Direction hole-filling method for a 3D view generator

Shwu-Huey Yen*, Hung-Ying Chai and Jen-Hui Tsao

Department of Computer Science and Information Engineering,
Tamkang University,
151 Ying-chuan Road, Tamsui,
New Taipei City 25137, Taiwan
Email: 105390@mail.tku.edu.tw
Email: 698420485@s98.tku.edu.tw
Email: 700410268@s00.tku.edu.tw
*Corresponding author

Abstract: Depth image-based rendering (DIBR) technology is an approach to creating a virtual 3D image from one single 2D image. A desired view can be synthesised at the receiver side using depth images to make transmission and storage efficient. While this technique has many advantages, one of the key challenges is how to fill the holes caused by disocclusion regions and wrong depth values in the warped left/right images. A common means to alleviate the sizes and the number of holes is to smooth the depth image. But smoothing results in geometric distortions and degrades the depth image quality. This study proposes a hole-filling method based on the oriented texture direction. Parallax correction is first implemented to mitigate the wrong depth values. Texture directional information is then probed in the background pixels where holes take place after warping. Next, in the warped image, holes are filled according to their directions. Experimental results showed that this algorithm preserves the depth information and greatly reduces the amount of geometric distortion.

Keywords: depth-image-based rendering; DIBR; hole-filling; 3D image; parallax.

Reference to this paper should be made as follows: Yen, S-H., Chai, H-Y. and Tsao, J-H. (2015) 'Direction hole-filling method for a 3D view generator', *Int. J. Computational Science and Engineering*, Vol. 11, No. 2, pp.167–175.

Biographical notes: Shwu-Huey Yen is currently an Associate Professor at the Computer Science and Information Engineering (CSIE) Department of Tamkang University, New Taipei City, Taiwan. She is also an author of over 50 journal papers and conference papers. Her academic interests are signal processing, multimedia processing and medical imaging.

Hung-Ying Chai received his Master degree in 2012 from CSIE, Tamkang University. He is serving the military service right now. His research interest is in image processing.

Jen-Hui Tsao is a graduate student in CSIE, Tamkang University. He is currently pursuing a Master degree and also a Software Engineering at YongMing Technology Consulting Company primarily responsible for the laboratory information system and electronic medical record system development. His research interest is in image processing.

1 Introduction

Three-dimensional images have gained popularity recently. Many movies nowadays are recorded and presented in stereoscopic 3D format. They bring a more life-like and visually immersive experience. People are no longer satisfied by the current 2D images or video contents and would like to pursue more visual effects through 3D images or videos.

A typical 3DTV system is composed of six main components: 3D image capturing and content, 3D content image coding, transmission, decoding the received sequences, generating virtual views, and displaying the stereoscopic images on the screen (Redert et al., 2002; Flack et al., 2003; Fehn et al., 2004). The stereoscopic images must be synthesised by at least two (left and right eye) images or multi-images which makes transmission and storage very inefficient for a real-time video system. Consequently, the European Information Society Technologies (European IST) started the Advanced Three-dimensional Television at the System Technologies (ATTEST) project (Redert et al., 2002). They proposed a depth image-based rendering (DIBR) technology in which the multiple images for 3D image synthesis can be generated from a single 2D image and the corresponding depth image (Fehn, 2004).

The DIBR scheme is based on preprocessing of the depth image which includes the following procedure: First, every pixel in the original colour image uses a corresponding depth map to calculate the new location in the virtual (left or right) images. Then, based on these parallax values, pixels are shifted to the virtual image. This procedure is called 'image warping'. As a result, the DIBR can not only reduce the transmission time and storage space but also synthesise any 3D image as if they were captured from several points of view.

However, DIBR technology has two serious problems in the warping step (Curti et al., 2002). The first problem is the occlusion problem. Occlusion occurs when two pixels are shifted to the same pixel location in the virtual image. The occlusion problem is resolved by choosing the pixel which has the least distance from the viewer (Nguyen et al., 2009). The other problem is the disocclusion problem. Owing to sharp depth transition in the depth image, newly exposed areas (so called 'holes') appear in the virtual image after warping. There are two common approaches to remove the disocclusion. One is to fill the disocclusion by using neighbour colour information such as interpolation, extrapolation (Vazquez et al., 2006), mirroring of background colour, or image in painting (Cheng et al., 2008). The other is to preprocess the depth image before image warping such as with Gaussian smoothing (Tam et al., 2004; Tam and Zhang 2004; Zhang and Tam, 2005). Although these methods can reduce the disocclusion areas, geometric distortion will appear in virtual images. This study proposes a directional hole-filling method based on the DIBR system for 3D views. The method also preprocesses the depth image; however, it is computationally efficient and effectively eliminates the disocclusion areas with most of the original depth information preserved.

The DIBR technology is introduced in Section 2 and the proposed method is described in Section 3. Section 4 presents some experiment results and a conclusion is given in Section 5.

2 Related work

As in the DIBR system, when the depth map and one monocular image are received, two or more virtual images are to be generated to reconstruct the 3D view. For the sake of simplicity, we assume that the original image is an intermediate image and the 3D image is a binocular one with the virtual left and the virtual right image half shifted. Also, the construction of the virtual right image is similar to that of the virtual left image. Therefore, we only discuss the construction of the virtual left image.

2.1 Parallax values

To display a 3D image on a dedicated screen, we have to evaluate the parallax value and the relative number of pixels to be shifted. Given one point P with the depth Z in the world, P is projected onto the image plane with the formula for 3D image warping described as follows:

$$x_l = x_c + \frac{1}{2}\left(\frac{t_x \cdot f}{Z}\right) \text{ and } x_r = x_c - \frac{1}{2}\left(\frac{t_x \cdot f}{Z}\right), \qquad (1)$$

where x_c is the coordinate of an intermediate image, x_l and x_r are the new coordinates in the virtual left and right image according to the depth value Z. t_x represents the baseline distance, usually equal to the interocular distance, and f is the focal length of the camera. In general, t_x and f are set to be constant. Therefore the disparity $\left(\frac{t_x \cdot f}{Z}\right)$ in (1) can also be expressed in the unit of the pixel as:

$$total\_parallax = M \cdot \left(1 - \frac{depth\_value}{255}\right), \qquad (2)$$

where M is the maximum parallax in the pixel, and *depth_value* is the grey value in the depth map. As in Figure 2(a), the depth map is a grey-valued image where each grey value is associated with a depth value. A point with a smaller (larger) grey value corresponds to a position farther (nearer) from the viewer. When the parallax value of (2) for each pixel is evaluated, the virtual left images can be created shifting the input image pixels by a half of (*total_parallax*). For the sake of simplicity, let *parallax* = (*total_parallax*)/2. Several studies (Redert et al., 2002) have suggested that the maximum parallax value should be within 5% of the width of a standard 4 × 3 image to have a comfortable view. Since *parallax* value is determined by the depth value and the width of the screen, we use *parallax* as the parameter in our method.

2.2 Preprocessing of depth image

As mentioned, the problems of occlusion and disoccolusion occur when pixels shift. The latter causes some positions in the virtual image to have no values. These 'holes' appear between two objects with different parallax values as shown in Figure 1. Many studies have been dedicated to solving the 'holes'. Tam et al. (2004) proposed a symmetric Gaussian smoothing filter to reduce the size and the number of holes. Figure 2 shows symmetric smoothing results for the sequence 'Interview'. However, this method results in geometric distortions in the virtual image, as shown in Figures 2(e) to 2(f). To improve upon this drawback, another study (Zhang and Tam, 2005) proposed an asymmetric Gaussian smoothing filter in which the vertical standard deviation (σ_v) is larger than the horizontal standard deviation (σ_μ) as shown in Figure 3. We can notice that the warped image has a good subjective quality, avoiding most of the geometric distortion and eliminating or reducing the holes. As suggested in their studies, the standard deviations in the Gaussian filters are $\sigma_\mu = \sigma_v = 30$ (Tam et al., 2004) and $\sigma_\mu = 10$, $\sigma_v = 90$ (Zhang and Tam, 2005) for the symmetric and asymmetric ones, respectively. The window size is usually three times that of the standard deviation. The computation cost is a concern since the smoothing is applied on every point in the depth map. In addition, the depth information is seriously degraded after smoothing. Some other proposed edge dependent methods are described as follows. In order to preserve the visual quality, these methods modify the depth map only on the boundary of two objects with different parallax values where holes would be present. Chen et al. (2005) proposed an edge dependent depth-based symmetric smoothing filter, as shown in Figure 4(a). Lee and Effendi (2011) proposed an edge-oriented method using both smoothing filters as shown in Figure 4(b). Their method first predicts locations of holes. An asymmetric Gaussian filter is used if there is enough vertical texture for a 'hole' region, and a symmetric Gaussian filter is used for the rest of the 'hole' regions.

Figure 1 The disoccolusion schematic diagram (see online version for colours)

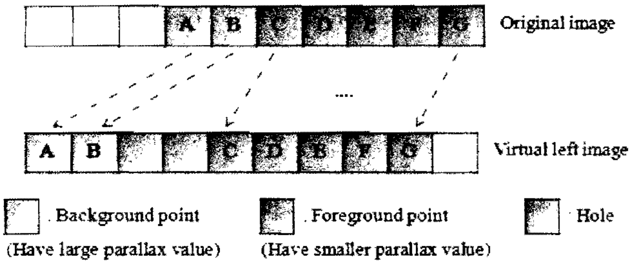


Figure 2 Symmetric smoothing preprocessing. (a) and (b) are the original depth map and image: (c) symmetric smoothed depth map: (d) virtual left view after 3D image warping: (e) and (f) geometric distortions shown in enlarged segments from (d)

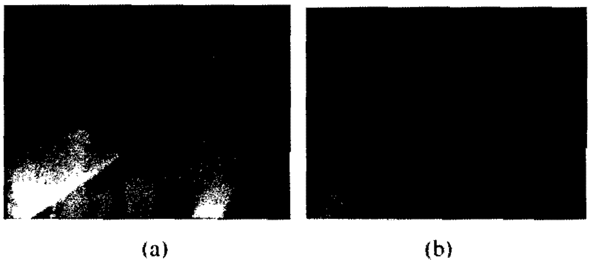
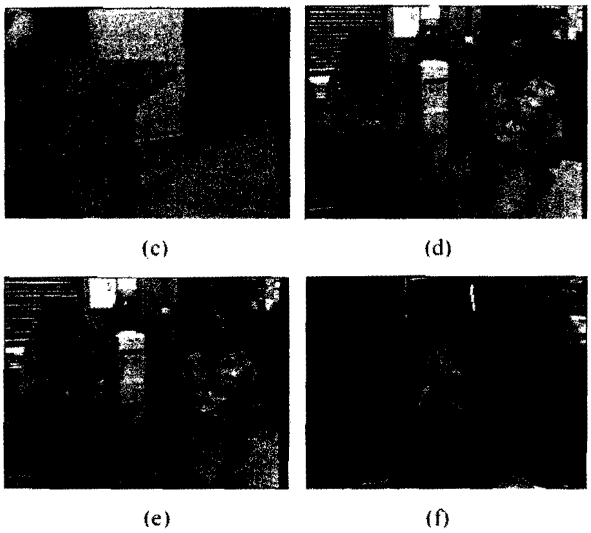
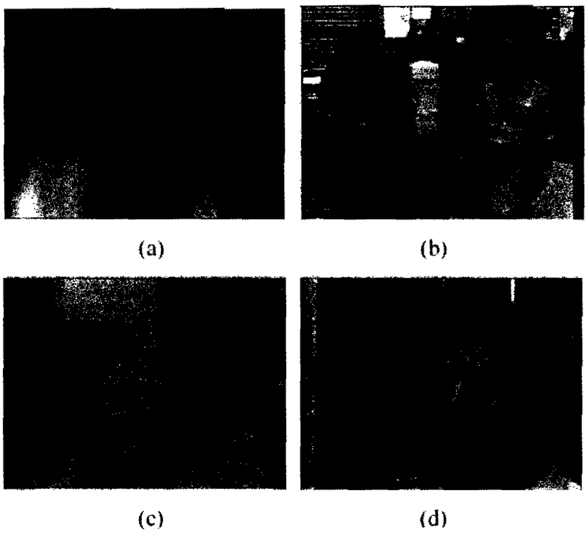

(a) (b)

Figure 2 Symmetric smoothing preprocessing. (a) and (b) are the original depth map and image: (c) symmetric smoothed depth map: (d) virtual left view after 3D image warping: (e) and (f) geometric distortions shown in enlarged segments from (d) (continued)


(c) (d)

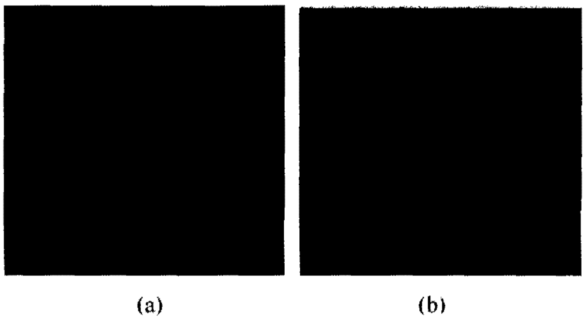
(e) (f)

Figure 3 Asymmetric smoothing preprocessing. (a) smoothed depth map: (b) virtual left view, (c) and (d) are the enlarged segments from (b) [as in Figure 3(e) and 3(f)] for comparison


(a) (b)

(c) (d)

Figure 4 Smoothed segments of the 'interview' depth map by the method of (a) Chen et al. (2005), and (b) Lee and Effendi (2011)


(a) (b)

3 Proposed method

To improve upon the shortcomings of smoothing depth map, we propose a direction hole-filling method. The method includes five steps: *parallax correction, hole region detection, texture detection, texture enhancement* and *hole-filling*. Figure 5 shows the diagram of the proposed scheme. For two adjacent points, P_1 and P_2 with P_1 is on the left of P_2, we say P_1 is a background pixel and P_2 is a foreground pixel (relatively) if P_1 has larger *parallax* than P_2 does. In fact, there would be a hole between P_1 and P_2 after warping.

3.1 Parallax correction

A depth image obtained from a database usually has some noise caused by absence of depth information as shown in Figures 6(a) and 6(c). These noise points with wrong depth value result in large *parallax* values and cause artefacts in a warped image.

Figure 5 Block diagram of the proposed scheme (see online version for colours)

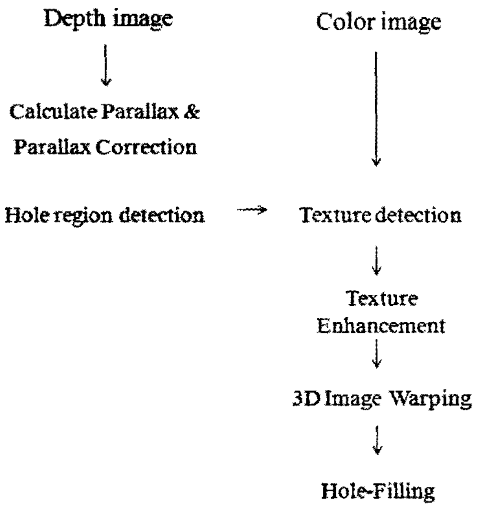
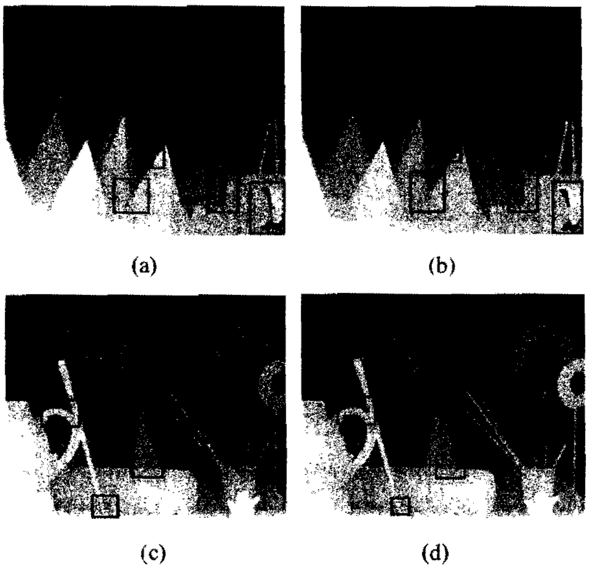
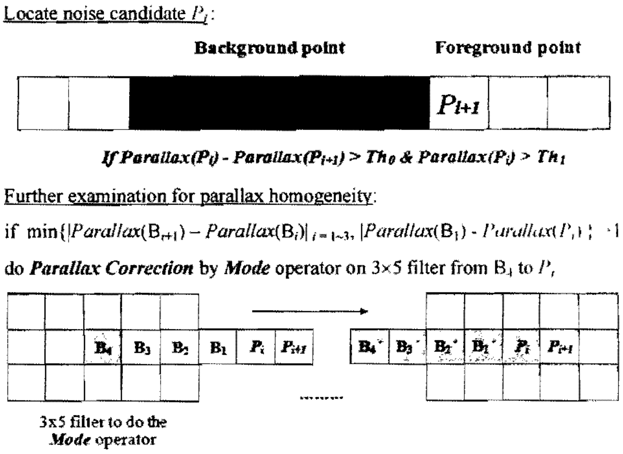


Figure 6 The parallax correction, (a) and (c) are before, and (b) and (d) are after (see online version for colours)



Observing the *parallax* values of image points row-wise, most of points are of the same value except those places where there are boundaries between the background area and foreground area. Based on this observation, we first identify that those points may be noise, and correct the *parallax* value by the homogeneity property if they indeed are noises. The method is explained as follows. If P_i is a background point relative to its right neighbour with large *parallax*, examination of the *parallax* values of the four immediate left neighbours will be executed. If P_i is not a noise, then its four immediate left neighbours should have the similar *parallax* as P_i's. On the other hand, if this is not the case, then correction of *parallax* should be done. To make the correction, a mode operator on 3 × 5 filter is applied to each of the four immediate left neighbours and P_i. Figure 7 depicts the correction scheme. To view the effect of *parallax correction*, we convert the modified *parallax* value back to the depth value, as shown in Figure 6(b) and Figure 6(d).

Figure 7 The parallax correction scheme (see online version for colours)


Locate noise candidate P_i:
Background point
Foreground point
P_{i+1}
If Parallax(P_i) - Parallax(P_{i+1}) > Th_0 & Parallax(P_i) > Th_1
Further examination for parallax homogeneity:
if min{|Parallax(B_{i+1}) – Parallax(B_i)|$_{i=1\sim3}$, |Parallax(B_1) - Parallax(P_i)|} > 1
do Parallax Correction by Mode operator on 3×5 filter from B_4 to P_i
B_4
B_3
B_2
B_1
P_i
P_{i+1}
B_4'
B_3'
B_2'
B_1'
P_i
P_{i+1}
3x5 filter to do the
Mode operator

3.2 Hole region detection

The homogeneous area in the depth image would not cause holes after 3D Image warping. In other words, the holes are usually near sharp depth discontinuity areas like object boundaries. For the virtual left view, holes occur on the position where *parallax* values are in transition from large to small. Thus, we can use formula (3) to predict the hole pixel after 3D image warping and label this pixel as $Hole(x, y) = 1$:

$$Hole(x,y)=\begin{cases}1, & \begin{array}{l} parallax(x+1,y)-parallax(x,y)>Th_2 \\ \quad\text{(for the left virtual view)} \\ parallax(x,y)-parallax(x-1,y)>Th_2 \\ \quad\text{(for the right virtual view)}\end{array} \\ 0, & \text{otherwise}\end{cases} \tag{3}$$

where the *parallax*(*x*, *y*) is the parallax value at position (*x*, *y*) which equals to (*total_parallax*)/2 from the formula (2).

3.3 Texture detection

After *hole region detection* step, we need to detect the texture feature and direction around of the hole neighbours using edge information. The texture features are divided into four directions: 0° (horizontal), 90° (vertical), 45°, and 135°. And they are determined by applying four directional *Sobel* filters on the original image.

Assuming P is located on (x, y) with $Hole(x, y) = 1$, let $\boldsymbol{k}$ be the strongest response direction ($\boldsymbol{k} \in \{0°, 90°, 45°, 135°\}$) determined of P by four directional Sobel operators. Due to the digitalised effect on natural images, a vertical line may have points located on its adjacent left or right) column. Therefore, the vertical (90°) response of a point (x, y) is the maximum of $\{|Sobel_{90}(x-1, y)|, |Sobel_{90}(x, y)|, |Sobel_{90}(x+1, y)|\}$. Similarly, the response on 45° texture is the maximum of $\{|Sobel_{45}(x-1, y-1)|, |Sobel_{45}(x, y)|, |Sobel_{45}(x+1, y+1)|\}$. A similar argument applies for determining texture in the directions of 0°, 135° as well.

After the strongest response $\boldsymbol{k}$ is determined, we examine whether this texture is sufficient on P. Along the direction $\boldsymbol{k}$, n consecutive points are examined. Let Q be such a point to be examined. First, only background points will be examined. This is done by

$$(parallax(Q)) \leq (parallax(P) - 1). \tag{4}$$

Table 1 The four pos/neg searching directions and corresponding points to be examined ($i = 1 \sim n$)

	0°	*90°*	*45°*	*135°*
Pos. direction	→	↑	↗	↘
Points	A_i	B_i	C_i	D_i
Neg. direction	←	↓	↙	↖
Points	A_{-i}	B_{-i}	C_{-i}	D_{-i}

Next, check whether Q also has a strong texture response on the direction $\boldsymbol{k}$. If both conditions are satisfied, then label $Se_k(Q) = 1$. As in Lee and Effendi (2011), if the sum of Se_k on these n points is large enough, then we record the texture on the hole $\boldsymbol{P}$ as $\boldsymbol{k}$. In particular, one texture may appear in one of the two directions. For example, when $\boldsymbol{k} = 90°$, the vertical texture may occur upward of P or downward of P. Thus, we examine n consecutive points in two passes for both directions. Table 1 summarises which points to be examined where $A_{1 \sim n}$ and $A_{(-1) \sim (-n)}$ represent two examination directions (positive and negative) of 0°, and the same to B_i, C_i, D_i for 90°, 45°, 135° textures, respectively.

Figure 8 describes the procedure of determining whether a hole point P possesses a texture property. To be more specific, we use $texture(P) = \boldsymbol{k}_1$ (or $\boldsymbol{k}_2$) to denote that a hole point P has a texture on the positive (or negative) $\boldsymbol{k}$ direction. Finally, assign $texture(P) = \delta$ if P cannot fulfil the requirement. Formula (5) summarises the procedure.

$$texture(P) = \begin{cases} k_1 & \sum_{i=1}^{n} Se_k(Q_i) > Th_4, \\ k_2 & \sum_{i=1}^{n} Se_k(Q_{-i}) > Th, \\ \delta & \text{otherwise} \end{cases} \tag{5}$$

where $(k, Q_i) \in \{(0°, A_i), (90°, B_i), (45°, C_i), (135°, D_i)\}$.

3.4 Texture enhancement

After texture detection, the vertical texture will be reinforced because human vision is more sensitive to vertical texture. In the following texture enhancement procedure, the comparison of textures is based on four directions only, regardless of whether the textures are positive or negative. For a given hole point P with $texture(P) = 90°$, if A and B are two hole-points above and below P, respectively, the following check up will be executed:

Case 1 texture(A) = texture(B) ≠ 90°

Starting from B, if these five consecutive points below P are hole-points, then their texture properties will be examined. If they all have the same texture, then $texture(P)$ is set to be the same as $texture(A)$ for the homogeneity of the texture in the neighbouring points.

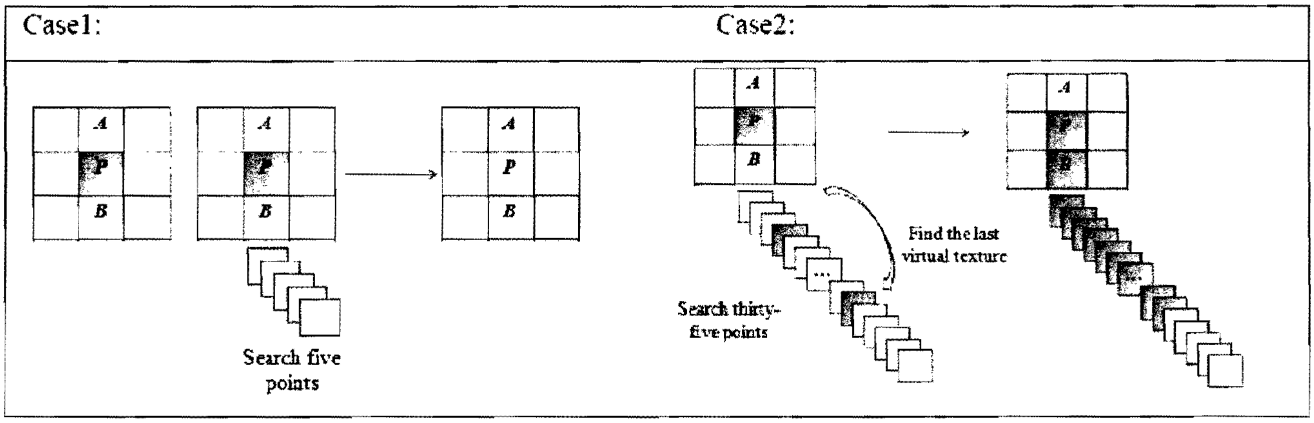
Case 2 texture(A) ≠ texture(B)

Starting from B, we look for downwards hole-points that have similar texture to P's (i.e., $texture = 90°$). If there are r consecutive hole-points below P and Q is the farthest one found that has the same texture as P's, we change the texture for every point between P and Q to be the same as $texture(P)$. According to our tests, check for 35 points below P gives satisfactory results. Figure 9 shows the *texture enhancement* scheme. We only check the points downwards since the texture enhancement is done in raster-scan order processing.

Figure 8 The texture search processing ($Th_2 = 1$)

```
Repeat the following process twice, one for i = 1 to n and the other for i = –1 to –n
    if   k = 0      then Qi = Ai     //horizontal direction
    if   k = 90     then Qi = Bi     //vertical direction
    if   k = 45     then Qi = Ci     //diagonal towards northeastern (southwestern) direction
    if   k = 135    then Qi = Di     //diagonal towards southeastern (northwestern) direction
    if $Parallax(Q_i) \geq (Parallax(P) - Th_2)$ and $|Sobel_k(Q_i)| > Th_3$ then $Se_k(Q_i) = 1$
```

Figure 9 The texture enhancement scheme (see online version for colours)



3.5 Hole filling

The virtual left image I' now can be created according to the parallax value on each point. Let P' be the point on I' corresponding to P of the original image I and the texture/parallax information of P' be inherited from P if there is any. Holes are presented on those hole-points detected before. As previously discussed, if P is a hole point, then $parallax(P) > parallax(Q)$ with Q is the right neighbour of P. Then the width of the hole between points P' and Q' on the warped image I' is

$$W(P) = parallax(P) - parallax(Q). \quad (6)$$

Assume $W(P) = w$ and these points are $H_1, \ldots, H_w$ located between P' and Q'. The contents of point Hi are absent since they are newly exposed. However, they can be regarded as an extension of the background point P. Therefore, $parallax(H_i)$ and $texture(H_i)$ are set to be the same as P's as shown in Figure 10. Then H_i can be filled according to its texture direction. The *hole-filling* method is processed as follows:

1 $W(P) = 1$:

Since it is hard to notice visually, thus the hole content is simply the average of P' and Q'.

2 $W(P) > 1$ and $texture(P) \neq \delta$:

The ideal content for H_i would be the same as the content of a background point located on the direction of $texture(P)$. Without loss of generality, let us assume $texture(P) = k_1$ with $k = 90°$. By looking upwards (direction k_1) from H_i, if C_i is the first point encountered with the conditions in (7) satisfied, then the content of H_i is set to be the same as C_i's as shown in the Figure 11.

$$\begin{gathered}(\text{Content of } C_i \text{ is not absent})\\ \text{and}\\ \left(parallax(C_i) \geq parallax(H_i) - 1\right).\end{gathered} \quad (7)$$

Figure 10 Hole texture detection process

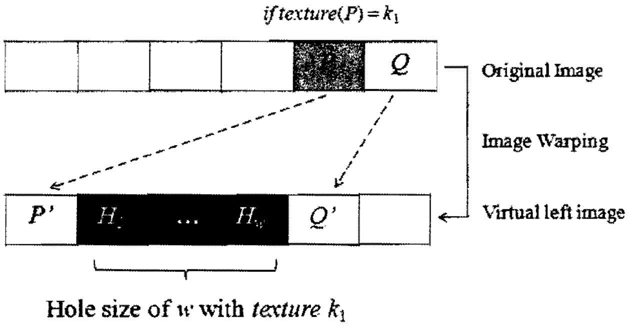
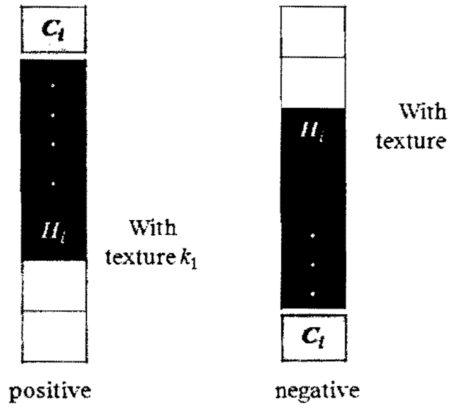


Figure 11 Hole-Filling process (k = 90° in this case)



3 The rest of the cases:

If H_i belongs to this case, then either $texture(H_i) = \delta$ or $texture(H_i) \neq \delta$ but no suitable background points are found. This is because they are missing apparent and/or consistent texture. Thus, we suggest using the simple horizontal mirror method. Assume $H_1, H_2, \ldots, H_w$, are consecutive points not yet filled. Towards the left of H_1, we look for consecutive background points $C_1, C_2, \ldots, C_r$, with the conditions in (8) satisfied. As described in Figure 12, $H_1, H_2, \ldots, H_w$, are filled by the following rule:

$$\begin{aligned}
&\text{If } r > w: \\
&H_i = C_{i-1} \text{ for } i = 1, 2, \ldots, w; \\
&\text{If } r \le w: \\
&H_i = \begin{cases} C_{i-1} & \text{for } i = 1, 2, \ldots, r-1; \\ C_{(i \bmod r)+1} & \text{for } i = r, r+1, \ldots, w. \end{cases}
\end{aligned} \tag{8}$$

4 Experimental results

The experimental results of the proposed method are presented in this section. The operating system used is 2.40 GHz Pentium Dual-Core processor and 512 MB RAM. The simulation compliers used are MATLAB 6.0 and Eclips. We tested three pairs of sequences. All image resolution is 463 × 370 pixels available on Middlebury stereo datasets (http://vision.middlebury.edu/stereo/data). The proposed method is compared with three other DIBR-based methods: *Sym* (the depth image is smoothed by symmetric Gaussian; Tam et al., 2004); *Asym* (the depth image is smoothed by asymmetric Gaussian; Zhang and Tam, 2005); *edge* (the depth image is edge-oriented smoothed by Lee and Effendi (2011). The evaluation is based on subjective visual quality and objective PSNR evaluation of depth images.

4.1 Parameter setting

Because 640 × 480 pixels is the common image resolution which results in the maximum *total_parallax* of 5% × 640 = 32 pixels, thus we also take this value for our experiment. This means the maximum *parallax* for the left/right image is 32/2 = 16 pixels.

In the *parallax correction* stage, we observed that the parallax differences between the background and foreground points near the object boundary are usually within 25% of the maximum of the *parallax* value. Thus, we set Th_0 as 4 (= 25% × 16). On the other hand, if the background point is a noise then its *parallax* value must be very large. Therefore, we set Th_1 to be 85% of the maximum *parallax* value which is 13.6 (= 85% × 16).

We set the threshold values in the hole region *detection* stage (Th_2) to 1. In the *texture detection* stage, the *Sobel* threshold (Th_3) search number of texture (n) and the sum of Se_k (Th_4) are 90, 20 and 15, respectively.

Figure 12 The horizontal mirror method to fill the hole (brown colour: foreground pixels; pink colour: background pixels) (see online version for colours)

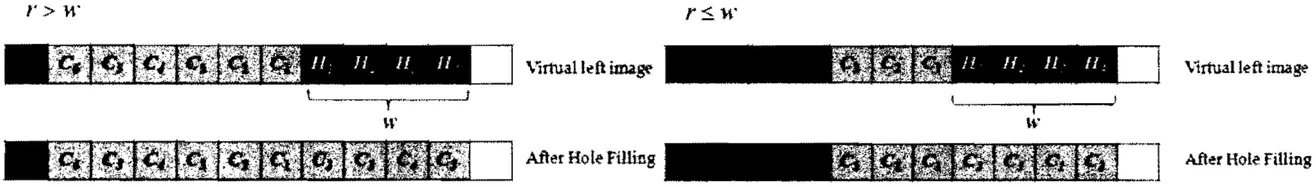


Figure 13 Processed depth image and corresponding virtual left image of 'interview' sequence. (a) original non-smoothing (b) Sym: symmetric Gaussian smoothing (c) Asym: asymmetric Gaussian smoothing (d) edge: edge-oriented smoothing (e) proposed (see online version for colours)

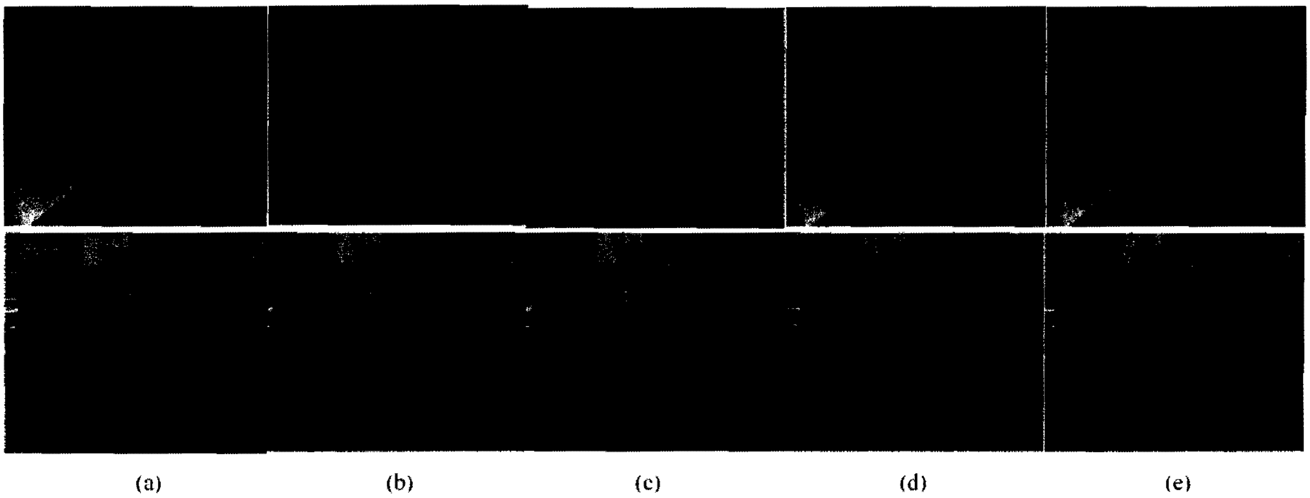


Figure 14 Processed depth image and corresponding virtual left image of 'cones' sequence, (a) original non-smoothing (b) Sym: symmetric Gaussian smoothing (c) Ssym: asymmetric Gaussian smoothing (d) edge: edge-oriented smoothing (e) proposed (see online version for colours)

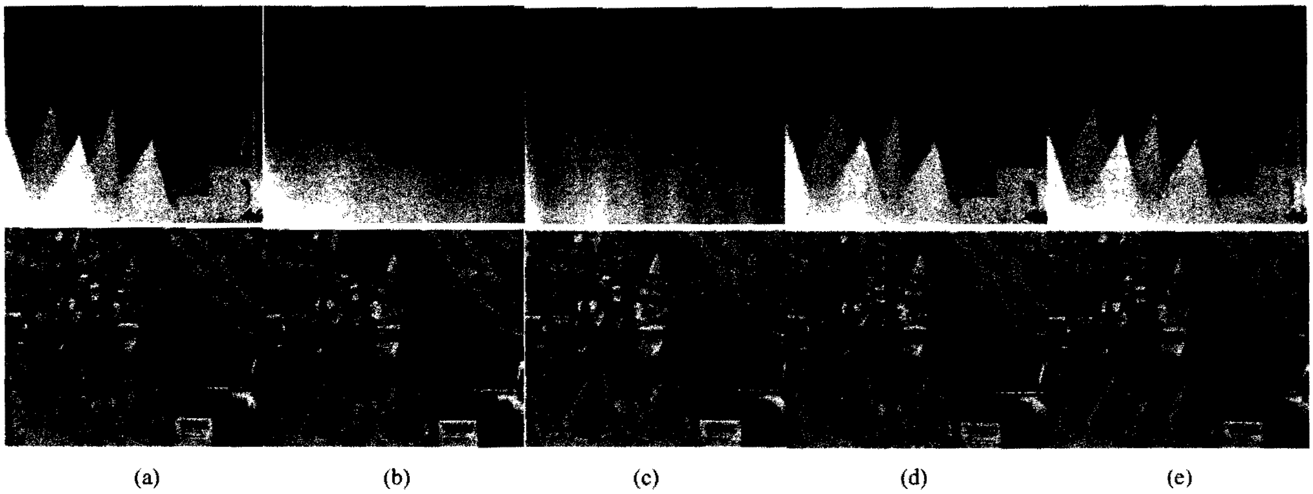
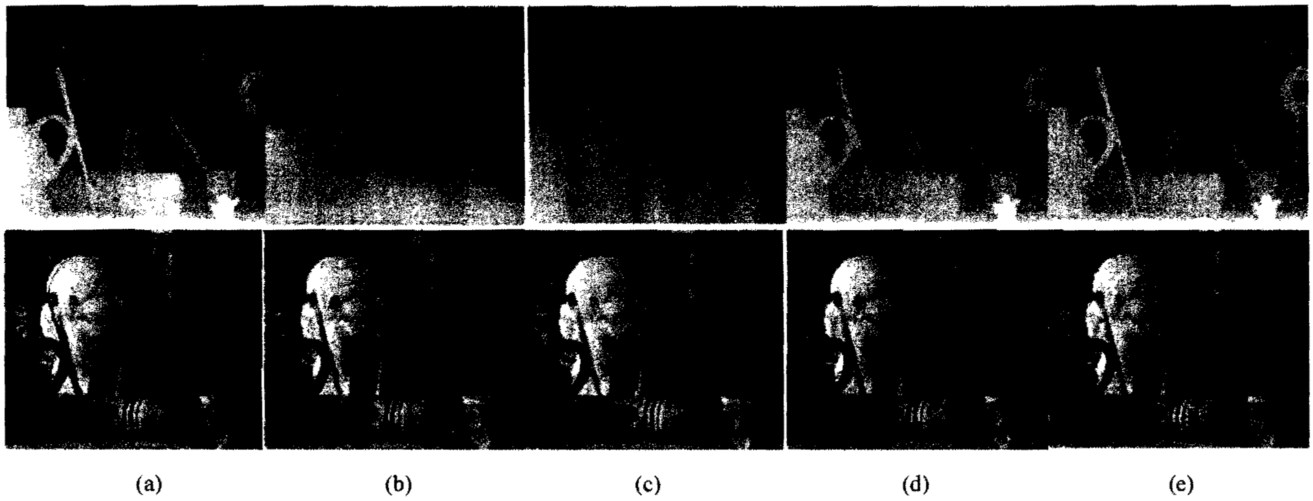


Figure 15 Processed depth image and corresponding virtual left image of 'art' sequence, (a) original non-smoothing (b) Sym: symmetric Gaussian smoothing (c) Asym: asymmetric Gaussian smoothing (d) edge: edge-oriented smoothing (e) proposed (see online version for colours)



4.2 Subjective evaluation

Figure 13 demonstrates the left image of the 'interview' sequence. We can find that symmetric Gaussian smoothing (Tam et al., 2004) generates a lot geometric distortion near the vertical object boundary in the virtual left image. Asymmetric Gaussian smoothing (Zhang and Tam, 2005), edge-oriented smoothing (Lee and Effendi, 2011) and our method can avoid most of the geometric distortion.

Figure 14 demonstrates the left image of the 'cones' sequence. Because the vertical texture feature is not obvious in this image and the original depth image has a lot of noise, a smoothing filter on the whole depth image, as with symmetric or asymmetric Gaussian smoothing, will be the best method for this image.

Figure 15 demonstrates the left image of the 'art' sequence. We can find that symmetric smoothing also has some geometric distortion problems in the crayon box and cones. Asymmetric smoothing performs the best. The results of edge-oriented smoothing also avoid most of the geometric distortion, but it has some colour penetration problems in the white wall. The proposed method uses texture direction to fill the hole, so the image looks more natural in the white wall.

4.3 Objective evaluation

We also evaluate the PSNR value for the depth images. The MSE and PSNR values are calculated as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right)$$
$$\text{with } MSE = \frac{1}{H \cdot W}\sum_{x=1}^{W}\sum_{y=1}^{H}\left(D_{orig}(x, y) - D_{method}(x, y)\right)^2, \tag{9}$$

where D_{orig} is the original depth map, D_{method} is the depth map modified by the indicated method, and H and W are the length and width of the image, respectively. The formula indicates that if the depth image suffers serious damage, then the PSNR value will be low. The experimental results (Table 2) show that smoothing filters on the whole depth image reduces the depth image quality. Thus, symmetric and asymmetric smoothing methods lead to low PSNR values. Our method modifies the depth values in regard to correction and enhancement only, so we can retain most of the depth information with the best PSNR values. The PSNR values of Lee's method fall between the smoothing method and ours, as the principle smoothing applied on the hole-points possess vertical texture.

Table 2 The PSNR values on depth images

Video *Method*	*Interview*	*Art*	*Cones*
Sym.	20.75	21.21	21.91
Asym.	19.85	19.64	21.10
Edge	31.46	28.66	28.18
Proposed	36.02	35.82	31.89

In addition to the good visual quality in the warped image and high PSNR values in the depth image, our method does not require large window-size convolution computation and thus is computationally efficient.

5 Conclusions

Solving the 'hole' problem caused by disocclusion is very critical in the DIBR of 3D images. Smoothing a given depth map to reduce 'holes' in size and number is a simple and effective solution. However, computational cost and depth information degradation come along with the smoothing procedure. The proposed method is designed to be computationally efficient as well as preserve most of the depth information. First, the depth map is corrected by MODE operation since available depth maps are prone to having noises. Expected 'holes' are located and texture information is detected then. Since human vision is sensitive to vertical texture, a vertical texture enhancement is further applied on the depth map. Then, the hole-filling process can be executed according to the detected texture information. We validate our method by comparing it to existing methods. The generated virtual left images have few geometric distortions and retain most of the depth information. Our method does not require large kernel convolution computation as is necessary in smoothing schema. Thus, it has the advantage of computational efficiency.

In the future, we will concentrate in applying the proposed method to transform conventional 2D movies into 3D movies. Also, we will study adjustable thresholds to facilitate more general application of the proposed method.

References

Chen, W.Y., Chang, Y.L., Lin, S.F., Ding, L.F. and Chen, L.G. (2005) 'Efficient depth image based rendering with edge dependent depth filter and interpolation', *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME 2005)*, pp.1314–1317.

Cheng, C.M., Lin, S.J., Lai, S.H. and Yang J.C. (2008) 'Improved novel view synthesis from depth image with large baseline', *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, pp.1–4.

Curti, S., Sirtori, D. and Vella, F. (2002) '3D effect generation from monocular view', *Proceedings of IEEE International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02)*, pp.550–553.

Fehn, C. (2004) 'Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV', *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems*, Vol. 5291, pp.93–104.

Fehn, C., Hopf, K. and Quante, Q. (2004) 'Key technologies for an advanced 3D-TV system', *Proceedings of SPIE, Three-Dimensional TV, Video and Display*, Vol. 5599, pp.66–80.

Flack, J., Harman, P. and Fox, S. (2003) 'Low bandwidth stereoscopic image encoding and transmission', *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems*, Vol. 5006, pp.206–214.

Lee, P. and Effendi, S. (2011) 'Nongeometric distortion smoothing approach for depth map preprocessing', *Proceedings of IEEE Transactions on Multimedia*, Vol. 13, No. 2, pp.246–254.

Middlebury stereo datasets [online] http://vision.middlebury.edu/stereo/data (accessed December 2011).

Nguyen, Q.H., Do, M.N. and Patel, S.J. (2009) 'Depth image-based rendering from multiple cameras with 3D propagation algorithm', *Proceedings of the 2nd International Conference on Immersive Telecommunications (IMMERSCOM '09), ICST*, pp.1–6.

Redert, A., de Beeck, M.O., Fehn, C., Ijsselsteijn, W., Pollefeys, M., Van Gool, L., Ofek, E., Sexton, I. and Surman, P. (2002) 'ATTEST – advanced three-dimensional television system techniques', *Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission*, pp.313–319.

Tam, W.J. and Zhang, L. (2004) 'Non-uniform smoothing of depth maps before image-based rendering', *Proceedings of SPIE, Three-Dimensional TV, Video, and Display*, Vol. 5599, pp.173–183.

Tam, W.J., Alain, G., Zhang, L., Martin, T. and Renaud, R. (2004) 'Smoothing depth maps for improved stereoscopic image quality', *Proceedings of SPIE, Three-Dimensional TV, Video, and Display*, Vol. 5599, pp.162–172.

Vazquez, C., Tam, W.J. and Speranza, F. (2006) 'Stereoscopic imaging: filling disoccluded areas in depth image-based rendering', *Proceedings of SPIE, Three-Dimensional TV, Video and Display*, Vol. 6392.

Zhang, L. and Tam, W.J. (2005) 'Stereoscopic image generation based on depth images for 3DTV', *IEEE Transactions on Broadcasting*, Vol. 51, No. 2, pp.191–199.