# Virtual View Synthesis Based on DIBR and Image Inpainting

Yuhan Gao[1], Hui Chen[1], Weisong Gao[2], and Tobi Vaudrey[3]

[1] School of Information Science and Engineering,
Shandong University, Jinan, Shandong, China
huichen@sdu.edu.cn
[2] Multimedia R & D Center, Hisense, Qingdao, China
gaoweisong@hisense.com
[3] Department of Computer Science
The University of Auckland, Auckland, New Zealand
t.vaudrey@auckland.ac.nz

**Abstract.** In 3DTV research, virtual view synthesis is a key component to the technology. Depth-image-based-rendering (DIBR) is an important method to realize virtual view synthesis. However, DIBR always results in hole problems where the depth and colour values are not known. Hole-filling methods often cause other problems, such as edge-ghosting and cracks. This paper proposes an algorithm that uses the depth and colour images to address the holes. It exploits the assumption of a virtual view between two laterally aligned reference cameras. The hole-filling method is performed on the blended depth image by morphological operations, and inpainting of the holes is obtained with the position information provided by the filtered depth maps. A new interpolation method to eliminate edge-ghosting is also presented, which additionally uses a post-processing technique to improve image quality. The main novelty of this paper is the unique image blending, which is more efficient than pre-processing depth maps. It is also the first method that is using morphological closing in the depth map de-noising process. The method proposed in this paper can effectively remove holes and edge-ghosting. Experimental quantitative and qualitative results show the proposed algorithm improves quality dramatically on traditional methods.

**Keywords:** Virtual View Synthesis, Depth-Image-Based-Rendering (DIBR), Image Inpainting, Interpolation.

## 1 Introduction

Recently, 3DTV has attracted considerable attention because of its wide application. Virtual view synthesis is one of the key steps to realize a full 3DTV system. The most widely used technique is depth-image-based rending(DIBR), which combines a reference image with its corresponding depth information to synthesize the virtual viewpoint in the scene. DIBR could not only render high quality images fast, but also new views of any position. However, DIBR would

R. Klette, M. Rivera, and S. Satoh (Eds.): PSIVT 2013, LNCS 8333, pp. 172–183, 2014.

cause holes due to the occlusion between the foreground/background, and also background that is only visible from the new view but not from the reference. Additionally, there is a problem that some false edges from the foreground object are projected to the new viewpoint. This is called edge-ghosting.

To deal with the hole and edge-ghosting problems, various algorithms have been proposed. Vazquez *et al.* [1] filled the holes based on the neighbouring pixels from the colour image only; it could not effectively address the issue of large holes. Domanski *et al.* [2] adopted a mapping method to generate a virtual view from each reference image independently and then blended them into one virtual view, but did not consider location of the holes. Narayanan *et al.* [3] proposed a depth and texture method to render complex scenes, while they did not propose an effective solution to the edge-ghosting problems.

The contributions of this paper are:

- a novel blending algorithm that is more efficient than preprocessing depth maps, but obtains similar quality
- a hole filling scheme that adopts a morphological closing operation along with a novel image inpainting method based on blended depth images
- an edge-ghosting eliminating method is proposed by interpolating pixels in the inpainting image
- the entire process addresses hole filling and edge-ghosting in an effective manor.

The remaining sections of this paper are organized as follows. Section 2 describes principles of DIBR. Section 3 explains the proposed algorithm. The experimental results are in Section 4. The paper is concluded in Section 5.

## 2 Depth-Image-Based-Rendering (DIBR)

Fehn [4] proposed that when using a pinhole camera model, if the camera coordinate system coincided with the world coordinate system of the reference view, it is possible to render the virtual view with DIBR technology. In order to realize virtual viewpoint rendering using DIBR, the following is required: a colour map, depth map, and camera parameters of the reference point, as well as the virtual viewpoint camera parameters.

### 2.1 3D Warping

3D Warping, which was proposed by McMilian [5] first, is the core of DIBR. It could mainly be divided into two steps: all the pixels in the reference image are projected to their corresponding three-dimensional space using the depth information, then the points of the three-dimensional space are projected to the virtual viewpoint image plane.

Fehn gave a transformation equation that is applicable to realistic settings:

$$\mathbf{m} = \mathbf{A}\mathbf{P}_\mathbf{n}\mathbf{C}\mathbf{M} \tag{1}$$

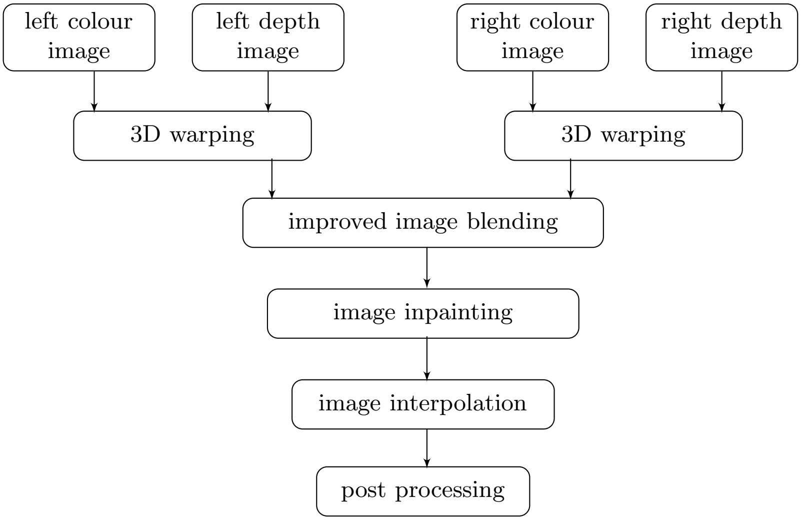


**Fig. 1.** The proposed view synthesis scheme

$\mathbf{M}$ is a world point (in world space coordinates) and $\mathbf{m} = (\mathbf{u}, \mathbf{v})$ is the respective projection in camera pixel coordinates. The matrix $\mathbf{A}$ denotes the intrinsic parameters that represent the inner structure of the camera (focal length, pixel size, skew, distortion, and principal point). The matrix $\mathbf{C}$ denotes the extrinsic parameters, which indicate the relationship between the world coordinates and the camera coordinates. $\mathbf{P_n}$ is the normalized perspective projection matrix from view $n$.

### 2.2 Problems of DIBR

When generating a new view image with 3D warping, there is a problem that several pixels may be projected to the same pixel on the virtual image, this is called a visibility issue. Moreover, some regions that are occluded in the reference image may become visible in the new virtual image. These new exposed regions become blank holes areas. Another issue is edge-ghosting, which is caused by the inaccurate edge matching of the colour and/or depth image. Therefore, the key problem is to cope with these regions correctly. This paper proposes a new method to fill holes and remove edge-ghosting.

## 3 Proposed Algorithm

The processing pipeline of the proposed algorithm consists of four steps: improved blending method, image inpainting based on depth image, image interpolation and post filtering. Fig. 1 shows the process of the proposed algorithm.

In the 3D warping step, the standard steps of DIBR are applied, i.e., the left and right colour images are warped to the virtual viewpoint using their corresponding depth images. Next, the traditional method is improved by blending warped colour images. In the image inpainting step, an image inpainting method based on filling holes via the depth image is proposed. A new interpolation algorithm is also presented to remove edge-ghosting during the image interpolation step. Finally, post processing is applied to improve image quality. Details are discussed in the following subsections.

### 3.1 3D Warping and Image Projection

In the first step, all 3D points $\mathbf{M}$ from the reference image are calculated by Eq. (1) using image points $\mathbf{m}$, the reference projection matrix $\mathbf{P}_{\mathrm{R}EF}$. The depth image $Z(u,v)$ is obtained from $\mathbf{M}$. In the second step, 3D point $\mathbf{M}$ is projected to warped virtual view with $\mathbf{P}_W$.

When obtaining the depth image in the warped projected virtual view $Z_W(u,v)$ conflicts can occur when two or more world points occupy the same pixel coordinate. One way to deal with this is to use the nearest point to the virtual camera, as used in Mori *et al.* [6]. The equation is shown below.

$$Z_W(u,v) = \underset{m_W}{\operatorname{argmin}} \Big( Z_W(u,v) \in (0,\infty) \Big) \tag{2}$$

in other words, the lowest depth value above zero (in front of camera) at warped pixel point $m_W(u,v)$ is used.

The warped colour image $I_W(u,v)$ is constructed using the colour value from the original reference image. For conflicts, the colour value of the original reference point selected by Eq. (2) is used. In other words, the colour of the closest point in front of the virtual camera is selected.

### 3.2 Image Blending

The starting point for image blending is one or more images reprojected to a virtual view. Once the reprojected views are obtained, there can be pixel conflicts from from the multiple reference images. The subsections below show how the proposed algorithm deals with these conflicts in a stereo camera environment when warping to a virtual view between the two reference cameras.

**Traditional Blending.** The virtual view image is rendered by blending the warped left and right image, which have been translated from the reference viewpoints to the virtual viewpoint using Eq. (1) (see Fig. 1). The traditional image blending approach from Li *et al.* [7] is illustrated by:

$$I(u,v) = (1-\alpha)I_L(u,v) + \alpha I_R(u,v) \tag{3}$$

$$\alpha = \frac{|t - t_L|}{|t - t_L| + |t - t_R|} \tag{4}$$

$I(u, v)$ is the blended virtual view image at pixel $(u, v)$. $I_L(u, v)$ and $I_R(u, v)$ are the warped virtual images from the left and right reference viewpoints, respectively . The blending factor $\alpha$ is calculated by Eq. (4) $t$ is the translation vector from the extrinsic matrix of the virtual view. $t_L$ and $t_R$ are the left and right translation vectors, respectively. This blending factor has more weight if the virtual viewpoint is closer to a particular reference image (in world coordinates).

**Improved Image Blending.** One problem of the traditional method is that some pixels may have conflicting overlaps from the reference images. Additionally, there are usually so many holes in the blended image. Several methods [8,6] pre-filter depth images before warping in an attempt to reduce noise. In practise this does not have any obvious improvements. For example, Mori *et al.* adopted median and bilateral filtering, and it provided a minor improvement (the result is shown in Table 1). Additionally, it caused some inaccurate pixels and lump holes when blending depth images. Therefore, pre-filtering the depth images is not used in the presented algorithm. Instead, the improved blending method from Eq. (5) is used. An example result is shown in Figure 3a. This heuristic blending is based on the assumption of a stereo camera configuration, where the project virtual image lies between the two cameras (in the height-depth plane). The blended image $I_B$ is calculated as follows:

$$I_B(u, v) = \begin{cases} I_L(u, v) & \text{if } c_1 \\ I_R(u, v) & \text{else if } c_2 \\ 0 & \text{else if } c_3 \\ (1-\alpha)I_L(u, v) + \alpha I_R(u, v) & \text{otherwise} \end{cases} \tag{5}$$

where

$$\begin{aligned} c_1 = & \Big(Z_L(u, v) \neq 0 \ \wedge \ Z_R(u, v) \neq 0 \ \wedge \ Z_R(u+1, v) = 0\Big) \\ & \qquad \vee \ \Big(Z_L(u, v) \neq 0 \ \wedge \ Z_R(u, v) = 0\Big) \\ c_2 = & \Big(Z_L(u, v) \neq 0 \ \wedge \ Z_R(u, v) \neq 0 \ \wedge \ Z_L(u-1, v) = 0\Big) \\ & \qquad \vee \ \Big(Z_L(u, v) = 0 \ \wedge \ Z_R(u, v) \neq 0\Big) \\ c_3 = & \qquad \ \ \Big(Z_L(u, v) = 0 \ \wedge \ Z_R(u, v) = 0\Big) \end{aligned} \tag{6}$$

$Z_L(u, v)$ and $Z_R(u, v)$ are the values of depth images projected from the original left and right depth maps to the virtual view. A value of $Z_*(u, v) = 0$ represents a hole (unknown) at $(u, v)$. The left image colour value is used in the blended image if a point in the left depth image is not a hole, while it is in the right image (second part of $c_1$). The left image value is also assumed if a point in both depth images is valid but the right neighbouring pixel in the right depth image is a hole, since holes in right depth image are often on the right side of real scene (first part of $c_1$). The exact opposite is assumed for the right hand image (see $c_2$). If both the left and right depth image points contain a hole, the corresponding pixel value in the blended image would also be a hole. Otherwise, the pixel value in blended image would be summed with the traditional weighting coefficient $\alpha$, see Eq. (4). This blending method reduces holes with marginal computational difference.

The depth image is blended $Z_B$ in a similar fashion to the colour images above. An example result is shown in Fig.2a:

$$Z_B(u,v) = \begin{cases} Z_L(u,v) & \text{if } c_1 \\ Z_R(u,v) & \text{else if } c_2 \\ 0 & \text{else if } c_3 \\ (1-\alpha)Z_L(u,v) + \alpha Z_R(u,v) & \text{otherwise} \end{cases} \tag{7}$$

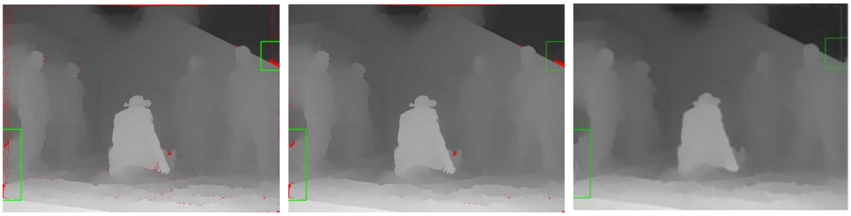
### 3.3 Image Inpainting

After creating the blended image, some holes still exist. This blended colour image does not remove all holes. There have been many research efforts to further fill the holes. Criminisi *et al.* [9] used exemplar-based inpainting to inpaint the image, this is computationally expensive and may result in edge-ghosting. Oh *et al.* [10] filled the holes with background pixels, this can result in foreground pixels mistakenly being assumed as background. Zhang *et al.* [8] improved the image, but with some holes still existing. Li *et al.* [7]adopted a blending image that still had some holes not yet removed. The following explains a new image inpainting method that utilizes the depth values to fill holes.

**Depth Image Filtering.** The depth image is a grey-scale image, and grey value represents the depth of the points in a world coordinate. Both the colour images and depth images are warped and blended, then the colour image is inpainted based on the blended depth image. Since the hole region has no depth values, the depth image requires filtering and interpolation. Mori *et al.* used median and bilateral filtering but could not remove the lump holes, Jung and Ho [11] copied neighbouring background depth values to hole regions. This process may change some background pixels to foreground, or vice-versa.

The method proposed in this paper adopts morphological operations on the depth images. A closing operation (dilation then erosion) that uses a diamond structured window is chosen to fill the holes in the depth image [12]. This closing operation fills any hole smaller than the chosen window, it also fills any cracks that are thinner than the window. The filtering results are shown in Fig. 2. The red regions represent holes. Fig. 2a is the original blended depth image. Fig. 2b is using the Mori *et al.* method with median and bilateral filtering. Fig. 2c is the result of this papers filtering. By comparing Fig. 2b with 2c, it is clear that small cracks and spot noise are removed with the median plus bilateral filtering, however the large lump holes are still present. Comparing those results with the presented method in Fig. 2c; all lump holes, cracks and spot noise are removed and the filtering result is as good if not better than the results in Fig. 2b.

**Depth-Based Image Inpainting.** The image blending above can remove most holes. However, larger holes still exist which are often located in areas where



(a) original depth image (b) Mori *et al.* [6] method (c) presented filtering

**Fig. 2.** Preprocessing result of depth image

occlusion occurs. Therefore, this paper proposes a new image inpainting method to fill holes. It uses the blended and filtered depth image to distinguish background and foreground. The pixel is blended with coloured pixels at the same depth within a specified region. When the background pixels are inpainted, it removes the effects from foreground pixels, and holes of background are weighted by points in background at the same depth, and vice versa. Each pixel is handled as Eq. (9). The depth value of the points around a pixel are compared, if the depth value is smaller than a threshold, it is determined that they are at the same depth as Eq. (10). The resulting inpainted colour image $I_J$ is calculated from the blended image $I_B$ from Eq. (5).

$$I_J(u,v) = \begin{cases} I_B(u,v) & \text{if } Z_B(u,v) = 0 \\ J(u,v) & (otherwise) \end{cases} \tag{8}$$

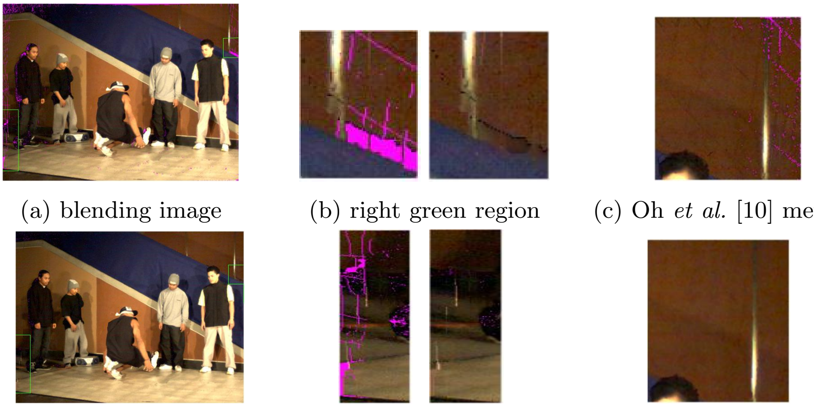
$$J(u,v) = \frac{\sum\limits_{(i,j)\in R} D(i,j)\, W(i,j)\, I_B(i,j)}{\sum\limits_{(i,j)\in R} D(i,j)\, W(i,j)} \tag{9}$$

$$D(i,j) = \begin{cases} 1 & \text{if } |Z_B(i,j) - Z_B(u,v)| < T_Z \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$W(i,j) = \frac{1}{(i-u)^2 + (j-v)^2} \tag{11}$$

where $J(u,v)$ is the hole inpainting function. The pixel $(i,j)$ is within the region $R$ about pixel $(u,v)$. $D(i,j)$ is depth weighting factor at pixel $(i,j)$, with $T_Z$ being the inpainting depth threshold. $W(i,j)$ is a weighting coefficient that reduces the weight given by quadratic distance from the original pixel. In other words, Eq. (8) will only fill in holes from the original blended image if there is a hole. From there, Eq. (9) is used to inpaint the image based on a local depth threshold and neighbourhood weighting factor.

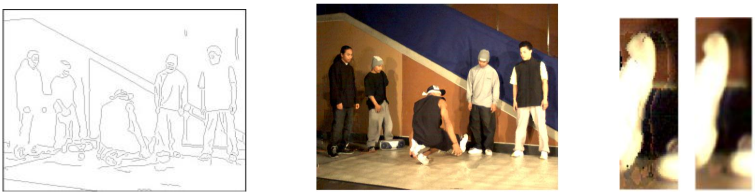
Fig. 3 demonstrates the process with an example. Fig. 3a shows a sample blended image $I_B$ where the pink regions are holes and the green frame is highlighting the hole regions. Fig.3d is the inpainting result $I_J$ . From analysing this figure, it is noticeable that the lump holes in green frame region are restored.



(a) blending image (b) right green region (c) Oh *et al.* [10] method

(d) inpainting image (e) left green region (f) This paper's method

**Fig. 3.** Result of inpainting



(a) edge of inpainting image (b) interpolation image (c) ghosting edge

**Fig. 4.** Result of interpolation process deployed in this paper

Fig. 3b and 3e are zoomed views of the green frame regions in Fig.3a and Fig.3d, respectively. The left hand image in each of these figures is the blended image, while the right is the inpainted image. The method proposed in Oh *et al.* is compared with this paper's inpainting method in Fig. 3c and 3f. From the figure, one could see the Oh *et al.* method has some speckles as the example pink region shows, while this paper's method could obtain a less noisy image.

### 3.4 Image Interpolation

In the process of view synthesis, some false edges of the foreground object are projected to new viewpoint. It is termed edge-ghosting. Mori *et al.* solved this problem by expanding the border of the occlusion, however, this would cause background points to become foreground points. The proposed method solves the problem of edge-ghosting by exploiting the edge analysis of the colour image in addition to using the depth image. A pixel is considered inaccurate if: the grey-value difference between its adjacent pixel is larger than a threshold, it does not belong to an edge, and if the depth of its adjacent pixel is below a threshold. An inaccurate pixel's colour is then averaged using a neighbourhood of pixels at a similar depth. The inpainted image $I_J$ is converted to a grey-scale image $I_G$ for the purposes of threshold calculations. To determine the edge, the Canny edge detector [13] is used on the grey-scale image $I_G$ to calculate the edge image $E_G$ (where 0 is not an edge and any positive value is an edge). Fig. 3d shows a sample of the Canny edge detection results. The resulting interpolation image $I_P$ is calculated from Eq. (12).

$$I_P(u,v) = \begin{cases} P(u,v) & \text{if } c_4 \\ I_J(u,v) & \text{otherwise} \end{cases} \tag{12}$$

$$\begin{aligned} c_4 = \Big(E_G(u,v) = 0\Big) \ \wedge \ & \Big(|I_G(u,v) - I_G(u,v-1)| > T_G\Big) \\ \wedge \ & \Big(|Z_B(u,v) - Z_B(u,v-1)| < T_Z\Big) \end{aligned} \tag{13}$$

$$P(u,v) = \frac{1}{4} \sum_{(i,j)\in N_4} I_J(i,j) \tag{14}$$

where $T_G$ and $T_Z$ represent the grey and depth thresholds, respectively. $P$ is the neighbourhood interpolation function with $N_4$ representing the four-neighbours of pixel $(u,v)$, excluding itself.

Fig. 4b is an example interpolation result. Fig. 4c show magnified areas showing the removal of edge-ghosting. The left shoe image (inpainted image) shows a very prominent ghost edge, while this is clearly removed in the right hand shoe image (interpolated image).

### 3.5 Post Processing

For clearing up the tiny spots to obtain a new view image with better quality, speckle reducing anisotropic diffusion (SRAD) [14] is used to post process the image. This method is widely used in processing radar imaging, and could erase speckles, at the same time, preserve edge parts. This method is applied here to eliminate tiny spots in the new view. Fig. 5c shows the results.

## 4 Experimental Results and Discussion

For testing the algorithm, a set of sequences named ‘breakdancers’ was used which are generated and distributed by Interactive Visual Group at Microsoft Research [15]. The sequences contains 8 viewpoints, the depth of the image is consistent with the resolution of the colour image 1024 × 768. Eight distinct video sequences from eight distinct viewpoints are provided. This includes the: colour images, corresponding depth images, and camera parameters for each view. For the experimental results below, the camera 2 and camera 4 are selected as the reference points of view. These views are used to recreate virtual camera viewpoint 3, where the raw data from the camera 3 sequence is used as the ground truth. The parameters used for these test sequences were: depth image closing operation used a diamond window with parameter of 7, depth based

**Table 1.** Experimental results for presented blending method

| measure | blending with depth preprocessing | presented method |
|---|---|---|
| PSNR | 30.4775 | 30.4047 |
| SSIM | 0.9080 | 0.9086 |

**Table 2.** Results for proposed inpainting, interpolation and post filtering

| measure | inpainting | interpolation | post filtering |
|---|---|---|---|
| PSNR | 32.5193 | 33.2696 | 33.7350 |
| SSIM | 0.9271 | 0.9334 | 0.9345 |

inpainting region $R = 10 \times 10$, depth threshold $T_Z = 5$, colour (grey) threshold $T_G = 5$, and Canny edge parameter $\sigma = 5$.

In order to measure the effectiveness of the proposed method, the following measures are used: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [16]. PSNR is an evaluation function measuring the quality of the reconstructed image, a higher value is better than a lower one. SSIM is used to evaluate the similarity between two images, a value of 0 means no similarity, while a value of 1 means exact similarity.

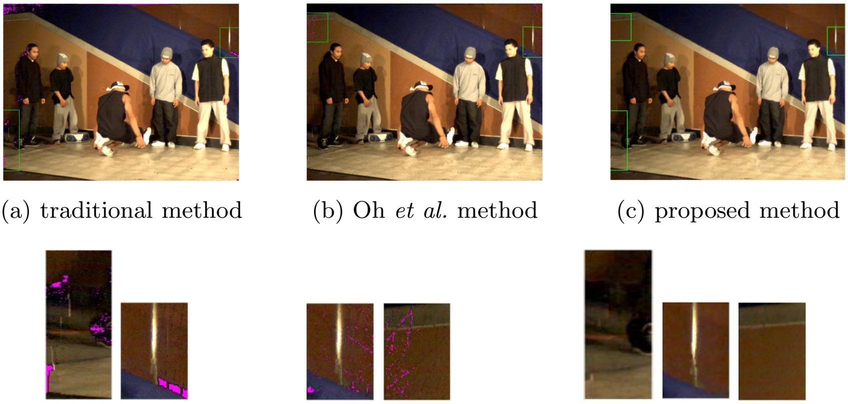
Firstly, blending using preprocessed depth maps is compared with this paper's blending method. We can see in Table 1 that using preprocessed depth maps has only marginal improvement over this paper's blending method, while adding a large computational cost. Therefore, it is not necessary to use preprocessing on the depth maps before blending the colour images.

Secondly, PSNR and SSIM are calculated in every step of this experiment. The results are given in Table 2. From this table we can see quantitatively that the presented inpainting method, interpolation method and post filtering method all improved quality of the synthesized virtual image. The inpainting alone provides an increase of 2.1 to PSNR and 0.02 for SSIM. Interpolation adds another 0.7 to the PSNR, and a further 0.01 to the SSIM. The post filtering process adds only marginal improvements (0.4 to PSNR and nearly nothing on the SSIM). The post filtering SRAD is not the novel part of this paper, but shows that it does provide some improvement in the post-process. The improvements are also obvious in qualitatively in the experimental figures. Fig. 3d highlights how well the image inpainting can fill holes, especially lump holes as the green frame region highlights. From the comparison in Fig. 4c, it is clear that this paper's interpolation method could remove edge-ghosting.

The proposed method is compared with the traditional synthesis method and Oh *et al.* in Table 3. The results quantitatively indicate that this paper's method improves image quality dramatically compared to the other methods. Fig. 5 shows the qualitative results of the different methods. Fig. 5d to 5f are amplifications of the green framed regions of the respective figures. From those figures,

**Table 3.** Comparative results for proposed method

| measure | proposed method | traditional synthesis | Oh *et al.* [8] |
|---|---|---|---|
| PSNR | 33.7350 | 30.4286 | 31.8150 |
| SSIM | 0.9345 | 0.8071 | 0.8365 |



(a) traditional method (b) Oh *et al.* method (c) proposed method

(d) green region of (b) (e) green region of (c) (f) green region of (d)

**Fig. 5.** Qualitative results highlighting lump holes

one can see that the traditional method could not remove lump holes, Oh *et al.* clears up big holes with some cracks still existing (the comparison is in Fig. 3c and 3f). The proposed method removes not only holes but also edge-ghosting and has a higher similarity to the original image.

## 5 Conclusion

In this paper, a new algorithm was described for realizing virtual view synthesis. The holes are filled by morphological image inpainting based on a depth image that is blended in a novel way without pre-processing. Meanwhile, the ghosting edges are eliminated with image interpolation. The experimental results clearly showed quantitatively and qualitatively that the presented method provides high quality results, and better than some other selected methods.

The test data that was selected (generating a virtual view between two cameras) is what the algorithm was designed to be used on. Further work and considerations need to be taken if wanting to synthesise views outside the assumptions of the algorithm. Making these blending assumptions more geometrically flexible is the aim of future work.

**Acknowledgments.** This work is supported by the Natural Science Found of Shandong NSFSD under No.2009ZRB01675 and No.ZR2013FM032 and Multimedia R & D Center of Hisense.

## References

1. Vazquez, C., Tam, W.J., Speranza, F.: Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering. In: Proceedings of SPIE, Orlando, FL, USA, vol. 6392 (2006)
2. Domaski, M., Gotfryd, M., Wegner, K.: View Synthesis For Multiview Video Transmission. In: International Conference on Computer Vision and Pattern Recognition, Florida, USA, pp. 433–439 (2009)
3. Narayanan, P., Kumar, P., Reddy, K.: Depth+Texture Representation For Image Based Rendering. In: Proceedings of Fourth Indian Conference on Computer Vision, Graphics and Image Processing, Kolkata, Indian, pp. 113–118 (2004)
4. Fehn, C.: Depth-image-based rendering(DIBR), Compression, and Transmission For a New Approach on 3DTV. In: Proceedings of the SPIE, San Jose, CA, USA, vol. 5291, pp. 93–104 (2004)
5. McMillan, L.: An Image-Based Approach to Three-Dimensional Computer Graphics. Technical Report. University of North Carolina at Chapel Hill, Chapel Hill, NC, USA (1997)
6. Mori, Y., Fukushima, N., Yendo, T., Fujii, T., Tanimoto, M.: View Generation With 3D Warping Using Depth Information for FTV. Signal Processing: Image Communication 24(1), 65–72 (2009)
7. Li, M., Chen, H., Li, R., Chang, X.: An Improved Virtual View Rendering Method Based on Depth Image. In: International Conference on Computer Communication, Jinan, China, pp. 381–384 (2011)
8. Zhang, L., Tam, J., Wang, D.: Stereoscopic Image Generation Based on Depth Images For 3DTV. IEEE Transactions on Broadcasting 51(2), 191–199 (2005)
9. Criminisi, A., Perez, P., Toyama, K.: Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE Transactions on Image Processing 13(9), 1200–1212 (2004)
10. Oh, K., Yea, S., Vetro, A., Ho, Y.: Virtual View Synthesis Method and Self Evaluation Metrics for Free Viewpoint Television and 3D Video. International Journal of Imaging Systems and Technology 20(4), 378–390 (2010)
11. Jung, J., Ho, Y.: Virtual View Synthesis Using Temporal Hole Filling with Bilateral Coefficients. In: IEEE International Conference on Research, Innovation and Vision for the Future, pp. 1–4 (2012)
12. Herk, M.V.: A Fast Algorithm for Local Minimum and Maximum Filters on Rectangular and Octagonal Kernels. Patt. Recog. Letters 13, 517–521 (1992)
13. Canny, J.: A Computational Approach To Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
14. Yu, Y., Acton, S.: Speckle Reducing Anisotropic Diffusion. IEEE Transactions on Image Processing 11, 1260–1270 (2002)
15. Microsoft Research, Image-Based Realities-3D Video Download, `http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/`.
16. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)