Depth Reconstruction Filter and Down/Up Sampling for Depth Coding in 3-D Video

Kwan-Jung Oh, Sehoon Yea, *Member, IEEE*, Anthony Vetro, *Senior Member, IEEE*, and Yo-Sung Ho, *Senior Member, IEEE*

***Abstract*—A depth image represents three-dimensional (3-D) scene information and is commonly used for depth image-based rendering (DIBR) to support 3-D video and free-viewpoint video applications. The virtual view is generally rendered by the DIBR technique and its quality depends highly on the quality of depth image. Thus, efficient depth coding is crucial to realize the 3-D video system. In this letter, we propose a depth reconstruction filter and depth down/up sampling techniques to improve depth coding performance. Experimental results demonstrate that the proposed methods reduce the bit-rate for depth coding and achieve better rendering quality.**

***Index Terms*—3-D video, depth down/up sampling, depth reconstruction filter, free viewpoint television (FTV).**

I. Introduction

A depth image represents a relative distance from a camera to an object in the three-dimensional (3-D) space, and is widely used in computer vision and computer graphics fields to represent 3-D scene information. Most image-based rendering (IBR) methods [1] utilize the depth image in conjunction with stereo or multiview video to realize 3-D and free-viewpoint video applications [2]–[7]. Efficient coding of both the depth image and the multiview video and rendering techniques are key issues for such applications.

The recently finalized multiview video coding (MVC) [8] standard supports inter-view prediction, but it does not include any particular provisions for depth coding. Recently depth related issues, such as depth estimation, rendering, and depth coding, are investigated in the context of the end-to-end system that includes 3-D data acquisition, transmission, and display in the MPEG 3DV group.

The main objective of depth coding is compression of the depth image to guarantee the decoded depth can synthesize the high-quality virtual view. That is, in texture coding, we try to improve coding performance of video data; however, in depth coding, we focus more on better rendering quality rather than depth quality. In this letter, we propose a depth reconstruction filter and depth down/up sampling methods. The proposed algorithms focus on coding of object boundaries that are sensitive to both depth coding and rendering.

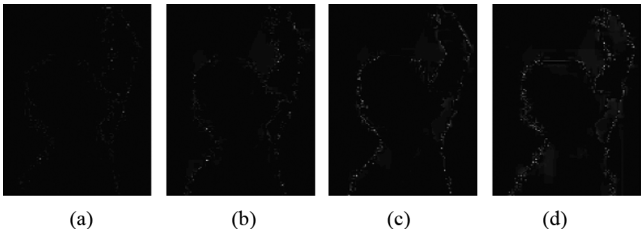

Fig. 1. Distribution of depth coding errors for various bit rates: (a) QP22, (b) QP27, (c) QP32, (d) QP37.

The proposed depth reconstruction filter is designed as an in-loop filter to recover object boundaries. It consists of a newly designed frequent-low-high filter and a bilateral filter. The proposed down/up sampling employs nonlinear scaling. The up-scaled image is post-filtered by combination of the median filter and the proposed depth reconstruction filter. Efficiency of the proposed algorithms is evaluated by the depth coding rate and rendering quality.

II. Proposed Depth Coding Methods

A. Depth Reconstruction Filter

Unlike natural images, the depth image has a well-defined object boundary and it is very important to maintain the object boundary for high-quality rendering. In general, the object boundary is weak to coding errors, since its intensity value is rapidly changed. The coding errors increase with lower bit-rates as shown in Fig. 1.

In this letter, we propose the depth reconstruction filter to compensate for depth coding errors by recovering distorted object boundaries. The proposed depth reconstruction filter consists of the frequent-low-high filter mainly to recover object boundaries and the bilateral filter to refine object boundaries by eliminating remaining errors. The frequent-low-high filter is a nonlinear filter and is designed as follows:

After we defined a $W(i,j)$ of $m \times n$ pixels, m and n are odd numbers, we sort all pixels in W by its intensity value, as shown in (1), and the median value for W is defined by (2):

Manuscript received March 02, 2009; revised May 05, 2009. First published June 05, 2009; current version published July 01, 2009. This work was supported in part by the ITRC through RBRC at GIST (IITA-2009-C1090-0902-0017). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitrios Tzovaras.

K.-J. Oh and Y.-S. Ho are with the School of Information and Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea (e-mail: kjoh81@gist.ac.kr; hoyo@gist.ac.kr).

S. Yea and A. Vetro are with the Mitsubishi Electric Research Laboratories (MERL), Cambridge MA 02139 USA (e-mail: yea@merl.com; avetro@merl.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2009.2024112

$$sort(W(i,j)) = \{I_0, I_1, I_2 \cdots I_{m\times n-1}\} \tag{1}$$

$$\text{median}(W) = I_{(m\times n+1)/2}. \tag{2}$$

The pixels in W are categorized into low and high groups by $median(W)$ as seen in (3).

$$W(i,j) \in \begin{cases} S_{low}, & \text{if } W(i,j) < \text{median}(W) \\ S_{high}, & \text{otherwise.} \end{cases} \tag{3}$$

For the pixel set $S_{low}(k)$, we count the number of occurrence (C_v) for each intensity value (v) belonging to S_{low} with (4).

$$C_v = \sum_{k=0}^{(m\times n-1)/2} \delta[v, S_{low}(k)] \quad with$$
$$\delta[a,b] = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Among v values, we choose the intensity value having the maximum number of occurrence and we define it as v_{low} as seen in (5). That is, v_{low} is the pixel intensity value of the most frequently occurring intensity value among the set of pixel values which are smaller than the median value of W.

$$C_{v_{low}} = \max\{C_{v_0}, C_{v_1}, \ldots C_{v_k}\}. \tag{5}$$

In the same manner, v_{high} is also defined. Finally, we decide the representative value for a given pixel by (6)

$$W(x,y) = \begin{cases} v_{low}, & \text{if } |W(x,y) - v_{low}| < |W(x,y) - v_{high}| \\ v_{high}, & \text{otherwise} \end{cases} \tag{6}$$

where $W(x,y)$ is a center point in W. $W(x,y)$ is replaced by the closer of the two representative values, v_{low} and v_{high}, in the proposed frequent-low-high filter.

The proposed frequent-low-high filter has following merits over other linear filters: 1) it is more robust against outliers; a noisy neighboring pixel does not affect the frequent-low-high value significantly and 2) the frequent-low-high does not create new unrealistic pixel values when the filter straddles an edge since the frequent-low-high value must actually be the value of one of the pixels in the neighborhood.

However, since the frequent-low-high filter dose not clean up the whole coding errors, some errors look like Gaussian noise still remain. To eliminate the remaining errors, we apply the bilateral filter [9]. The bilateral filter is nonlinearly designed and is a representative edge-preserving filter. The bilateral filter extends the concept of Gaussian smoothing by weighting the filter coefficient with the intensity value of the corresponding relative pixel. Pixels that are very different in intensity from the central pixel are weighted less even though they may be in close proximity to the central pixel. This is effectively a convolution with a nonlinear Gaussian filter, with weights based on pixel intensities as in (7):

$$BF\,[I]_p = \frac{1}{W_p} \sum_{q\in S} G_{\sigma_s}\left(\|p-q\|\right) G_{\sigma_r}\left(|I_p - I_q|\right) I_q \tag{7}$$

where I_p is a central pixel to be bilateral filtered and I_q is a neighboring pixel of I_p in the window W_p. The filter parameters, space sigma (σ_s) and range sigma (σ_r), which define the spatial extent of the kernel and threshold for which color differences in the image, respectively. The depth reconstruction filter is implemented as an in-loop filter and it is located after the deblocking filter in H.264/AVC compatible codecs.

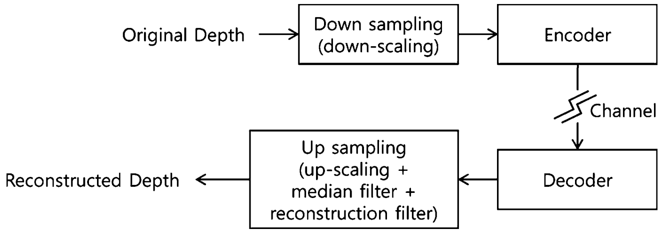
Original Depth
Down sampling (down-scaling)
Encoder
Channel
Decoder
Up sampling (up-scaling + median filter + reconstruction filter)
Reconstructed Depth

Fig. 2. Flow diagram of the depth down/up sampling.

B. Depth Down/Up Sampling

Encoding a reduced resolution depth can reduce the bit-rate substantially, but it also degrades the quality of the depth map, especially in high frequency regions such as the object boundary. The rendered image from the subsampled image has visible rendering artifacts. In general, conventional down/up samplers use a low-pass filter and an interpolation filter to resize the image without quality degradation. However, these traditional techniques are not suitable for depth down/upsampling since they smooth out the object boundaries.

Considering the above, we design a new down/up sampler for depth. In down-sampling a 2-D image, a representative value among the values in a certain window must be selected; we choose a median value as in (8):

$$img_{down}(x,y) = median(W_{s\times s}) \tag{8}$$

where $W_{s\times s}$ represents a $s \times s$ block and s is a scaling factor for down-sampling. That is, the down-sampling reduces the image size by selecting the median value for each $W_{s\times s}$.

The up-sampling process consists of following three steps: 1) image up-scaling, 2) 2-D median filtering, 3) proposed depth reconstruction filtering. The up-scaling is an inverse process of the image down-sampling as depicted in (9). We use the same scaling factor used in the down-sampling process.

$$img_{up}(x,y) = img_{down}\left(\left\lfloor \frac{x}{s} \right\rfloor, \left\lfloor \frac{y}{s} \right\rfloor\right). \tag{9}$$

After the up-scaling, we apply the 2-D median filter to smooth the blocking artifact caused by image down-sampling. Finally, the proposed depth reconstruction filter is applied to reconstruct the object boundary smoothed by median filtering. Fig. 2 shows a flow diagram of the depth down/up sampling processes. As you can see, down/up sampling is a pre/post-processing.

C. Evaluation of Depth Coding

Coding efficiency of the texture image is measured by the encoding bit-rate and a peak signal-to-noise ratio (PSNR) value between the original and the reconstructed images as in (10):

TABLE I
EXPERIMENTAL RESULTS FOR RECONSTRUCTION FILTER (BALLET)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	943.50	922.68	32.15	32.23
25	705.07	692.70	32.07	32.18
28	526.68	521.25	31.95	32.08
31	391.52	395.90	31.80	31.93

TABLE II
EXPERIMENTAL RESULTS FOR RECONSTRUCTION FILTER (BREAKDANCERS)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	1174.53	1158.18	31.78	31.88
25	863.78	859.05	31.73	31.86
28	616.88	621.78	31.65	31.81
31	432.15	442.97	31.53	31.73

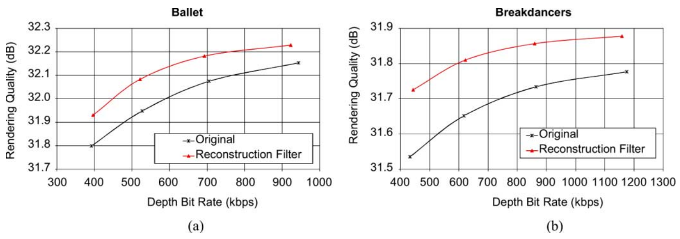

Fig. 3. RD curves for depth reconstruction filter: (a) Ballet, (b) Breakdancers.

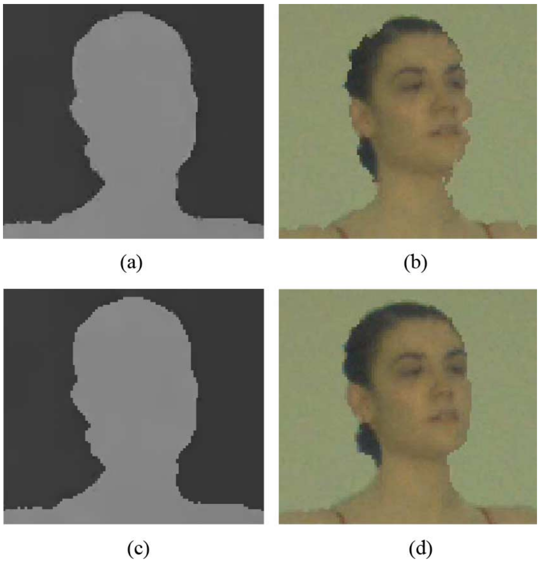

Fig. 4. Effect of the depth reconstruction filter (QP31, 33rd frame): (a) without depth reconstruction filter, (b) rendering result for (a), (c) with depth reconstruction filter, (d) rendering result for (c).

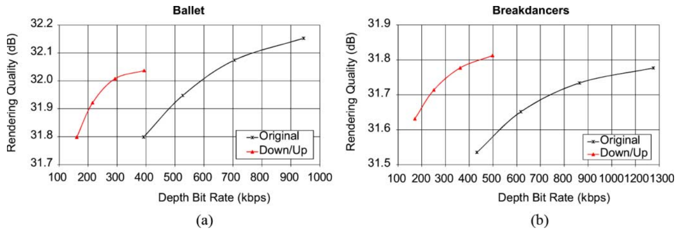

Fig. 5. RD curves for depth down/up sampling: (a) Ballet, (b) Breakdancers.

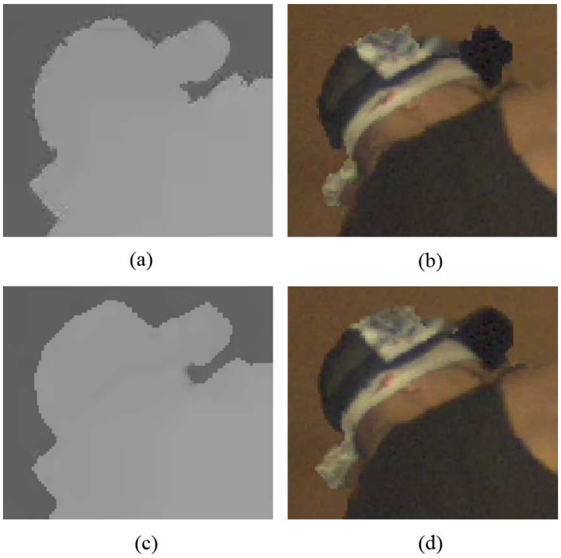

Fig. 6. Effect of the depth down/up sampling (QP31, 1st frame): (a) without depth down/up sampling, (b) rendering result for (a), (c) with depth down/up sampling, (d) rendering result for (c).

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right). \quad (10)$$

However, the depth image is 3-D information to support the virtual view synthesis, thus its quality should be evaluated in terms of rendering quality. In this letter, we measure the rendering PSNR by mean squared error (MSE) between the original image (I_{org}) and the rendered image (I_{ren}) as in (11):

$$MSE_{ren} = \frac{1}{w \times h}\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}\|I_{org}(i,j) - I_{ren}(i,j)\|^2. \quad (11)$$

III. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested the proposed methods on "Breakdancers" and "Ballet" sequences [10]. Among the eight views, view 3 and view 5 were selected as reference views and view 4 was set as a virtual view to be synthesized. Depth views were encoded using JMVC 3.0 [11] with QP 22, 25, 28, and 31. The delta QP for the hierarchical-B picture structure was set as zero in all layers. We used own rendering method [12] and it was fairly used and texture videos were not encoded. The original coding means coding results without the proposed method.

The 7 × 7 frequent-low-high filter was used and bilateral filter parameters, the color sigma and space sigma, were set as 10 and 1, respectively. Experimental results for depth reconstruction filter are given in Tables I and II. The rate-distortion (RD) curves are shown in Fig. 3. Fig. 4 shows the effect of the depth reconstruction filter by comparing decoded depth images and rendered images for "Ballet" sequence coded with QP 31.

Experimental results for the down/up sampling filter in terms of the depth coding rate and rendering quality are given in Tables III and IV. The 5 × 5 median and 13 × 13 frequent-low-high filters were used for up-sampling. RD curves are illustrated

TABLE III
EXPERIMENTAL RESULTS FOR DOWN/UP SAMPLING (BALLET)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	943.50	393.09	32.15	32.04
25	705.07	292.59	32.07	32.01
28	526.68	215.12	31.95	31.92
31	391.52	160.55	31.80	31.80

TABLE IV
EXPERIMENTAL RESULTS FOR DOWN/UP SAMPLING (BREAKDANCERS)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	1174.53	497.67	31.78	31.81
25	863.78	361.36	31.73	31.78
28	616.88	250.52	31.65	31.71
31	432.15	170.50	31.53	31.63

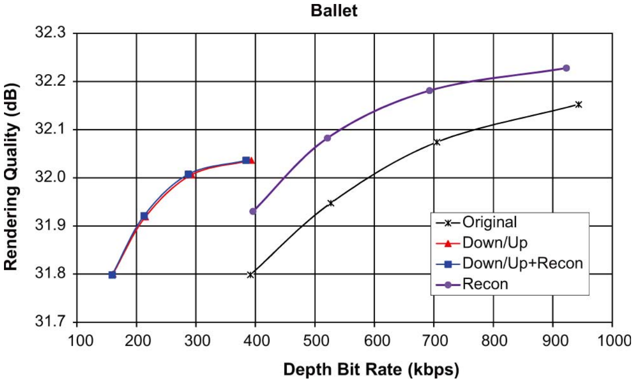

Fig. 7. Comparison of RD curves for four coding methods (Ballet).

in Fig. 5. Fig. 6 shows decoded depth images and rendered images for depth down/up sampling.

From experimental results, we confirmed that the proposed depth reconstruction filter and down/up sampling achieve better depth coding performance and better rendering quality.

Tables V and VI show experimental results for both the depth reconstruction filter and down/up sampling methods used. The combined scheme was slightly better than the down/up sampling case. This is due to the effect of the reconstruction filter being offset by the up-sampling process. As shown in Figs. 7 and 8, the down/up sampling method achieved a good gain by reducing the bit-rate, whereas the reconstruction filter improved rendering quality.

IV. CONCLUSIONS

In this letter, we have proposed a depth reconstruction filter and depth down/up sampling methods. The proposed depth reconstruction filter consists of the frequent-low-high filter and the bilateral filter. The depth reconstruction filter recovers the object boundary and gives advantages to both depth coding and rendering. In addition, the depth down/up sampling methods were designed by considering the characteristics of the depth image and enormously reduced the depth bit-rate without degradation of rendering quality. By experiments, we confirmed that the proposed schemes achieved better coding performance and rendering quality.

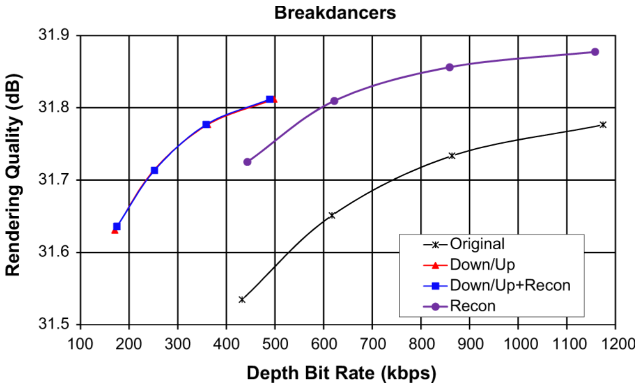

Fig. 8. Comparison of RD curves for four coding methods (Breakdancers).

TABLE V
EXPERIMENTAL RESULTS FOR COMBINED METHOD (BALLET)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	943.50	384.39	32.15	32.04
25	705.07	287.29	32.07	32.01
28	526.68	212.83	31.95	31.92
31	391.52	159.07	31.80	31.80

TABLE VI
EXPERIMENTAL RESULTS FOR COMBINED METHOD (BREAKDANCERS)

QP	Depth rate (kbps)		Rendering quality (dB)	
	original	proposed	original	proposed
22	1174.53	489.07	31.78	31.81
25	863.78	358.35	31.73	31.78
28	616.88	252.04	31.65	31.71
31	432.15	174.71	31.53	31.64

REFERENCES

[1] H. Y. Shum, S. B. Kang, and S. C. Chan, “Survey of image-based representations and compression techniques,” *Proc. of IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[2] A. Smolic and D. McCutchen, “3DAV exploration of video-based rendering technology in MPEG,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 348–356, Mar. 2004.

[3] F. Isgro, E. Trucco, P. Kauff, and O. Schreer, “3D image processing in the future of immersive media,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 288–303, 2004.

[4] A. Smolic and P. Kauff, “Interactive 3D video representation and coding technologies,” *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, pp. 99–110, 2005.

[5] M. Tanimoto, “Overview of free viewpoint television,” *Signal Process.: Image Commun.*, vol. 21, pp. 454–465, Jul. 2006.

[6] A. Smolic, H. Kimata, and A. Vetro, “Development of MPEG standards for 3D and free viewpoint video,” in *Proc. SPIE Optics East, Three-Dimensional TV, Video, and Display IV*, Boston, MA, Oct. 2005.

[7] ISO/IEC JTC1/SC29/WG11, Description of Exploration Experiments in 3-D Video Coding, Doc. N10173 2008.

[8] ISO/IEC JTC1/SC29/WG11, Survey of Algorithms Used for Multi-View Video Coding (MVC), Doc. N6909 2005.

[9] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE Int. Conf. Computer Vision*, Bombay, India, 1998.

[10] MSR 3D Video Sequences [Online]. Available: http://www.research.microsoft.com/vision/ImageBasedRealities/3DVideoDownload/

[11] ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Joint Multiview Video Coding (JMVC) 3.0, Doc. JVT-AC207 2008.

[12] K. J. Oh, S. Yea, A. Vetro, and Y. S. Ho, “Virtual view synthesis method and self evaluation metrics for free viewpoint television and 3D video,” *Int. J. Imag. Syst. Technol. (IJIST)*, unpublished.