WEIGHTED MODE FILTERING AND ITS APPLICATIONS TO DEPTH VIDEO ENHANCEMENT AND CODING

Dongbo Min[†], *Jiangbo Lu*[†], *Viet-Anh Nguyen*[†] *and Minh N. Do*[§]

Advanced Digital Sciences Center (ADSC), Singapore[†]
University of Illinois at Urbana-Champaign, IL, USA[§]
dbmin99@gmail.com, Jiangbo.Lu@adsc.com.sg, vanguyen@adsc.com.sg, minhdo@illinois.edu

ABSTRACT

This paper presents a novel approach for improving the quality of depth video. Given a high-quality color image and its corresponding low-quality depth image, we handle various artifacts which may exist on the depth video by applying a weighted mode filtering method based on a joint histogram. When the histogram is generated, the weight based on color similarity between reference and neighboring pixels on the color image is computed and then used for counting each bin on the joint histogram of the depth map. A final solution is determined by seeking a global mode on the histogram. Experimental results show that the proposed method has outstanding performance and is very efficient in various applications such as depth video enhancement and compression.

Index Terms— Weighted mode filtering, joint histogram, depth video enhancement, depth coding

1. INTRODUCTION

With the recent development of 3D multimedia/display technologies and the increasing demand for realistic multimedia, 3D video has gained more attentions as one of the most dominant video formats with a variety of applications such as 3DTV or freeview point TV (FTV). For successful development of 3D video systems, many technical issues should be resolved, e.g. capturing and analyzing stereo or multiview images, compressing and transmitting the data, and rendering 3D images on various 3D displays.

The main challenging issues are depth estimation, virtual view synthesis, and 3D video coding. Depth maps are used to synthesize virtual views at the receiver side, so providing a high-quality depth map is an important issue in the 3D video system. Recently, active depth sensors such as Time-of-Flight (ToF) cameras have been widely used to provide 3D video. For instance, given a hybrid system which consists of one color camera and one depth sensor, depth maps provided from the depth sensor, which may be noisy and of low-resolution, can be enhanced by using the corresponding color image. The depth video is then encoded and transmitted with the corresponding color images together. At the receiver side, the decoded color-plus-depth data are utilized to synthesize virtual views. Thus, the depth video should be efficiently compressed to reduce the depth bit rate as much as possible.

In this paper, we focus on developing an efficient solution to improve the quality of the depth video. For that, an weighted mode filtering (WMF) is proposed based on a joint histogram. Given a high-resolution color video and its corresponding low-quality depth video, the weight based on similarity measure between reference and neighboring pixels is used to construct the histogram, and a final solution is then determined by seeking a global mode on the histogram. We will show the performance of the proposed method in various applications such as depth video enhancement and compression.

2. WEIGHTED MODE FILTERING ON HISTOGRAM

A histogram is a function that represents a probability distribution of continuous (or discrete) values in a given data set. The global histogram of an entire image can be exploited to represent global characteristics of the image in some applications such as contrast enhancement. A *localized* histogram $H(p,d)$ of an image for a reference pixel p and the d^{th} bin is computed using a set of its neighboring pixels inside a window, which was introduced by Weijer and Boomgaard [1].

In this paper, we introduce this concept in a generic formulation and focus on developing a post-processing algorithm with the localized histogram [2]. Specifically, given a discrete function $f(p)$ whose value ranges from 0 to $L-1$, the localized histogram $H(p,d)$ is defined at the pixel p and the 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 a rectangular (or irregularly-shaped) region. We propose a novel filtering approach which seeks the global mode on the histogram by leveraging the similarity measure between the data of two pixels. When the histogram $H(p,d)$ for each pixel p is generated, the data of each pixel inside the rectangular (or irregularly-shaped) region is adaptively counted on its corresponding bin by using the data similarity function $w(p,q)$ between the reference and neighboring pixels as

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

where $w(p,q)$ is a non-negative function which defines the correlation between the pixels p and q. $N(p)$ is the set of neighboring pixels of the pixel p. The neighboring pixel which exhibits a stronger correlation with the reference pixel p has a larger weighting value $w(p,q)$. Gaussian spreading function G_r models errors that may exist on the input data $f(p)$.

We call this histogram-based approach the weighted mode filtering, in which the final solution is obtained by seeking the highest mode of the weighted distribution $H(p,d)$ [2]. In this paper, the weighted mode filtering is extended into the joint filtering framework by using a guide signal $g(p)$ different from the reference signal $f(p)$ to be filtered as follows:

$$H(p,d) = \sum_{q \in N(p)} G_I(g(p) - g(q)) G_S(p - q) G_r(d - f(q)) \,, \tag{2}$$

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

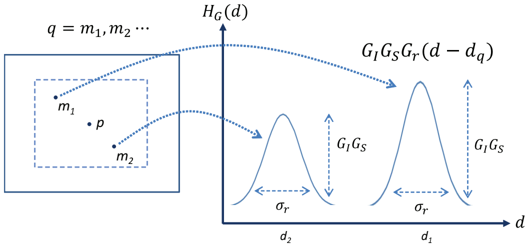


Fig. 1. Joint histogram generation: The joint histogram H of the reference pixel p is calculated by adaptively counting the neighboring pixels m_1, m_2 with a form of Gaussian function according to their disparity values d_1 and d_2.

where $G_I(x)$, $G_S(x)$, and $G_r(x)$ are Gaussian functions where means are 0 and standard deviations are σ_I, σ_S, and σ_r, respectively. The final solution $f_G(p)$ for the weighted mode filtering can be computed as follows:

$$f_G(p) = \arg\max_d H(p, d) \ . \tag{3}$$

Here $g(p)$ is a $2D$ function where each pixel p has a specific data. Since a guide signal is employed for calculating the data-driven adaptive weight G_I, the proposed filtering is contextualized within the joint bilateral filtering framework [3].

Fig. 1 explains the procedure that generates the joint histogram H. The neighboring pixels m_1, m_2 of the reference pixel p are adaptively counted with a form of Gaussian function on each bin corresponding to their disparity values. The bandwidth and magnitude of Gaussian function are defined by the standard deviation σ_r of G_r and the magnitude of $G_I G_S$, respectively. The standard deviation σ_r of Gaussian spreading function G_r is used for modeling errors that may exist on the input depth data. In other words, the neighboring pixels are adaptively accumulated on the joint histogram H by using color (G_I) and spatial (G_S) similarity measures and Gaussian error model (G_r).

In general, depth values vary smoothly inside objects and has sharp discontinuities on the object boundaries. When the depth is enhanced by a conventional joint bilateral filtering [3], the filtered output depth value is provided by using an adaptive summation based on color information. Although an adaptive weight based on color information is used for preserving edges, it still results in unnecessary blur due to its *summation*.

3. APPLICATIONS

In this section, we verify the performance and effectiveness of the proposed method by applying it to enhancing a low-quality depth map provided from a ToF depth sensor and suppressing coding artifacts generated from depth video compression based on a typical transform-based motion-compensated video codec. In these applications, $g(p)$ is the color image $I(p)$ and $f(p)$ is the depth image $d(p)$.

3.1. Depth Video Enhancement

The quality of depth video can be measured by the amount of noise, spatial resolution and temporal consistency. Therefore, the depth video can be improved by suppressing the noise, increasing its spatial resolution and handling the temporal flickering problem. In this section, we show how to enhance the depth map obtained from a ToF camera based on the weighted mode filtering for achieving these goals.

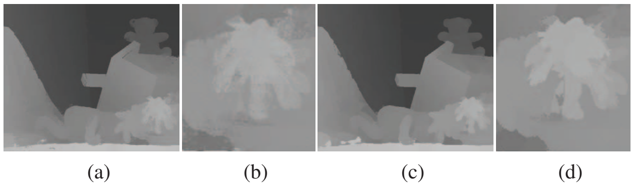

(a) (b) (c) (d)

Fig. 2. Aliasing effect in the depth upsampling: (a) Aliased depth map without MCM. (b) Cropped image of (a). (c) Proposed method with MCM. (d) Cropped image of (c).

3.1.1. Weighted Mode Filtering Using Multiscale Color Measure

Given a low resolution depth map and a high resolution color image, the sparse original depth values mapped into the color camera coordinate only are used for preventing the output depth value from being blurred on the depth boundaries. In order to include this notation in Eq. (2), we define a binary function $R(p)$, whose value is 1 when p has an original depth value, 0 otherwise. The histogram $H(p, d)$ can be expressed as follows:

$$H(p, d) = \sum_{q \in N(p)} R(q) G_I(I_p - I_q) G_S(p - q) G_r(d - d(q)) \ . \tag{4}$$

As shown in Fig. 2, however, if we use the sparse original depth values for the depth upsampling directly, it may cause the aliasing artifact due to different sizes between the original depth and color images. In Eq. (4), since the color distance $G_I(I_p - I_q)$ of neighboring pixels are calculated by using sparse pixels only where they have depth values ($R(p) = 1$), this color measure cannot represent the distribution of color information inside the window $N(p)$. The existing methods [3][4] have handled this problem by applying prefiltering methods such as bilinear or bicubic interpolation. However, this initial depth map contains contaminated values which may cause serious blur on the depth boundaries. In this paper, we handle this problem by using a multiscale color measure [2], instead of applying the prefiltering techniques. This method can provide an aliasing-free upsampled depth map as well as preserve the depth discontinuities well.

Before explaining the method in detail, we define some parameters for helping readers to understand it. Specifically, let the resolution difference between the original low-resolution depth map and the high-resolution color image be $S_{dc} = min(H_c/H_d, W_c/W_d)$, where H_d, W_d and H_c, W_c are the height and width of the depth and color images, respectively. The number of level L ($l : L - 1 \sim 0$) on the multiscale framework is set to $log_2 S_{dc}$. The window $N(p)$ on the l^{th} level is defined as $\{q| \ |p - q|_\infty \leq 2^l S_W\}$, where S_W is the size of the window on the original small depth domain. Namely, the actual size of the window $N(p)$ is dependent on the upsampling ratio, since the sparse original depth values only are used. We also define a new Gaussian filtered color image $I_G = G * I$, which is used for calculating the color distance on each level of the multiscale framework.

The sparse depth map can be upsampled by using Eq. (4) in a coarse-to-fine manner. The Gaussian lowpass-filtered color image I_G is first computed for each level, and the weighted mode filtering is performed on each level by using the original and upsampled depth values only. For instance, if L is 3, the upsampling procedure starts for every $4 (= 2^2)$ pixels on the coarsest level, and the

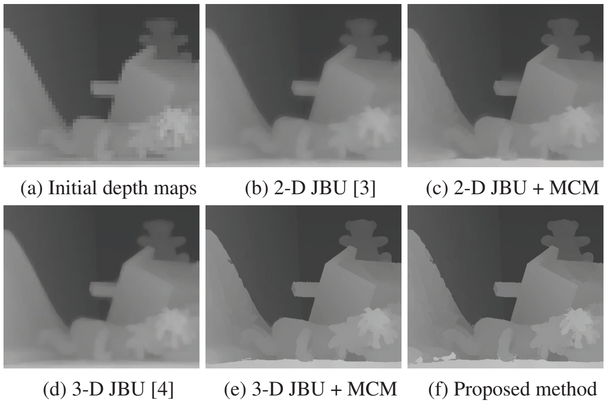


Fig. 3. Depth upsampling results for test bed images: the downsamping ratio is 8 in each dimension. The processing times are (b) $0.95s$, (c) $0.34s$, (d) $220.6s$, (e) $65.8s$, and (f) $0.55s$. The processing time of the proposed method is 0.25% of that of 3-D JBU, while it has the best edge-preserving performance.

binary function $R(p)$ of these pixels is set to 1. In other words, the depth value of pixels upsampled on the current level can be used on the next level again. The variance of the Gaussian function for low-pass filtering of I_G is proportional to the ratio of downsampling on each level. Different from the conventional downsampling procedure where low-pass filtering is first applied and an image is then downsampled, the Gaussian low-pass filtered color image I_G is first computed and the color distance $G_I(I_G(p) - I_G(q))$ is then calculated on the *full resolution grid* (not coarse resolution grid). In other words, the filtered color image I_G is not downsampled.

3.1.2. Depth Upsampling Results

For the objective evaluation, we performed experiments with ground truth depth maps provided by the Middlebury test bed [5]. Low-quality depth map, which is generated by downsampling the ground truth depth map, is upsampled by the proposed method. The weighting parameters σ_I, σ_S and σ_r in Eq. (4) are set to 6.0, 7.0 and 2.9, respectively. The size of the window S_W on the original small depth domain is 2.

Fig. 3 shows the results of the proposed method for the test bed images, when the downsampling ratio is 8 in each dimension. Due to the lack of space, we showed the results of Teddy only. The results of the 2-D JBU [3] and the 3-D JBU [4] were included for a visual evaluation. Note that these methods used the bilinear interpolation technique for computing the initial input (dense) depth maps. The size of the window $N(p)$ is set to 11×11, since the multiscale color measure is not used. In order to fairly compare the performance of the filtering-based methods by using the same input depth maps, the results which were upsampled with the multiscale color measure (MCM) were included as well: 2-D JBU + MCM, 3-D JBU + MCM.

The proposed method yields the superior results over the existing methods, especially on the depth discontinuities. The performance of the 3-D JBU + MCM is very similar to that of the weighted mode filtering, since the MCM prevented the upsampled depth map from being blurred on the depth discontinuities. The 2-D JBU + MCM does not improve the edge-preserving performance, even compared with the 2-D JBU which used the blurred depth maps as the initial value. The processing time of the 2-D JBU + MCM is the smallest among all methods, but its quality is worse. Although the 3-D JBU + MCM has a similar accuracy to the proposed method, the computational complexity of the proposed method is nearly 0.8% of that of the 3-D JBU + MCM. Since the 3-D JBU performs the joint bilateral filtering for all depth candidates repeatedly, it results in huge computational complexity.

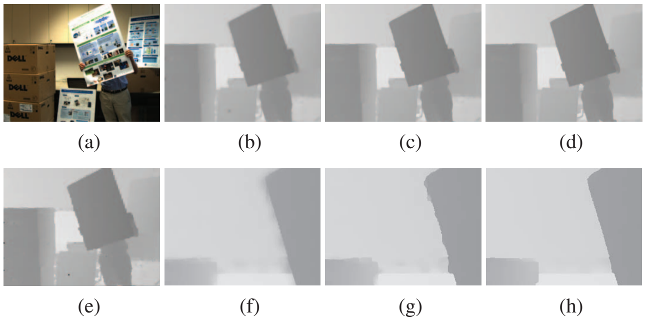


Fig. 4. Upsampling results for low-quality depth image (from 'Mesa Imaging SR4000') with corresponding color image (from 'Point Grey Flea'). (a) Color image. (b) 2-D JBU. (c) 3-D JBU. (d) Proposed method. (e) Initial depth map. (f) Cropped image of (b). (g) Cropped image of (c). (h) Cropped image of (d).

The experiments were also performed using depth and color videos, captured by a 'Mesa Imaging SR4000' depth sensor and a 'Point Grey Flea' color camera. As shown in Fig. 4, the proposed method was evaluated by comparing the upsampled depth images with those of the 2-D JBU [3] and the 3-D JBU [4]. The sizes of the input depth and color images are 176×144 and 1024×768, respectively. We found that the proposed method provides the best edge-preserving performance on the depth discontinuities. The processing times are $6.8s$ for the 2-D JBU, $1592.3s$ for the 3-D JBU, and $5.6s$ for the proposed method, respectively.

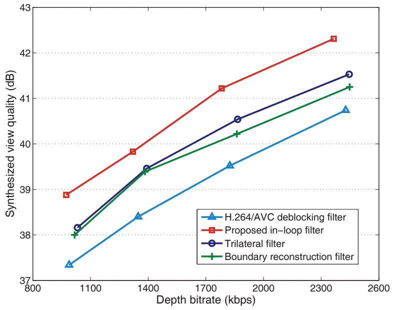
3.2. Depth Video Coding

In this section, we describe a novel scheme that compresses the depth video efficiently using the framework of a conventional video codec [6]. The depth video is encoded by a typical transform-based motion compensated video encoder such as H.264/AVC, and compression artifacts are addressed by utilizing the weighted mode filtering as an in-loop filter.

3.2.1. Weighted Mode Filtering-Based Depth Coding

Containing homogeneous regions separated by sharp edges, a 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. In order to address these artifacts, the weighted mode filtering is employed to design an in-loop edge-preserving denoising filter. By selecting the global mode on the histogram, it avoids unnecessary blur and provides a solution with the highest occurrence [6]. We will show the depth video coding based on the weighted mode filtering outperforms the existing post-processing based coding methods [7][8].

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 color video pixels. Inspired by this observation, we utilize the color video pixels $I(p)$ as the guided



(a) Ballet

Fig. 5. RD curves obtained by encoding the depth maps using the proposed and existing in-loop filters.

function $g(p)$ to denoise the depth data $d(p)$ as 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.

3.2.2. Depth Coding Results

We have conducted various experiments with the Ballet test sequences with resolutions of 1024×768, of which both the color video and depth map are provided from Microsoft Research [9]. 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 [10]. The conventional H.264/AVC deblocking filter in the reference software was replaced with the proposed in-loop filter. For the 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.

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 [7] and the trilateral filter [8], which are also utilized as the in-loop filter. Fig. 5 shows RD curves obtained by the proposed and existing in-loop filters for the two test sequences. 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 [11] compared with that of the existing filters for the Ballet sequences. In addition, the proposed filter obtained a better visual quality of the synthesized view, as shown in Fig. 6.

4. CONCLUSION

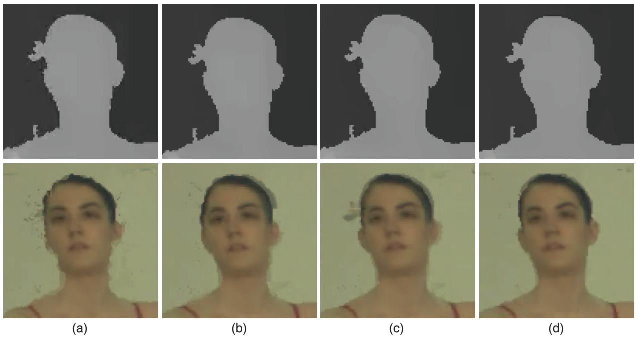


Fig. 6. Sample frames of the reconstructed depth map and rendered view for the Ballet sequence obtained by different in-loop filters: (a) H.264/AVC Deblocking filter, (b) Boundary reconstruction filter [7], (c) Trilateral filter [8], (d) Proposed in-loop filter.

In this paper, we have presented an weighted mode filtering method based on joint histogram and then verified its performance and effectiveness in the depth video enhancement and compression. Considering the guide color image in the joint filtering framework, the proposed method effectively handles several artifacts caused by inherent limits of the ToF depth sensor or the typical transform-based motion-compensated video codec (H.264/AVC). It provides the results that have better edge-preserving performance. The multiscale color measure (MCM) was also proposed for suppressing the aliasing effect on the depth upsampling. Moreover, the proposed method is highly parallelizable on GPUs or FPGA thanks to its pixel-wise independent processing.

5. REFERENCES

[1] J. Weijer and R. Boomgaard, "Local Mode Filtering," in *IEEE Proc. CVPR*, 2001.

[2] D. Min, J. Lu, and M. N. Do, "Depth Video Enhancement Based on Weighted Mode Filtering," *IEEE Trans. on Image Processing*, 2011, to appear.

[3] J. Kopf, M. F. Cohen, D. Lischinski and M. Uyttendaele, "Joint bilateral upsampling," *ACM SIGGRAPH* 2007.

[4] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *IEEE Proc. CVPR*, 2007.

[5] http://vision.middlebury.edu/stereo

[6] V.-A Nguyen, D. Min, and M. N. Do, "Efficient Techniques for Depth Video Compression Using Weighted Mode Filtering," *IEEE Trans. on Circuits and Systems for Video Technology*. (submitted)

[7] K.-J. Oh, A. Vetro, and Y.-S. Ho, "Depth Coding Using a Boundary Reconstruction Filter for 3-D Video Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 350-359, 2011.

[8] S. Liu, P. Lai, D. Tian, and C. W. Chen, "New Depth Coding Techniques With Utilization of Corresponding Video," *IEEE Trans. on Broadcasting*, vol. 57, no. 2, pp. 551-561, 2011.

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

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

[11] "An excel add-in for computing Bjontegaard metric and its evolution," document VCEG-AE07, ITU-T SG16 Q.6, Jan. 2007.