Codes of the paper can be downloaded from
http://nma.web.nitech.ac.jp/fukushima/research/depthmap_postfilter.html

REMOVING DEPTH MAP CODING DISTORTION BY USING POST FILTER SET

Norishige Fukushima, Tomohiko Inoue, Yutaka Ishibashi

Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
{fukushima, ishibasi}@nitech.ac.jp, inoue_t@mcl.nitech.ac.jp

ABSTRACT

We propose a fast post filter set for removing coding distortions of depth maps. Lossy coding generates various distortions, such as mosquito and block noises, edge blurs, and over quantization. These distortions seriously deteriorate image quality of synthesized views in free viewpoint image rendering. Thus, we propose the post filter set which includes median filter, Gaussian filter, min-max blur remove filer, and binary weighted range filter to remove these noises. In experiments, we use various codecs, which are JPEG, JPEG-LS, JPEG2000, and H.264/AVC, for depth map coding, and synthesize views with the coded depth maps. Experimental results show that our post filter set can improves every codecs performance. Improvement of PSNR is larger than the conventional post filter, and especially JPEG is large. The computational time of the filter set, which is implemented by C++ with SIMD optimization, is within 5.2 ms at high bit rate cases, and within 15.3 ms at low bit rate cases.

Index Terms— Depth map coding, Post filter, Image based rendering, Edge preserving filter.

1. INTRODUCTION

Free viewpoint image rendering can synthesize views on an arbitrary viewpoint by users' input [1]. The rendering requires multi-view images and its depth information. These source data becomes huge, thus effective compression methods for the novel 3DTV applications are proposed [2].

Multi-view video coding (MVC) [3] can encode multi-view videos effectively by using not only temporal redundancy but also spatial one. Using the MVC for free viewpoint rendering, depth maps associated with multi-view videos are estimated after decoding at a decoder side. The process flow requires huge computational cost for users to decode videos.

Another approach is depth image based rendering [4] and it's coding [5]. In the approach, we obtain views and depth maps at first, and then encode them. The coded source are decoded at a decoder side, thus, we do not need depth estimation at the decoder side. In this approach, the quality of synthesized views depends on the accuracy of depth information as much as the quality of multi-view video. Figure 1 (left side) shows coding distortion of the depth map and its effect for view synthesis. Depth values in the depth map are scattered and blurred at the object boundary, and over quantized at the flat region. Then, the shape of the synthesized view from the depth map is distorted (see a chair's edge). In addition, when we change a viewpoint of the synthesized image, the view has card-board like movement.

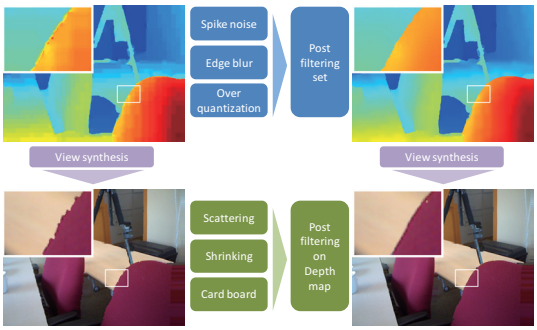


Fig. 1. Coded depth maps have distortions (coded by H.264/AVC at qp = 39, 0.021 bpp). Our post filter set recovers these distortions, and improves quality of synthesized views from the depth map.

There are techniques which optimize the trade-off between bits for depth maps and views for view synthesis to minimize distortions, such as [6]. Such optimization is effective, but directly minimizing depth map distortion itself is also important for view synthesis.

Thus, this paper proposes a post filter set to recover the distorted depth map at a decoder side. The post filter set contains four filters which remove spike noises and blurs at object boundaries, and recover over-quantized regions in slanted surface. After the filtering, depth map distortions are removed, and then image quality of the synthesized view is improved (Fig. 1 right). In addition, these filters are designed to save decoding computational cost.

Organization of this paper is as follows. In Sec. 2, we summarize related works of depth map coding and post filters. The post filter set is proposed in Sec. 3. Section 4 demonstrates the coding performance of the proposed method. Finally, we conclude this paper in Sec. 5.

This work was partly supported by JSPS KAKENHI 22700174, SCOPE 122106001, and Foundation of Public Interest Tatematsu Foundation.

2. RELATED WORKS

To keep depth shapes in depth map coding, there are three types of coding; the first is lossless or near lossless coding, the second is object based coding, and the third is conventional coding with post filtering. Our method is the third type.

Examples of lossless or near lossless coding are PNG, JPEG-LS, and JPEG2000 (near lossless mode). These codecs keep signal distortions within allowable ranges $(0, \pm 1, \pm 2, ...)$. These codecs can keep edge of depth maps, but coding efficiency is not high.

Object based coding is useful for keeping shapes of object boundaries. A codec [7] fits object shapes by quad trees plus line-segments, thus the depth maps do not have mosquito noises. However the codec always has approximation errors in fitting. Palettes with binary shape coding [8] can compress depth maps well without distortions of object boundaries. The codec splits sub regions of depth maps into foreground and background regions, and encodes them as shape maps. The foreground and background depth values in the palettes of the sub region are assigned for each region. The codec has the higher coding performance than the H.264/AVC, but quantization of the depth value is inevasible.

Considering backward compatibility, usual codecs with post filters is a convenient solution. Some filters are used for in/out loop filters in depth map coding. Bilateral filter [9] for the coded depth map can reduce noises, while keeps object edges. However, the filter cannot recover object boundaries which are highly noised. Depth boundary reconstruction filter [10] can recover the areas using neighborhood depth frequency, space and depth value nearness information. The filter can remove mosquito noises on object boundaries well, while a slight blurs are remained. The position of depth edges in the coded depth map are damaged, and the bilateral filter and depth boundary reconstruction filter cannot correct the edges' position, because we do not know the correct position.

Joint image and depth filters, which use correlation between RGB images and depth maps, are also proposed [11, 12, 13] to correct edge position. Trilateral filter [11], which is a variant of bilateral filter, uses adaptive kernels. The kernels are defined by nearness of depth values between a focusing pixel and supporting pixels, and are also defined nearness of intensities/color values of the associated RGB view. The filter can recover shapes of edges which are broken in coding process. A remaining issue of the method is a little blur on the edges of the depth map. Weighted mode filter [12] adds frequency of depth values and color values information to the trilateral filter's kernel computation. The filter do not blend pixel values, thus the filter can suppress blurring around object boundaries. The filter requires local histogram construction, thus its computational cost is higher than the trilateral filter. Weighted joint bilateral filter and blur remove filter [13] can reconstruct depth shape without blur and its computational cost is almost same as the trilateral filter.

The joint filtering is possible, only when the RGB image can be captured on the depth maps. Basic stereo matching from stereo images can generate such data. Recent depth sensors, such as Kinect, do not have same posited depth and color sensors. Detail of Kinect is reviewed by [15].

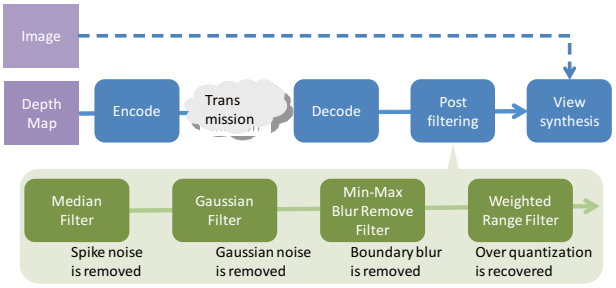


Fig. 2. Flow of depth map transmission and post filter set.

Needless to say, we can arbitrarily move the depth map's viewpoint by 3D warping, but the warped depth map loses information on occluding regions. In addition, there are depth only depth sensors which do not have RGB sensors, such as ASUS Xtion, MESA Swiss Ranger SR4000. In addition, effect of the joint filters for accurate depth maps, which are captured by Kinect or semi-automatic depth estimation method [14], is lower than the inaccurate case. Therefore we focus on non-joint image type filtering for universal usages. Comparison with the joint filters is our future work.

Comparing with boundary reconstruction filter, which is a representative filter of the non-joint type filter, advantages of our filter has; (1) blur removing ability, (2) quantization recovering ability, and (3) low computational cost.

3. PROPOSED METHOD

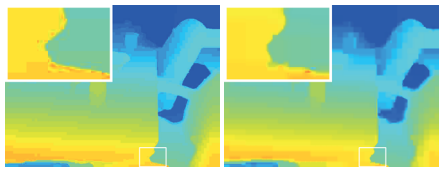
We encode depth maps by various encoders, which are JPEG, JPEG-LS, JPEG2000, and H.264/AVC. The decoded depth maps are filtered by proposing four post filters including median filter, Gaussian filter, min-max blur remove filter and weighted range filter (see Fig. 2). The first filter removes spike noises, the second filter removes blurs around object boundaries, and the third filter recovers over-quantized areas. These filters are performed introduced order.

In this section, we review distortion types of depth map coding by these codecs in Sec. 3.1, and then we propose the post filter set in Sec. 3.2.

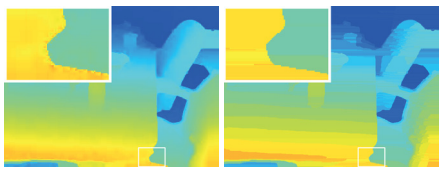
3.1. Coding Distortion

Figure 3 shows distortions of a depth map which is coded by JPEG, JPEG2000, JPEG-LS and H.264/AVC. We categorize types of depth map distortions as three factors which are spike noise and edge blur from mosquito noise and over-quantization. Strength and frequency of these noises depend on codec types.

JPEG and H.264/AVC intra-coding (De-blocking filter is turned off) are block based transform coding (8×8 discrete cosine transform (DCT) for JPEG and 4×4 integer DCT for H.264/AVC). These codecs cause block bounded mosquito



(a) JPEG (0.070 bpp) (b) H.264/AVC (0.079 bpp)



(c) JPEG2000 (0.017 bpp) (d) JPEG-LS (0.149 bpp)

Fig. 3. Coding distortion of various codecs.

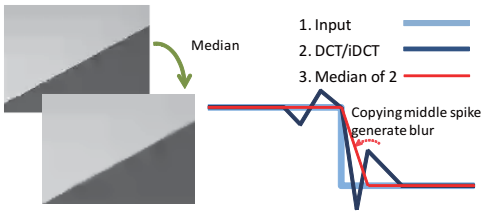


Fig. 4. Median filter for distorted depth map.

noises and quantization. In Fig. 3 upper part, coded depth maps by JPEG and H.264/AVC are shown. Focusing on the zoom pictures of a diagonal edge, block bounded mosquito noises are observed. Comparing the two codecs, H.264/AVC has the smaller area of mosquito noises, because macro-block size is smaller than the JPEG. Switching the focusing point to the outside of the zoomed area, we can observe that slant surface parts are quantized and as see block noise.

In Fig. 3 of left under part, coded depth map by JPEG2000 is shown. There is no block noise, because JPEG2000 is non-block based coding, but mosquito noises still exist. JPEG2000 uses transform coding of discrete wavelet transform, and quantization in wavelet domain also causes mosquito noise. In addition, these noises are not bounded by blocks, thus size of the area will be large. In addition, the coded images tend to be blurred and then lose sharp edges.

In Fig. 3 of right under part, coded depth map by JPEG-LS (near lossless mode) is shown. JPEG-LS is a lossless or near lossless codec. If we use lossless mode, the coded depth map is not distorted, but the coding efficiency is low. In the near lossless mode, the codec quantizes pixel prediction error. This fact generates over-quantization on slant surface when quantization factor is high. The codec does not use transform coding. Thus the coded depth map of the codec does not have mosquito noises, and does not distort edge shapes.

3.2. Post Filter Set

3.2.1. Median Filter

At first, we apply median filter to distorted depth maps. In object boundaries, mosquito noises are occupied by transform

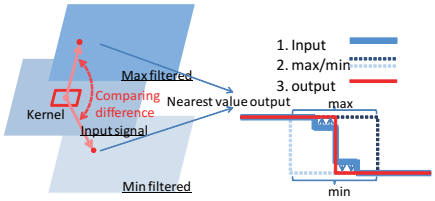


Fig. 5. Min-Max blur remove filter for boundary areas.

based coding, and noised parts will have spike noises. Figure 4 shows an example of a coded depth map and a median filtered one. We can remove the spikes by the filter. Usually, median filter does not blur edges, but the filter for coded depth maps by transform based coding makes blurs around the edge. Because the coded depth map contains mixed pixels (Fig. 4). The median filter is only performed around object boundaries which have high contrasted edges. The areas are computed after median filter, thus the other areas are copied from non-filtered values.

We use a fast median filtering [16], which is independent of kernel size. The implementation is in OpenCV, and is coded by C++ with SIMD (SSE4.1) optimizations.

3.2.2. Gaussian Filter

Secondly, we use Gaussian filter. To remove Gaussian noise, Gaussian filter is mode suitable than median filter. But Gaussian filter explicitly makes blurs around object boundaries. Therefore, we set kernel size of Gaussian small. The next blur remove filter can remove this type of blur, too.

3.2.3. Min-Max Blur Remove Filter

Thirdly, we perform a novel deblurring filter. The filter enhances edges, and is similar to shock filter [17]. The filter is defined as follows;

$$D_{out}(p) = \underset{d \in \{d_{\max}, d_{\min}\}}{\arg\min} \left(|D(p) - \max_{q \in N} D(q)|, |D(p) - \min_{q \in N} D(q)| \right), \quad (1)$$

where D, D_{out} are input/output depth values on pixel p, q, $\min$ / $\max$ are max and min filters (kernel size is N) and these outputs are $d_{\max}, d_{\min}$. The operator $\arg\min(a, b)$ gets an argument minimum value in a or b. The filter assigns a value to the max/min value in the kernel. The min-max replacing makes steep edge, and it removes blurred pixels.

The deblurring filter is applied to only near the object boundaries, because this filter strongly enhances edges including, slight slant surface. The blurred regions in transformed coding with median filtering are not so large, thus we can apply the filter only the object boundaries.

The proposed blur remove filter is almost independent of filter kernel size. We use streaming maximum-minimum filter [18], which simultaneously computes max and min filtered values with low cost, for fast implementation. The filter weakly depends on kernel size and is low latency processing. In our blur remove filter, we add just a process of comparing max and min values per pixel for maximum-minimum filter. Therefore our filter is fast.

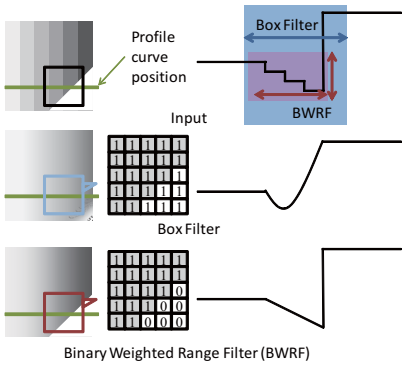


Fig. 6. Slant surface reconstruction by box and binary weighted range filter.

3.2.4. Binary Weighted Range Filter

The last filter is weighted range filter for quantization recovery or super-quantization. The simplest approach of the recovering quantized values is just box filtering. The blur filter can convert stair steps to slant plane. Figure 6 upper side shows a recovering result of stair steps, but the box filter also blurs sharp edges. Thus we use an edge preserving filter of the binary weighted range filter. The filter is a simplified bilateral filter [9], which omits domain kernel of the bilateral filter and limits range kernel values of one to be binary. The filter is defined as follows;

$$out(p) = 1/K \sum_{q \in N} bw_p(q) D(q), \tag{2}$$

$$bw_p(q) = \begin{cases} 1 & (|(D(p) - D(q)| \leq Th_D) \\ 0 & (otherwise), \end{cases} \tag{3}$$

where normalizing factor is $K = \sum_{q \in N} bw_p(q)$ and $bw_p(q)$ is a binary (0 or 1) weight at a pixel q around a pixel p. The weight is computed for every pixels p, and the weights are assigned one, if difference between a value at a pixel p and q is within a threshold Th_D. The others are zero.

We apply the horizontal and vertical filter separately, such as approximated bilateral filter [19]. If the kernel size is $K_W \times K_H$, the computational order of usual implementation is $O(K_W K_H)$. In the case of the separable filter, the order is reduced to $O(K_W + K_H)$. The filter is not separable, but the separated filter is well approximated. When kernel size is large, we compute the weights by cross skeleton based filter [20], which can ignore enclave areas in the filtering kernel, for robust computation.

4. EXPERIMENTAL RESULTS

4.1. Experimental Environment

In our experiment, we use four codecs, which are JPEG, JPEG2000, JPEG-LS and H.264/AVC intra coding (the implementations of the each codecs are libjpeg-turbo with arithmetic coding instead of Huffman encoding, JasPer, CharLS (ver. 1.0), x264 main profile, respectively. Depth maps videos are coded by these codecs, and then the proposed post filter set is applied. We use Kinect's depth map and image data sets [21]. The data have single view's depth and RGB video data. In view synthesis evaluation, we do not have ground truth signal on the synthesized view, thus we compare synthesized view from non-compressed depth maps with one from compressed depth maps by using Peak Signal to Noise Ratio (PSNR).

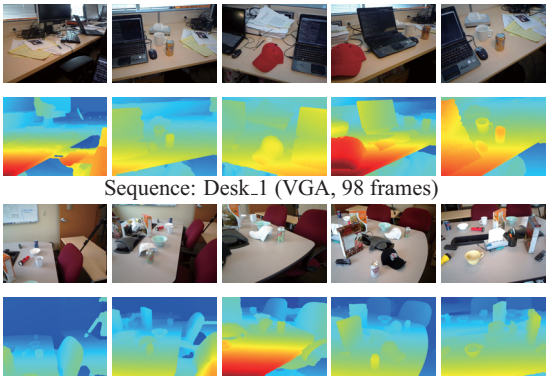


Fig. 7. Example of RGB image and depth map [21].

We synthesize views on 2 cm and 8 cm away from Kinect's RGB image sensor. The view synthesis process is depth image based rendering [4] of single view version. In the experiment, RGB images are not compressed to focus on evaluation of depth map distortions.

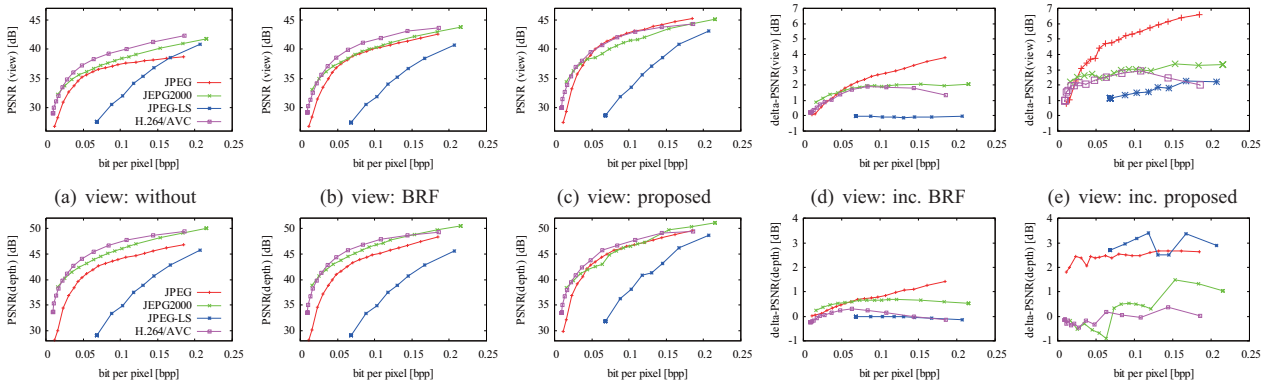
Kinect's depth maps have usually holes in the depth map, thus we fill the holes by the lowest (farthest) value around pixels on the hole iteratively. Examples of RGB images and depth maps are shown in Fig. 7.

4.2. Coding Results

In this experiment, we compare coding performance of various codecs and compare our post filter set with the conventional post filter of boundary reconstruction filter (BRF) [10].

Figure 8(a)-(e) shows rate-distortion curves of *synthesized views* from distorted depth maps by various codecs with/without post filters in Desk_1 sequence. Table 1 shows selected points of coding results of another sequence (Meeting_small) with difference viewpoint interval. In depth map coding without post filter set, H.264/AVC is the best codec, and then the 2nd best codec is JPEG2000. The 3rd one is JPEG, and the last one is JPEG-LS. After our post filter set or the competitive filter, synthesized image quality is improved. Comparing our post filter set with the boundary reconstruction filter, our method has higher PSNR than the conventional method in all codecs and over all bit rates. Focusing on each codec, improvement of image quality is different (Fig. 8(d),(e)). JPEG is the most improving codecs. This is because the codec strongly quantizes input signals.

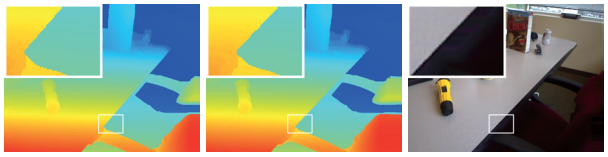
Figure 8(f)-(j) shows rate-distortion curves of *depth maps*. Our post filter cannot improve the PSNR of the depth map coded by H.264/AVC and JPEG2000. In low bit case, performance of both codecs is decreasing. However, improving



(a) view: without (b) view: BRF (c) view: proposed (d) view: inc. BRF (e) view: inc. proposed

(f) depth: without (g) depth: BRF (h) depth: proposed (i) depth: inc. BRF (j) depth: inc. proposed

Fig. 8. Synthesized view's rate-distortion of various codecs with/without post filters (PFs)((a)-(e)) and depth map's one ((f)-(j)) in Desk_1; (a,f) without PF, (b,g) with PF [10], (c,h) with our PF, (d,i) Δ PSNR (b-a) and (g-f), (e,j) Δ PSNR (c-a) and (h-f).



(a) input depth map (b) warped depth map (c) warped view

Fig. 9. Input depth map and view, and the synthesized result of them on virtual viewpoint using RAW depth map.

PSNR of synthesized views is the objective of this paper.

Figure 9 shows input image and depth map and the view synthesis result of the warped image and depth map. Figure 10 shows the coded/filtered depth map and the synthesized result of the warped image and depth map. In this experiment, we compare warped image in Fig. 9 with warped images in Fig. 10 of various codecs with post filters, and compare depth map in Fig. 9 with non-warped depth maps in Fig. 10. In Fig. 10, the depth map distortions on the edges degrade quality of the synthesized images. The conventional filter cannot recover edge and slant surface, but our post filter set can remove visible noises well.

5. CONCLUSION

In this paper, we proposed a post filter set for removing distortions of coded depth maps, and evaluated coding efficiency of various codecs, which are JPEG, JPEG2000, JPEG-LS and H.264/AVC intra coding. The proposed filter set had four filters which ware median filter, Gaussian filter, min-max blur remove filter and binary weighted range filter. The filter set could remove spike and Gaussian noises, blurs around object boundaries and recover over-quantization regions in any coded depth maps.

Experimental results showed that the filter set improved every codec's performance in our experiments. Especially, improvement of JPEG codec is notable at high bit rate cases.

The most efficient codecs, in low bit case, were H.264/AVC and JPEG2000 with our post filter set. In high bit case, the performance of JPEG codec reaches one of H.264/AVC, and then H.264/AVC and JPEG with our post filter set were the best codecs.

Table 1. Coding result of various codecs with/without post filters (Meeting_small sequence).

Codec (bpp)	cm	without	BRF (Δ) [10]	proposed (Δ)
JPEG (0.022)	2	30.53	31.19 (0.66)	33.72 (3.19)
JPEG (0.186)	2	38.37	42.26 (3.89)	44.67 (6.29)
JPEG (0.022)	8	23.40	23.65 (0.25)	25.57 (2.17)
JPEG (0.186)	8	31.06	33.72 (2.67)	35.12 (4.06)
JPEG-LS (0.077)	2	29.98	29.93 (-0.05)	32.20 (2.22)
JPEG-LS (0.200)	2	40.49	40.34 (-0.15)	42.51 (1.02)
JPEG-LS (0.077)	8	22.63	22.61 (-0.02)	24.35 (1.73)
JPEG-LS (0.200)	8	29.56	29.77 (0.20)	32.09 (2.53)
JPEG2000 (0.025)	2	34.08	35.16 (1.09)	36.17 (2.10)
JPEG2000 (0.184)	2	41.13	42.98 (1.85)	44.06 (2.93)
JPEG2000 (0.025)	8	25.41	27.11 (1.70)	28.55 (3.14)
JPEG2000 (0.184)	8	29.06	31.88 (2.83)	32.38 (3.32)
H.264/AVC (0.021)	2	33.86	34.43 (0.58)	35.54 (1.68)
H.264/AVC (0.185)	2	41.97	43.02 (1.05)	43.67 (1.70)
H.264/AVC (0.021)	8	26.16	27.58 (0.42)	27.30 (1.14)
H.264/AVC (0.185)	8	32.09	32.82 (0.73)	33.29 (1.20)

As a future work, we use other depth sequence which is not captured by Kinect to evaluate dependency of our filter set for input depth type. In addition, we will use 8×8 DCT in H.264/AVC for effective coding.

6. REFERENCES

[1] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, "FTV for 3-d spatial communication," *Proc. the IEEE*, vol. 100, no. 4, pp. 905 –917, Apr. 2012.

[2] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV – a survey," *IEEE Trans. CSVT*, vol. 17, no. 11, pp. 1606 –1621, Nov. 2007.

[3] A. Vetro, T. Wiegand, and G.J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. of the IEEE*, vol. 99, no. 4, pp. 626 –642, Apr. 2011.

[4] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for ftv," *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 65 –72, Jan. 2009.

[5] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. ICIP*, Sep.-Oct. 2007, pp. 201-204.

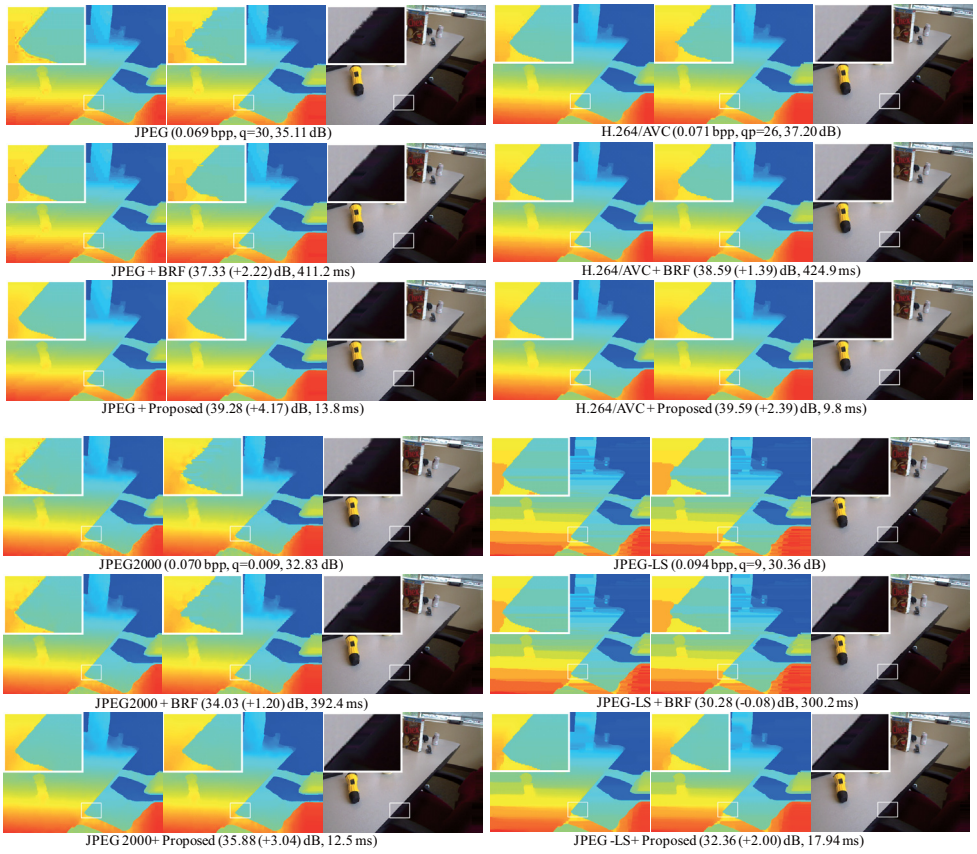


Fig. 10. Result of coded/filtered depth map, virtual viewpoint depth map, and virtual viewpoint image. The first column and fourth column show coded/filtered depth maps. The second column and fifth column show virtual viewpoint depth maps. The third column and sixth column show virtual viewpoint depth maps.

[6] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Proc. SPIE, Visual Infor. Proces. and Comm.*.

[7] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Proc. SPIE SD & A*, vol. 6055, Jan. 2006.

[8] S. Shimizu, H. Kimata, S. Sugimoto, and N. Matsuura, "Block-adaptive palette-based prediction for depth map coding," in *Proc. ICIP*, pp. 117 –120, Sep. 2011.

[9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, pp. 839 –846, Jan. 1998.

[10] K.-J. Oh, A. Vetro, and Y.-S. Ho, "Depth coding using a boundary reconstruction filter for 3-d video systems," *IEEE Trans. CSVT*, vol. 21, no. 3, pp. 350 –359, Mar. 2011.

[11] S. Liu, P. Lai, D. Tian, and C.W. Chen, "New depth coding techniques with utilization of corresponding video," *IEEE Trans. Broadcasting,*, vol. 57, no. 2, pp. 551–561, June 2011.

[12] D. Min, J. Lu, and M.N. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Transactions on IP*, vol. 21, no. 3, pp. 1176 –1190, Mar. 2012.

[13] T. Matsuo, N. Fukushima, Y. Ishibashi, "Weighted joint bilateral filter with slope depth compensation filter for depth map refinement," in *Proc. International Conference on Computer Vision Theory and Applications*, Feb. 2013.

[14] M. O. Wildeboer, N. Fukushima, T. Yendo ; M. P. Tehrani, T. Fujii, M. Tanimoto, "A semi-automatic multi-view depth estimation method," in *Proc. SPIE Visal Communications and Image Processing*, July 2010.

[15] K. Khoshelham and S.O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[16] S. Perreault and P. Hebert, "Median filtering in constant time," *IEEE Trans. IP*, vol. 16, no. 9, pp. 2389 –2394, Sep. 2007.

[17] S.J. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numerical Analysis*, vol. 27, pp. 919 –940, Aug. 1990.

[18] D. Lemire, "Streaming maximum-minimum filter using no more than three comparisons per element," *Nordic J. of Computing*, vol. 13, no. 4, pp. 328 –339, Dec. 2006.

[19] T. Q. Pham and L. J. Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proc. ICME*, pp. 1 –4, July 2005.

[20] Ke Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. CSVT*, vol. 19, no. 7, pp. 1073 –1079, July 2009.

[21] K. Lai, L .Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset.," in *Proc. ICRA*, pp. 1817 – 1824, May 2011.