# Efficient Techniques for Depth Video Compression Using Weighted Mode Filtering

Viet-Anh Nguyen, Dongbo Min, *Member, IEEE,* and Minh N. Do, *Senior Member, IEEE*

***Abstract*—This paper proposes efficient techniques to compress a depth video by taking into account coding artifacts, spatial resolution, and dynamic range of the depth data. Due to abrupt signal changes on object boundaries, a depth video compressed by conventional video coding standards often introduces serious coding artifacts over object boundaries, which severely affect the quality of a synthesized view. We suppress the coding artifacts by proposing an efficient postprocessing method based on a weighted mode filtering and utilizing it as an in-loop filter. In addition, the proposed filter is also tailored to efficiently reconstruct the depth video from the reduced spatial resolution and the low dynamic range. The down/upsampling coding approaches for the spatial resolution and the dynamic range are used together with the proposed filter in order to further reduce the bit rate. We verify the proposed techniques by applying them to an efficient compression of multiview-plus-depth data, which has emerged as an efficient data representation for 3-D video. Experimental results show that the proposed techniques significantly reduce the bit rate while achieving a better quality of the synthesized view in terms of both objective and subjective measures.**

***Index Terms*—3-D video, depth coding, depth down/upsampling, depth dynamic range, weighted mode filtering.**

## I. Introduction

With the recent development of 3-D multimedia/display technologies and the increasing demand for realistic multimedia, 3-D video has gained more attentions as one of the most dominant video formats with a variety of applications such as 3-D TV or freeview point TV (FTV). 3-D TV aims to provide users with 3-D depth perception by rendering two (or more) views on stereoscopic (or auto-stereoscopic) 3-D display. FTV can give users the freedom of selecting a viewpoint, different from conventional TV where the viewpoint is determined by an acquisition camera. 3-D freeview video can also be provided by synthesizing multiple views at the selected viewpoint according to a user preference [1]. For successful development of 3-D video systems, many technical issues should be resolved, e.g. capturing and analyzing the stereo or multiview images, compressing and transmitting the data, and rendering multiple images on various 3-D displays.

The main challenging issues of 3-D TV and FTV are depth estimation, virtual view synthesis, and 3-D video coding. The depth maps are used to synthesize the virtual view at the receiver side, so accurate depth maps should be estimated in an efficient manner for ensuring a seamless view synthesis. Since the performance of 3-D TV and FTV heavily depends on the number of multiple views, virtual view synthesis is an important technique in 3-D video systems as well. In other words, synthesizing virtual view with the limited number of original views leads to reducing the cost and bandwidth required to capture a number of viewpoints and to transmit the huge amount of data over network.

For 3-D video coding, Joint Video Team (JVT) from the Moving Pictures Experts Group (MPEG) of ISO/IEC and the Video Coding Experts Group (VCEG) of ITU-T jointly standardized multiview video coding (MVC) with an extension of H.264/AVC standard [2], [3]. The MPEG-FTV division, an adhoc group of MPEG, has made a new standard for FTV [4], [5]. Also, a multiview-plus-depth data format was proposed to make the 3-D video systems more flexible [6], [7]. The depth maps are precalculated at the transmitter side and encoded with the corresponding color images together. At the receiver side, the decoded multiview-plus-depth data are utilized to synthesize the virtual view. While a number of methods have been proposed to efficiently compress the multiview video using view redundancies, the depth video coding has not been studied extensively. The depth video coding aims to reduce a depth bit rate as much as possible while ensuring the quality of the synthesized view. Thus, its performance is determined by the quality of the synthesized view, not the depth map itself.

In general, the depth map contains a per-pixel distance between camera and object, and it is usually represented by 8-bit grayscale value. The depth map has unique characteristics such that: 1) the depth value varies smoothly except object boundaries or edges; 2) the edges of the depth map usually coincide with those of the corresponding color image; and 3) object boundaries should be preserved in order to provide the high-quality synthesized view. Thus, the straightforward compression of the depth video using the existing video coding standards such as H.264/AVC may cause serious coding artifacts along the depth discontinuities, which ultimately affect the synthesized view quality.

The depth video coding approaches can be classified into two categories according to coding algorithms: transform

Manuscript received September 12, 2011; revised December 27, 2011 and April 1, 2012; accepted April 13, 2012. Date of publication June 6, 2012; date of current version February 1, 2013. This work was supported by a research grant from the Human Sixth Sense Program, Advanced Digital Sciences Center from Singapore's Agency for Science, Technology, and Research (A*STAR). This paper was recommended by Associate Editor F. Wu.

V.-A. Nguyen and D. Min are with the Advanced Digital Sciences Center, Singapore 138632 (e-mail: vanguyen@adsc.com.sg; dbmin99@gmail.com).

M. N. Do is with the University of Illinois at Urbana-Champaign, Urbana, IL 61820 USA (e-mail: minhdo@illinois.edu).

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/TCSVT.2012.2203212

based coding and postprocessing based coding. Morvan *et al.* proposed a platelet-based method that models depth maps by estimating piecewise-linear functions in the subdivisions of quadtree with variable sizes under a global rate-distortion constraint. They showed that the proposed method outperforms JPEG-2000 encoder with a 1–3 dB gain [8], [9]. Maitre and Do proposed a depth compression method based on a shape-adaptive wavelet transform by generating small wavelet coefficients along depth edges [10]. Although these methods have better performance than the existing image compression methods, they are difficult to be extended into video domain for exploiting temporal redundancies, and are not compatible with the conventional video coding standards such as H.264/AVC. New intraprediction in H.264/AVC was proposed to encode depth maps by designing an edge-aware intraprediction scheme that can reduce a prediction error in macroblocks [11]. Different from the platelet or wavelet based coding methods [8], [10], this scheme can be easily integrated with H.264/AVC. However, the performance of the coding algorithm was evaluated by measuring the depth map itself, not the synthesized view.

In order to meet the compatibility to the advanced H.264/AVC standard, depth video coding algorithms have moved interest on reducing compression artifacts that may exist on depth video which is encoded by H.264/AVC. Kim *et al.* proposed a new distortion metric that considers camera parameters and global video characteristics, and then used the metric in the rate-distortion optimized mode selection to quantify the effects of depth video compression on the synthesized view quality [12]. Lai *et al.* showed that a rendering error in the synthesized view is a monotonic function of the coding error, and presented a method to suppress compression artifacts using a sparsity-based de-artifacting filter [13]. Some approaches have been proposed to encode a downsampled depth map and to use a special upsampling filter after decoding to recover the depth edge information [14]–[16]. The work proposed in [14] 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. Oh *et al.* proposed a new coding scheme based on a depth boundary reconstruction filter which considers occurrence frequency, similarity, and closeness of pixels [17], [18]. Liu *et al.* utilized a trilateral filter, which is a variant of bilateral filter, as an in-loop filter in H.264/AVC and a sparse dyadic mode as an intramode to reconstruct depth map with sparse representations [19].

In this paper, we propose a novel scheme that compresses the depth video efficiently using the framework of a conventional video codec. In particular, an efficient postprocessing method for the compressed depth map is proposed in a generalized framework, which considers compression artifacts, spatial resolution, and dynamic range of the depth data. The proposed postprocessing method utilizes additional guided information from the corresponding color video to reconstruct the depth map while preserving the original depth edge. The depth video is encoded by a typical transform-based motion compensated video encoder, and compression artifacts are addressed by utilizing the postprocessing method as an in-loop filter. In addition, we design a down/upsampling coding approach for both the spatial resolution and the dynamic range of the depth data. The basic idea is to reduce the bit rate by encoding the depth data on the reduced spatial resolution and depth dynamic range. The proposed postprocessing filter is then utilized to efficiently reconstruct the depth video.

For the postprocessing of the compressed depth map, we utilize a weighted mode filtering (WMF) [20], which was proposed to enhance the depth video obtained from depth sensors such as time-of-flight (ToF) camera. Given an input noisy depth map, a joint histogram is generated by first calculating the weight based on spatial and range kernels and then counting each bin on the histogram of the depth map. The final solution is obtained by seeking a mode with the maximum value on the histogram. In this paper, we introduce the concept of the weighted mode filtering in generic formulation tailored to the depth image compression. We will also describe the relation with the bilateral and trilateral filtering methods [19], which have been used in depth video coding, and show the effectiveness of the proposed method with a variety of experiments. The main contributions of this paper over [20] can be summarized as follows.

1) Theoretically analyze the relation between the WMF and the existing approaches in a localized histogram framework to justify its superior performance in the perspective of robust estimation.
2) Effectively utilize the WMF in various proposed schemes for depth coding by considering important depth properties for a better synthesized view quality.
3) Thoroughly evaluate the effectiveness of the WMF in the depth coding context, where the noise characteristics and the objective measure are different from [20].

The remainder of this paper is organized as follows. Section II briefly introduces the weighted mode filtering concept. Section III presents the proposed techniques for depth video compression based on the weighted mode filtering. Experimental results of the proposed coding techniques are presented in Section IV. In Section V, we conclude the paper by summarizing the main contributions.

## II. Weighted Mode Filtering on Histogram

Weighted mode filtering was introduced in [20] to enhance the depth map acquired from depth sensors. In this paper, we utilize such a filter in an effective manner for depth compression. This section provides an insight of the weighted mode filtering over [20] in relations with the existing filters such as the frequent occurrence filter and the bilateral filter, which have been recently used in the postprocessing based depth coding [17]–[19]. Specifically, we shall mathematically show the existing filters are the special cases of the weighted mode filtering and justify the superior performance of the weighted mode filtering for depth compression. For completeness, we also provide a brief redefinition of the weighted mode filtering based on the localized histogram concept.

A *localized* histogram $H(p, d)$ for a reference pixel $p$ and $d^{th}$ bin is computed using a set of its neighboring pixels inside a window, which was introduced by Weijer *et al.* [21].

Specifically, given a discrete function $f(p)$ whose value ranges 0 to $L-1$, the localized histogram $H(p,d)$ is defined at the pixel $p$ and $d^{th}$ bin ($d \in [0, L-1]$). The localized histogram means that each bin has a likelihood value which represents an occurrence of neighboring pixels $q$ inside rectangular (or any shape) regions. The likelihood value is measured by adaptively counting a weighting value computed with a kernel function $w(p,q)$ as

$$H(p,d) = \sum_{q \in N(p)} w(p,q) E_r(d - f(q)) \tag{1}$$

where $w(p,q)$ is a nonnegative function which defines the correlation between the pixels $p$ and $q$. $N(p)$ is the set of neighboring pixels in a window centered at $p$. The weighting function represents the influence of the neighboring pixels on the localized histogram. In essence, the neighboring pixel which exhibits a stronger correlation with the reference pixel $p$ has a larger weighting value $w(p,q)$. Such correlation is generally described by the geometric closeness and the photometric similarity between different pixels. A spreading function $E_r$ models errors that may exist on the input data $f(p)$. A weighted mode filtering is then obtained by seeking the highest mode of the weighted distribution $H(p,d)$, where the final filtered solution $\hat{f}(p)$ is calculated by

$$\hat{f}(p) = \arg\max_d H(p,d). \tag{2}$$

Note that unlike [21], the final solution here is obtained by seeking the highest mode rather than the local mode on the localized histogram. In addition, various filter designs can be realized with different selections of the weighting and spreading functions. In what follows, we shall show the relations of the weighted mode filtering with the existing filters recently used for depth compression by examining different weighting and spreading functions on the localized histogram.

### A. Frequent Occurrence Filter

Consider the case where all the neighboring pixels have the equal weights, i.e., $w(p,q) = 1$. Let the spreading function $E_r(m)$ be represented by a Dirac function $\delta_r(m)$ where $\delta_r(m)$ is 1 if $m = 0$, and 0 otherwise. The filtered result for the weighted mode filtering will be computed as

$$\hat{f}(p) = \arg\max_d \sum_{q \in N(p)} \delta(d - f(q)). \tag{3}$$

In this case, the traditional mode is found, where the reference pixel is replaced with the value with the most frequent occurrence within the neighborhood. Interestingly, it is easy to show that the most frequent occurrence is optimal with respect to $L_0$ norm minimization, in which the error norm is defined as

$$\hat{f}(p) = \arg\min_d \lim_{\alpha \to 0} \sum_{q \in N(p)} |d - f(q)|^{\alpha}. \tag{4}$$

The occurrence frequency is utilized in the depth boundary reconstruction filter for suppressing coding artifacts in compressed depth [17], [18]. However, such operation is very noise dependent and unstable due to the noisy samples in a small neighborhood, which may not be able to preserve the original information of the signal.

### B. Bilateral Filter

To reduce the noise dependence and stabilize the mode operation, the spreading function is utilized to partially relate to the neighboring values. Let the spreading function $E_r(m)$ be represented by a quadratic function $E_B(m) = C - am^2$, where $C$ and $a$ represent arbitrary constant values satisfying $a > 0$. The weighted mode filtering can be rewritten as

$$\begin{aligned} H_B(p,d) &= \sum_{q \in N(p)} w(p,q)(C - a(d - f(q))^2) \\ \hat{f}(p) &= \arg\max_d H_B(p,d). \end{aligned} \tag{5}$$

This equation has some similarities with the weighted least square optimization, which is optimal with respect to $L_2$ norm minimization. To maximize (5), we take the first derivative with respect to $d$

$$\frac{\partial H_B(p,d)}{\partial d} = -2a \sum_{q \in N(p)} w(p,q)(d - f(q)). \tag{6}$$

Assuming $w(p,q) > 0$, the value of $d$ that maximizes $H_B(p,d)$ can be found by solving $\dfrac{\partial H_B(p,d)}{\partial d} = 0$. Thus, we obtain the filtered result as

$$\hat{f}(p) = \frac{\sum_{q \in N(p)} w(p,q) f(q)}{\sum_{q \in N(p)} w(p,q)}. \tag{7}$$

Consider $w(p,q) = G_s(p-q)G_f(f(p) - f(q))$, where $G_s$ is the spatial Gaussian kernel to describe the geometric closeness between two pixels, while $G_f$ is the range Gaussian kernel to measure the photometric similarity. The final solution can be formulated as

$$\hat{f}(p) = \frac{\sum_{q \in N(p)} G_s(p-q) G_f(f(p) - f(q)) f(q)}{\sum_{q \in N(p)} G_s(p-q) G_f(f(p) - f(q))}. \tag{8}$$

This equation is in fact the bilateral filtering expression, which can be considered as a special case of the weighted mode filtering by using a quadratic function to model the input error. Obviously, by computing an appropriate weighting function $w(p,q)$ with an associated function $g(p)$ different from the function $f(p)$ to be filtered, the joint bilateral and trilateral filtering in [19] can be realized by the weighted mode filtering concept. However, the joint bilateral (or trilateral) filtering for depth denoising may still result in unnecessary blur due to its summation [19].

### C. Weighted Mode Filter

In this paper, we adopt a Gaussian function $G_r(m) = e^{-m^2/2\sigma_r^2}$ to represent the spreading function $E_r(m)$ as

$$\begin{aligned} H(p,d) &= \sum_{q \in N(p)} w(p,q) G_r(d - f(q)) \\ \hat{f}(p) &= \arg\max_d H(p,d). \end{aligned} \tag{9}$$

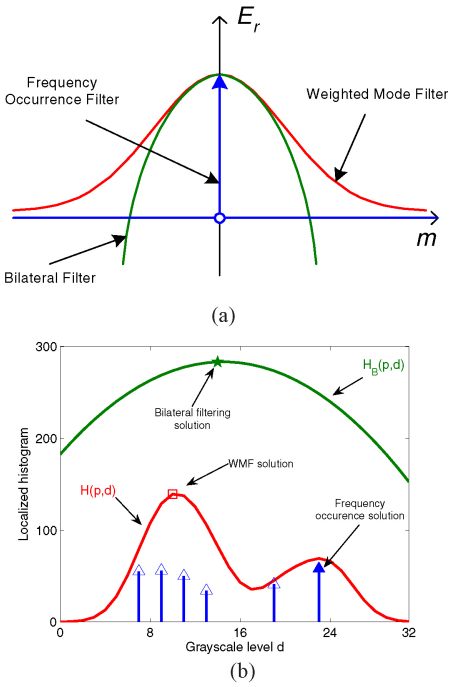



Fig. 1. Realization of various existing filtering approaches using different spreading functions in the localized histogram. (a) Spreading function $E_r(m)$. (b) Localized histograms and the final filtered solutions.

The Gaussian parameter $\sigma_r$ is determined by the amount of noise in the input data $f(p)$. For $\sigma_r \to 0$ the spreading function becomes the Dirac function and the most frequent occurrence value is found. In addition, using a Taylor series expansion of an exponential function, we can represent

$$G_r(d - f(q)) = \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!} \left( \frac{(d - f(q))^2}{2\sigma_r^2} \right)^n \qquad (10)$$

Consider the case when $\sigma_r \to \infty$. As $\lim_{\sigma_r \to \infty} \left( \frac{(d-f(q))^2}{2\sigma_r^2} \right)^n = 0$, by keeping only the two most significant terms in the Taylor series expansion, the localized histogram can be approximated as

$$H(p,d) \approx \sum_{q \in N(p)} w(p,q) \left\{ 1 - \frac{(d - f(q))^2}{2\sigma_r^2} \right\}. \qquad (11)$$

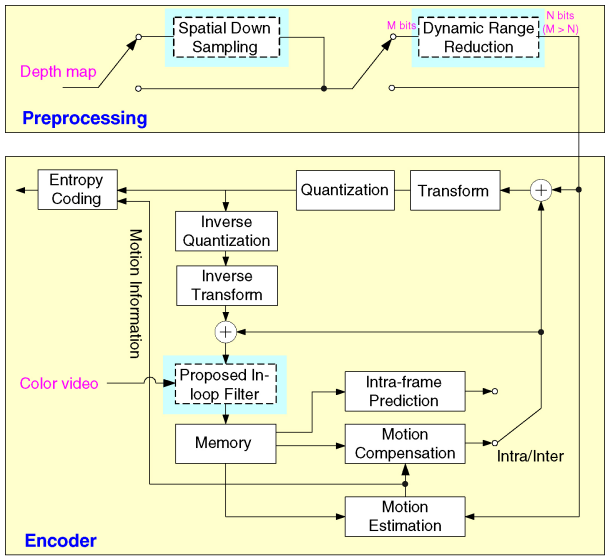
By setting $a = \frac{1}{2\sigma_r^2}$ and $C = 1$ in (5), the representation of $H(p,d)$ in (11) is in fact similar to $H_B(p,d)$ of (5). Thus, for $\sigma_r \to \infty$ the filtered solution of the bilateral filter, which can be obtained through (5) as mentioned in Section II-B, will be the same as that obtained by the weighted mode filtering through (2) and (11). In other words, the bilateral filter is a special case of the weighted mode filtering employing the Gaussian spreading function when the Gaussian parameter $\sigma_r \to \infty$.

For illustration, Fig. 1 shows the plots of different spreading functions and the localized histograms corresponding to various existing filtering approaches through a toy example. Intuitively, the main advantage of using a Gaussian function to model the spreading function is that the fall-off rate is higher than the quadratic function. In the context of the robust estimation, the Gaussian function is more robust and stable than the quadratic function. Specifically, the influence of the outliers is determined by the derivative of the given spreading function, which is called an influence function [22], [23]. The influence function $\Phi_B(m) = \nabla E_B(m)$ of the bilateral filter is $-2am$, which is linearly proportional to the distribution of the neighboring measurements, so that this function is very sensitive to the outliers. In contrast, an influence function $\Phi_M(m) = \nabla G_r(m) = \frac{m}{\sigma_r^2} e^{-\frac{m}{2\sigma_r^2}}$ of the weighted mode filter increases until $m$ ranges from 0 to $\sigma_r$, and then converges to 0 when $m \to \infty$. Therefore, the influence of the strong neighboring outliers is diminished faster by down-weighting, while the influence of the neighboring inliers where the magnitude of the local difference ($|d - f(q)|$) is smaller than $\sigma_r$ maintains. The spreading function $\delta_r(m)$ of the frequent occurrence filter in (4) can be interpreted as having the similar form to the spreading function $G_r$ of the weighted mode filter when $\sigma_r \to 0$. Thus, the influence function of the frequent occurrence filter always ignores the influence of the slightly corrupted inliers in the neighborhood, regardless of the magnitude of the local difference ($|d - f(q)|$). As a result, in depth compression the bilateral filter may cause blur on the depth boundaries due to the sensitivity to the outliers in the weighted summation operation; while the frequent occurrence filter may ignore the influence of the slightly corrupted inliers and attain less accurate and noise-dependent filtered depth values. This will not be able to result in sharp and artifact-free depth edges with a proper alignment with color video, which in turn will severely degrade the synthesized view quality as discussed in details later in Sections III and IV.

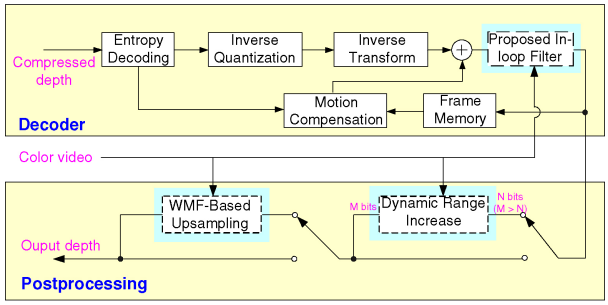
In this paper, the effectiveness and performance of the weighted mode filtering are verified by applying to the compressed depth map for the coding artifact suppression. Using the theoretical analysis in this section together with extensive experiments, we will show that the proposed weighted mode filtering based depth coding scheme has a superior performance in comparison with those of the frequent occurrence filter and the bilateral filter.

## III. Proposed Weighted Mode Filtering-Based Depth Coding

Fig. 2 shows the architecture of the proposed depth map encoder and decoder based on a typical transform-based motion compensated video codec. By using the framework of the conventional video coding standard, it is efficient to implement the proposed depth encoder and decoder by utilizing off-the-shelf components in practice. The encoder contains a preprocessing block that enables the spatial resolution and dynamic range reduction of depth signal, if necessary, for an efficient depth map compression. The motivation is that with an efficient upsampling algorithm, encoding the depth data on the reduced resolution and dynamic range can reduce the bit rate substantially while still achieving a good synthesized view quality. In addition, a novel WMF-based in-loop filter will be introduced to suppress the compression artifacts, especially on object boundaries, by taking the depth characteristics into




(a)




(b)

Fig. 2. Block diagrams of the proposed weighted mode filtering based depth map encoder and decoder using a typical transform-based motion compensated coding scheme. (a) Encoder. (b) Decoder.

account. For the decoding process, the WMF-based method is utilized to upsample the spatial resolution and the dynamic range of the decoded depth map, if necessary. In what follows, we present the three key components of our proposed depth map encoder and decoder: 1) WMF-based in-loop filter; 2) WMF-based spatial resolution upsampling; and 3) WMF-based dynamic range upsampling.

### A. In-Loop Filter

Containing homogeneous regions separated by sharp edges, transform-based compressed depth map often exhibits large coding artifacts such as ringing artifacts and blurriness along the depth boundaries. These artifacts in turn severely degrade the visual quality of the synthesized view. Fig. 3 shows the sample frames of the depth video compressed at different quantization parameters (QPs) and the corresponding synthesized view. Obviously, coding artifacts introduced in the compressed depth create many annoying visual artifacts in the virtual view, especially along the object boundaries.

Existing in-loop filters such as the H.264/AVC deblocking filter and Wiener filter [24], which are mainly designed for the color video, may not be suitable for the depth map compression with different characteristics. In this paper, the


Fig. 3. Sample frames of the depth video compressed at different QPs and the corresponding synthesized view.

weighted mode filtering concept is employed to design an in-loop edge-preserving denoising filter. In addition, we also extend the concept to use a guided function $g(p)$ different from the function $f(p)$ to be filtered in the weighting term as follows:

$$H(p,d) = \sum_{q \in N(p)} w(p,q) G_r(d - f(q)) \tag{12}$$

$$w(p,q) = G_g(g(p) - g(q)) G_f(f(p) - f(q)) G_s(p - q)$$

where two range kernels $G_g$ and $G_f$ are introduced here to measure a similarity between data of two pixels $p$ and $q$, $G_s$ is the spatial kernel to indicate the geometric closeness. By selecting such a weighting function, the weighted mode filtering is contextually similar to joint bilateral and trilateral filtering methods [26], [27], since the guided function $g(p)$ and the original function $f(p)$ are used to calculate the weighting value. Liu *et al.* has observed that the weighting functions $G_g$ and $G_f$ may still cause unnecessary blur on the depth boundaries due to its summation [19]. In contrast, by selecting the global mode on the localized histogram, it may help to reduce an unnecessary blur along the object boundaries observed in the case of the bilateral filtering. We will show the depth video coding based on the weighted mode filtering outperforms the existing postprocessing based coding methods.

In general, the compressed depth map is often transmitted together with the associate color video in order to synthesize the virtual view at the receiver side. In addition, two correlated depth pixels along the depth boundaries usually exhibit a strong photometric similarity in the corresponding video pixels. Inspired by this observation, we utilize the color video pixels $I(p)$ as the guided function $g(p)$ to denoise the depth data $D(p)$ of the original function $f(p)$. It should be noted that both color and depth video information can be used as guided information in the weighting term to measure the similarity of pixels $p$ and $q$. However, through extensive experiments it is observed that using the color video information only as guided information generally provides a better performance in comparison with incorporating the guided depth information in the weighting term. This can be explained by the fact that the

input depth map already contains more serious coding artifacts around the sharp edges than the color videos. Thus, using the noisy input depth to guide the noise filtering of its own signal may not be effective. In contrast, color frame consistently provides an effective guided information even when it is encoded heavily lossy as shown later in the experimental results. Furthermore, in view synthesis, the distortion in depth pixels will result in a rendering position error. By using the similarity of color pixels to guide the filtering process, it may diminish the quality degradation of the synthesized view due to the rendering position error.

In this paper, we shall only utilize the color video information as guided information in the proposed weighted mode filtering. Specifically, the localized histogram using guided color information can be formulated as

$$H_I(p,d) = \sum_{q \in N(p)} w_I(p,q) G_r(d - D(q)) \tag{13}$$

where the weighting term, $w_I$, incorporates the photometric similarity in the corresponding color video and is defined as

$$w_I(p,q) = G_I(I(p) - I(q)) G_s(p - q) \tag{14}$$

The range filter of the color video, $G_I$, is chosen as a Gaussian filter. For a fast computation, look-up tables may be utilized for approximating float values of Gaussian filters. Note that the "bell" width of the spreading function $G_r$ is controlled by the filter parameter $\sigma_r$. To reduce the complexity, when the localized histogram is computed using (13), only the $d^{th}$ bin satisfying $|d - D(q)| \leq B$ should be updated for each neighboring pixel $q$, where the threshold $B$ is determined to meet the condition $G_r(B/2) = 0.3$.

As mentioned in Section II, the width ($\leftrightarrow \sigma_r$) is determined by the amount of noise in the input data $f(p)$. In addition, the larger the value of $\sigma_r$ is, the higher the computational complexity is. Furthermore, too large value of $\sigma_r$ may cause blur on the object boundaries due to the fact that the weighted mode filtering approaches to the solution of the bilateral filtering as mentioned in Section II. To select an optimal value of $\sigma_r$, we compressed the depth video at different QPs by using the proposed in-loop filter in the encoder with different values of $\sigma_r$. An objective performance is measured indirectly by analyzing the quality of the synthesized view (refer to Section IV for the simulation setup details). Fig. 4 shows the peak-signal-to-noise ratio (PSNR) of the synthesized view obtained by using different values of $\sigma_r$. The results show that with different amounts of noise introduced by the quantization artifact, setting $\sigma_r$ to 3 generally provides the best synthesized view quality while maintaining low computational complexity.

Meanwhile, serious coding artifacts generally appear around sharp edges. Thus, it may be more efficient to detect and apply the proposed in-loop filter only to these regions for an efficient implementation. Basically, it is adequate to determine regions containing strong edges rather than accurate edges in the depth video. Hence, we propose to use a simple first-order method to detect the depth discontinuities by calculating an image gradient. Specifically, the estimates of first-order derivatives, $D_x$ and $D_y$, and the gradient magnitude of a pixel in the depth

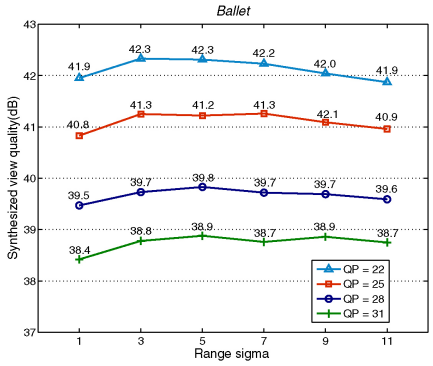



Fig. 4. PSNR (dB) results of the synthesized view obtained by encoding the depth video at different QPs using the proposed in-loop filter with different values of range sigma $\sigma_r$.

(a) (b)

Fig. 5. Edge maps obtained from the depth map of the *Ballet* test sequence based on the classified (a) edge pixels and (b) edge blocks.

map $D$ are computed as

$$\begin{aligned} D_x(m,n) &= D(m,n+1) - D(m,n-1) \\ D_y(m,n) &= D(m+1,n) - D(m-1,n) \\ |\nabla D(m,n)| &= \sqrt{D_x(m,n)^2 + D_y(m,n)^2}. \end{aligned} \tag{15}$$

Each pixel of the depth map is then classified as an edge pixel if the gradient magnitude $|\nabla D(m,n)|$ is greater than a certain threshold. We then partition the depth map into nonoverlapping blocks of $N \times N$ pixels. A block is classified as an edge block if there exists at least a certain number of edge pixels in the block. The proposed in-loop filter is then applied only for the pixels in these edge blocks to reduce the computational complexity.

Fig. 5 shows the classification of edge pixels and edge blocks obtained by the proposed method for a depth image of the *Ballet* test sequence. Conceivably, the computational complexity gain would likely depend on the size of partitioned blocks. Intuitively, partitioning the depth map into blocks of smaller size would result in less total number of pixels inside the classified edge blocks that need to be filtered. Thus, it is expected to achieve a higher gain in complexity reduction, but may reduce the noise removal performance. Note that the smallest transform block size in the conventional video coding standard is a $4 \times 4$ integer transform in H.264/AVC, thus the size of partitioned blocks should not be less than 4. In our simulation, different partitioned block sizes are used to select an optimal trade-off between the complexity gain and the effectiveness of artifact suppression.

### *B. Depth Down/Upsampling*

It has been shown that a downsampled video when compressed and later upsampled, visually beats the video compressed directly on high resolution at a certain target bit rate [28], [29]. Based on this observation, many have proposed to encode a resolution-reduced depth video to reduce the bit rate substantially [14]–[16]. However, down/upsampling process also introduces the serious distortion as some high frequency information is discarded. Without a proper down/upsampling scheme, important depth information in the object boundary regions will be distorted and affect the visual quality of the synthesized view. In this section, we show how to employ the filtering scheme proposed in Section III-A to upsample the decoded depth map while recovering the original depth edge information.

Depth Downsampling: Traditional downsampling filter, consisting of a low-pass filter and an interpolation filter, will smooth the sharp edges in depth map. Here, we employ a simple median downsampling filter proposed in [15] as

$$D_{\text{down}}(p) = \text{median}(W_{s\times s}) \tag{16}$$

where $s$ is a downsampling factor and each pixel value in the downsampled depth is selected as the median value of the corresponding block $W_{s\times s}$ of size $s \times s$ in the original depth.

Depth Upsampling: The proposed weighted mode filtering is tailored to upsample the decoded depth video. Coarse-to-fine upsampling approach proposed in our previous work for the depth superresolution task is employed here [20]. The advantage of the multiscale upsampling approach is to prevent an aliasing effect in the final depth map. For simplicity, let $s = 2^K$ and the upsampling will be performed in $K$ steps. Initially, only pixels at position $p \mid p\%2^K = 0$ in the upsampled depth will be initialized from the downsampled depth as

$$D_{\text{up}}(p \mid p\%2^K = 0) = D_{\text{down}}(p/2^K). \tag{17}$$

The other missing pixels will be gradually initialized and refined in every step by applying the proposed weighted mode filtering scheme. Specifically, at step $0 \le k \le K-1$, pixels at positions $p \mid p\%2^k = 0$ will be updated. Note that only pixel $q$ in the neighboring pixels $N(p)$ that is initialized in the previous steps will be re-used to compute the localized histogram. In addition, the size of the neighboring window $N(p)$ will be reduced by half in each step.

### *C. Depth Dynamic Range Reduction*

For the dynamic range of the depth map, we propose to design a similar approach to the spatial down/upsampling method to further reduce the encoding bit rate. The basic idea is to first reduce the number of bits per depth sample prior to encoding and then reconstruct the original dynamic range after decoding. The main motivation is that the depth map consists of less texture details compared with the color video, enabling us to reconstruct efficiently the original dynamic range of the depth map from the lower dynamic range data. In addition, there exists a close relationship among depth sample, camera baseline, and image resolution. For instance, small number of bits per sample is sufficient to represent the depth map well at low spatial resolution and provide a seamless synthesized view [33]. As dynamic range reduction will severely reduce the depth precision, an appropriate upscaling technique is required to retain the important depth information (e.g., precision on the depth edge) without spreading the coding artifacts from the compressed range-reduced depth map.

Considering the above, we design a new down/upscaling approach for the dynamic range compression. In the downscaling process, a simple linear tone mapping is employed to reduce the number of bits per depth sample prior to encoding as shown in the preprocessing block of Fig. 2. The new depth sample value is computed as

$$D_{\text{N\_bits}}(p) = \left\lfloor \frac{D_{\text{M\_bits}}(p)}{2^{M-N}} \right\rfloor \tag{18}$$

where $\lfloor\cdot\rfloor$ denotes the floor operator, $M$ and $N$ specify the original and new number of bits per depth sample, respectively. In our simulation, $M$ is equal to 8.

The upscaling process consists of two parts: an initialization of the downscaled depth map and the weighted mode filtering based upscaling scheme. At first, a linear inverse tone mapping is used to obtain an initial depth map with the original dynamic range as follows:

$$D^{\text{rec}}_{\text{M\_bits}}(p) = D^{\text{rec}}_{\text{N\_bits}}(p) * 2^{M-N}. \tag{19}$$

The weighted mode filtering is then applied to reconstruct a final solution $\hat{f}(p)$ with the original dynamic range. By using the guided color information, the weighted mode filtering may suppress the distortion from the dynamic range down/upscaling process by filtering the upscaling depth value based on the neighborhood information without degrading much the synthesized view quality. In addition, the weighted mode filtering will also reduce the spread of any coding artifacts in the compressed range-reduced depth map into the reconstructed depth map at the original dynamic range. Note that since less information is presented in the low dynamic range, we also apply the upscaling process in the multiscale manner. Specifically, we increase the number of bits per depth sample by only 1 in each step and apply the proposed weighted mode filtering. For example, we obtain 7-bit depth data in an immediate step when reconstructing 8-bit depth from 6-bit depth.

## IV. Experimental Results

We have conducted a series of experiments to evaluate the performance of the proposed depth compression techniques. We have tested with the *Breakdancers* and *Ballet* test sequences with resolutions of $1024 \times 768$, of which both the color video and depth map are provided from Microsoft Research [30].

The experiments were conducted by using the H.264/AVC Joint Model Reference Software JM17.2 to encode the depth map of each view independently [31]. The conventional H.264/AVC deblocking filter in the reference software was replaced with the proposed in-loop filter. For each test sequence, we encoded two (left and right) views for both color and depth videos using various QPs.

To measure the performance of the proposed method, we analyzed the quality of the color information for the synthesized intermediate view. Among 8 views, view 3 and view 5 were selected as reference views and a virtual view 4 was generated using the View Synthesis Reference Software (VSRS) 3.0 provided by MPEG [32]. For an objective comparison, the PSNR of each virtual view generated using compressed depth maps was computed with respect to that generated using the original depth map. Rate distortion (RD) curves were obtained by the total bit rate required to encode the depth maps of both reference views and the PSNR of the synthesized view. Note that the captured original view at the synthesis position was not used here as a reference to measure the quality of the virtual view since such a measure generally incorporates more than one source of distortion. For instance, there are a lot of distortion sources such as slightly different camera response function, exposure time, and lighting condition among multi-view images. It has also been shown in [34] that the distortion introduced by VSRS widely masks those due to depth map compression, which can result in a misleading study in order to justify the effectiveness of depth compression.

### A. In-Loop Filter

In the first set of experiments, we evaluated the performance of the proposed in-loop filter in comparison with the existing in-loop filters. Besides the conventional H.264/AVC deblocking filter, we have also compared with the depth boundary reconstruction filter [17] and the trilateral filter [19], which are also utilized as the in-loop filter. In addition, to evaluate the effectiveness of using the weighted mode filtering as an in-loop filter, we have also implemented it as an out-loop filter, in which the weighted mode filtering is applied to the decoded depth map at the receiver side. Note that the H.264/AVC deblocking filter was completely replaced by the proposed in-loop filter in our methods. Fig. 6 shows RD curves obtained by the proposed and existing in-loop filters for the two test sequences. The depth bit rates and PSNR results are shown in Tables I and II. The results show the superior performance of the proposed filter compared with the existing filters. Specifically, by using the proposed filter, we achieved about 0.8-dB and 0.5-dB improvement in PSNR of the synthesized view quality in terms of average Bjontegaard metric [35] compared with that of the existing filters for the *Ballet* and *Breakdancers* sequences, respectively. Not surprisingly, having used the guided color information and a more stable spreading function, the proposed filter outperformed the boundary reconstruction filter, which only employs a noise dependent frequent occurrence filter and a noisy depth signal to guide the filtering process. Meanwhile, using the weighted mode filter as an out-loop filter achieved a slightly inferior performance compared with that of using as an in-loop filter. This is because using the weighted mode filter as an in-loop filter will suppress the coding artifacts of the reconstructed frame and result in a better reference frame, which will reduce the required bit rate to code the future frames. In addition, the proposed filter not only achieved objective improvement, but also obtained a better visual quality of the synthesized view. Fig. 7 shows the sample frames of the filtered depth video

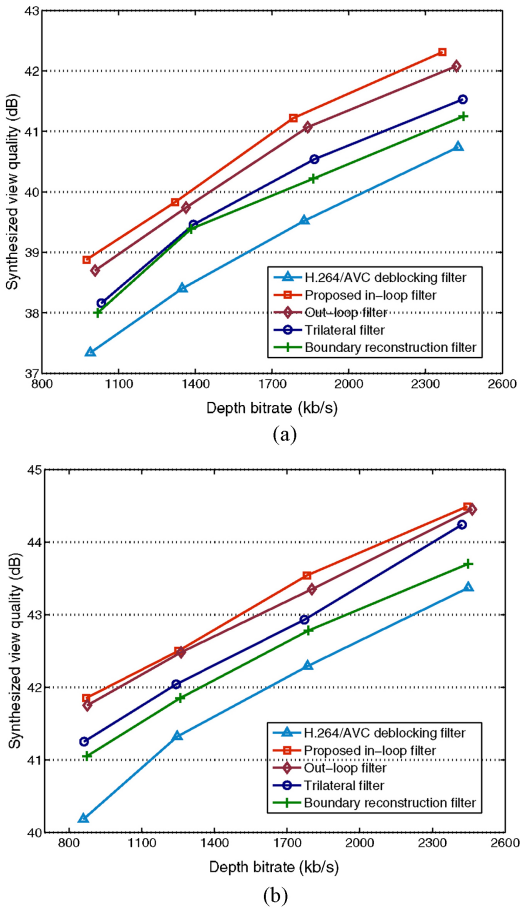



Fig. 6. RD curves obtained by encoding the depth maps using the proposed and existing in-loop filters. (a) *Ballet*. (b) *Breakdancers*.

and the corresponding synthesized view. The figures show that the proposed filter efficiently suppressed the coding artifacts around the object boundaries, which resulted in a better visual quality of the virtual view in comparison with the existing filters.

In another set of experiments, the proposed in-loop filter was used together with the edge region detection to speed up the filtering process. Fig. 8 shows the RD curves obtained by applying the proposed in-loop filter only in the edge regions, which are detected by using different sizes of edge blocks. Table III shows the complexity comparison in term of the average number of $4 \times 4$ blocks per frame that need to be filtered. Not surprisingly, using the smaller size of edge block obtained higher gain in the complexity reduction at the cost of degrading the virtual view quality. However, it was also evident that incurring only marginal quality degradation, the edge block size of 8 reduced the complexity by a factor of 9. In other words, the edge block size of 8 provided the best trade-off between the computational complexity and the virtual view quality.

To further evaluate the performance of the proposed in-loop filter, we examined the effect of using lossy color videos as guided information. In particular, we considered different input sets of the compressed color videos, which have different levels of lossy quality. These input sets were obtained by

TABLE I

EXPERIMENTAL RESULTS OBTAINED BY THE PROPOSED AND EXISTING IN-LOOP FILTERS FOR THE *Ballet* SEQUENCE

| | Depth bitrate (kbps) | | | | | Synthesized view quality (dB) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| QP | H.264/AVC | Proposed filter | Outloop filter | Trilateral filter | Boundary rec. filter | H.264/AVC | Proposed filter | Outloop filter | Trilateral filter | Boundary rec. filter |
| 22 | 2426.71 | 2365.12 | 2420.70 | 2445.67 | 2447.86 | 40.74 | 42.31 | 42.08 | 41.53 | 41.25 |
| 25 | 1824.46 | 1782.77 | 1838.90 | 1865.29 | 1861.42 | 39.52 | 41.22 | 41.07 | 40.54 | 40.22 |
| 28 | 1347.74 | 1320.48 | 1363.01 | 1392.29 | 1383.34 | 38.40 | 39.83 | 39.74 | 39.46 | 39.39 |
| 31 | 988.88 | 973.91 | 1007.58 | 1032.24 | 1017.81 | 37.34 | 38.88 | 38.70 | 38.16 | 38.00 |

TABLE II

EXPERIMENTAL RESULTS OBTAINED BY THE PROPOSED AND EXISTING IN-LOOP FILTERS FOR THE *Breakdancers* SEQUENCE

| | Depth bitrate (kbps) | | | | | Synthesized view quality (dB) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| QP | H.264/AVC | Proposed filter | Outloop filter | Trilateral filter | Boundary rec. filter | H.264/AVC | Proposed filter | Outloop filter | Trilateral filter | Boundary rec. filter |
| 22 | 2447.70 | 2443.59 | 2464.16 | 2422.07 | 2447.56 | 43.37 | 44.49 | 44.45 | 44.24 | 43.70 |
| 25 | 1784.42 | 1781.85 | 1802.18 | 1771.50 | 1787.87 | 42.29 | 43.54 | 43.35 | 42.93 | 42.78 |
| 28 | 1246.30 | 1251.10 | 1262.66 | 1242.03 | 1258.64 | 41.32 | 42.50 | 42.48 | 42.04 | 41.85 |
| 31 | 859.04 | 870.89 | 875.20 | 861.89 | 874.10 | 40.18 | 41.85 | 41.75 | 41.25 | 41.05 |

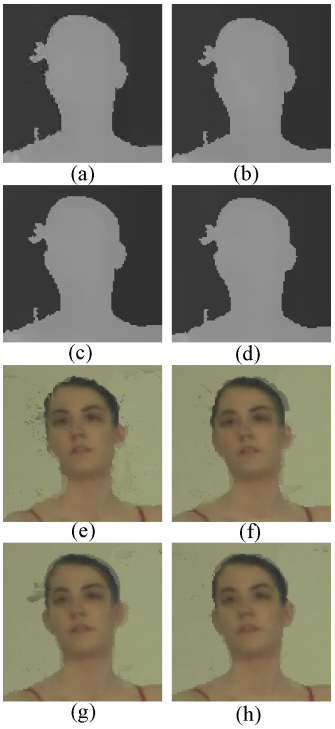



Fig. 7. Sample frames of the reconstructed depth map and rendered view for the *Ballet* sequence obtained by different in-loop filters: (a) H.264/AVC deblocking filter, (b) boundary reconstruction filter, (c) trilateral filter, (d) proposed in-loop filter, (e) synthesized image from (a), (f) synthesized image from (b), (g) synthesized image from (c), and (h) synthesized image from (d).

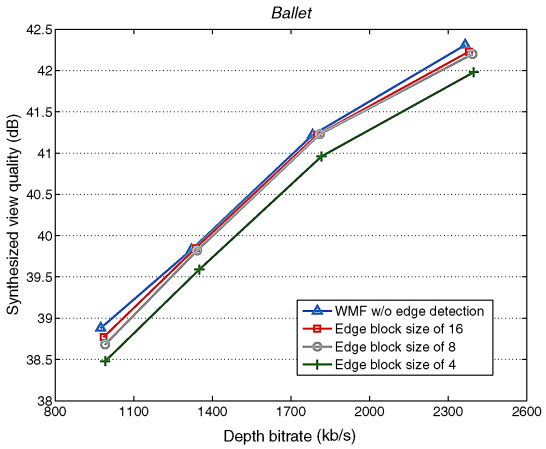



Fig. 8. RD curves obtained by applying the proposed in-loop filter to only the edge regions detected by using different sizes of edge blocks for the *Ballet* test sequence.

encoding the original color videos at different QPs of 22 and 31. Table IV shows the depth bit rates and PSNR results of the synthesized view obtained by the proposed in-loop filter and the H.264/AVC deblocking filter for different color video input sets. The results show that the proposed in-loop filter always achieved a better synthesized view quality compared to the H.264/AVC deblocking filter even in the case the guided color videos were compressed at high QP. Specifically, for the *Ballet* test sequence we achieved about 1.64 dB and 1.67 dB PSNR gains in terms of average Bjontegaard metric for the color videos compressed at QPs of 22 and 31, respectively. Meanwhile using the color input sets compressed at QPs of 22 and 31 provided about 1.26 dB and 1.06 dB improvement in PSNR, respectively, for the *Breakdancers* sequence. Intuitively, one would expect the PSNR gain will likely be reduced in the case the color videos are heavily compressed. However, it is observed that the PSNR improvement is still significant when using heavily lossy color information. In other words, color frame consistently provides an effective guided information even when it is encoded heavily lossy.

### *B. Depth Down/Upsampling*

In this set of experiments, the input depth maps were downsampled by a factor of 2 in both horizontal and vertical directions. The resolution-reduced depth videos were encoded

TABLE III
COMPLEXITY COMPARISON BY USING DIFFERENT SIZES OF EDGE BLOCKS TO DETECT EDGE REGIONS

| | Average number of 4 × 4 filterred blocks per frame | | | |
|---|---|---|---|---|
| QP | W/o edge detection | Edge block size = 16 | Edge block size = 8 | Edge block size = 4 |
| 22 | 49 152 | 11 080 | 5580 | 2610 |
| 25 | 49 152 | 10 816 | 5396 | 2512 |
| 28 | 49 152 | 10 496 | 5176 | 2396 |
| 31 | 49 152 | 10 168 | 5014 | 2321 |
| Average | 49 152 | 10 640 | 5291 | 2460 |
| Gain | 1 | 4.62 | 9.29 | 19.98 |

without enabling the proposed in-loop filter. In addition, we have also implemented and compared with the existing depth upsampling filters [15], [16], [26], [27], [36]. Fig. 9 shows the RD curves obtained by different upsampling filters and the regular depth coding at the original resolution. The results show that the proposed down/upsampling approach outperformed the regular depth coding over a wide scope of bit rates. The significant performance gain is due to two main reasons. One, encoding a resolution-reduced depth video reduced a bit rate substantially. Two, having employed the weighted mode filtering with the guided color information, which was also used in the view synthesis, the proposed method not only preserved the important depth information for rendering virtual view, but also suppressed coding artifacts along depth edges. Furthermore, in comparison with the existing upsampling filters, the proposed upsampling filter also achieved more than 1-dB PSNR gain of the virtual view in terms of average Bjontegaard metric.

As mentioned in Section II, by using a Gaussian function as the spreading function together with a mode operation, the proposed upsampling filter suppressed the boundary artifacts efficiently and obtained sharp depth edges. In contrast, based on the frequent occurrence filtering [15] and the joint bilateral filtering [26], [27], [36], the existing filters are less robust and stable in coding artifact suppression and may cause unnecessary blur on the depth boundaries (see Section II-C). Meanwhile, [16] lacks the adaptive aggregation using multiple inlier pixels in the neighborhood and may not handle the coding artifacts along the depth boundaries efficiently. Such blur and artifacts along depth edges are undesirable in the reconstructed depth map as it will result in serious disparity errors and false edges in the synthesized view. Note that besides the depth boundary artifacts, the misalignment between color and depth boundaries also introduces serious synthesized artifacts in the virtual view. By using the guided color information in an efficient manner, the proposed upsampling approach not only maintained the sharp depth boundaries, but also ensured the edge alignment between color and depth videos. As a result, the fidelity of the synthesized view obtained by using the reconstructed depths from the proposed approach is closer to that of the reference synthesized view, which results in a lower distortion compared with the existing approaches.

To further justify the performance of the down/upsampling approach, which may be influenced by the limited spatial resolution of the original coding materials, we have conducted another experiment using the *Undo-Dancer* test sequence [37] with a higher spatial resolution of 1920 × 1088. The experimental results are shown in Fig. 10. As can be seen from the figure, the results are consistent with that observed in the *Ballet* and *Breakdancers* sequences, which indicates the

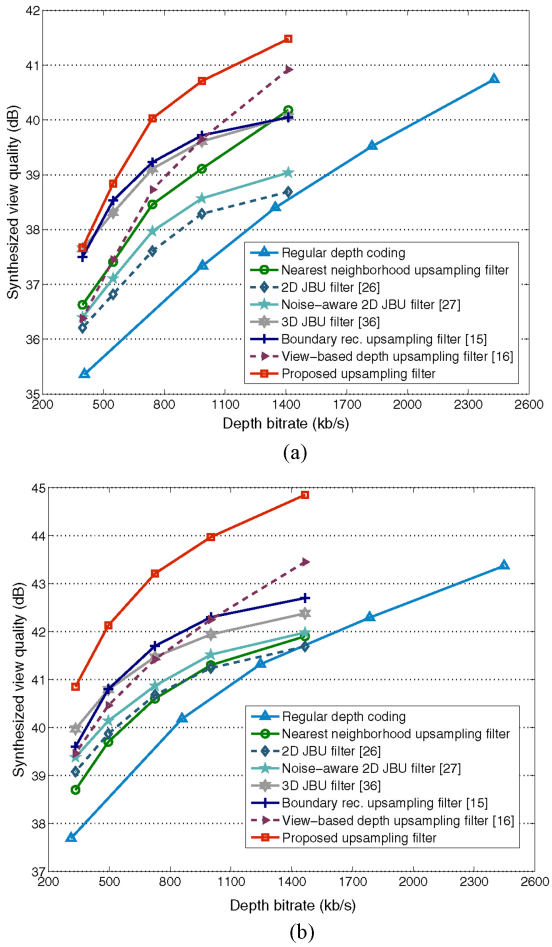



Fig. 9. RD curves obtained by reconstructing the decoded depth video using different upsampling filters and the regular depth coding at the original resolution for the *Ballet* and *Breakdancers* sequences. (a) *Ballet*. (b) *Breakdancers*.

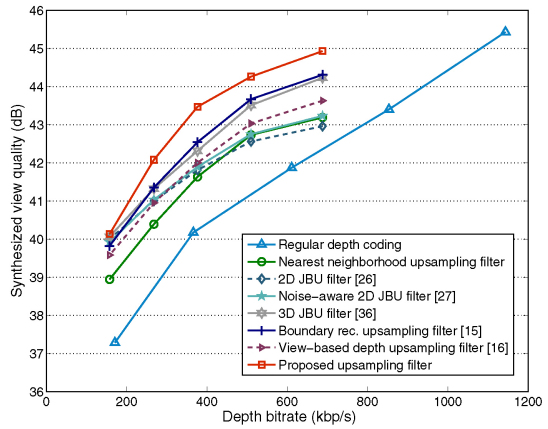



Fig. 10. RD curves obtained by reconstructing the decoded depth video using different upsampling filters and the regular depth coding at the original resolution for the *Undo-Dancer* sequence.

TABLE IV
EXPERIMENTAL RESULTS OBTAINED BY USING LOSSY COLOR VIDEOS WITH DIFFERENT QUALITIES

(a) *Ballet*

| | Color videos compressed at QP = 22 | | | | Color videos compressed at QP = 31 | | | |
|---|---|---|---|---|---|---|---|---|
| | Depth bitrate (kbps) | | Synthesized view (dB) | | Depth bitrate (kbps) | | Synthesized view (dB) | |
| QP | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter |
| 22 | 2426.71 | 2365.12 | 40.74 | 42.31 | 2426.71 | 2391.73 | 41.12 | 42.59 |
| 25 | 1824.46 | 1782.77 | 39.52 | 41.22 | 1824.46 | 1810.24 | 39.89 | 41.65 |
| 28 | 1347.74 | 1320.48 | 38.40 | 39.83 | 1347.74 | 1340.92 | 38.67 | 40.32 |
| 31 | 988.88 | 973.91 | 37.34 | 38.88 | 988.88 | 995.73 | 37.65 | 39.13 |
| | Average Bjontegaard PSNR gain: 1.64 dB | | | | Average Bjontegaard PSNR gain: 1.67 dB | | | |

(b) *Breakdancers*

| | Color videos compressed at QP = 22 | | | | Color videos compressed at QP = 31 | | | |
|---|---|---|---|---|---|---|---|---|
| | Depth bitrate (kbps) | | Synthesized view (dB) | | Depth bitrate (kbps) | | Synthesized view (dB) | |
| QP | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter | H.264/AVC filter | Proposed filter |
| 22 | 2447.70 | 2443.59 | 43.37 | 44.49 | 2447.70 | 2429.38 | 44.00 | 44.89 |
| 25 | 1784.42 | 1781.85 | 42.29 | 43.54 | 1784.42 | 1778.27 | 42.87 | 44.08 |
| 28 | 1246.30 | 1251.10 | 41.32 | 42.50 | 1246.30 | 1250.37 | 41.86 | 42.79 |
| 31 | 859.04 | 870.89 | 40.18 | 41.85 | 859.04 | 869.30 | 40.74 | 41.96 |
| | Average Bjontegaard PSNR gain: 1.26 dB | | | | Average Bjontegaard PSNR gain: 1.06 dB | | | |

better performance of the proposed method compared with the existing methods.

For visualization, Fig. 11 shows the sample frames of the depth video obtained by the different upsampling filters and the regular depth coding. The figures show that the proposed upsampling filter reconstructed efficiently the depth map at the original resolution. Being able to preserve and recover the depth edge information, we achieved a better visual quality of the synthesized view.

### C. Depth Dynamic Range Reduction

In the evaluation of the down/upscaling approach for the dynamic range, the original bit depths of the depth video were reduced to 6 bits and 7 bits without spatial downsampling and encoded without enabling the proposed in-loop filter. After decoding, the proposed depth dynamic range upscaling was used to reconstruct the original dynamic range of depth data. Fig. 12 shows the RD curves obtained by this approach with different number of bits per sample and the regular depth coding using original depth data. Not surprisingly, by encoding the depth video with the lower dynamic range, the total depth bit rate was reduced notably. Furthermore, with the effectiveness of the proposed filter, we reconstructed the depth video at the original dynamic range well, which is reflected by the high-quality virtual view. Specifically, by using the proposed approach with the 7-bit depth video, we achieved a bit rate savings of approximately 27.8% and 46.1% in terms of average Bjontegaard metric for the *Ballet* and *Breakdancers* sequences, respectively, while retaining the same virtual view quality. However, the 6-bit depth video only provided modest PSNR improvement compared with the 7-bit depth video over a narrow bit rate range (i.e., from 200 to 300 kbps). It is due to the fact that too much information is discarded during the tone mapping from 8 bits to 6 bits. Thus, even with the guided information from the color video in the weighted mode filtering, it was not sufficient to recover some of important depth details. Fig. 13 shows that the proposed method with dynamic range compression provided a better visual quality of the virtual view than the regular depth coding at the same bit rate.

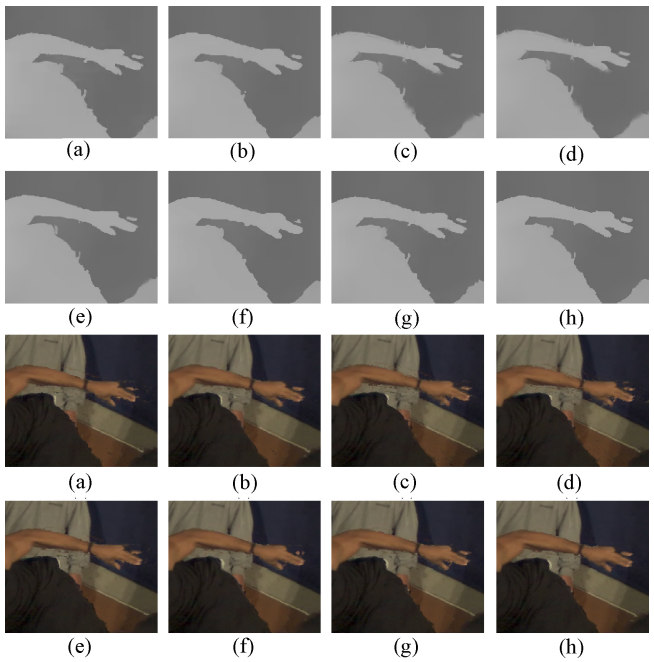



Fig. 11. Sample frames of the reconstructed depth map and rendered view for the *Breakdancers* test sequence. (a) Regular depth coding. (b) Nearest neighborhood upsampling filter. (c) 2D JBU filter [26]. (d) Noise-aware 2D JBU filter [27]. (e) 3-D JBU filter [36]. (f) Depth upsampling boundary reconstruction filter [15]. (g) View-based depth upsampling filter [16] and (h) Proposed upsampling filter.

### D. Down/Upsampling with Dynamic Range Reduction

In the last set of experiments, we analyzed the performance of the down/upsampling approach when used together with

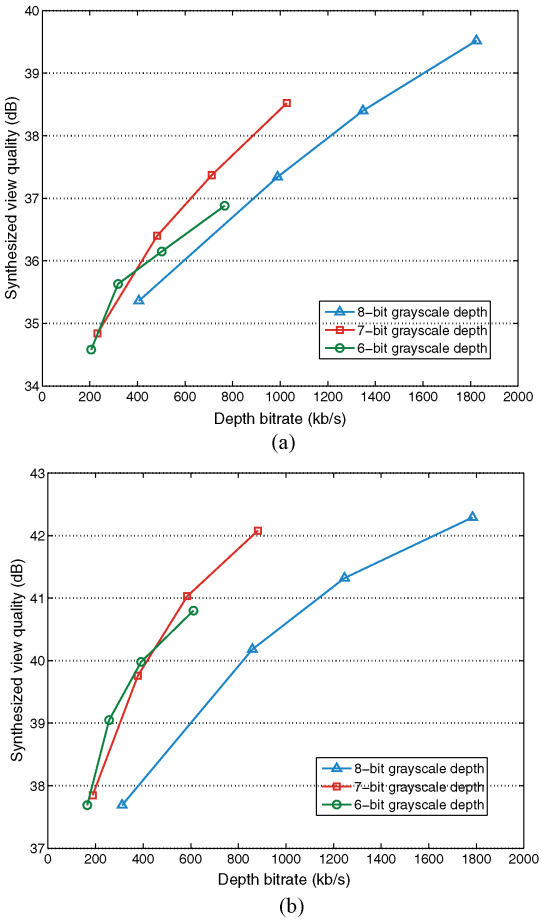



Fig. 12. RD curves obtained by encoding the reduced dynamic range of depth data and reconstructing the decoded depth video at the original dynamic range using the proposed filter and the regular depth coding. (a) *Ballet*. (b) *Breakdancers*.

the dynamic range reduction. Specifically, the depth videos were spatially downsampled by a factor of 2 and followed by the dynamic range reduction process to 7 bits per sample prior to encoding. In addition, we also evaluated the performance of the proposed down/upsampling approaches for the spatial resolution and dynamic range reduction when used together with the proposed in-loop filter. Fig. 14 shows the RD curves obtained by encoding the depth videos with different approaches in comparison with the regular H.264/AVC depth coding. The figures show that enabling the proposed in-loop filter could not provide a significant performance improvement in the proposed down/upsampling approaches for the spatial resolution and dynamic range reduction. This is because the upsampling stages of the spatial resolution and the dynamic depth range also utilize the weighted mode filtering with the guided color information. Thus, it can also suppress the coding artifacts while reconstructing the original spatial resolution and dynamic depth range. In our view, the additional memory and computation required for the proposed in-loop filter cannot justify for such an incremental performance in these cases. Thus, we propose to disable the proposed in-loop filter in the proposed down/upsampling approaches for the spatial resolution and dynamic range reduction. This also makes the

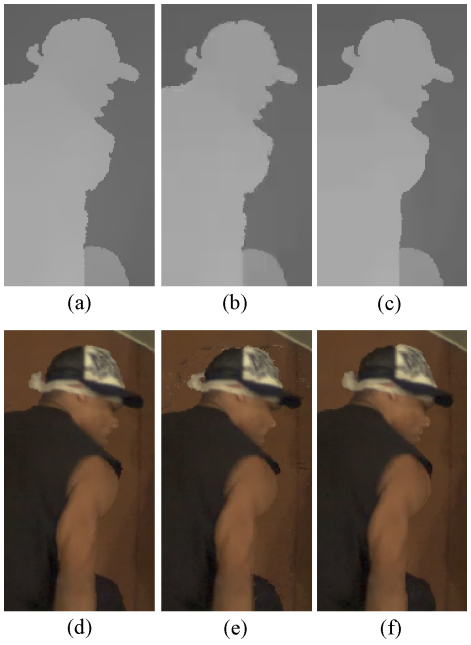



Fig. 13. Sample frames of the reconstructed depth map and rendered view for the *Breakdancers* sequence at the same bit rate of 880 kb/s. (a) Uncompressed depth map. (b) Regular depth coding. (c) 7-bit depth video. (d) Synthesized image from (a). (e) Synthesized image from (b), and (f) synthesized image from (c).

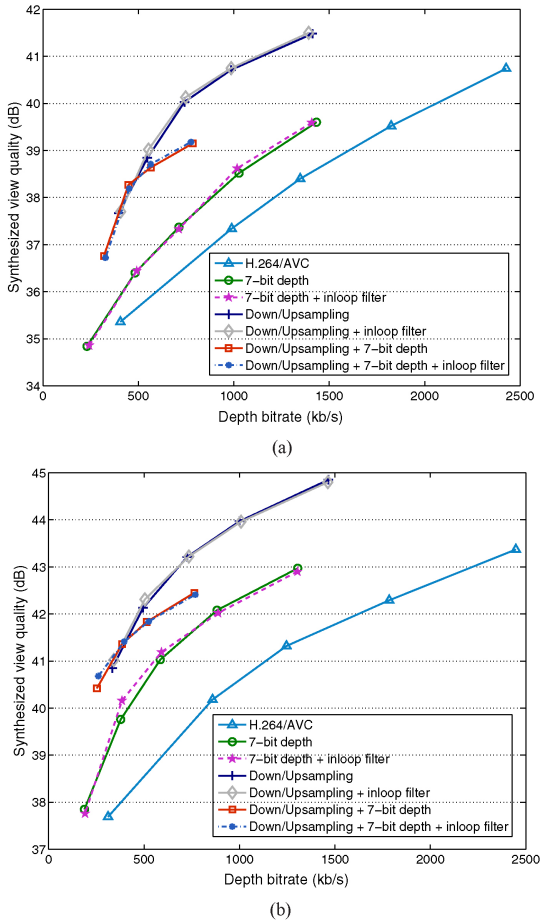



Fig. 14. RD curves obtained by encoding the depth videos using different proposed approaches in comparison with the regular H.264/AVC depth coding. (a) *Ballet*. (b) *Breakdancers*.

encoder/decoder block standard-compliant to easily utilize off-the-shelf components for the implementation of the proposed depth encoder/decoder.

In addition, the results show that the spatial down/upsampling approach outperformed the depth dynamic range reduction approach in both test sequences. Interestingly, combining the down/upsampling approach with the dynamic range reduction generally could not provide a better performance than using only the down/upsampling approach. Obviously, too much information is lost during the tone mapping and downsampling processes and makes it inefficient to reconstruct high quality of upsampled depth videos.

## V. Conclusion

We presented novel techniques to compress the depth video by taking into account the coding artifacts, the spatial resolution, and the dynamic range of depth data. Specifically, an efficient postprocessing method was proposed to suppress the coding artifacts based on the weighted mode filtering and utilized as an in-loop filter. We also presented the spatial resolution sampling and the dynamic range compression to reduce the coding bit rate. The novelty of the proposed approach comes from the efficiency of the proposed upsampling filters, which has been tailored from the in-loop filter based on the weighted mode filtering. The experimental results showed the superior performance of the proposed filters compared with the existing filters. The proposed filters can efficiently suppress the coding artifacts in the depth map as well as recover depth edge information from the reduced resolution and the low dynamic range. As a result, and incurring much lower coding bit rate, we can achieve the same quality of the synthesized view.

## Acknowledgment

The authors would like to thank the Associate Editor and the reviewers for their thoughtful comments and suggestions that helped improve the quality of this paper.

## References

[1] D. Min, D. Kim, S. Yun, and K. Sohn, "2D/3D freeview video generation for 3DTV system," *Signal Process: Image Commun.*, vol. 24, no. 1–2, pp. 31–48, 2009.

[2] MPEG document, N9760, Text of ISO/IEC 14496-10:2008/FDAM 1 "Multiview Video Coding," Oct. 2008, Busan, Korea.

[3] 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. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1606–1621, Nov. 2007.

[4] MPEG document, N9992, "Results of 3-D Video Expert Viewing," Jul. 2008, Hannover, Germany.

[5] MPEG document, w11061, "Applications and requirements on 3-D video coding," MPEG, Xi'an, China, Oct. 2009.

[6] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process.: Image Commun.*, vol. 22, no. 2, pp. 217–234, 2007.

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

[8] Y. Morvan, P. With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," *Proc. SPIE, Stereoscopic Displays Appl.*, vol. 6055, pp. 93–100, 2006.

[9] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Commun.*, vol. 24, no. 1–2, pp. 73–88, 2009.

[10] M. Maitre and M. N. Do, "Joint encoding of the depth image based representation using shape-adaptive wavelets," in *Proc. IEEE ICIP*, Oct. 2008.

[11] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *Proc. IEEE ICIP*, Sep. 2010, pp. 3393–3396.

[12] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. IEEE ICIP*, Nov. 2009, pp. 721–724.

[13] P. Lai, A. Ortega, C. C. Dorea, P. Yin, and C. Gomila, "Improving view rendering quality and coding efficiency by suppressing compression artifacts in depth-image coding," in *Proc. SPIE VCIP*, 2009.

[14] E. Ekmekcioglu, M. Mrak, S. Worrall, and A. Kondoz, "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering," in *Proc. IEEE ICIP*, Nov. 2009, pp. 733–736.

[15] K. J. Oh, S. Yea, A. Vetro, and Y. S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-D video," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 1–4, Sep. 2009.

[16] M. O. Wildeboer, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Depth up-sampling for depth coding using view information," in *Proc. 3DTV-CON*, May 2011, pp. 1–4.

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

[18] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter for depth coding," *IEEE Electron. Lett.*, vol. 45, no. 6, pp. 305–306, Mar. 2009.

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

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

[21] J. Weijer and R. Boomgaard, "Local mode filtering," in *IEEE Proc. Comput. Vision Pattern Recognit.*, vol. 2, Dec. 2001, pp.428–433.

[22] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*," New York: Wiley, 1986.

[23] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, Mar. 1998.

[24] T. Watanabe, N. Wada, G. Yasuda, A. Tanizawa, T. Chujoh, and T. Yamakage, "In-loop filter using block-based filter control for video coding," in *Proc. IEEE ICIP*, Nov. 2009, pp. 1013–1016.

[25] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[26] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *Proc. ACM SIGGRAPH*, vol. 26, no. 3, Aug. 2007.

[27] D. Cham, H. Buisman, C. Theobalt, and S. Thrun, "A noise-aware filter for real-time depth upsampling," in *Proc. Workshop MMSFAA*, 2008, pp. 1–12.

[28] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-scaling for better transform compression," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1145, Sep. 2003.

[29] V. A. Nguyen, W. Lin, and Y. P. Tan, "Downsampling/upsampling for better video compression at low bit rate," in *Proc. IEEE ISCAS*, May 2008, pp. 1–4.

[30] *MSR 3-D Video Sequences* [Online]. Available: http://www.research.microsoft.com/vision/ImageBasedRealitites/3DVideoDownload.

[31] JM Reference Software Version 17.2 [Online]. Available: http://bbs.hhi.de/suehring/tml/download.

[32] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 3.0 (VSRS3.0)," Tech. Rep. Document M16090, ISO/IEC JTC1/SC29/WG11, Feb. 2009.

[33] K. Muller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," in *Proc. IEEE*, vol. 99, no. 4, Apr. 2011, pp. 643–656.

[34] N. A. El-Yamany, K. Ugur, M. M. Hannuksela, and M. Gabbouj, "Evaluation of depth compression and view synthesis distortions in multiview-video-plus-depth coding systems," in *IEEE Proc. 3DTV-CON*, Jun. 2010, pp. 1–4.

[35] "An excel add-in for computing Bjontegaard metric and its evolution," document VCEG-AE07, ITU-T SG16 Q.6, Jan. 2007.
[36] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Proc. CVPR*, Jun. 2007 pp. 1–8.
[37] *Undo-Dancer Video Sequences*. [Online] Available: http://mpeg3dv.research.nokia.com.

**Viet-Anh Nguyen** (M'10) received the B.S and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2004 and 2010, respectively.

He is currently working with the Advanced Digital Sciences Center (ADSC), Singapore, which was jointly founded by University of Illinois at Urbana-Champaign, Urbana, and the Agency for Science, Technology and Research, a Singapore government agency. Before joining ADSC, he was with the School of Electrical and Electronic Engineering, NTU as a Research Fellow from 2008 to 2010. His current research interests include image and video processing, media compression and delivery, computer vision, and real-time multimedia systems.

**Dongbo Min** (M'09) received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2003, 2005 and 2009, respectively.

Since July 2010, he has been with the Advanced Digital Sciences Center, Singapore, which was jointly founded by University of Illinois at Urbana-Champaign, Urbana, and the Agency for Science, Technology and Research, a Singapore government agency. From 2009 to 2010, he was with the Mitsubishi Electric Research Laboratories, Cambridge, MA, as a Post-Doctoral Researcher, where he developed a prototype of 3-D video systems. His current research interests include 3-D computer vision, GPU-based real-time systems, 3-D modeling, hybrid sensor systems, and computational photography.

**Minh N. Do** (M'01–SM'07) was born in Vietnam in 1974. He received the B.E. degree in computer engineering from the University of Canberra, Australia, in 1997, and the Ph.D. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland, in 2001.

Since 2002, he has been on the faculty of the University of Illinois at Urbana-Champaign (UIUC), Urbana, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering, and holds joint appointments with the Coordinated Science Laboratory, Beckman Institute for Advanced Science and Technology, Urbana, IL, and the Department of Bioengineering. His current research interests include image and multidimensional signal processing, wavelets and multiscale geometric analysis, computational imaging, augmented reality, and visual information representation.

Dr. Do received a Silver Medal from the 32nd International Mathematical Olympiad in 1991, a University Medal from the University of Canberra in 1997, a Doctorate Award from the EPFL in 2001, a CAREER Award from the National Science Foundation in 2003, and a Young Author Best Paper Award from IEEE in 2008. He was named a Beckman Fellow at the Center for Advanced Study, UIUC, in 2006, and received a Xerox Award for Faculty Research from the College of Engineering, UIUC, in 2007. He is a member of the IEEE Signal Processing Theory and Methods and Image, Video, and Multidimensional Signal Processing Technical Committees, and an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.