Optical Engineering

SPIEDigitalLibrary.org/oe

Edge-preserving down/upsampling for depth map compression in high-efficiency video coding

Huiping Deng
Li Yu
Juntao Zhang
Bin Feng
Qiong Liu

SPIE

Edge-preserving down/upsampling for depth map compression in high-efficiency video coding

Huiping Deng
Wuhan University of Science and Technology
School of Information Science and Engineering
Wuhan 430074, China
and
Huazhong University of Science and Technology
Wuhan National Laboratory for Optoelectronics
Department of Electronics and Information Engineering
Wuhan 430074, China
E-mail: denghuiping.hust@gmail.com

Li Yu
Juntao Zhang
Bin Feng
Qiong Liu
Huazhong University of Science and Technology
Wuhan National Laboratory for Optoelectronics
Department of Electronics and Information Engineering
Wuhan 430074, China

Abstract. An efficient down/upsampling method to compress a depth map efficiently within the high-efficiency video coding (HEVC) framework is presented. A different edge-preserving depth upsampling method is proposed by using both the texture and depth information. We take into account the edge similarity between depth maps and their corresponding texture images as well as the structural similarity among depth maps to build a weight model. Based on the weight model, the optimal minimum mean square error upsampling coefficients are estimated from the local covariance coefficients of the downsampled depth map. The upsampling filter is combined with HEVC to increase coding efficiency. The objective results demonstrate that we achieve a maximum bit rate saving of 32.2% compared to full resolution method and 27.6% compared to a competing depth down/upsampling method on depth bit rate. The subjective evaluation showed that our proposed method achieves better quality in synthesized views than existing methods do. [DOI: 10.1117/1.OE.52.7.071509]

Subject terms: three-dimensional video; high efficiency video coding; view synthesis; depth coding; edge-preserving upsampling.

Paper 121592SS received Oct. 31, 2012; revised manuscript received Jun. 8, 2013; accepted for publication Jun. 11, 2013; published online Jul. 17, 2013.

1 Introduction

The research and development in three-dimensional (3-D) video are capturing the attention of the research community, application developers, and the game industry. Many interesting applications of 3-D video—such as 3-D television (3DTV), free-viewpoint television, 3-D cinema, gesture recognition systems, and other consumer electronics products—have been developed. An attractive 3-D video representation is a multiview video plus depth (MVD) format,[1] which allows rendering numerous viewing angles from only two to three given input views. However, MVD results in a vast amount of data to be stored or transmitted, and efficient compression techniques for MVD are vital for achieving high 3-D visual experience with constrained bandwidth. In addition, the introduction of MVD format allows generating an arbitrary number of intermediate views with low-cost depth image–based rendering[2] techniques, but the quality depends on the accuracy of the depth maps.[3,4] Thus, in this article, we concentrate on the compression of depth information in an MVD format.

A new video coding standard for high efficiency video coding (HEVC)[5] is now being finalized with a primary focus on efficient compression of monoscopic video. Preliminary results have already demonstrated that this new standard provides the same subjective quality at 50% of the bit rate compared to H.264/AVC High Profile. Recently, JCT-3DV has been formed for the development of new 3-D standards, including extensions of HEVC. Since depth maps generally have more spatial redundancy than natural images, the depth down/upsampling can be combined with HEVC framework to increase coding efficiency. There have been some works proposed to compress a downsampled depth map at the encoder in the H.264/AVC framework.[6–9] MPEG 3DV experiments also demonstrate that this down/upsampling-based depth coding approach can improve the depth map coding efficiency.[10] At the same time, 3D-AVC Test Model[11] successfully exploits the possibility of subsampling depth data by the factor of 2, which substantially increases compression efficiency. Since the quality of the synthesized views depend on the accuracy of the depth map information, depth coding-induced distortion not only affects the depth quality but also the synthesized view quality. Therefore, depth down/upsampling method at the decoder needs to be carefully designed to guarantee synthesized view quality.

Classical techniques, such as pixel repetition, bilinear, or bicubic interpolation cause jagged boundaries, blurred edges, and annoying artifacts around edges. Bilateral filter is a widely used edge-preserving filtering technique, where the weights of the filter are selected as a function of a photometric similarity measure of the neighboring pixels. Besides that, a joint bilateral filter[12] is proposed by using auxiliary information from high-resolution images, which is beneficial for edge preserving. The concepts of bilateral and joint bilateral filter have been used for in-loop filtering[13–15] and postfiltering[16–18] on reconstructed depth images. Liu et al.[15] designed a joint trilateral in-loop filter to reconstruct the depth map that takes into account both the similarity among depth samples and that among corresponding texture pixels. Wildeboer et al.[16] proposed a joint bilateral upsampling algorithm by utilizing the high-resolution texture video in the process of depth upsampling; they calculated a weight-cost based on pixel positions and intensity similarities. Ekmekcioglu et al.[17] exploited an adaptive depth map upsampling algorithm with a corresponding color image in order to obtain coding gain while maintaining the quality of the synthesized view. Recently, Schwarz et al.[18] introduced an adaptive depth filter utilizing an edge information from the texture video to improve HEVC efficiency. However, the texture-assisted joint bilateral filter for depth image suffers from the texture copy problem. The edge-directed interpolation techniques recover sharp edges while suppressing pixel jaggedness and blurring artifacts by imposing accurate source models. Li and Orchard[19] proposed a new edge-directed interpolation (NEDI) algorithm for natural images, which exploits image geometric regularity by using the covariance of a low-resolution image to estimate that of a high-resolution image. Asuni and Giachetti[20] improved the stability of NEDI by using edge segmentation. Zhang et al.[21] estimated the low-resolution covariance adaptively with improved nonlocal edge-directed interpolation. Since NEDI needs a relatively large window to compute the covariance matrix for each missing sample, it may introduce spurious artifacts in local structures due to nonstationary structures and result in incorrect covariance estimate.

Preserving the edges of depth maps is important for improving the synthesized view quality. This article proposes a novel edge-preserving depth upsampling method for down/upsampling-based depth coding using both the texture and depth information. The optimal minimum mean square error (MMSE) upsampling coefficients are estimated from the local covariance matrix of the downsampled depth map. By using an adaptive weight model, which takes into account both the structural similarity within the depth map and the edge similarity between the depth map and its corresponding texture image, our proposed method is capable of suppressing artifacts caused by the different geometry structures in a local window.

The remainder of this article is organized as follows. Section 2 describes the depth map coding framework and details the proposed down- and upsampling algorithms. Section 3 presents some experimental results and comparative studies and Sec. 4 concludes the article.

2 Proposed Method

Figure 1 shows the framework of the proposed depth map encoder and decoder based on a HEVC codec. We utilize the efficiency of HEVC and concentrate on depth down/upsampling to increase coding efficiency and synthesized view quality. The encoder contains a preprocessing block that enables the spatial resolution reduction of depth data. Then the resulting depth map is encoded with HEVC. For the decoding process, a novel edge-preserving upsampling (EPU) is utilized to upsample the spatial resolution of the decoded depth map, especially on object boundaries, by taking the depth and texture characteristics into account. The motivation is that, on one hand, with an efficient HEVC codec, encoding the depth data on the reduced resolution can reduce the bit rate substantially. On the other hand, with an efficient upsampling algorithm, encoding the depth data on the reduced resolution can still achieve a good synthesized view quality. The novelty of this approach is the two key components of the proposed depth map coding framework: reliable median downsampling and EPU filter. In what follows, we give a detailed description of the down/upsampling algorithm.

2.1 *Depth Prefiltering*

We use an edge detection–based prefiltering before downsampling to preserve important objection boundaries and remove potential high frequencies in constant depth regions. Figure 2 illustrates a block diagram of the prefiltering. It contains three blocks of boundary layer detection, Gaussian blur, and boundary enhancement. A Canny edge detector[22] divided the input depth map into the smooth region and the boundary layer. The filtered depth map contains the enhanced boundaries and the blurred smooth region.

The smooth depth region is then filtered using a bilateral filter. The bilateral filter is an edge-preserving filtering technique where the kernel filter weights are modified as a function of the photometric similarity between pixels, thus giving

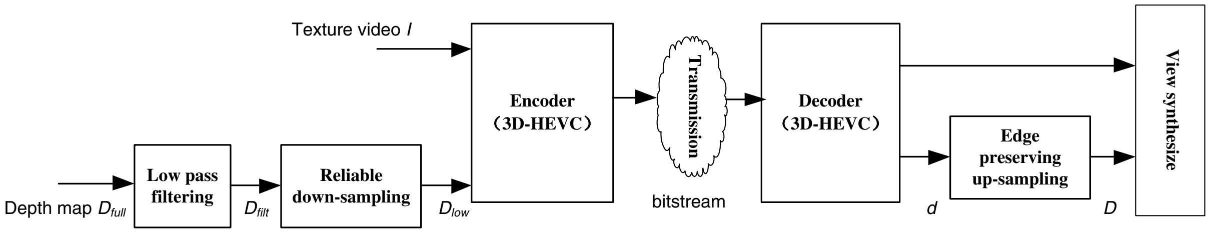


Fig. 1 Down/upsampling coding scheme in HEVC.

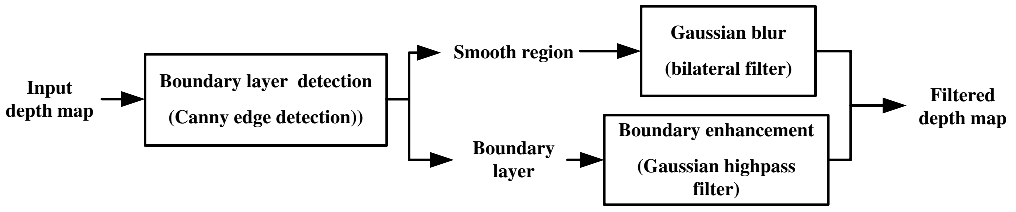


Fig. 2 Block diagram of the depth map prefiltering.

higher weights to pixels belonging to similar regions and reducing the blurring effect in the edges, where photometric discontinuities are present. Let us consider $D_{\mathrm{full}}(p)$ as the intensity of the pixel at position p and Ω^p its neighborhood and the resulting filtered pixel $D_{\mathrm{filt}}(p)$ obtained with the bilateral filter is:

$$D_{\mathrm{filt}}(p) = \frac{1}{k_p} \sum_{q \in \Omega^p} D_{\mathrm{full}}(p) f(p, q) g[\|D_{\mathrm{full}}(p) - D_{\mathrm{full}}(q)\|], \tag{1}$$

where

$$f(\cdot) = \exp\left(-\frac{\|p-q\|^2}{2\sigma_f^2}\right)$$

is a two-dimensional (2-D) smoothing kernel also known as the domain term that measures the closeness of the pixels, and

$$g(\cdot) = \exp\left\{-\frac{[D_{\mathrm{full}}(p) - D_{\mathrm{full}}(q)]^2}{2\sigma_g^2}\right\}$$

is the range term that measures the intensity similarity of the pixels. The scalar $k_p = \sum_{q \in \Omega^p} f(p,q) g(\|D_{\mathrm{full}}(p) - D_{\mathrm{full}}(q)\|)$ is a normalization factor. In our experiment, the filter size is 15×15, and $\sigma_f = 3.5$ and $\sigma_g = 15$.

The boundary layer is enhanced by a Gaussian high-pass filtering. We mark a 7-pixel wide area along depth edges as the boundary layer which includes foreground and background boundary information. In our experiment, the boundary layer is enhanced by a Gaussian high-pass filter with a size of 3×3 and $\sigma = 0.5$.

2.2 *Depth Downsampling*

Reducing the resolution of encoding depth can reduce the bit rate substantially, while the loss of resolution also degrades the quality of the depth map. Therefore, the downsampling method should be designed for better recovering of the quality of high-resolution depth after decoding. Conventional linear downsampling filters create new unrealistic pixel values which will spread to the entire depth map in the upsampling procedure, further causing distortion in the synthesized view.

Considering the above, we propose a reliable median filter for depth downsampling. The proposed reliable median filter is a nonlinear downsampling filter. The downsampled results are obtained in two steps:

Step 1 We obtain those reliable depth values $R_{m\times n}$ of a block $W_{m\times n}$ of the depth map in detail as follows.

Define $W_{m\times n}$ as a $m \times n$ block of the depth map, we sort all pixels in $W_{m\times n}$ by e intensity value and the mean value for $W_{m\times n}$ is defined by

$$\begin{aligned} \mathrm{sort}[W(x,y)] &= \{D_1, D_2, \ldots D_{m\times n}\} \\ D_{\mathrm{ave}} &= \mathrm{mean}(W_{m\times n}). \end{aligned} \tag{2}$$

The pixels in $W_{m\times n}$ are categorized into low and high groups by D_{ave} as

$$W(x,y) \in \begin{cases} S_{\mathrm{fg}}, & \text{if } W(x,y) > D_{\mathrm{ave}} \\ S_{\mathrm{bg}}, & \text{otherwise} \end{cases}. \tag{3}$$

Let $\max(W_{m\times n})$ and $\min(W_{m\times n})$ be the maximum and minimum values of the block $W_{m\times n}$, respectively. If the maximum value $\max(W_{m\times n})$ and minimum value $\min(W_{m\times n})$ are very close, then the local window $W_{m\times n}$ is a smooth region, so all pixels in the block $W_{m\times n}$ are reliable candidates; otherwise, the local window $W_{m\times n}$ contains foreground and background regions, so only the pixels belonging to the foreground region are chosen as the reliable candidates in order to avoid background covering foreground. The reliable candidates formulated as follows:

$$R_{m\times n} = \begin{cases} W_{m\times n}, & \max(W_{m\times n}) - \min(W_{m\times n}) \\ S_{\mathrm{fg}}, & \text{otherwise} \end{cases}, \tag{4}$$

where the threshold $T_0 = 10$ in our experiment.

Step 2 The median of the reliable data is the filtering results. The reliable median filter for depth downsampling is

$$D_d(x,y) = \mathrm{median}(R_{m\times n}). \tag{5}$$

The proposed reliable depth downsampling filter has the following merits over other linear filters: (1) it is more robust against outliers; a noisy neighboring pixel does not affect the median value significantly and (2) the median filtering does not create new unrealistic pixel values when the filter straddles an edge since the median value must actually be the value of one of the pixels in the same object.

The proposed downsampling excludes the nonsimilar neighbor pixels from the filtering process, thus discriminating from pixels that belong to different objects. It is a generalized form of a 2-D median downsampling filter.[6,7] When the downsampling factor is 2, the reliable-based median filter can be simplified as the 2-D median downsampling filter.

2.3 *Edge-Preserving Depth Upsampling*

After HEVC encoding and decoding, the downsampled depth map d is needed to be recovered to the original full resolution for rendering virtual views. An EPU is proposed for depth map reconstruction, utilizing edge information from the corresponding texture frame.

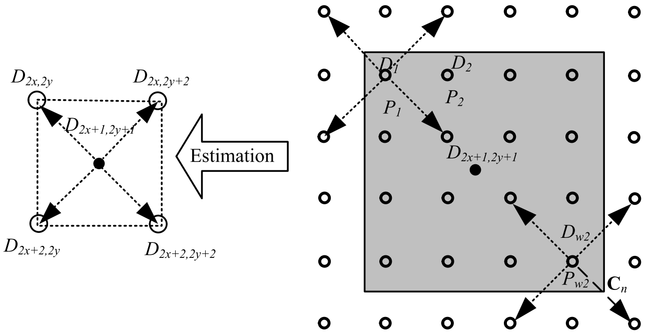


Fig. 3 Covariance estimation based on local statistics from a local window.

Figure 3 gives the sketch map of the upsampling process. Let d denotes the input low-resolution depth map of size $M \times N$. We start with the simplest case of upsampling by a factor of 2 and assume D is the high-resolution depth map after upsampling to size $2M \times 2N$. We first copy the low-resolution depth map d directly to its high-resolution version D, i.e., $D_{2x,2y} = d_{x,y}$ and then interpolate $D_{2x+1,2y+1}$, $D_{2x+1,2y}$, and $D_{2x,2y+1}$ from D in two steps. The first step is to interpolate $D_{2x+1,2y+1}$ from its four nearest neighbors $D_{2x,2y}$, $D_{2x+2,2y}$, $D_{2x,2y+2}$, and $D_{2x+2,2y+2}$ along the diagonal directions of a square lattice. The second step is to interpolate other missing samples $D_{2x+1,2y}$ and $D_{2x,2y+1}$ from a rhombus lattice in the same way after a 45-deg rotation of the square grid. Therefore, the implementation of all the pixels is almost identical. For example, $D_{2x+1,2y+1}$ is calculated as

$$D_{2x+1,2y+1} = k_0 D_{2x,2y} + k_1 D_{2x,2y+2} + k_2 D_{2x+2,2y} + k_3 D_{2x+2,2y+2}, \tag{6}$$

where k_0, k_1, k_2, and k_3 are interpolation coefficients.

Since natural images typically consist of smooth areas, textures, and edges, they are not globally stationary. A reasonable assumption is that the sample mean and variance of a pixel are equal to the local mean and variance of all pixels within a fixed range surrounding. The validity of the assumption is applied in most statistical image representations in previous work as shown by Kuan[23] and Lee.[24] Moreover, compared with natural images, depth maps are more homogenous mostly, therefore, it is reasonable to treat depth maps as being locally stationary. Furthermore, optimal MMSE linear interpolation is successful in the image recovery in that it effectively removes noise while preserving important image features (e.g., edges). Thus, under the assumption that depth image can be modeled as a locally stationary Gaussian process, according to classical Wiener filtering theory, the optimal MMSE linear interpolation coefficients $\mathbf{K} = [k_0, k_1, k_2, k_3]^{\mathbf{T}}$ are given by

$$\mathbf{K} = \mathbf{R}^{-\mathbf{1}}\mathbf{r}, \tag{7}$$

where $\mathbf{R} = E[\mathbf{D}\mathbf{D}^T]$, $\mathbf{D} = [D_{2x,2y}, D_{2x+2,2y}, D_{2x,2y+2}, D_{2x+2,2y+2}]^T$, and $\mathbf{r} = [D_{2x+1,2y+1}\mathbf{D}]$ are the local covariance at the high-resolution level.

By exploiting the similarity between the high-resolution covariance and the low-resolution covariance, $\mathbf{R}$ and $\mathbf{r}$ can be estimated from a local window of its low-resolution depth map. As shown in Fig. 3, we estimate $\mathbf{R}$ and $\mathbf{r}$ based on the local statistics from a local $w \times w$ window centered at the interpolated pixel location, leading to

$$\hat{\mathbf{R}} = p_1 \mathbf{c}_1^T \mathbf{c}_1 + p_2 \mathbf{c}_2^T \mathbf{c}_2 + \ldots + p_{w^2} \mathbf{c}_{w^2}^T \mathbf{c}_{w^2} = \sum_{n=1}^{w^2} p_n \mathbf{c}_n^T \mathbf{c}_n$$

$$\hat{\mathbf{r}} = p_1 \mathbf{c}_1^T D_1 + p_2 \mathbf{c}_2^T D_2 + \ldots p_{w^2} D_{w^2}^T \mathbf{c}_{w^2} = \sum_{n=1}^{w^2} p_n \mathbf{c}_n^T D_n, \tag{8}$$

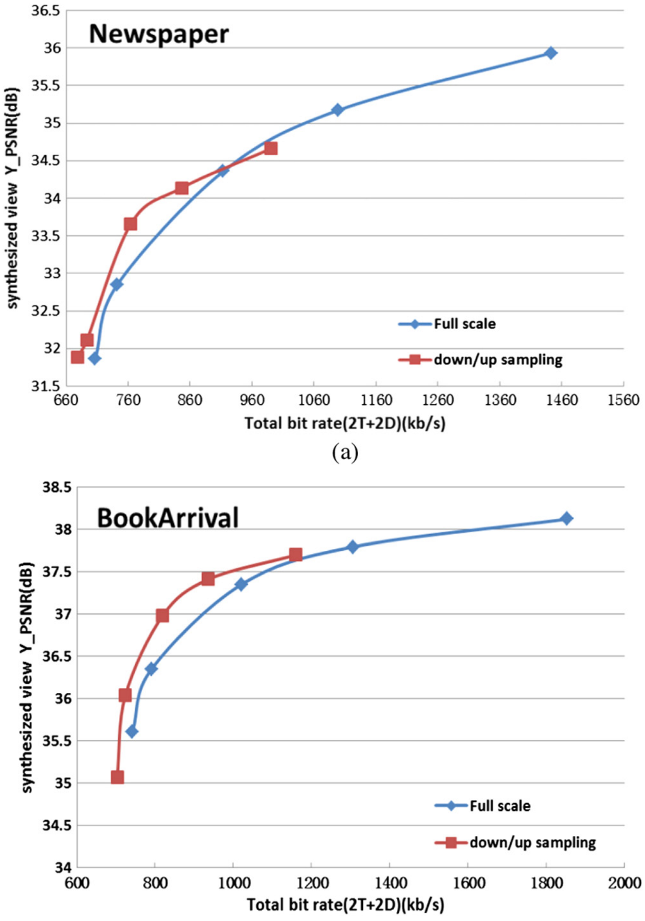
where D_n is the known pixel from low resolution $d(D_{2x,2y} = d_{x,y})$,p_n is the weighting of sample D_n, and $\mathbf{c}_n$ is a 4×1 matrix whose samples are the four neighbors of D_n along the diagonal directions, as shown in Fig. 3.

We note that the covariance estimation in NEDI (Ref. 19) with each sample inside the $w \times w$ window having the same weight $p_n = 1/w^2$ is a special case of ours. In edge-preserving depth map upsampling, the samples $\mathbf{D}_0 = [D_1, D_2, \ldots\ldots, D_{w^2}]^{\mathbf{T}}$ used to calculate coefficients should have similar geometric structure (i.e., edge direction) with the region centered in the interpolated pixel $D_{2x+1,2y+1}$. Otherwise, in the presence of a sharp edge, if a sample is interpolated across instead of along the edge direction, large and visually disturbing artifacts will be introduced. In this article, we introduce a weight model for each sample and make samples adaptive to the local characteristics of the depth map.

Aiming to take advantage of the geometric similarity within depth maps as well as the photometric similarity between the depth map and its corresponding texture sequence, we propose to use the pixel distance, intensity difference, and texture similarity to build a weight model with

$$p_n = \frac{p_n^c + p_n^d + p_n^t}{3}, \tag{9}$$

where p_n^c depends on the distance between the current pixel position (x_n, y_n) and the center pixel position (x_c, y_c), which is measured by the Euclidean distance as



(a)

(b)

Fig. 4 Rate distortion (RD) performance comparison of encoding depth maps between full scale and down/upsampling based method. (a) Book Arrival and (b) Newspaper.

$$\text{dist}(n) = \sqrt{(x_c - x_n)^2 + (y_c - y_n)^2}, \tag{10}$$

and given by

$$p_n^c = \frac{\max\_\text{dist} - \text{dist}(n)}{\max\_\text{dist} - \min\_\text{dist}}, \tag{11}$$

where max _dist and min _dist are the maximum and minimum pixel distance within the window W, respectively.

The quantity p_n^d in Eq. (9) is a function of the absolute difference dif $D(n) = |D_n - D_c|$ between the current pixel value D_n and center pixel value D_c in a depth map, and given by

$$p_n^d = \frac{\max\_\text{dif}D - \text{dif}D(n)}{\max\_\text{dif}D - \min\_\text{dif}D}, \tag{12}$$

where max _dif$D$ and min _dif$D$ indicate the maximum and minimum depth intensity difference within the window W, respectively.

Different from texture sequences, the depth maps usually come with their accompanying texture video. It is known that they share similar structures, especially along the edges. Therefore, an additional term p_n^t measuring this similarity is introduced in Eq. (9). Similar to depth samples' similarity p_n^d, the third subcost function p_n^t means the similarity of texture intensity between the current texture pixel value I_n and the center texture pixel value I_c in texture image. It is measured by the absolute difference dif$T(n) = |I_n - I_c|$ as given in

$$p_n^t = \frac{\max\_\text{dif}T - \text{dif}T(n)}{\max\_\text{dif}Tt - \min\_\text{dif}T}, \tag{13}$$

where max _dif$T$ and min _dif$T$ indicate the maximum and minimum texture intensity differences within the window W, respectively. With this texture similarity, even if the reconstructed depth map has certain artifacts around the edges, we can still utilize the corresponding texture information to provide help with depth boundaries.

With the weight model in Eq. (9), we can estimate **R** and **r** using Eq. (8). Consequently, the interpolation coefficients

Table 1 Performaces (bitrate versus synthesized view PSNR) of full scale and down/upsampling depth map coding.

		Full scale			Down/upsampling		
Quantization parameter (QP)	$T1+T2$ (QP32)	$D1+D2$ (kb/s)	$2T+2D$ (kb/s)	Y_PSNR (dB)	$D1+D2$ (kb/s)	$2T+2D$ (kb/s)	Y_PSNR (dB)
				S1			
24	667.6	1186.9	**1854.5**	**38.12**	493.1	**1160.7**	**37.69**
28		638.3	**1305.9**	**37.79**	268.5	**936.1**	**37.41**
32		353.9	**1021.5**	**37.35**	151.9	**819.5**	**36.98**
40		122.8	**790.4**	**36.34**	56.2	**723.8**	**36.04**
44		74.9	**742.5**	**35.61**	35.4	**703**	**35**
				BDrate $(2T+2D) = 8.9\%\%$			
				BDrate $(2D) = 32.2\%$			
				S2			
QP	T1+T2 (QP32)	D1+D2 (kb/s)	2T+2D (kb/s)	Y_PSNR (dB)	D1+D2 (kb/s)	2T+2D (kb/s)	Y_PSNR (dB)
24	651.6	791.1	**1442.7**	**35.93**	339.3	**990.9**	**34.67**
28		447.9	**1099.5**	**35.17**	194.6	**846.2**	**34.14**
32		261.8	**913.4**	**34.35**	112.8	**764.4**	**33.67**
40		91	**742.6**	**32.85**	41.6	**693.2**	**32.12**
44		55.2	**706.8**	**31.86**	26.3	**677.9**	**31.89**
				BD rate $(2T+2D) = 5.3\%$			
				BDrate $(2D) = 27.6\%\%$			

$\mathbf{K} = [k_0, k_1, k_2, k_3]^\mathbf{T}$ needed in Eq. (6) can be obtained from Eq. (7) as

$$\mathbf{K} = \left(\sum_{n=1}^{w^2} p_n \mathbf{c}_n^T \mathbf{c}_n \right)^{-1} \sum_{n=1}^{w^2} p_n \mathbf{c}_n^T y_n. \tag{14}$$

3 Experimental Results

We study the performance of the proposed depth down/upsampling method for depth map coding using two types of test sequences in resolutions (1920 × 1088 pixels: Poznan_Street,[25] Undo_Dancer and 1024 × 768 pixels: Newspaper, Bookarrival[26]), with YUV 4:2:0 8 bits per pixel (bpp) format. The test materials are provided by MPEG and depth maps have been estimated from original video based on the depth estimation reference software.[27] For Poznan_Street sequence, view 3 and view 5 are selected as reference views. For Undo_Dancer sequence, view 4 and view 6 are selected as references and view 5 as the target view. For Book-Arrival sequence, view 8 and view 10 are selected as references and view 9 as the target view.

For each reference depth map, we downsample it by a factor of two before encoding using the 3-D-HEVC test model (HTM) version 4.1[28] with quantization parameters (QP) 24, 28, 32, 40, and 44. The texture video sequences have a fixed QP 32. Thirty frames are coded for each sequence. Other encoder configurations follow those specified in the common test conditions[29] for 3-D video coding. No multiview video coding is applied. After the decoding is finished, the intermediate view is synthesized by view synthesis reference software.[30] The efficiency of the proposed method is evaluated through rate distortion (RD) performance and subjective quality of synthesized view. For the RD curves, the x-axis stands for the total bit rate for the two depth maps and two texture sequences, and the y-axis is the Y_PSNR of the synthesized views compared to the original view.

3.1 *Coding Performance*

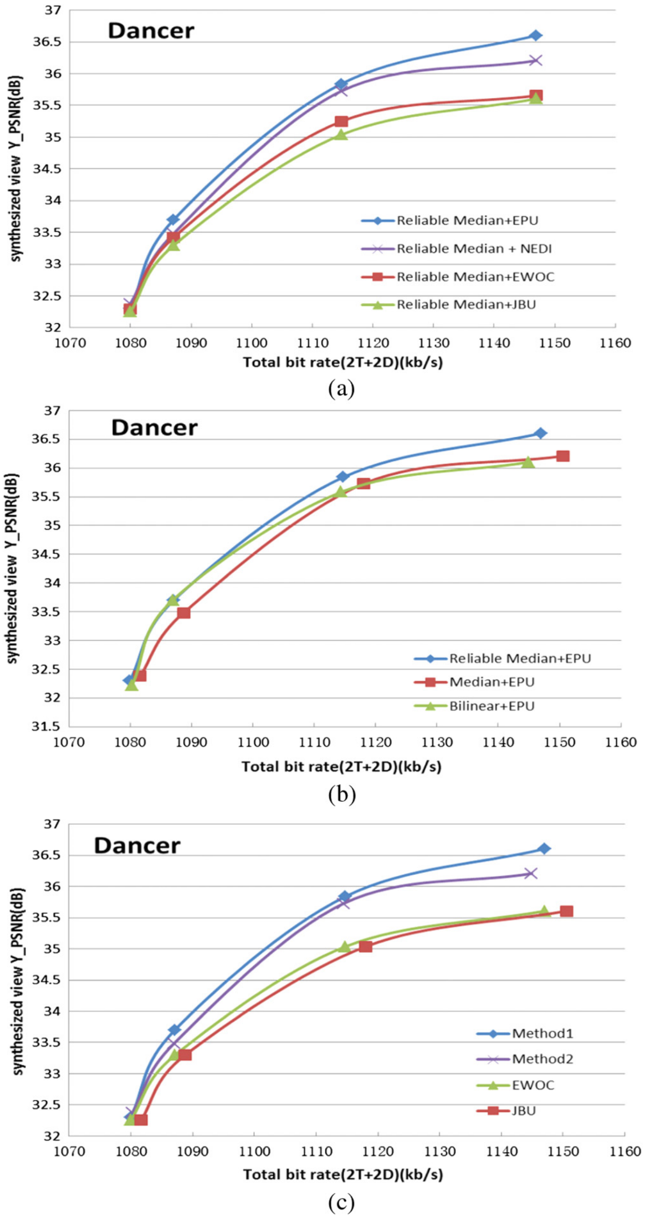
First, the performance of the down/upsampling-based depth coding scheme is compared to that of full scale. For the full scale method, the depth maps are encoded without down/upsampling using HTM reference software. Figure 4 shows the RD curves comparison between the proposed method and the full resolution method.

It can be seen that the down/upsampling-based depth maps coding scheme outperforms full-scale depth map coding at lower bit rates. Specially, as shown in Table 1, bit rate saving is up to 32.2% for "BookArrival" and 27.6% for "Newspaper" on depth maps, whereas it is 8.9% for "BookArrival" and 5.3% for "Newspaper" on total bit rates. Since the bit rates of depth maps are only about 10 to 20% that of texture sequences, the gain of bit rate saving is less for total bit rate than that for depth bit rate. At higher bit rates, the frames are encoded with larger QP and preserve much more details in texture. Therefore, the influence of down/upsampling distortion becomes larger. The RD performance is below the full scale case with high bit rate.

Second, we evaluate the performances of the proposed downsampling, upsampling, and prefiltering method separately. In order to test the effectiveness of the proposed downsampling algorithm, the original depth maps are downsampled using different downsampling methods while being upsampled with the same EPU algorithm. Figure 5(a) shows the RD curves of different depth downsampling method, "Median downsc." stands for the downsampling as proposed by Oh in Ref. 6 and "Reliable Median downsc." that described in Sec. 2.2.

In order to test the effectiveness of the proposed upsampling algorithm, the decoded depth maps are upsampled using different interpolation algorithms while downsampled using the same reliable median downsampling before encoding. Figure 5(b) shows the RD curves of different upsampling methods, where "EPU upsc." stands for the upsampling method as described in Sec. 2.3, "NEDI upsc." stands for the upsampling method in Ref. 9, and "EWOC upsc." and "JBU upsc." stand for the recent published upsampling algorithms in Refs. 17 and 18, respectively.

Figure 5(c) shows the RD curves to compare the coding efficiency of the proposed methods against two advanced



(a)

(b)

(c)

Fig. 5 RD performance of proposed (a) downsampling, (b) upsampling, and (c) down/upsampling method.

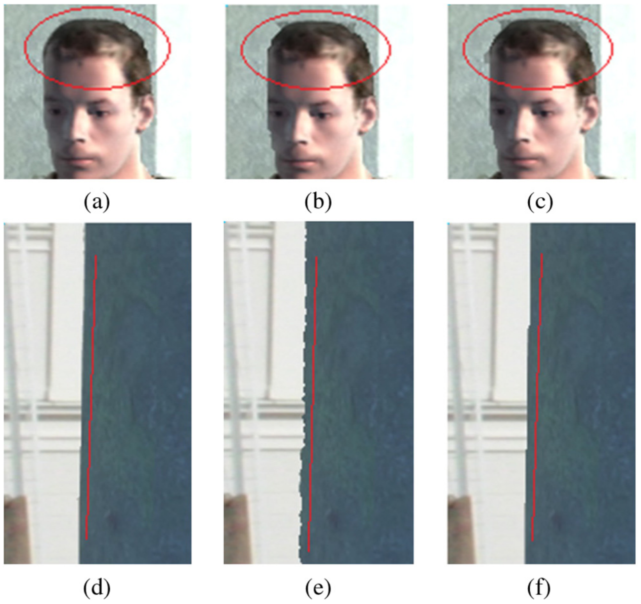


Fig. 6 The synthesized view [Undo_dancer, quantization parameter (QP) = 40 of view 5] with depth map upsampled with (a) (d) proposed, (b) (e) JBU upsampling, and (c) (f) EWOC method at the decoder.

down/upsampling-based depth coding methods. “Method 1” is the combined method, where depth maps are preprocessed as described in Sec. 2.1, then downsampled as described in Sec. 2.2 and upsampled as described in Sec. 2.3. “Method 2” is the result with the proposed downsampling and upsampling. “EWOC” stands for the depth map coding method in Ref. 18. “JBU” stands for the down/upsampling algorithm for depth maps in Ref. 17, where depth maps are downsampled with median filtering and upsampled with JBU. No prefiltering is applied to either “Method 2” or JBU method.

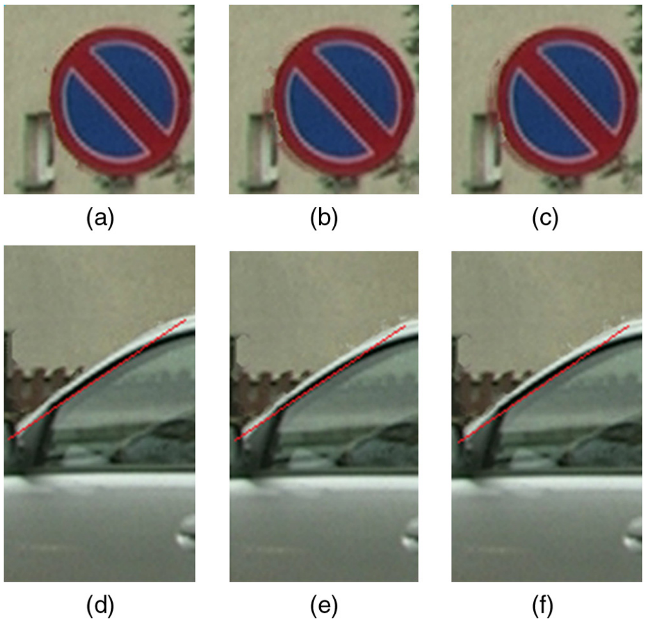


Fig. 7 The synthesized view (Poznan_street, QP = 28 of view 4) with depth map upsampled with (a) (d) proposed, (b) (e) JBU upsampling, and (c) (f) EWOC method at the decoder.

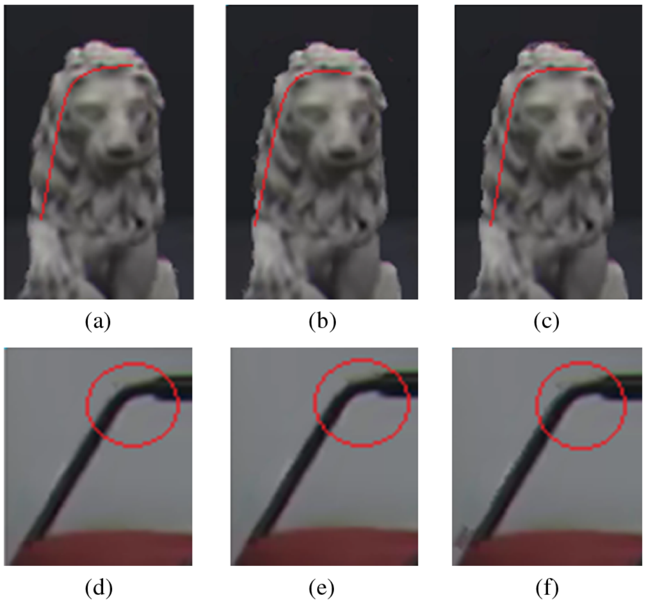


Fig. 8 The synthesized view (Bookarrival, QP = 28 of view 8) with depth map upsampled with (a) (d) proposed, (b) (e) JBU upsampling, and (c) (f) EWOC method at the decoder.

We can see that both the proposed upsampling method and downsampling show good performance as shown in Fig. 5(a) and 5(b). By combining the proposed prefiltering, downsampling and the upsampling methods, additional gain can be achieved as shown in Fig. 5(c).

3.2 *Synthesized View Quality*

Depth map downsampling and upsampling directly impact the subjective quality of synthesized views. Figures 6–8 compare our proposed upsampling method with EWOC upsampling and the JBU in terms of the subjective quality of the synthesized views at the decoder after depth map encoding at the same rate. It is seen that the synthesized images with EWOC interpolation and JBU upsampling exhibit strong jaggedness around object edges. On the other hand, for our proposed upsampling method, it employs texture image which provides the edge information in the upsampling procedure; therefore, our method obtains clearer and smoother edges along object boundaries.

Table 2 Processing times of full scale coding and down/upsampling coding.

	Full scale		Down/upsampling			
	Enc T [s]	Dec T [s]	Down T [s]	Enc T [s]	Up T [s]	Dec T [s]
S1	71646	289	2620	18288	3360	94
	Sum $T = 71935$ s		Sum $T = 24362$ s			
S2	89712	314	4014	22931	13800	144
	Sum $T = 90026$ s		Sum $T = 40899$ s			

3.3 *Computational Complexity Analysis*

We show the processing times in the Table 2. Depth map encoding time for the proposed method contains downsampling time (low-pass filtering and downsampling procedures), HEVC encoding time, decoding time, and proposed upsampling time. Depth coding time for full scale contains HEVC encoding time and decoding time. $S1$ and $S2$ denote the Newspaper and Book-Arrival sequences.

Since the resolution of encoding video in down/upsampling-based method is less than full scale method, the encoding time of downsampled is far less than that of full scale. Although additional downsampling and upsampling procedures are needed for the down/upsampling method, the overall computation time of down/upsampling-based method is less than the full scale method as shown in Table 2.

4 Conclusions

We have presented an edge-preserving depth upsampling method for down/upsampling-based depth coding within the HEVC framework. Different from the NEDI algorithm of Ref. 19, we introduced a weight model for each sample that incorporates geometric similarity as well as intensity similarity in both the depth map and its corresponding texture sequence, thus allowing an adaptation of interpolation coefficients to the edge orientation. An evaluation of performance in terms of coded data and synthesized views has been provided. Experimental results show that our proposed interpolation method for down/upsampling-based depth coding improves both the coding efficiency and synthesized view quality.

Acknowledgments

This work is supported in part by the National Science Foundation of China under Grant Nos. 61231010 and 61202301.

References

1. K. Muller, P. Merkle, and T. Wiegand, “3-D video representation using depth maps,” *Proc. IEEE* **99**(4), 643–656 (2011).
2. C. Fehn, “Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” *Proc. SPIE*, **5291**, 93–104 (2004).
3. K. Sharma, I. Moon, and S. G. Kim, “Depth estimation of features in video frames with improved feature matching technique using Kinect sensor,” *Opt. Eng.* **51**(10), 107002 (2012).
4. S. S. Zhang and S. Yan, “Depth estimation and occlusion boundary recovery from a single outdoor image,” *Opt. Eng.* **51**(8), 087003 (2012).
5. B. Bross et al., “High Efficiency Video Coding (HEVC) text specification draft 8,” ITU-T SG16 WP3, and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-J1003, Stockholm, CE (2012).
6. K. J. Oh et al., “Depth reconstruction filter for depth coding,” *Electron. Lett.* **45**(6), 305–306 (2009).
7. K. J. Oh et al., “Depth reconstruction filter and down/up sampling for depth coding in 3-D video,” *IEEE Signal Process. Lett.* **16**(9), 747–750 (2009).
8. H. P. Deng et al., “A joint texture/depth edge-directed up-sampling algorithm for depth map coding,” in *Proc. 2012 IEEE Int. Conf. Multimedia and Expo(ICME’12)*, pp. 646–650, IEEE, Melbourne (2012).
9. M. O. Wildeboer et al., “Color based depth up-sampling for depth compression,” in *Proc. IEEE Conf. Picture Coding Symposium (PCS2010)*, pp. 170–173, IEEE, Nagoya (2010).
10. K. Klimaszewski, K. Wegner, and M. Domanski, “Influence of views and depth compression onto quality of synthesized views,” ISO/IEC JTC1/SC29/WG11, M16758, UK (2009).
11. M. Hannuksela, Y. Chen, and T. Suzuki, “AVC Draft Text 3,” in *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 Doc. JTC3V-A1002, 1st Meeting*, Stockholm, SE (2012).
12. J. Kopf et al., “Joint bilateral upsampling,” *ACM Trans. Graph.* **26**(3), 96 (2007).
13. K. J. Oh, A. Vetro, and Y. S. Ho, “Depth coding using a boundary reconstruction filter for 3-D video systems,” *IEEE Trans. Circ. Syst. Video Technol.* **21**(3), 350–359 (2011).
14. D. Min, J. Lu, and M. N. Do, “Depth video enhancement based on weighted mode filtering,” *IEEE Trans. Image Process.* **21**(3), 1176–1190 (2012).
15. S. J. Liu et al., “New depth coding techniques with utilization of corresponding video,” *IEEE Trans. Broadcast.* **57**(2), 551–561 (2011).
16. M.O. Wildeboer et al., “Depth up-sampling for depth coding using view information,” in *Proc. 3DTV Conf.: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, IEEE (2011).
17. E. Ekmekcioglu et al., “Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering, ” in *Proc. 2009 16th IEEE Int. Conf. Image Processing (ICIP)*, pp. 733–736, IEEE (2009).
18. S. Schwarz et al., “Adaptive depth filtering for HEVC 3D video coding,” in *Proc. 2012 Picture Coding Symposium (PCS 2012)*, pp. 49–52, IEEE (2012).
19. X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Trans. Image Process.* **10**(10), 1521–1527 (2001).
20. N. Asuni and A. Giachetti, “Accuracy improvements and artifacts removal in edge based image interpolation,” in *Proc. 3rd Int. Conf. Computer Vision Theory and Applications (VISAPP’08)*, pp. 58–65, Springer (2008).
21. X. F. Zhang et al., “Nonlocal edge-directed interpolation,” in *Proc. 2009 Pacific Rim Conference on Multimedia (PCM’09)*, pp. 1197–1120, Springer, Bangkok, Thailand (2009).
22. J. F. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 1521–1527 (1986).
23. D. T. Kuan et al., “Adaptive noise smoothing filter for images with signal-dependent noise,” *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**(2), 165–177 (1985).
24. J. S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**(2), 165–168 (1980).
25. M. Domanski et al., “Poznan Multiview Video Test Sequences and Camera Parameters,” MPEG Doc. m17050, ISO/IEC JTC1/SC29/WG11 (2009).
26. I. Feldmann et al., “HHI Test Material for 3D Video,” MPEG/M15413, Archamps, France (2008).
27. M. Tanimoto et al., “Depth Estimation Reference Software(DERS) 4.0,” ISO/IEC JTC1/SC29/WG11, MPEG 2008/M16605, London, UK (2009).
28. “HEVC software,” SVN repository for HTM 4.1, https://hevc.hhi.frauhofer.de/svn/svn_3DVCSofware/tags/HTM-4.1.
29. D. Rusanovskyy, K. Müller, and A. Vetro, “Common test conditions of 3DV core experiments,” in *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 Doc. JTC3V-A1100, 1st Meeting*, Stockholm, SE (2012).
30. M. Tanimoto, T. Fujii, and K. Suzuki, “View Synthesis Algorithm in View Synthesis Reference Software 3.0 (VSRS3.0),” MPEG Doc. M16090, ISO/IEC JTC1/SC29/WG11 (2009).

Huiping Deng received a BS degree in electronics and information engineering, an MS degree in communication and information system from Yangtze University, Jingzhou, China, in 2005 and 2008, respectively. She is currently working toward the PhD degree in the Electronics and Information Engineering Department, HUST. Her research interests are video coding and computer vision, currently focusing on three-dimensional video (3DV).

Li Yu received the BS degree in electronics and information engineering, the MS degree in communication and information system and the PhD degree in electronics and information engineering, all from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995, 1997, and 1999, respectively. In 2000, she joined the Electronics and Information Engineering Department, HUST, where she has a professor since 2005. She is a co-sponsor of China AVS standard special working group and working as the key member of China AVS standard special working group. Her team has applied more than 10 related patents and submitted 79 proposals to AVS standard organization. Her current research interests include multimedia communication and processing, computer network, wireless communication.

Biographies and photographs of the other authors are not available.