

## An Asymmetric Filter Based 3D Synthesis Algorithm with Depth Region Classification

PEI-JUN LEE, HONG-LI LIN AND TRONG-AN BUI

*Department of Electrical Engineering*

*National Chi Nan University*

*Nantou, 545 Taiwan*

*E-mail: pjlee@ncnu.edu.tw; honglilin@hotmail.com; trongan93@gmail.com*

To reduce geometric distortion in three dimensional (3D) synthesized images and high computation time in the depth map preprocessing, this paper proposes a region classification based asymmetric filter for depth map preprocessing. First, the depth map can be classified into three types of region: depth continuous region, sharp depth transition regions with, and without vertical lines. To maintain 3D experience and reduce computation time, the depth value in the depth continuous region is left unchanged, such as the background region. For the sharp depth transition regions, we determine the presence or lack of vertical lines. Next, an adaptive Horizontal Gaussian filter is used to reduce rubber-sheet holes generated in the region which does not contain vertical lines. Additionally, an adaptive Gaussian filter based on the vertical line direction is used to reduce geometric artifacts and further reduce the computation time compared to other asymmetric Gaussian filters used in the region which has vertical lines. The proposed algorithm can improve the quality of the Depth-image-based rendering (DIBR) virtual views by using depth sensors. According to the experiment result, the computation time is reduced by 90-95% compared with the asymmetric Gaussian filter. Moreover, the vertical line comparison and the skewness comparison show that geometric distortion is reduced obviously and the 3D experience is preserved successfully.

**Keywords:** 3D, depth region, depth image based rendering, DIBR, synthesized virtual view

### 1. INTRODUCTION

In recent years, 3D technologies have been developed very rapidly. Virtual Reality (VR) [1] is a trend of multimedia application. Therefore, the research topics related to 3D content are very focused. However, the cost of traditional 3D video captured by multiple image sensors is very high. Depth-image-based rendering (DIBR) is the most popular technology to generate virtual views [2] for 3D displays by using a depth sensor [3]. DIBR technology is based on a stereo camera configuration where a color image and its corresponding depth map are used to develop a warping formula as Eq. (1).

$$x_l = x_c + \frac{t_x}{2} \frac{f}{Z}, \quad x_r = x_c - \frac{t_x}{2} \frac{f}{Z}, \quad (1)$$

where  $x_l$  and  $x_r$  are the pixel coordinates of the generated virtual left and right images, respectively.  $x_c$  is the pixel coordinate of the intermediate view.  $t_x$  represents the baseline distance between the left and right cameras,  $f$  is the focal length and  $Z$  is the depth distance from the object point to the left ( $c_l$ ) or right ( $c_r$ ) cameras.

---

Received May 21, 2019; revised July 24, 2019; accepted September 27, 2019.  
Communicated by Wen-Liang Hwang.

A block diagram of the DIBR system is shown in Fig. 1. However, the depth value distribution is non-linear so holes or rubber-sheet artifacts occur in the sharp depth transition (big hole) regions and the depth discontinuity (creek hole) regions after image warping [4], especially in the depth image which is captured by depth sensor. To improve the 3D quality of the virtual views and reduce the computation time for the hole-filling process [5], depth map pre-processing [6, 7] are used before the 3D image warping to reduce hole occurrences.

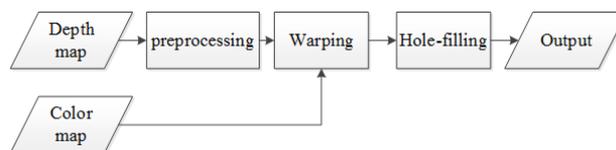


Fig. 1. Block diagram of DIBR.

The most popular method of depth map pre-processing is to design a smoothing filter and apply it to the hole depth map. The smoothing filter can reduce the small depth discontinuities and soften the sharp depth transitions, thereby to decrease the subsequent generation of creek holes and big holes in the virtual views. The symmetric Gaussian filter [8] and the symmetric bilateral filter [9] are designed for depth pre-processing. Both filters cause the depth values of the object boundaries to become smoother than the original depth map. However, the pre-processing of the depth map usually generates other artifacts, such as rubber sheet artifacts, geometric distortion and even the degradation of depth perception. Geometric distortions in the smoothed region are generated after applying the symmetric smoothing filter to the hole depth map. Based on observations made during previous 3D studies, geometric artifacts easily occur in vertical lines in the foreground and in background areas of sharp depth transition. Therefore, an asymmetric Gaussian filter [10] is proposed to reduce the geometric distortion.

The asymmetric filter is designed using the vertical smoothing strength [10] which is greater than the horizontal smoothing strength. However, the computation time of the asymmetric filter when the applied entire depth map is higher than that of the symmetric filter. To preserve the 3D experience and reduce the computation time, a region-based smoothing filter [11] is proposed to smooth the depth map, which divides the depth map into hole region and non-hole region then they adopted different smoothing strength asymmetric filters for the depth map. Therefore, the depth information in the non-hole region is preserved such that the 3D experience can be improved. Even though the computation time is reduced, the geometric distortion still occurs when using these methods, especially in the hole regions containing edges. A directional Gaussian filter is used to reduce the geometric distortion in the hole-regions, but the 3D effect is worse than that in [10]. An edge-based asymmetric Gaussian filter [12] is used to reduce the geometric distortion in the hole region, such that the 3D experience in the non-hole region is preserved and the computation time could be reduced.

To reduce the computation time, the depth information is preserved to maintain the 3D experience in the non-hole region. In the hole region, an adaptive edge-oriented depth map smoothing filter (AEODF) is proposed in [13], [14] proposed an edge-dependent depth filter, and [15] proposed an adaptive edge-oriented smoothing filter with adaptive

content (AAEODF) to reduce the hole occurrence. Moreover, in hole regions with vertical lines, the constant depth value is used in [12], and the asymmetric Gaussian filter is used in [13, 15] to reduce the geometric distortion, respectively. However, these algorithms generate new depth discontinuities, such as depth value mixing on background and foreground boundary regions and over smoothing in shape depth transition regions. Therefore, the regions of the depth map should be classified according to detail.

This paper proposes a region classification based asymmetric filter for depth map pre-processing. In order to maintain 3D experience and to reduce computation time, the proposed algorithm does not change the depth value in-depth continuous regions, such as background regions. Firstly, in sharp depth transition regions, the proposed algorithm determines whether the regions contain vertical lines or not. Secondly, in the region without vertical lines, this paper proposes an adaptive horizontal Gaussian filter to reduce the rubber-sheet hole generation. Thirdly, in the region with vertical lines, we reduce depth value mixing at background/foreground boundaries and over-smoothing in sharp depth transition regions. This paper proposes an adaptive Gaussian filter based on the direction in which the background vertical lines extend from the edge of the foreground objects in order to reduce geometric artifact generation and reduce the computation time even more than other asymmetric Gaussian filters did. According to the experiment results, the computation time is reduced by 90~95% compared with the asymmetric Gaussian filter. Moreover, the vertical line comparison and the skewness comparison show that the geometric distortion is noticeably reduced and the 3D experience is preserved.

This paper is organized as follows. The proposed algorithm is presented in Section 2. The experiment results are shown in Section 3. The conclusion is given in Section 4.

## 2. THE PROPOSED ALGORITHM

As we know, most of the depth map can be classified as a non-hole region due to its continuous depth value, as indicated by the orange squares in Figs. 2 (a) and (b). The orange square is a related position of the final synthesized [16] view [17], so it is clear to be classified into the non-hole regions. To preserve the depth quality of depth continuous regions in the depth map, this paper does not change the initial depth value. Thus, the 3D experience is preserved and the computation time is not increased in these regions. In the depth transition regions, geometric distortion will occur in synthesized virtual views with vertical lines as shown in the areas denoted by red squares in Figs. 2 (c) and (d). To improve the quality of the synthesized virtual views and reduce the computation time, this paper proposes a region classification-based algorithm to apply adaptive Gaussian filters in different regions.

The proposed adaptive horizontal Gaussian filter is used solely in hole regions without vertical lines (as denoted by the green squares in Figs. 2 (a) and (b)) to reduce the depth transition in the depth map, then the quality of the synthesized virtual views in this region, which is called a "Type A" region, is improved. Since the smoothing strength and the smoothing region are adaptive to the hole length, the computation time and the distortion will be reduced in this region.

Geometric distortion is the main problem when depth map pre-processing is used, especially in hole regions that contain vertical lines which are called "Type B" regions.

Therefore, in general, a vertical Gaussian filter is used in hole regions with vertical lines to reduce the geometric distortion, the drawback of this method is that the computation time is greatly increased. An adaptive symmetric Gaussian filter is proposed to smooth sharp depth transition in Type B regions. The vertical smoothing area can be determined and the smoothing strength of the proposed vertical Gaussian filter is based on foreground object protection such that the proposed algorithm can reduce the geometric distortion in the synthesized virtual views in hole regions which have vertical lines.

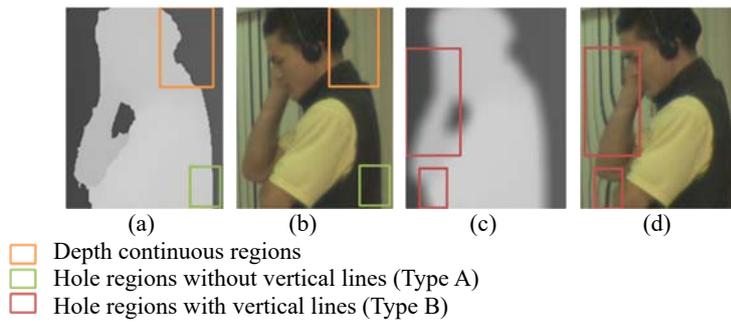


Fig. 2. The depth map region classification and its corresponding synthesized virtual view.

The flow chart of the proposed algorithm is shown in Fig. 3. First, hole detection is used to divide the depth map into depth continuous regions (non-hole regions) or depth transition regions (hole regions). Then the proposed adaptive horizontal Gaussian filter is used in the depth transition regions to reduce the hole occurrence. Then, vertical line detection is applied to the corresponding color image of the depth map to determine Type A or Type B regions. A vertical smoothing area determination and a foreground protected vertical Gaussian filter are proposed in Type B regions to reduce the geometric distortion.

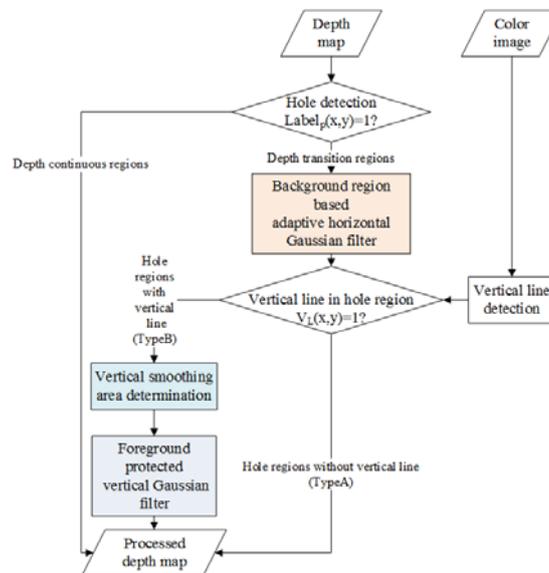


Fig. 3. The flow chart of the proposed algorithm.

## 2.1 Depth Transition Region (Hole Region) Determination

Holes occur in synthesized virtual views where a sharp depth transition exists in the depth map. Therefore, smoothing these sharp depth transition regions of the depth map can reduce the rate of occurrence of geometric distortions. In depth continuous regions, the depth values remain unchanged and then the computation time of the depth map pre-processing is not increased and the 3D experience is preserved in these regions.

The depth transition regions are detected by Eq. (2):

$$Label_p(x, y) = \begin{cases} 1, \forall p(x+1, y) - p(x, y) > Th_0, \text{ for the left view,} \\ \quad \forall p(x-1, y) - p(x, y) > Th_0, \text{ for the right view,} \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

$$p(x, y) = M \times (1 - s(x, y) / 255)$$

where  $p(x, y)$  is the disparity value (parallax) of  $(x, y)$  which is estimated using the fast parallax look-up table [15].  $M$  is the largest parallax. Usually,  $M$  is equal to 5% of the image width.  $s(x, y)$  is the initial depth value of  $(x, y)$ . Here we use the method proposed in [15] to detect the depth transition regions and label these regions  $Label_p(x, y) = 1$ .

Further, we set the threshold  $Th_0 = 20$  to determine the hole regions. For the left view image, holes occur on the depth map in areas of low to high depth transition and vice versa for the right view image. If  $Label_p(x, y) = 0$ , then the region is determined as a depth continuous region, and the processed depth value  $\hat{s}(x, y)$  is equal to the initial depth value.

$$\hat{s}(x, y) = s(x, y) \quad (3)$$

In depth transition regions, rubber-sheet artifacts will occur in the synthesized views. Therefore, a horizontal smoothing filter is proposed to reduce the size of any creek holes in the synthesized views. In addition, a vertical Gaussian filter is used in hole regions which contain vertical lines to reduce the geometric distortion in the synthesized views.

## 2.2 The Proposed Adaptive Horizontal Gaussian Filter

In general, the strength of the horizontal smoothing filter which is used to eliminate the depth sharp transition in the depth map is constant, thus the size of the hole in the synthesized virtual views is reduced. However, if there are two objects in the foreground and the distance between these two objects is less than the filter strength, such as the two objects hand and head as shown in Fig. 4, then these two objects and the background gap

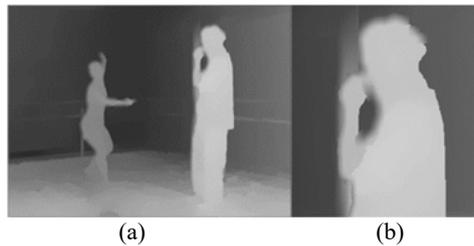


Fig. 4. (a) A depth map after the application of a horizontal smoothing filter of constant strength; (b) enlarged detail of (a).

will be assigned with the same depth values in the processed depth map. Therefore, some errors will occur in this region of the synthesized virtual image. For instance, errors will occur in the right side of the image because the difference in depth value between the hand and the head of the figure on the right side of Fig. 4 is very small.

To solve the above problem, an adaptive horizontal Gaussian filter based on the background region width is proposed to reduce the depth discontinuity in the depth map and consequently, the size of the creek holes in the synthesized views is also reduced. To decrease the computation time in advance, this paper proposes an adaptive horizontal Gaussian filter Eq. (4) which is used only in the depth transition regions. Moreover, the smoothing strength  $\sigma$  is dependent on the estimated hole length and background region width as shown in Eq. (5). Then the processed depth value  $\hat{s}(x, y)$  is obtained from Eq. (6).

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{\sigma^2}\right) \quad (4)$$

$$\sigma = \begin{cases} p(x+1, y) - p(x, y), & \text{for left view} \\ p(x-1, y) - p(x, y), & \text{for right view} \end{cases} \quad (5)$$

$$\hat{s}(x+m, y) = \begin{cases} \frac{\sum_{i=-3\sigma}^{3\sigma} s(x+m+i, y)g(i, \sigma)}{\sum_{i=-3\sigma}^{\sigma} g(i, \sigma)}, & \forall \text{Label}_p(x, y) = 1, \hat{s}(x+m, y) > s(x+m, y) \\ s(x+m, y), & \forall s(x+m, y) > \hat{s}(x+m, y) \end{cases}, \quad (6)$$

where  $m \in [s_b, s_f]$ ,  $s_b = \min(2\sigma, \text{background width})$ ,  $s_f = \sigma \cdot g(x, \sigma)$  is the horizontal Gaussian smoothing filter.  $\hat{s}(x, y)$  is the smoothed depth value,  $s(x, y)$  denotes the original depth value. Here the window size is  $[-i, i]$ ,  $i$  is set to  $3\sigma$ . Furthermore, the smoothed region is from the pixel  $s_b$  to the pixel  $s_f$ , where  $s_b$  belongs to the smoothed region of the background and  $s_f$  belongs to the smoothed region of the foreground.

An illustration of the proposed horizontal smoothing filter is shown in Fig. 5. The green points are the depth transition pixels which are detected by Eq. (2) with  $\text{Label}_p(x, y) = 1$ . The red boxes show the window size of the horizontal filter, and the blue boxes show the smoothed regions. It is more important in terms of the viewing experience to preserve the depth information in the foreground, then when the width of the background gap between two foreground objects is larger than the filter strength, as in the row of point A in Fig. 5, the number of pixels which are altered by the filter should be smaller in the foreground than in the background.

Therefore, the width of the smoothing region is set at  $2\sigma$  in the background and at  $\sigma$  in the foreground. If the width of the background gap between two foreground objects is less than the filter strength as in the row of point B in Fig. 5, the smoothing region for the background would be changed based on the background gap between the two foreground objects. The proposed adaptive horizontal smoothing filter can reduce the rubber sheet artifacts which occur in the background gaps between foreground objects.

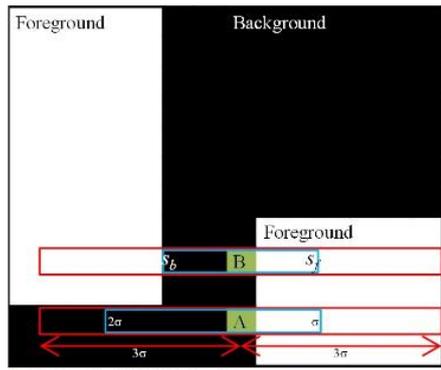


Fig. 5. Adaptive horizontal Gaussian filter for left boundary.

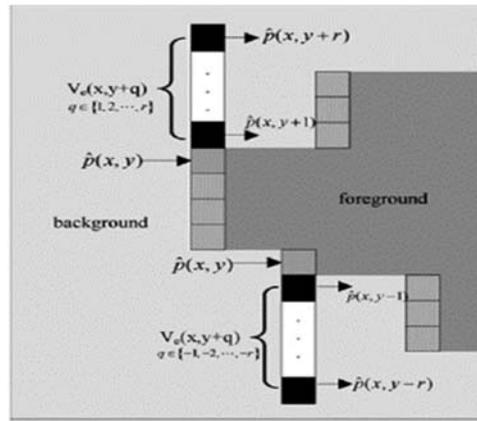


Fig. 6. Vertical line detection.

### 2.3 Vertical Line Detection

The depth map is processed using a horizontal smoothing filter to reduce hole occurrence for DIBR generated virtual views. Geometric distortion will occur in the depth transition regions which have vertical lines in the background of the depth map. To preserve the vertical lines in the depth transition regions, the proposed algorithm first detects the locations of the vertical lines in the background of the depth map, then applies an adaptive vertical Gaussian filter to smooth this region.

The vertical line detection method proposed in [13] is also used in this paper. Fig. 6 illustrates the vertical line detection performed on the corresponding color image of the depth map. First, the edge detection operation is performed in the background, then we check the pixels  $\hat{p}(x, y+q)$ ,  $q = 1, \dots, r$ , with vertical vectors and mark them as  $V_e(x, y+q)$  using Eq. (7),

$$V_e(x, y+q) = \begin{cases} 1, & \text{if } |p_q * G| > Th_1, \text{ where } s(x, y+q) < s(x, y) + Th_0 \text{ and } Label_p(x, y) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

for  $q \in \{1, 2, \dots, r\}$ , or  $q \in \{-1, -2, \dots, -r\}$ . Here,  $p_q = \hat{p}(x, y+q)$  is in the background in which  $q \in \{1, 2, \dots, r\}$  are for the upward vertical direction of the highest predicted hole pixel  $\hat{p}(x, y)$  and  $q \in \{-1, -2, \dots, -r\}$  are for the downward vertical direction of the lowest predicted hole pixel  $\hat{p}(x, y)$  with  $Label_p(x, y) = 1$  in a column. The threshold  $Th_1$  is set to 1 and “\*” is the convolution operator, and  $G$  in Eq. (7) is a  $3 \times 3$  vertical Sobel convolution kernel.

Based on Eq. (7), the proposed method uses Eq. (8) to identify the pixels  $V_L(x, y)$  in the background of the depth map which contains vertical lines on the hole pixel located at  $(x, y)$ . Eq. (8) is used to search the pixels with vertical vectors with up or down directions and to count the number of pixels with the vertical vector as Fig. 6.

$$V_L(x, y) = \begin{cases} 1, & \forall \frac{\sum_{q=1}^r V_e(x, y+q)}{r} \geq Th_2, \text{ or } \forall \frac{\sum_{q=1}^{-r} V_e(x, y+q)}{r} \geq Th_2 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

If the number of pixels with the vertical vector is larger than the threshold  $Th_2$ , the label  $V_L(x, y)$  is given 1. The threshold  $Th_2$  is set to 1, and the searching length  $r$  is set to 20. The thresholds setting is referred to in the paper [18].

## 2.4 Vertical Smoothing Area Determination

In depth transition regions containing vertical lines, geometric distortion in the synthesized views will be reduced by the vertical Gaussian filter [13, 15]. Their purpose is to smooth the change of the depth value in the vertical line region. However, if the vertical line area is not detected accurately, a new depth discontinuity will be generated in this region. An example is shown in Fig. 7 (a), where the depth value of the background/foreground boundary adjacent region is easily over-smoothed by the vertical Gaussian filter. Here, the area which needs vertical smoothing should be determined, and then the vertical smoothing filter is applied only near the vertical line and in the background. A comparison of the results yielded by the proposed algorithm with an extended smoothing region and those yielded by the methods in [13, 15] are shown in Fig. 7 (b). Thus, the 3D experience in the foreground is preserved and, due to the smaller smoothing region and the smaller smoothing mask, the computation time is also reduced.

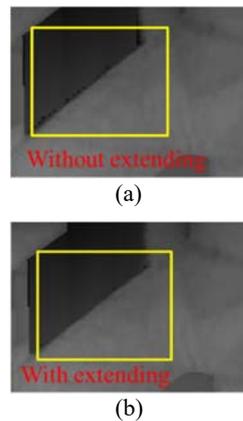


Fig. 7. Depth pre-processing results; (a) without extending the vertical smoothing region; (b) with extending the vertical smoothing region.

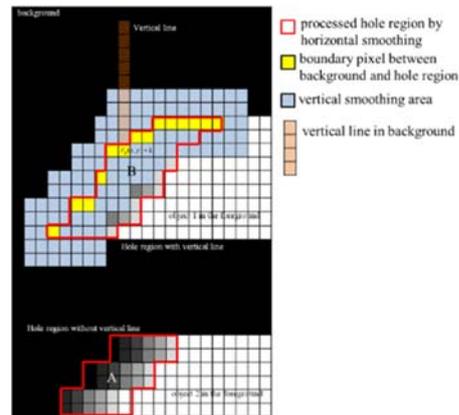


Fig. 8. Vertical smoothing area determination.

Two vertical smoothing filters will be adopted into up-direction parts and down-direction parts of vertical smoothing regions, respectively. Which are in order to preserve the depth information in the foreground and reduce the computation time. An illustration of the smoothing region determination is shown in Fig. 8 where the up-direction part is explained and vice versa for the down-direction. The red lines surround the depth transition regions which were smoothed using the horizontal smoothing filter (5). The two red outlined regions shown in Fig. 8 represent different possible region types. Region A is a depth transition region without vertical lines. Region B is a depth transition region with a vertical line (shown in orange)  $V_L(x, y)$  (7). In order to decrease the computation time, only the regions of depth transition containing vertical lines (Type B regions) will need to be

smoothed by the vertical Gaussian smoothing filter. Regions of depth transition containing no vertical lines (Type A regions) will not require vertical smoothing. Therefore, these regions are not smoothed by the vertical Gaussian filter. If vertical lines exist in a region smoothed by the horizontal smoothing filter, then the region is determined to be a vertical smoothing (Type B) area and the vertical filter is applied.

The upper- or lower-most pixels in a Type B region are labeled by  $h(x, y) = 1$  as in the yellow points in Fig. 8. These points are boundary pixels between a hole region and the background. After the boundary pixels are found, these pixels will be extended to  $k \times k$  block regions and labeled by  $h(x, y) = 1$  as in the blue areas shown in Fig. 8. The window length  $k$  is set to 5. The reason for extending the smoothing region beyond the topmost pixel is to reduce the generation of new depth discontinuities along the boundaries as shown in Fig. 7 (a). Finally, all yellow points and blue points ( $h(x, y) = 1$ ) in the Type B regions are the pixels which will be smoothed by the proposed vertical Gaussian filter.

### 3. FOREGROUND PROTECTED VERTICAL GAUSSIAN FILTER

To reduce geometric distortion in Type B regions of the synthesized views, a vertical Gaussian filter based on foreground protection is proposed to smooth Type B regions of the depth map. In order to avoid destroying the depth information in the foreground, only the background and regions of depth transition are smoothed. Moreover, the proposed vertical Gaussian filter in Eq. (9) is used only in the regions which have been determined to contain vertical lines so that not only the generation of new depth discontinuities to reduce hole and to increase 3D experience but also the computational time can be reduced as shown in Fig. 7 (b). Then the processed depth values  $s_v(x, y)$  are obtained from Eq. (10).

$$g(y, \mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{\mu^2}\right), \quad (9)$$

$$s_v(x, y + n) = \frac{\sum_{j=-3\mu}^{3\mu} \hat{s}(x, y + n + j)g(j, \mu)}{\sum_{j=-3\mu}^{3\mu} g(j, \mu)}, \quad \forall h(x, y) = 1, \quad (10)$$

where,  $n \in [s_b, s_f]$ ,  $s_b = 3\mu$ ,  $s_f = 0$ , the vertical smoothing strength  $\mu$  is much greater than the horizontal smoothing strength.  $\mu$  is set to 90. Here the vertical smoothing window size is  $[-j, j]$ , and  $j$  is set to  $3\mu$ .

The region to be smoothed by the proposed vertical smoothing filter is from the boundary of the object in the foreground to the background. Therefore, the region to be smoothed in the foreground is zero, and the region in the background in Fig. 9 shows a comparison of the regions smoothed by vertical Gaussian filters using the proposed algorithm as Fig. 9 (a) and using the algorithm in [13] as Fig. 9 (b). In Fig. 9, the background is the field to be shown with black color, the foreground is the field to be shown with the gray color. The region between these 2 colors, which needs to be smoothed. Finally, the processed depth value is obtained from Eq. (11).

$$\hat{s}(x, y) = \begin{cases} s_v(x, y), & \text{if } s_v(x, y) > \hat{s}(x, y) \\ \hat{s}(x, y), & \text{if } s_v(x, y) < \hat{s}(x, y) \end{cases} \quad (11)$$

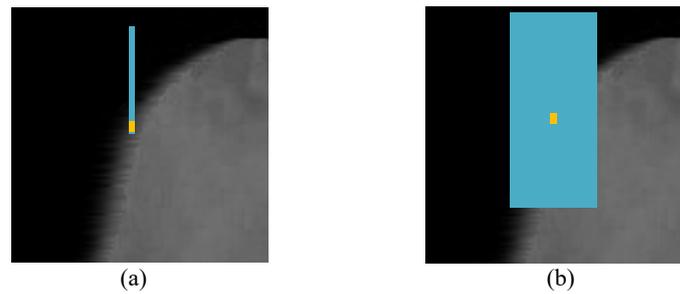


Fig. 9. The region to be smoothed using the proposed algorithm in (a), and the algorithm in [13] in (b).

#### 4. EXPERIMENT RESULT

In this section, a series of test sequences provided by the Advanced Three-dimensional Television System Technology (ATTEST) [18] are used to evaluate the performance of the proposed smoothing method in the DIBR system. In this paper, we use 5 test video sequences with different view parallax, vertical lines and high or low motion in picture. We found there are two kinds of sequences. The test sequences are divided into two categories. The first contains the sequences which have vertical lines and includes the Interview, Ballet, and Café sequences in Figs. 10 (a)-(c). The second category contains the sequences which do not have vertical lines and includes the Balloon and Break-dancer sequences in Figs. 10 (d) and (e). The picture size of all these five sequences is  $1024 \times 768$ . In addition, we also have a version of the Café sequence with the picture size  $720 \times 576$ . The color images and their corresponding depth maps are shown in Fig. 10. The environment for this experiment is as follows. The configuration of the test computer includes a CPU with 4 cores, 8 threads; 16GB of RAM; and the GPU with GDDR5 2GB standard memory, 1033 MHz Boost Clock.

The proposed method will be compared with the following methods; the symmetric Gaussian filter [8], the asymmetric Gaussian filter [10], the AEODF [13] and the AAEODF [15]. The main comparison will be on regions with holes and vertical lines processed using the AEODF and the AAEODF. There are five scores including the PSNR of the virtual view, skewness, vertical line, computation time, and MOS score for 3D view experience [19] to be compared.

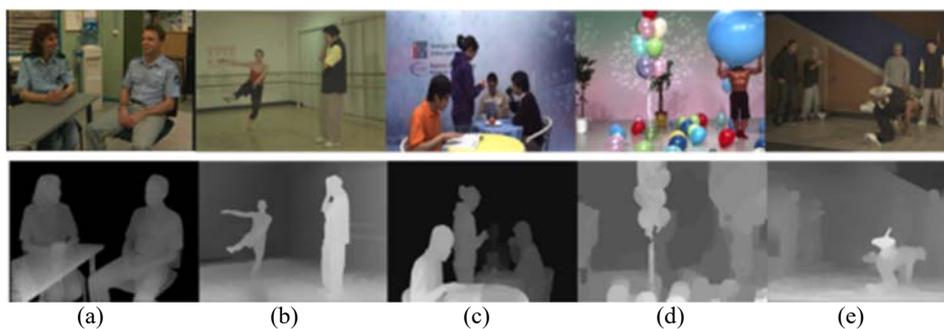


Fig. 10. Original and depth map of the five sequences.

Fig. 11 shows the depth maps of the sequence “Interview 720×576” produced by the symmetric Gaussian filter [8], asymmetric Gaussian filter [10], AEODF [13], AAEODF [15], and the proposed fast non-geometric smoothing method, respectively. Fig. 11 is the rendered left-view images using DIBR. The generated depth maps shown in Figs. 11 (d) and (e) have new depth discontinuities within the yellow squares, but the proposed method produces a more continuous and smooth image as shown in Fig. 11 (f). For ease of observation, the sections of the images in the yellow squares of Figs. 11 (d)-(f) are enlarged and presented in (g), (h), and (i), respectively.

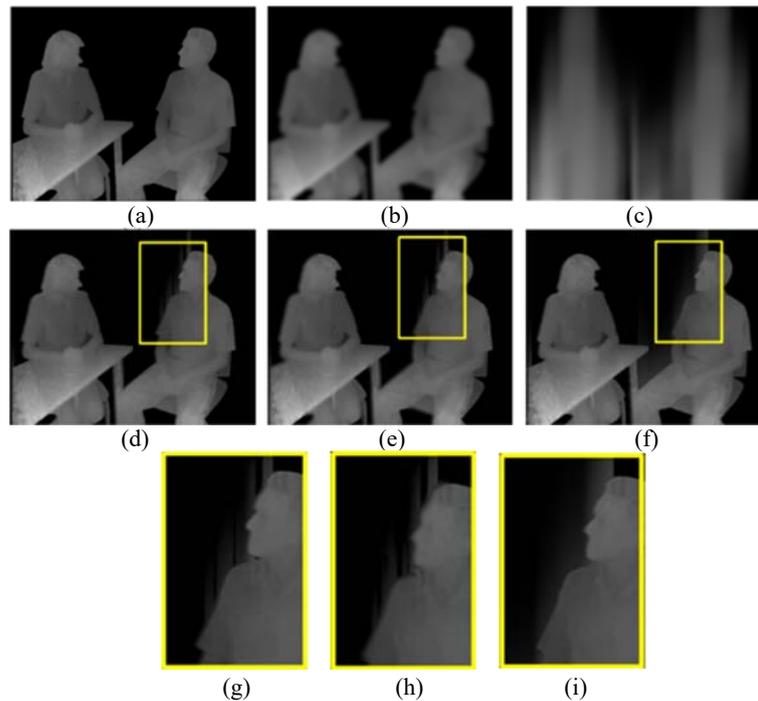


Fig. 11. Produced depth map of the sequence “Interview”; (a) Non-smoothing; (b) Symmetric Gaussian filter [8]; (c) Asymmetric Gaussian filter [10]; (d) AEODF [13]; (e) AAEODF [15]; (f) Proposed algorithm; (g), (h), and (i) enlarged detail of the AEODF [13], the AAEODF [15] and the proposed algorithm, respectively.

Fig. 12 (a) is the generated left-view image with the initial, unprocessed depth map, in which a rubber-sheet artifact occurred due to the large size of the hole. Fig. 12 (b) is the left-view image generated using symmetric smoothing, in which geometric distortion is produced due to the changed depth values at the boundaries in the background. The virtual view using the depth map generated through asymmetric Gaussian smoothing is shown in Fig. 12 (c), in which the geometric distortion is reduced in the left-view images. Because of the vertical smoothing area determination process, the left-view image generated by the proposed algorithm shown in Fig. 12 (f) is smoother than each of the left-view images generated by the AEODF and the AAEODF as shown in Figs. 12 (d) and (e), respectively. It is seen that new depth discontinuities generate when using AEODF and AAEODF.

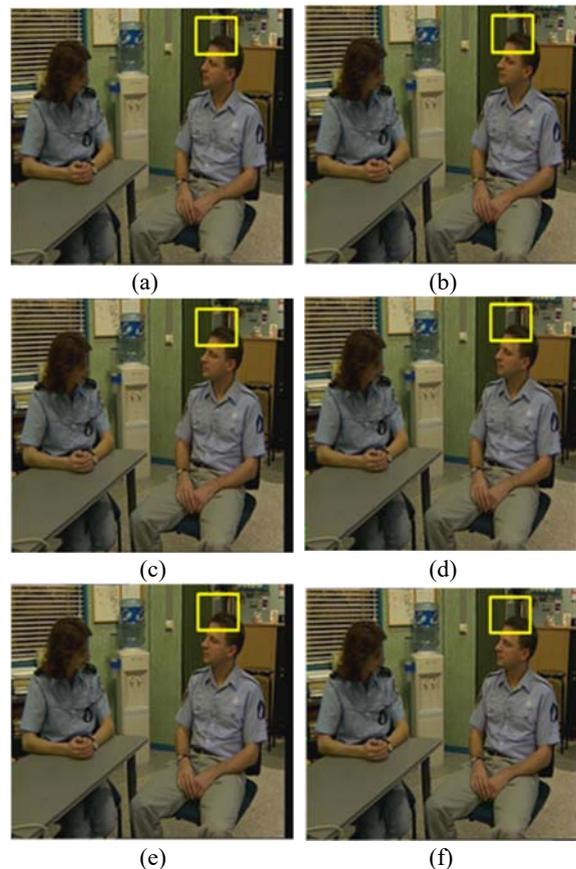


Fig. 12. Virtual left-view image of the sequence “Interview” produced from a processed depth map; (a) Non-smoothing; (b) Symmetric Gaussian filter [8]; (c) Asymmetric Gaussian filter [10]; (d) AEODF [13]; (e) AAEODF [15]; (f) Proposed algorithm.

Figs. 13-16 are side by side comparisons of left view synthesized images taken from the test sequences “Interview 1024×768”, “Break-dancer1024×768”, “Ballet1024×768”, and “Café1024×768”. Each of these figures shows an image generated using the original unprocessed depth map, as well as those generated using depth maps which have been preprocessed in the following five ways; the symmetric depth smoothing method, the asymmetric depth smoothing method, AEODF, AAEODF, and the proposed fast non-geometric smoothing method in DIBR, respectively. The rubber-sheet artifact is obvious when the initial depth map is used. These results are shown in Figs. 13-16 (a). Geometric distortion occurs in the images synthesized using the symmetric smoothing filter as shown in Figs. 13-16 (b). There is a significant reduction of geometric distortion when using the asymmetric smoothing method shown in Figs. 13-16 (c). Although the effect of the proposed algorithm is worse than the effect of the asymmetric Gaussian filter, the computation time of the proposed algorithm is less than that of [10] while the reduction of geometric distortion is more significant compared to [13, 15]. The geometric distortion is still generated when using AEODF or AAEODF algorithms. The results are shown in Figs. 13-16 (d)

and Figs. 13-16 (e). Because of the inaccurate vertical line detection, the asymmetric Gaussian filter used in AEODF and AAEODF methods are not applied to all of the hole regions which have vertical lines. However, the proposed method still reduces the geometric distortion because of the large vertical smoothing area near the vertical lines. The significant results are shown in Figs. 13-16 (f).

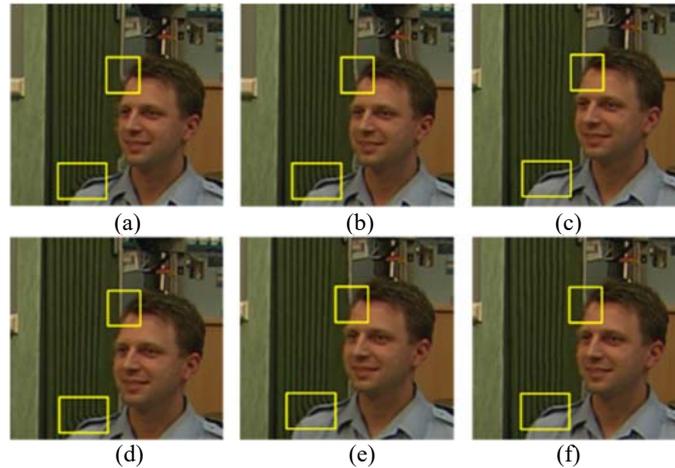


Fig. 13. Virtual left-view image of the sequence “Interview” produced from a processed depth map; (a) Non-smoothing; (b) Symmetric Gaussian filter [8]; (c) Asymmetric Gaussian filter [10]; (d) AEODF [13]. (e) AAEODF [15]; (f) Proposed algorithm.

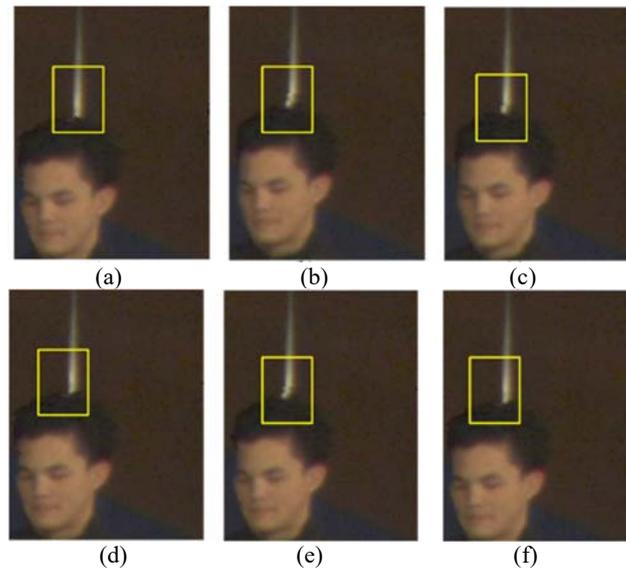


Fig. 14. Virtual left-view image of the sequence “Break-dancer” produced from a processed depth map; (a) Non-smoothing; (b) Symmetric Gaussian filter [8]; (c) AEODF [13]; (d) AAEODF [15]; (e) Asymmetric Gaussian filter [10]; (f) Proposed algorithm.

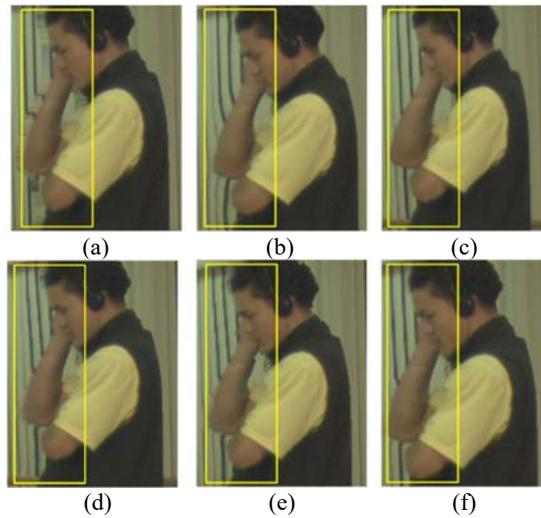


Fig. 15. Virtual left-view image of the sequence “Ballet” produced from a processed depth map; (a) non-smoothing; (b) Symmetric Gaussian filter [8]; (c) Asymmetric Gaussian filter [10]; (d) AEODF [13]; (e) AAEODF [15]; (f) Proposed algorithm.

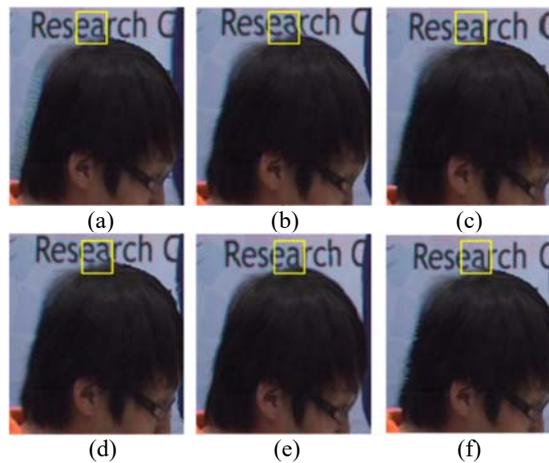


Fig. 16. Virtual left-view image of the sequence “Café” produced from a processed depth map; (a) non-smoothing; (b) Symmetric Gaussian filter [8]; (c) Asymmetric Gaussian filter [10]; (d) AEODF [13]; (e) AAEODF [15]; (f) Proposed algorithm.

Among the methods of depth map pre-processing, the asymmetric Gaussian filter has the best performance in solving the problem of geometric distortion. Therefore, the reference image for the objective scores comparison is the virtual view image generated by the asymmetric Gaussian filter. The objective scores of the generated virtual views are MSE and PSNR in Eqs. (12) and (13).

$$MSE_{method} = \frac{1}{m \times n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} |I_{asy}(x, y) - I_{[method]}(x, y)|^2 \quad (12)$$

$$PSNR_{[method]} = 10 \log_{10} \left( \frac{255^2}{MSE_{method}} \right) \quad (13)$$

The PSNR value ( $PSNR_{[method]}$ ) for each method is obtained by calculating the  $MSE_{method}$  between the generated depth map ( $I_{asy}(x, y)$ ) with asymmetric Gaussian filter [10] and the generated depth map ( $I_{[method]}(x, y)$ ) with symmetric Gaussian filter, AEODF, AAEOFD, and the proposed method. The illustration of the PSNR comparison is shown in Fig. 17.

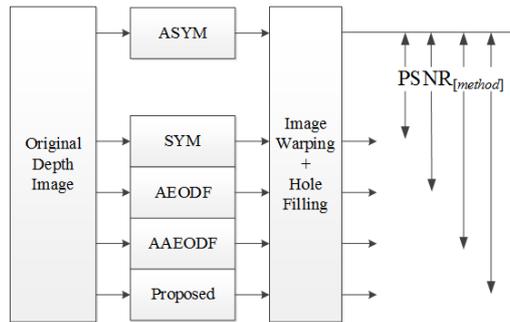


Fig. 17. Illustration of  $PSNR_{[method]}$ .

Table 1 is the PSNR comparison in which the reference image is the virtual view image generated using the asymmetric filter. The virtual views generated using the proposed algorithm have satisfactory results in the sequences “Interview”, “Ballet”, and “Café”. There are a lot of vertical lines in these three sequences, so the smoothing regions are larger than those when using AEODF or AAEOFD. The effects of the depth map pre-processing are more similar to those of the asymmetric Gaussian filter.

**Table 1. PSNR comparison.**

Algorithm	Interview	Interview	Ballet	Café	Balloon	Break dancer
	720×576	1024×768	1024×768	1024×768	1024×768	1024×768
Symmetric [8]	29.189	31.277	36.460	29.494	32.573	37.913
AEODF [13]	28.658	30.568	35.349	28.498	31.785	37.406
AAEOFD [13]	28.707	30.508	35.506	28.495	31.894	37.473
proposed	29.044	30.819	35.646	28.553	31.803	37.425

Because the PSNR of the virtual view image cannot absolutely express the effect of the 3D experience. The geometric distortion reduction of the different methods for depth map pre-processing, the calculations of the vertical lines and the skewness will also be compared.

The vertical line comparison is used to compare the geometric distortion in the regions with vertical lines. The concept of vertical line detection is adopted from [13].

In Eq. (14), pixels with a vertical vector are labeled  $label(x, y) = 1$ . Fig. 18 is an illustration of vertical line detection.  $O$  is the pixel at the location  $(x, y)$ ,  $a$  and  $b$  are the numbers of pixels through which the vertical vector continuously going upward and downward, respectively, corresponding to Eqs. (15) and (16).



Fig. 18. Vertical line detection.

$$label(x, y) = 1, \forall |p * G| > Th_3, \quad (14)$$

$$\begin{cases} a = a + 1, \forall label(x, y - k) = 1, \text{ for upward vertical vector} \\ b = b + 1, \forall label(x, y + k) = 1, \text{ for downward vertical vector} \end{cases}, \quad (15)$$

$$a + b > Th_4. \quad (16)$$

If the total amount of the pixels containing the vertical line is larger than the threshold  $Th_4$ , then the vertical line exists at the pixel  $O$ . In this paper, we use 5 sequences with completely different lighting and environmental conditions. In this paper, we use 5 test video sequences with different view parallax, vertical lines and high or low motion in picture. In Table 2, we use 3 different threshold value to do the testing. Set  $Th_3 = 40$  for the sequence Interview with size  $1024 \times 768$ ; the  $Th_3 = 60$  for the sequence Interview with size  $720 \times 576$ , Balloon and Breakdancer with size  $1024 \times 768$  and  $Th_3 = 75$  for sequence Ballet and Café with size  $1024 \times 768$ . Those thresholds are found to satisfy [14] such that the possibility is about 80%. The  $Th_4$  is set to 20. The higher the score, the better the effect on the geometric distortion.

**Table 2. Vertical line comparison.**

Algorithm	(1) 720×576	(1) 1024×768	(2) 1024×768	(3) 1024×768	(4) 1024×768	(5) 1024×768
Threshold ( $Th_3$ )	60	40	75	75	60	60
Symmetric [8]	17594	38816	17245	8681	12441	6156
Original	17301	38997	17373	8490	12137	6003
Asymmetric [10]	18032	39893	17828	8624	12465	6166
AEODF [13]	17556	39170	17383	8477	12167	6054
AAEODF[15]	17871	39137	17276	8428	12175	6045
proposed	17920	39085	17255	8562	12140	6045

(1): Interview      (2): Ballet      (3): Café      (4): Balloon      (5): Breakdancer

Table 2 shows the comparison results of vertical lines. The vertical view generated by the asymmetric Gaussian filter has the highest scores. The proposed method has nice scores in some sequences such as “Interview 720×576” and “Café1024×768”.

An illustration of the skewness comparison is shown in Fig. 19. The reference image is the initial image, and the compared images are the virtual view images generated by different methods.

The equations are as follows.

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}, \theta = \arctan(m), \Delta\theta = |\theta_{ori} - \theta_{method}|, \quad (17)$$

where  $m$  is the gradient,  $\theta_{ori}$  is the angle of the reference image which provides from test sequence, and  $\theta_{method}$  is the angle of the generated virtual view image by using difference DIBR methods.

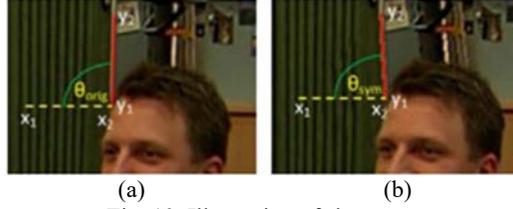


Fig. 19. Illustration of skewness.

Table 3 is a skewness comparison. The smaller the angle, the less geometric distortion occurs in the virtual view image. The angle is always the smallest in the proposed algorithm. The results shown in Table III mean that geometric distortion is reduced much more by the proposed algorithm than by the other algorithms. But, the result of sequence (without vertical lines) Ballet  $1024 \times 768$  by the proposed method is not better than Asymmetric Gaussian filter-based algorithm.

**Table 3. Skewness comparison.**

Algorithm	(1) 720×576	(1) 1024×768	(2) 1024×768	(3) 1024×768	(4) 1024×768	(5) 1024×768
Symmetric [8]	10°	15°	15°	25°	0°	15°
Asymmetric [10]	1°	1°	2°	0°	0°	5°
AEODF [13]	1°	1°	5°	0°	0°	0°
AAEODF [15]	1°	2°	5°	15°	0°	10°
proposed	1°	1°	5°	0°	0°	0°
(1): Interview	(2): Ballet	(3): Café	(4): Balloon	(5): Breakdancer		

In this paper, the main goal is reducing the vertical line geometric, because of the vertical line is one of un-comfortable reason for 3D viewing. From the simulation result, the proposed method has the lowest computation time in all of the  $1024 \times 768$  test sequences except “Café  $1024 \times 768$ ”. This is due to the extended area in which the smoothing filter is applied despite the fact that there are actually fewer vertical lines detected in this sequence. However, even though the computation time is higher than those in [13, 15], the scores of the vertical line and skewness are also higher than those in [13, 15]. In this paper, because the one-dimensional vertical smoothing mask and the vertical smoothing region are only at the boundary of the hole regions without vertical lines, and in the background, the computed pixels for the hole regions with vertical lines are reduced. It is seen that the computational time of the proposed algorithm is reduced about 90~95% compared to the asymmetric Gaussian filter [10]. Table 4 is the computation time comparison in the unit of second.

In the proposed algorithm, the region to be smoothed is divided into non-hole regions, hole regions without vertical lines, and hole regions containing vertical lines. In hole regions without vertical lines, the adaptive horizontal Gaussian filter is proposed. In the hole regions containing vertical lines, the vertical Gaussian filter is proposed. This is due to the extended area where the smoothing filter is applied despite there are actually fewer vertical lines detected in this sequence. Since the smoothing filter is applied in the extended area, even there are actually fewer vertical lines detected in this sequence. In hole regions without vertical lines, the adaptive horizontal Gaussian filter is proposed. In the hole regions containing vertical lines, the vertical Gaussian filter is proposed.

Due to the above reason, from the simulation result in Table 4 shows that the sequence “Café1024×768” does not have the lowest computation time.

Table 5 presents the Mean Opinion Score (MOS) comparison. The MOS comparison with different algorithms for different sequences is shown in Figs. 20 (a)-(e). The 3D experience of the virtual view images generated by the proposed method is the most comfortable for the human visual system in the sequences “Café” and “Break-dancer”.

**Table 4. Computation time (second) comparison.**

Algorithm	(1)	(1)	(2)	(3)	(4)	(5)
	720×576	1024×768	1024×768	1024×768	1024×768	1024×768
Symmetric [8]	85	180	180	180	184	179
Asymmetric [10]	1188	2320	2524	2375	2321	2359
AEODF [13]	356	858	1653	68	3	932
AAEODF [15]	357	249	1263	24	15	230
proposed	100	120	89	79	1	8

(1): Interview (2): Ballet (3): Café (4): Balloon (5): Breakdancer

**Table 5. MOS comparison.**

Algorithm	Interview	Ballet	Café	Balloon	Break dancer
	1024×768	1024×768	1024×768	1024×768	1024×768
original	6.75	5	7.25	6.25	5.5
Symmetric [8]	7.5	6	7.5	7	6
Asymmetric [10]	<b>7.25</b>	<b>6</b>	<b>7.75</b>	<b>7</b>	<b>5.75</b>
AEODF [13]	6.25	5	7	6.5	5
AAEODF [15]	6	6.5	7.25	6.25	5.75
proposed	7	6.25	7.5	6.75	6.25

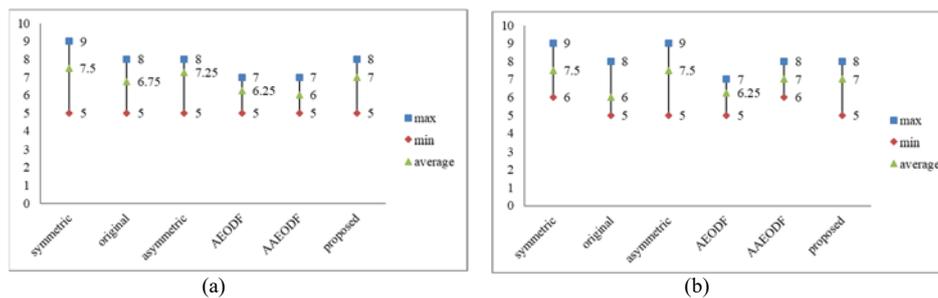


Fig. 20. MOS comparison with different algorithms; (a) and (b) for Interview; (c) and (d) Ballet; (e) and (f) for Café; (g) and (h) for Balloon; (i) and (j) for Break-dancer.

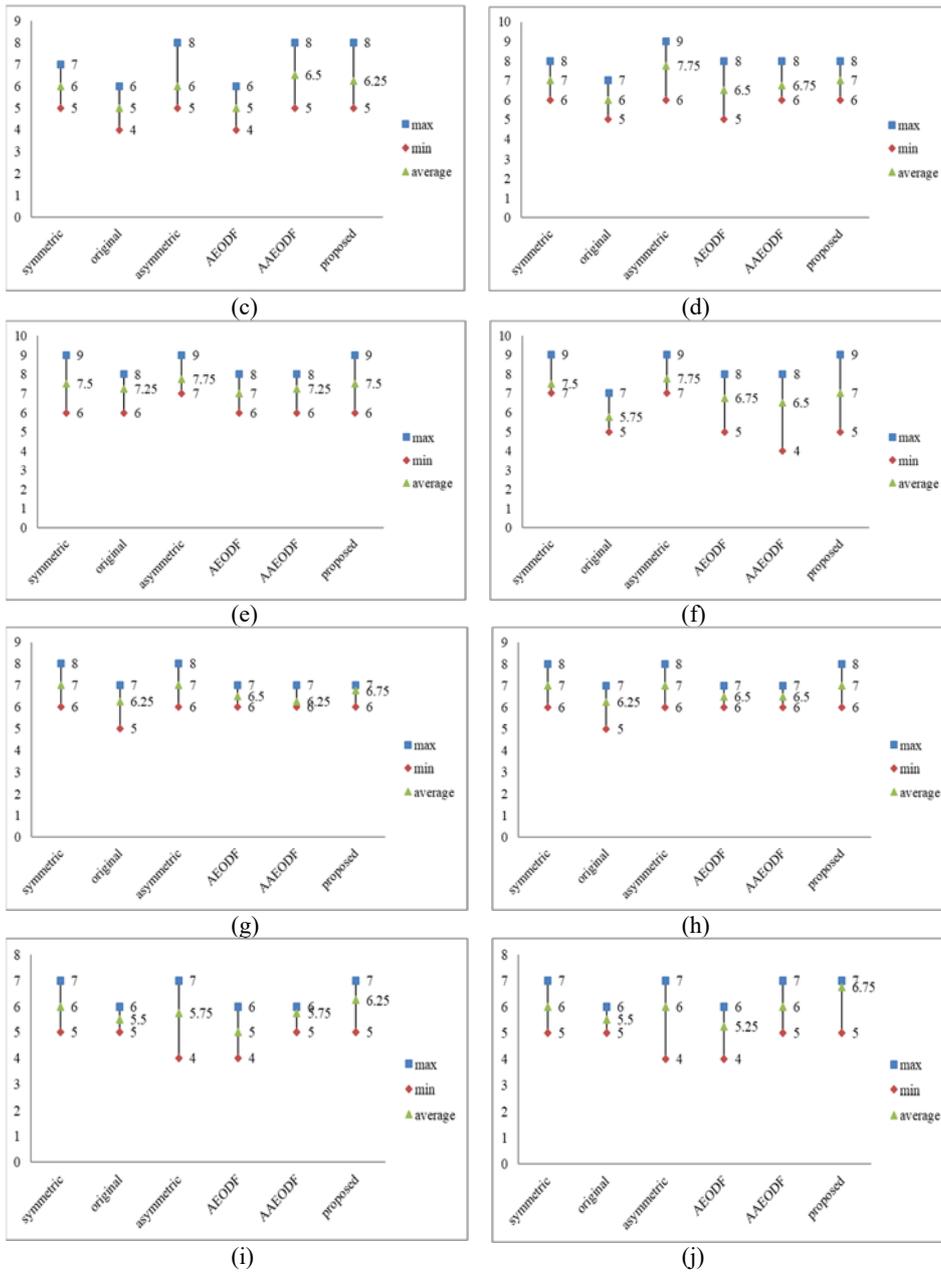


Fig. 20. (Cont'd) MOS comparison with different algorithms; (a) and (b) for Interview; (c) and (d) Ballet; (e) and (f) for Café; (g) and (h) for Balloon; (i) and (j) for Break-dancer.

From the simulation results, the proposed algorithm provides better performance on geometric distortion reduction on the synthesized views and reduces the computational time most significantly on depth preprocessing.

## 5. CONCLUSION

DIBR is a popular technology for advanced 3D-TV systems. Depth map pre-processing is always used to improve the quality of the synthesized virtual view. Based on region classification, this paper has proposed an asymmetric filter to solve the geometric distortion and the high computation time problems in the depth map pre-processing. The depth map can be classified into 3 regions in the proposed algorithm. The adaptive horizontal Gaussian filter is proposed to reduce the hole size in the generated virtual view's hole regions. Therefore, the 3D experience is preserved in the non-hole regions and the computation time is reduced significantly. The vertical Gaussian filter is proposed to reduce the geometric distortion in the hole regions containing vertical lines. The size of the region to be smoothed by the proposed vertical smoothing filter is decreased by the vertical smoothing region determination.

Therefore, the proposed algorithm can preserve the 3D experience in each or in different regions and reduce the computational time in depth preprocessing. Experiment results have shown that the computation time is reduced about 90~95% compared with the asymmetric Gaussian filter. Though the computation time is decreased, the 3D quality of the synthesized virtual views is still preserved.

## REFERENCES

1. P. Dias, R. Silva, P. Amorim, J. Lains, E. Roque, I. Serôdio, F. Pereira, and B. S. Santos, "Using virtual reality to increase motivation in poststroke rehabilitation," *IEEE Computer Graphics and Applications*, Vol. 36, 2019, pp. 64-70.
2. P.-J. Lee, T.-A. Bui, and C. Su, "Artifact reduction based multi-view generation algorithm for sparse camera configuration," *Journal of Information Science and Engineering*, Vol. 35, 2019, pp. 23-39.
3. A. L. S. Kawamoto and F. S. C. D. Silva, "Depth-sensor applications for the elderly: A viable option to promote a better quality of life," *IEEE Consumer Electronics Magazine*, Vol. 7, 2018, pp. 47-56.
4. L. Ahn and C. Kim, "Depth-based disocclusion filling for virtual view synthesis," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2012, pp. \_\_\_.
5. Y. Mao, G. Cheung, and Y. Ji, "On Constructing z-dimensional DIBR-Synthesized Images," *IEEE Transactions on Multimedia*, Vol.18, 2016, pp. 1453-1468.
6. T. Tezuka, M. P. Tehrani, K. Suzuki, K. Takahashi, and T. Fujii, "View synthesis using superpixel based inpainting capable of occlusion handling and hole filling," in *Proceedings of Picture Coding Symposium*, 2015, pp. 124-128.
7. M. Schmeing and X. Jiang, "Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting," *IEEE Transactions on Multimedia*, Vol. 17, 2015, pp. 2160-2173.
8. W. J. Tam, G. Alain, L. Zhang, T. Martin, R. Renaud, and D. Wang, "Smoothing depth maps for improved stereoscopic image quality," in *Three-Dimensional TV, Video, and Display III*, B. Javidi and F. Okano, Vol. 5599, 2004, pp. 162-172.
9. C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th International Conference on Computer Vision*, 1998, pp. 839-846.

10. L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Transactions on Broadcasting*, Vol. 51, 2005, pp. 191-199.
11. I. Daribo and H. Saito, "Bilateral depth-discontinuity filter for novel view synthesis," in *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, 2010, pp. 145-149.
12. Y. M. Feng, D. X. Li, K. Luo, and M. Zhang, "Asymmetric bidirectional view synthesis for free viewpoint and three-dimensional video," *IEEE Transactions on Consumer Electronics*, Vol. 55, 2009, pp. 2349-2355.
13. P. J. Lee and Effendi, "Nongeometric distortion smoothing approach for depth map preprocessing," *IEEE Transactions on Multimedia*, Vol. 13, 2011, pp. 246-254.
14. W. Y. Chang, Y. L. Lin, S. F. Ding, L. F. Chen, and L. G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2005, pp. 1314-1317.
15. C. H. Hsia, "Improved depth image-based rendering using an adaptive compensation method on an autostereoscopic 3-D display for a kinect sensor," *IEEE Sensors Journal*, Vol. 15, 2015, pp. 994-1002.
16. Y. Ren, L. Sun, G. Wu, and W. Huang, "DIBR-synthesized image quality assessment based on local entropy analysis," in *Proceedings of IEEE International Conference on the Frontiers and Advances in Data Science*, 2017, pp. 86-90.
17. Y. Ren, L. Sun, G. Wu, and W. Huang, "DIBR-synthesized image quality assessment based on local entropy analysis," in *Proceedings of IEEE International Conference on the Frontiers and Advances in Data Science*, 2017, pp. 86-90.
18. C. Fehn, K. Schüür, I. Feldmann, P. Kauff, and A. Smolic, "Distribution of ATTEST test sequences for EE4 in MPEG 3DAV," ISO/IEC JTC1/SC29/WG11, M9219 doc., 2002.
19. 3D Video Coding Technology ISO/IEC JTC1/SC29/WG11, MPEG2011/N12036, Geneva, Switzerland, March 2011.



**Pei-Jun Lee (李佩君)** received the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University in 2004. Since August 2015, she has been Distinguished Professor of the Department of Electrical Engineering, National Chi Nan University, Taiwan. She became a Senior Member of IEEE in 2014. In 2008, she was the recipient of FIRST RUNNER-UP and outstanding advisor of the Macronix Golden Silicon Awards Semiconductor Design and Application Competition from MXIC Co., Ltd. Her research interests include 2D/3D image conversion, 3D/multi-view video coding, video communication, video compression, image processing, robot fish and FPGA system application.



**Hong-Li Lin (林宏祿)** received the MS degree from Department of Electronic Engineering National Chi Nan University in the Master of Electric Engineering at National Chi Nan University in 2015. His research interest's fast depth map pre-processing.



**Trong-An Bui (裴重恩)** is a Ph.D. student in National Chi Nan University, Taiwan. In 2015, he graduated from Ho Chi Minh University of Education, Vietnam with the B.Sc. degree from the Faculty of Information Technology. In 2018, he also received M.S. degree with a research topic in Image Processing, National Chi Nan University, Taiwan. His current research interests include neural networks and artificial intelligence, computer vision and image processing, video coding and data compression, satellite and UAV.