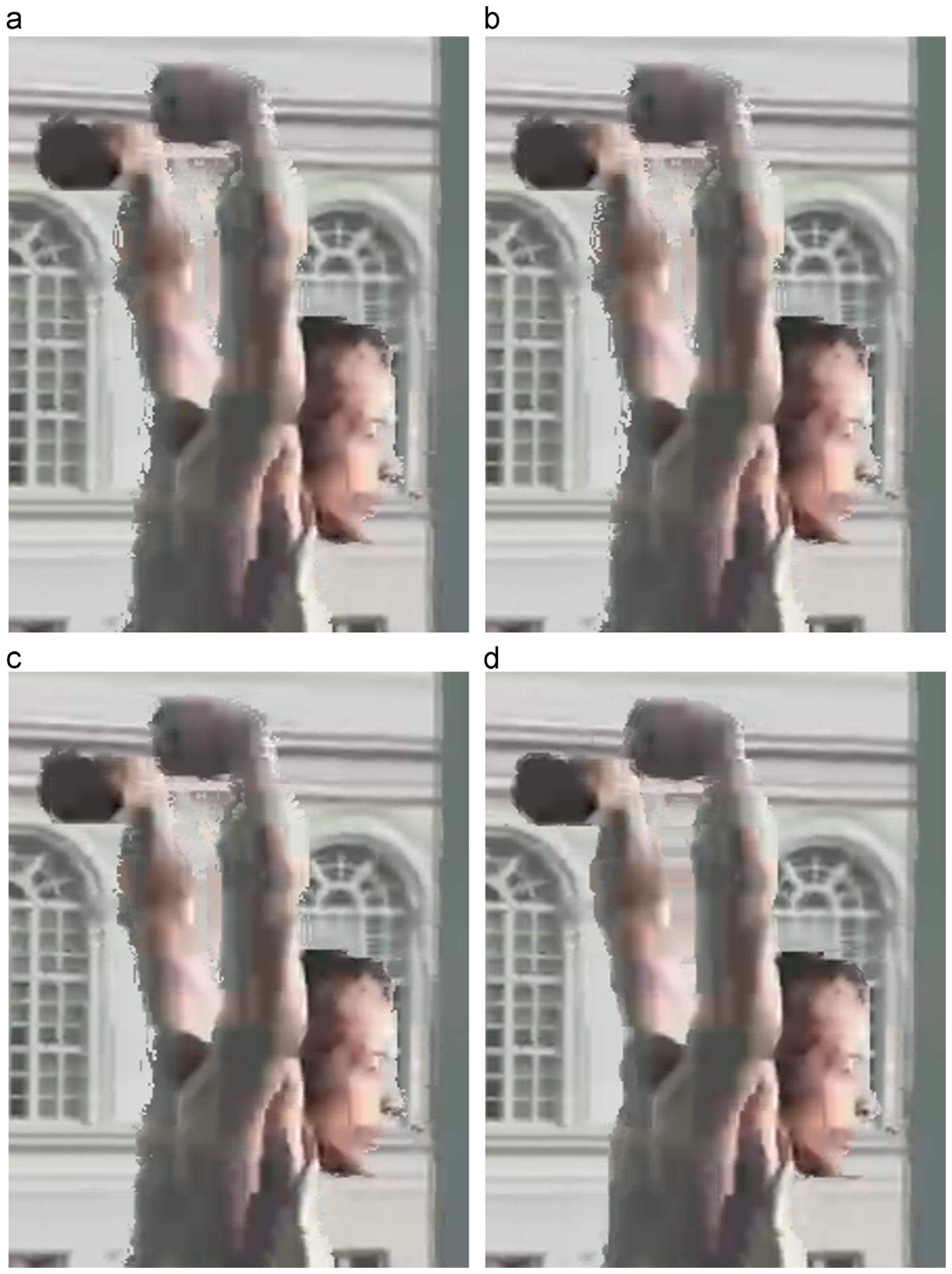

Fig. 11. Synthesized view of 148th frame view 3 of *Undo_dancer* based on (a) compressed texture and depth ($QP=41$), (b) compressed texture and depth using AJBF, (c) compressed texture and depth using ASF, (d) compressed texture and depth using ADTF.

by the proposed filter (ADTF) and these two depth images are more similar to their original depth images without downsampling. By comparing the results of synthesized views from AJBF and ASF, Fig. 10(e) and Fig. 12(e) show that the proposed filter can significantly reduce the boundary artifacts caused by the blurred depth edges. For larger *QP* (quantization parameter) value, there are more distortions in the edges regions, which increase the difficulty in recovering sharp edges. The proposed ADTF can also recover the sharp edges when the *QP* value is large, and the synthesized boundary artifacts can be removed since the sharp depth edges can be recovered as Figs. 7, 11 and Figs. 9 and 13.

Table 1 presents the objective evaluation based on PSNR improvement of decoded depth images with AJBF, ASF and the proposed ADTF. From Table 1, it shows the enhanced depth images by AJBF and ASF have significant improvement in terms of PSNR. Results of AJBF attain lower PSNR values compared with those of ASF. However, the enhanced depth images by AJBF and ASF still contain blurred depth edges that result in annoying boundary artifacts in the synthesized views. Thus, the PSNR of synthesized views through AJBF and ASF are very similar to the results of ATM anchor as in Table 2. From Tables 1 and 2, the results show that depth image filtered by ADTF and the synthesized texture views from depth images enhanced by the ADTF filter are more similar to the reference views compared with AJBF and ASF. Sharp depth edges are recovered by ADTF and the boundaries artifacts are removed in the synthesized virtual views. The proposed method has improvement up to 3.25 dB in the depth image and bitrate reduction of 3.06% in the synthesized views. This exists in *Undo_dancer* sequence, in which depth distances between background and foreground are large and its depth images satisfy our assumption that depth image exhibits a high degree of spatial correlation except at object boundaries. Table 3 shows a summary of ADTF compared with 3DV-ATM anchor, in which the average PSNR improvement for decoded depth image is 1 dB and the average bitrate reduction in synthesized views is 1.75%. Performance for small *QP* value (*QP* 26 or 31) is obviously for both depth image and synthesized views, while for large *QP* value, like 36 or 41, the compressed depth images have PSNR improvement, but there is almost no improvement for the synthesized views. Although sharp depth edges are recovered by ADTF and the boundary artifacts are removed as example of *Undo_dancer* and *Balloon* in Figs. 7, 11, 9 and 13, the PSNR tabulated in Table 2 has no obviously improvement for the synthesized virtual view. It is because the quality of the synthesized view not only depends on the depth image, but also relies upon the compressed texture image. The

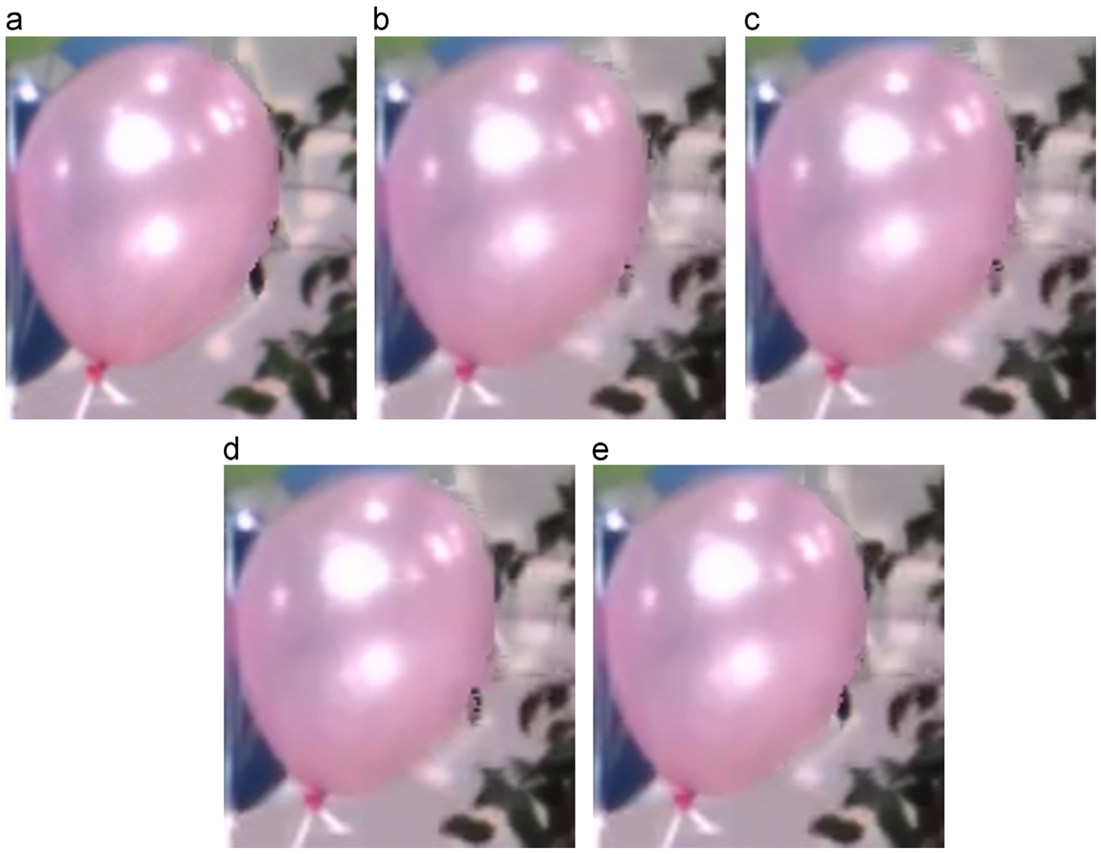

Fig. 12. Synthesized view of 169th frame view 1.5 of *Balloon* based on (a) original texture and depth, (b) compressed texture and depth ($QP=31$), (c) compressed texture and depth using AJBF, (d) compressed texture and depth using ASF, (e) compressed texture and depth using ADTF.

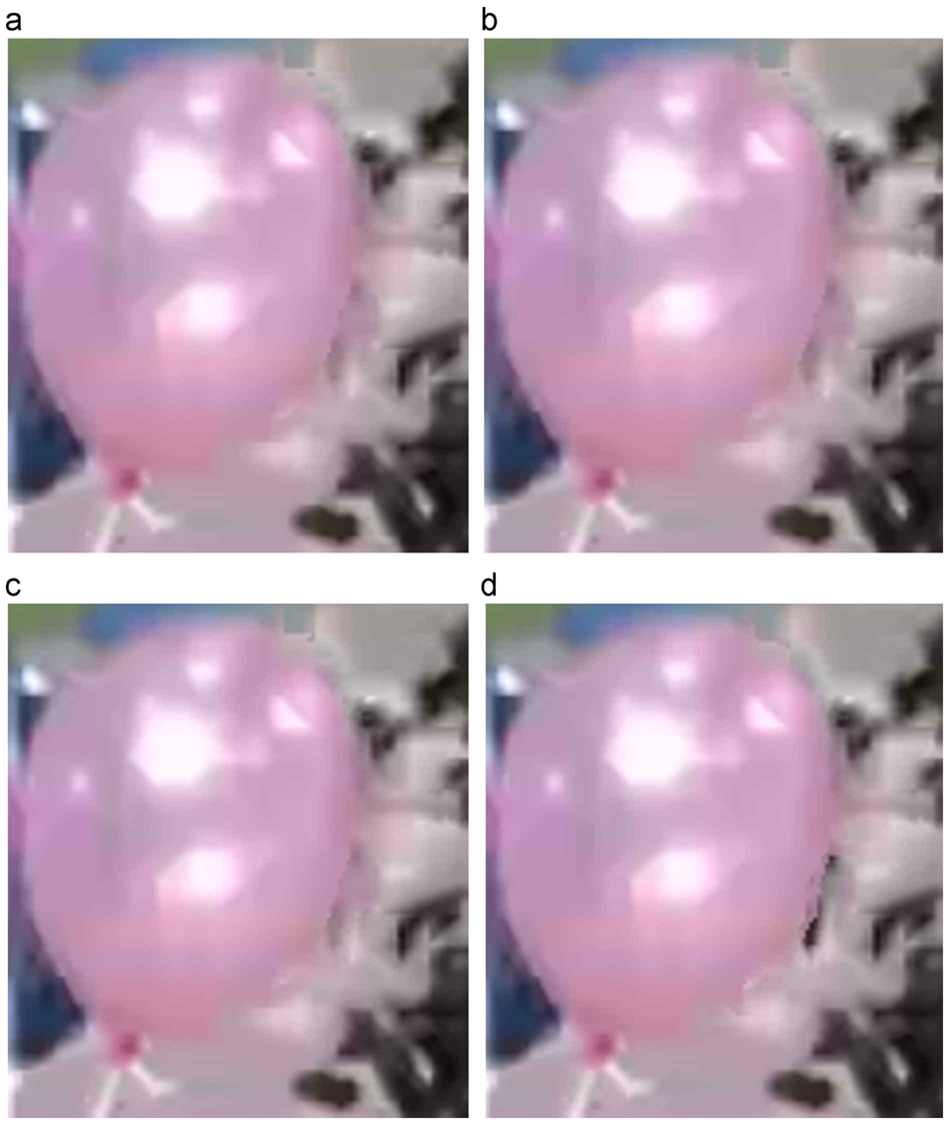

Fig. 13. Synthesized view of 169th frame view 1.5 of *Balloon* based on (a) compressed texture and depth (*QP*=41), (b) compressed texture and depth using AJBF, (c) compressed texture and depth using ASF, (d) compressed texture and depth using ADTF.

Table 1
PSNR improvement of depth image compared with ATM 5.0 anchor.

QP	ΔPSNR (dB)		
	AJBF	ASF	ADTF
Poznan_hall2			
26	0.45	0.51	0.83
31	0.27	0.28	0.42
36	0.08	0.10	0.17
41	0.00	0.00	0.05
Average PSNR	0.20	0.22	0.37
Poznan_street			
26	1.49	1.48	1.54
31	0.76	0.83	0.88
36	0.37	0.42	0.43
41	0.18	0.21	0.17
Average PSNR	0.70	0.74	0.76

Table 1 (*continued*)

QP	ΔPSNR (dB)		
	AJBF	ASF	ADTF
Undo_dancer			
26	3.38	3.48	6.00
31	2.61	2.65	4.09
36	1.57	1.60	2.17
41	0.63	0.66	0.74
Average PSNR	2.05	2.10	3.25
GT_Fly			
26	1.73	1.74	2.31
31	0.96	0.99	1.11
36	0.41	0.46	0.47
41	0.10	0.11	0.12
Average PSNR	0.80	0.83	1.00
Newspaper			
26	2.80	2.83	2.87
31	1.48	1.60	1.83
36	0.70	0.80	0.98
41	0.35	0.38	0.46
Average PSNR	1.33	1.40	1.53
Kendo			
26	0.01	0.02	0.07
31	0.02	0.01	0.04
36	0.04	0.04	0.01
41	0.04	0.04	0.00
Average PSNR	0.03	0.03	0.03
Balloons			
26	1.52	1.84	1.88
31	0.80	0.86	1.06
36	0.32	0.48	0.52
41	0.17	0.20	0.25
Average PSNR	0.70	0.84	0.93

Table 2
PSNR improvement of synthesized views compared with ATM 5.0 anchor.

QP	ΔPSNR (dB)		
	AJBF	ASF	ADTF
Poznan_hall2			
26	0.00	0.04	0.14
31	0.00	0.02	0.07
36	0.00	0.01	0.03
41	0.00	0.00	0.01
BD-Bitrate (%)	−0.05	−0.51	−1.82
Poznan_street			
26	0.01	0.01	0.02
31	0.00	0.01	0.00
36	0.00	0.00	0.01
41	−0.01	0.00	−0.01
BD-Bitrate (%)	+0.07	−0.08	−0.10
Undo_dancer			
26	0.02	0.08	0.34
31	0.01	0.05	0.12
36	0.00	0.01	0.03
41	0.00	0.00	0.01
BD-Bitrate (%)	−0.19	−1.05	−3.06
GT_Fly			
26	−0.01	0.01	0.09
31	−0.01	0.00	0.06
36	−0.01	0.00	0.02
41	−0.01	−0.01	0.01
BD-Bitrate (%)	+0.27	−0.04	−1.23
Newspaper			
26	−0.02	0.03	0.22
31	−0.01	0.01	0.10
36	0.00	−0.01	0.02

Table 2 (*continued*)

QP	ΔPSNR (dB)		
	AJBF	ASF	ADTF
41	0.00	−0.01	−0.01
BD-Bitrate (%)	+0.23	−0.06	−1.98
Kendo			
26	0.01	0.04	0.14
31	0.01	0.00	0.16
36	−0.03	−0.02	0.00
41	−0.02	−0.01	−0.02
BD-Bitrate (%)	+0.01	+0.02	−1.74
Balloons			
26	0.02	0.07	0.24
31	0.00	0.02	0.07
36	−0.01	−0.01	0.02
41	−0.01	−0.01	0.00
BD-Bitrate (%)	+0.16	−0.33	−1.45

Table 3
Summary of ADTF compared with ATM 5.0 anchor.

Video sequence	Depth image	Synthesized views	
	ΔPSNR (dB)	ΔPSNR (dB)	BD-Bitrate (%)
Poznan_hall2	0.37	0.06	−1.82
Poznan_street	0.76	0.01	−0.10
Undo_dancer	3.25	0.13	−3.06
GT_FLY	1.00	0.05	−1.23
Newspaper	1.53	0.08	−1.98
Kendo	0.03	0.07	−1.74
Balloons	0.93	0.08	−1.45
Average	1.00	0.07	−1.75

Table 4
Decoding complexity of ADTF compared with ATM 5.0 anchor.

Video sequence	Time ration (%)
Poznan_hall2	104.43
Poznan_street	103.20
Undo_dancer	110.00
GT_FLY	113.91
Newspaper	103.29
Kendo	103.59
Balloons	110.36
Average	106.97

texture image has large distortion after quantization using large *QP* value, which makes the synthesized views resulted with much noise. Although the boundary artifacts are removed in this situation, the PSNR improvement is not obvious. However, the synthesized virtual views are perceived better due to the removal of boundary artifacts. In addition, the proposed ADTF has a low complexity as tabulated in Table 4. There is only around 7% increase in runtime when comparing the decoding time of the 3DV-ATM with the decoding time of 3DV-ATM with ADTF. However, the ADTF can achieve significant improvement for the compressed depth image and synthesized view.

4. Conclusion

As transform coding based H.264/AVC and H.265/HEVC were adopted in the recent 3DV coding standards development using MVD format, sharp depth edges of the decoded depth images sometimes are blurred or even with ringing artifacts. Annoying boundary artifacts are caused in the synthesized views. To recover these sharp depth edges from the decoded depth images, a new adaptive block truncation filter with low complexity is proposed. It uses a very effective depth edge block repositioning and expansion method to achieve higher quality depth values refinement. It is because the adaptive block positioning and expansion can increase the sharp depth edges recoverability since it contains sufficient information of each depth layer inside the block. Experimental results show sharp depth edges can be recovered and the boundary artifacts in the synthesized views can also be removed. In addition, the objective evaluation also has improvement in terms of PSNR performance.

Acknowledgements

This work was supported by the City University of Hong Kong, Hong Kong, under a SRG grant with project 7004058.

References

[1] K. Müller, P. Merkle, T. Wiegand, 3-D video representation using depth maps, Proc. IEEE 99 (4) (2011) 643–656.
[2] M. Tanimoto, M.P. Tehrani, T. Fujii, T. Yendo, Free-viewpoint TV, IEEE Signal Process Mag. 28 (1) (2011) 67–76.
[3] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, B R. Tanger, Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability, Signal Process. Image Commun. Special Issue on 3DTV 22 (2) (2007) 217–234.
[4] ISO/IEC JTC1/SC29/WG11, Tentative Test Model for AVC-based 3D Video Coding, Doc. M22842, Nov. 2011.
[5] ISO/IEC JTC1/SC 29/WG 11, Test Model Under Consideration for HEVC based 3D Video Coding, Doc. M12350, Nov. 2011.

[6] D.V.S.X. De Silva, W.A.C. Fernando, H. Kodikaraarachchi, S.T. Worall, A.M. Kondoz, Improved depth map filtering for 3D-TV systems, Proc. IEEE Int. Conf. Consum. Electron. (ICCE) (2011) 645–646.

[7] K.J. Oh, S.Y. Yea, A. Vetro, Y.S. Ho, Depth reconstruction filter and down/up sampling for depth coding in 3-D video, IEEE Signal Process Lett. 16 (9) (2009) 747–750.

[8] D.V.S.X. De Silva, W.A.C. Fernando, H. Kodikaraarachchi, S.T. Worall, A.M. Kondoz, Adaptive sharpening of depth maps for 3D-TV, Electron. Lett. 46 (23) (2010) 1546–1548.

[9] D.V.S.X. De Silva, W.A.C. Fernando, H. Kodikaraarachchi, S.T. Worall, A.M. Kodoz, A depth map post-processing framework for 3D-TV systems based on compression artifact analysis, IEEE J. Sel. Top. Sign. Process. (2011). (accepted for publication, pp (99), pp. 1,1,0).

[10] E. Ekmekcioglu, M. Mrak, S. Worrall, A. Kondoz, Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering, Proc. IEEE Int. Conf. Image Process. (ICIP) (2009) 733–736.

[11] K.J. Oh, A. Vetro, Y.S. Ho, Depth coding using a boundary reconstruction filter for 3-D video systems, IEEE Trans. Circuits Syst. Video Technol. 21 (3) (2011) 350–359.

[12] ISO/IEC JTC1/SC29/WG11, Common Test Conditions for 3DV Experimentation, Doc. N12745, Geneva, Switzerland, May. 2012.

[13] ISO/IEC JTC1/SC29/WG11, 3D-CE8. A Results on View Synthesis Optimization Using Distortion in Synthesized Views by Samsung, Doc. M24826, Geneva, Switzerland, May. 2012.