

Effective Removal of Artifacts from Views Synthesized using Depth Image Based Rendering

Jiangbin Zheng, Danyang Zhao

Dept. of Computer Science and Engineering, School of
Computer, Northwestern Polytechnical University
Xi'an, China
zhengjb0163@163.com

JinChang Ren

Dept. of Electronic and Electrical Engineering, University
of Strathclyde Glasgow
G1 1XW, United Kingdom
jinchang.ren@strath.ac.uk

Abstract—Depth Image Based Rendering (DIBR), as a free-viewpoint synthesis technique, enables interactive selection of the view for watching. However, many rendering methods based on DIBR usually bring contour, crack and disocclusion artifacts. To address these problems, we propose effective methods to remove these artifacts. Firstly, before warping, the reference depth maps and color images are analyzed to find the regions causing contour artifacts. A combination of depth map edge detection and color consistency correction is applied to the analysis. By omitting warping the found regions, the synthesized views contain no contour artifacts and edges of foreground objects are well preserved. Secondly, cracks are filled using surrounding non-crack pixels rather than filtering for local consistency and smoothness. Thirdly, we apply a method based on texture extrapolation with depth information to inpaint the disocclusions. For two well-known sequences, ‘Ballet’ and Breakdancers’, we obtain large Peak Signal to Noise Ratio (PSNR) gains in comparison to state-of-the-art techniques. In addition, the proposed method also obtains good results in Structural Similarity Index Measurement (SSIM) and visual quality.

Keywords- Depth Image Based Rendering (DIBR); depth map edge detection; color consistency correction; contour artifacts; crack artifacts; disocclusion artifacts

I. INTRODUCTION

With the development of computer techniques, transmission and display of digital images and videos become increasingly popular in a number of applications such as 3D video display [1] and 3D reconstruction. To satisfy the new requirements in terms of free-view and immersive experiencing, multi-view imaging (MVI) has attracted much more attention. As one of the most important applications of MVI, free-viewpoint TV (FTV) [2] brings a new visual experience where users can interactively select the view whilst watching videos.

Recently, Depth Image Based Rendering (DIBR) is emphasized, which involves the projection of a viewpoint into another. In general, two surrounding reference images are used in DIBR for view synthesis[3-12]. But there are different kinds of artifacts in the synthesized images based on DIBR. Three major artifacts need to be solved in such cases, which include contours caused by pixels at boundary of high depth discontinuities, cracks due to sampling rate of the reference image and disocclusions remained after blending projected images. Fig.1 shows the artifacts before processed.

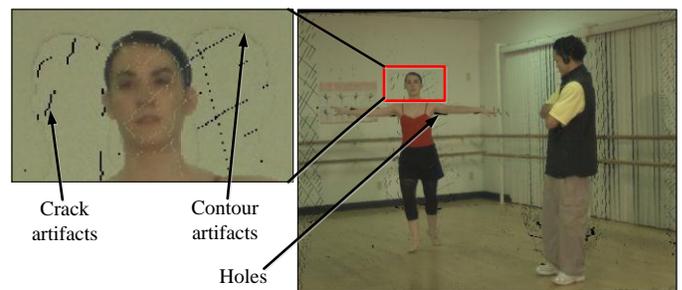


Figure 1. Artifacts in the synthesized view before processing

In this paper, we propose efficient algorithms to solve these artifacts. Firstly, depth map edge detection and color consistency correction are used in labeling pixels from contours, and only unlabeled pixels are used in 3D warping and blending images. Secondly, the depth differences are used in identifying crack regions and non-crack regions and cracks are filled using surrounding non-crack pixels. Thirdly, the hole artifacts are filled using texture extrapolation with depth information. Experimental results have demonstrated the efficacy of the proposed approaches. Detailed description of the proposed approaches and relevant results are reported in the next sections.

This paper is organized as follows. The related work is summarized in section 2. An overview of proposed algorithm is presented in section 3. The main removal approaches of artifacts are described in detail in section 4. In section 5, we show the experimental results and have a discussion. Finally, the conclusion is made in section 6.

II. RELATED WORK

Depth Image Based Rendering (DIBR) techniques use depth-plus-video data for free-viewpoint rendering. The 3D sense representation based on depth-plus-video combine the advantages of geometry based representation and image based representation. Comparing to geometry and image based representation, depth-plus-video 3D sense representation does not use explicit 3D models and a large number of images and only depth map and fewer images need to be processed [8]. Previous research on DIBR from one reference image has two inherent limitations, which are viewpoint dependency of textures and disocclusions. To overcome these limitations, most recent methods employ warping from two surrounding

reference images to a virtual viewpoint. Disocclusions from one reference are compensated by the other view. Zitnick et al. [9] pointed out that three main artifacts need to be removed in rendering a high quality virtual viewpoint. First of all, pixels at high discontinuities tend to cause contour artifacts which need to be fixed. Secondly, empty pixels and holes due to insufficient sampling of the reference images need to be filled. Thirdly, the remaining disocclusions after blending the warped images need to be generated.

To overcome these artifacts, recent research involved different methods. We classify the techniques into three categories according to different kind artifacts. We will review some typical techniques as follows.

2.1 Techniques of contour artifacts removing

In the reference image, because of borders staying in high discontinuities, the blended images always contain contour artifacts. Muller et al. [13] provided a free-viewpoint rendering algorithm based on a layered depth map presentation. They defined three layers: foreground boundary layer, background boundary layer and main layer. They first project the main layer and get the blended image. Secondly, they projected foreground and background boundary layer and they used a simple depth test to avoid contour artifact existing. The quality of this algorithm is not measured. It also requires amount of pre-process and post-process.

Mori et al. [5] used boundary matting method to remove contour artifacts. After 3D warping, they expanded the boundary to background direction and successfully removed contour artifacts in background, but this method is not efficient to the contour artifacts in foreground.

Luat et al. [6] first detected the foreground boundary in depth map and labeled the unreliable regions which corresponding to Fig. 3. When they did 3D warping they only used the unlabeled regions. This method may remove the contour artifacts, but boundary shrinks to the foreground direction which damages the foreground objects.

2.2 Techniques of crack artifacts removing

Crack artifacts often occur in the virtual image after forward 3D warping. Each point from an original image is projected separately into the virtual view, and falls in general onto a floating point coordinate. This position is quantized to the nearest neighbor position of the integer sample raster. Unfortunately, quantization may leave some samples unfilled being visible as thin black lines in Fig. 5(a). In some case, such cracks in foreground regions are filled by background information.

Mori et al. applied a bilateral filter. This method is efficient to the cracks in background. But this method didn't consider depth information and was not efficient to cracks filled by background information in foreground regions. Based on the algorithm of Mori et al., Luat et al. proposed to process the virtual depth image with a median filter. Afterwards they compared the input and output of the median filter and performed an inverse warping when pixels have changed. This

method can fill cracks both in foreground and background, but its inverse warping may increase computational errors.

2.3 Techniques of disocclusion inpainting

The blended virtual images often contain some disocclusion regions which are visible in virtual view but occluded in both reference views.

Mori et al. inpainted disocclusion regions with the method proposed by Telea [14]. This method used only texture information and no depth information. Using this method, the inpainted regions between background and foreground may contain blurs. Luat et al. introduced inpainting method with depth information on the basis of Telea's method. In [15], each virtual view image featuring disocclusions is compensated using image information from a causal picture neighborhood via a background sprite. Residual uncovered areas are initially coarsely estimated and then refined using texture synthesis. Chen [16] proposed edge dependent Gaussian depth filter and interpolation to fill holes.

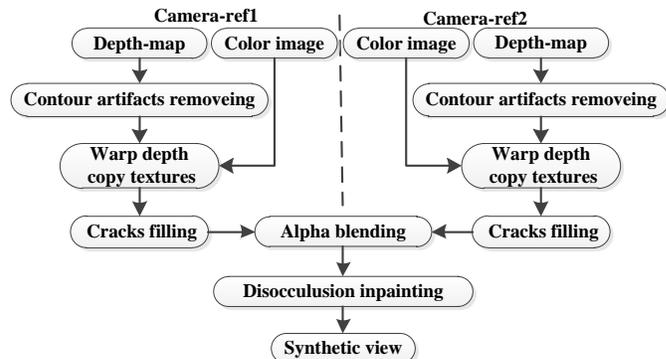


Figure 2. Block diagram of proposed rendering algorithm

III. PROPOSED FREE-VIEWPOINT RENDERING ALGORITHM

An overview of proposed algorithm is given in Fig. 2. The synthesis approach can be treated as a mapping process with some processing steps which are used to remove the artifacts during the map process. In our algorithm, the inputs are two reference views with depth and color images.

The first step is depth reliability analysis, which is used to analyze the misalignment of sharp depth map edges as detailed in Section 3.1. The misalignment of sharp depth map edges is the major cause of the contours in the synthesized images, and our solution to solve this problem is presented in Section 3.2. The second step is warping. In order to reduce the computational errors, when the depth map of the reference plane is projected to virtual view, the textures corresponding to the depth map are directly copied.

The third step is crack-filling. The insufficient sampling of the reference image introduces cracks in the synthesized images. To preserve the original warped information and achieve good filling quality, we propose a crack filling method which need not filter and re-projection. Instead, we use the surrounding depth and color information of non-crack pixels for crack-filling, and relevant method is discussed in Section 3.3. The next step is alpha blending which used to synthesize the warped images together.

The last step is disocclusion inpainting. After get synthesized depth maps and textures, the images still contain disocclusions, due to the fact that some parts of the scene can be seen at the synthesized view but invisible in the reference views. The surrounding non-hole pixels with depth information are used to fill these disocclusions.

IV. ARTIFACTS ANALYSIS AND SOLUTIONS

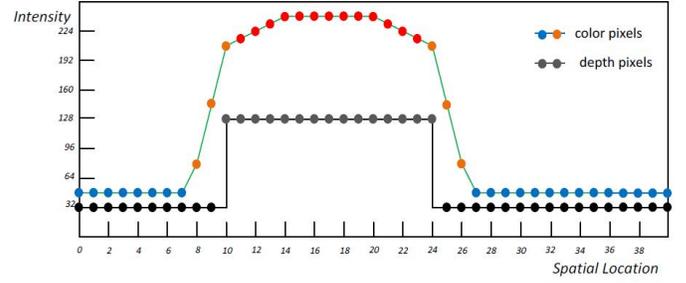
In this section, we will give detailed description of proposed approaches for solving the major artifacts in synthesized images.

4.1 Depth map reliability analysis

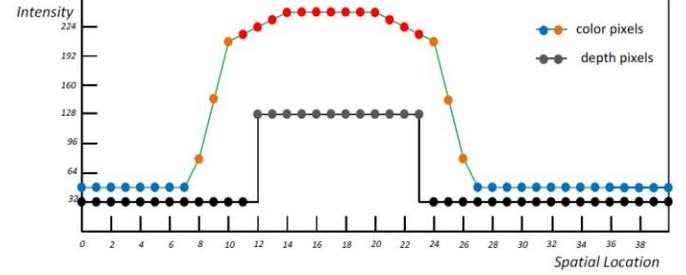
Generally, the depth map captured by depth camera or estimated from video frames may not align with color view correctly. Depth map is a piece-wise image that has large homogeneous regions within scene object and contains sharp changes at object boundaries. However, the edges in color image usually have intensity changing smoothly over transition regions where the object boundaries contain combination of foreground and background. There are detailed descriptions in [17]. Generally, if the depth map aligns with the color image as showed in Fig. 3(a), the object's sharp boundary edge will be at the transition region of color image. However, if the depth map misaligns with the color image, there are two kinds of misalignments between depth map and color image. The first kind misalignment is showed in Fig. 3(b) where the object's sharp boundary edges of depth map align with the foreground of color image. The second kind misalignment is showed in Fig. 3(c) where the object's sharp boundary edges of depth map align with the background of color image.

The synthesized image will contain contour artifacts if these misalignments are not processed. Mori et al. applied a boundary matting method. After 3D warping of depth maps, they expanded the occlusion region to background direction, this erased the mixture of foreground and background color in the background and the occlusion region were filled by the other warped depth map. As this can only remove contours in the background, and fail to deal with contours in the foreground in Fig. 4(b).

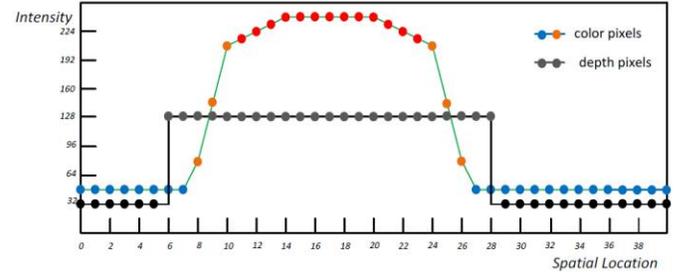
Luat et al. proposed a depth map preprocess method which detected the foreground boundary, and labeled the boundary as unreliable regions. When they did the 3D warping, they only warped the pixels that are not labeled. This method can get a good result for the alignment in Fig. 3(a). For another two kinds of misalignment showed in Fig. 3(b) and Fig. 3(c), this method still can't yield satisfactory results. For the first kind misalignment when the edge of depth aligns with foreground of color image, this method can remove the contour, but it also erased the foreground of color image. This becomes more apparent especially when the foreground object is thin, such as fingers showed in Fig. 4(a). For the second kind misalignment, this method will bring contours in the foreground region as shown in Fig. 4(b). To deal with all three cases of misalignment and remove the contours we propose a new method based on depth map edge detection and color consistency correction. This is discussed in detail the next subsection.



(a) Depth map boundaries aligned with transition region.



(b) Depth map boundaries misaligned with foreground region.



(c) Depth map boundaries misaligned with background region.

Figure 3. Color pixel intensity values and depth values for a horizontal line in video-plus-depth image format[17]

4.2 Contour artifacts removing

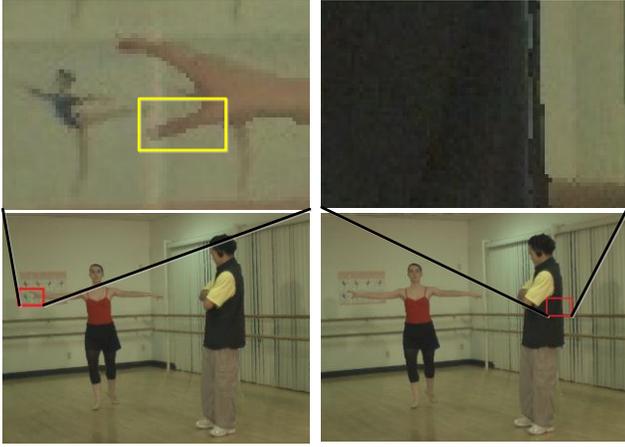
According to the above analysis, the depth map can be refined in the preprocessing stage in order to remove the contour artifacts. For this artifact, we need process all the three kinds of alignment or misalignments. To this end, combination of depth map edge detection and color consistency correction is employed.

In the proposed method, firstly we analyze the depth maps and find out the unreliable regions. If pixels in depth map are satisfied (1) or (2), they are labeled as unreliable pixels. Pixels satisfied (1) correspond to foreground and pixels satisfied (2) correspond to background, respectively.

$$\forall_{u,v} \in S, \left(\sum_{i=-2}^2 \sum_{j=-2}^2 D(u+i, v+j) \right) - (5*5) * D(u, v) > T_d \quad (1)$$

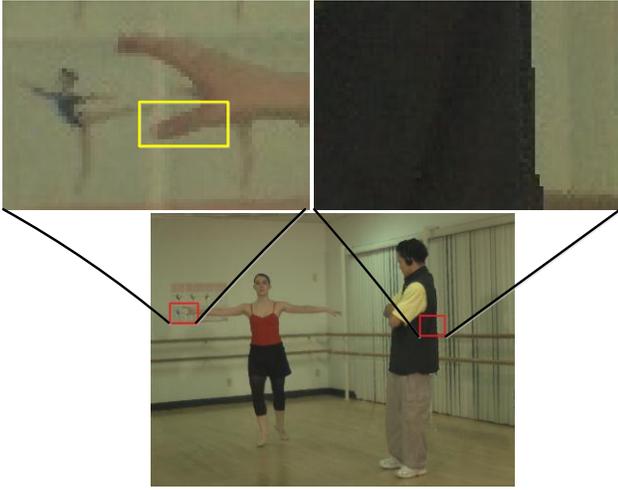
$$\forall_{u,v} \in S, \left(\sum_{i=-2}^2 \sum_{j=-2}^2 D(u+i, v+j) \right) - (5*5) * D(u, v) < -T_d \quad (2)$$

Where S is the image space, D denotes the depth map of the reference camera and T_d is predefined threshold. Generally the mixture region of foreground and background has a width of two to three pixels. To detect such regions, the image block to be examined should be larger than this size, and this is why we use a 5x5 block above.



(a) Result of Luat et al.

(b) Result of Mori et al. and Luat et al.



(c) Result of proposed method

Fig.4: Results comparison. In (a), object boundary shrinks to the foreground direction and finger become thinner. In (b), contours still exist in foreground. In (c), no contour exists in images

We only warp the unlabeled pixels in the depth map and thus there will be no contour artifacts in background of the synthesized view. This process is effective for contours in the background. We needn't worry about excessively erasing the background of depth map as the erased region will be filled by the other reference view. Although it may also remove contours in the foreground, the object boundary shrinks to the foreground direction which damages the foreground objects, especially those small or thin ones such as fingers showed in Fig. 4(a).

To preserve the foreground boundary information and remove contours in foreground at the same time, further work based on color consistency correction are used as follows. First, for every pixel that satisfied (1), we search its neighbor pixels in eight directions and find the nearest unlabeled pixels in each direction. Second, we calculate the differences of color values between current pixel and the unlabeled pixels that in foreground. We re-label the pixels to reliable if (3) is satisfied.



(a) Image with cracks

(b) Image after cracks filled

Figure 5. Processing image crack

$$\forall_{u,v} \in R, \left| \frac{\sum_{i=1}^N c_i * r_i}{\sum_{i=1}^N r_i} - c(u,v) \right| < T_c \quad (3)$$

$$\text{with } r_i = \begin{cases} 1 & , \text{if reliable \& foreground} \\ 0 & , \text{others} \end{cases}$$

In (3), R is the unreliable region which satisfy (2); parameter $N=8$, denotes the eight directions; c_i denotes the color value of the nearest unlabeled pixel in direction i . T_c is a predefined threshold. As we can see, if the unreliable pixels in depth map satisfy (3) which belong to the foreground of color image, they should be relabeled as reliable regions. This process preserves foreground boundary and removes contours in foreground at the same time. The results are given in Fig. 4 (c). As can be seen, our method has the advantage of removing contours from both foreground and background whilst preserving the foreground boundary information.

4.3 Cracks filling

Small cracks often occur in the entire image area and are introduced by the forward mapping nature of image-based 3D warping. In this section, we will present our method for crack filling. To fill the cracks efficiently and preserve the original warped depth information of non-crack regions, we propose a method that only fills cracks and does nothing to non-crack regions. Usually the width of crack is one to two pixels. First, we used (4) to detect the cracks.

$$\forall_{u,v} \in S, crack_{u,v} = \begin{cases} 1 & , \text{if } \sum_{i=-2}^2 \sum_{j=-2}^2 f_{ij} \geq T_n \\ 0 & , \text{else} \end{cases} \quad (4)$$

$$\text{where } f_{ij} = \begin{cases} 1 & , \text{if } |D(u+i, v+j) - D(u,v)| > T_d \\ 0 & , \text{else} \end{cases}$$

Where S is the image space, D denotes the depth map of the synthesized view, T_d and T_n are predefined thresholds, f_{ij} denotes the difference between the current pixel and each surrounding pixel. $crack_{u,v}$ is a flag, indicating whether current pixel is in crack or not. We check the depth difference between the current pixel and each surrounding pixel in a 5×5 block. If the current pixel is in crack regions, most of the surrounding pixels are not in crack regions, we will get a large summation. On the contrary, if the current pixel is not in crack regions, summation will be small.

After detecting cracks, we use the surrounding pixels which are not in crack regions to calculate average values to fill pixels in crack regions, specified by (5).

$$\forall_{u,v} \in S, c(u,v) = \frac{\sum_{i=1}^N c_i * (1 - crack_i)}{\sum_{i=1}^N (1 - crack_i)} \quad (5)$$

Where S is the image space, again c_i denotes the color value of the nearest non-crack pixel in direction i , parameter $N = 8$, denotes the pixels in eight directions, $crack_i$ defined in (4) is the crack label of c_i . After this step, as shown in Fig. 5, cracks in foreground and background are successfully filled and the original information of non-crack regions are preserved.

4.4 Alpha blending

After the warped images were removed contour artifacts and filled cracks, we blend the two images to one using the (6)

$$C(u,v) = \begin{cases} (1-\alpha)C_L(u_L, v_L) + \alpha C_R(u_R, v_R) & (occ_L(u,v) = 0, occ_R(u,v) = 0) \\ C_L(u_L, v_L) & (occ_L(u,v) = 0, occ_R(u,v) = 1) \\ C_R(u_R, v_R) & (occ_L(u,v) = 1, occ_R(u,v) = 0) \\ 0 & (occ_L(u,v) = 1, occ_R(u,v) = 1) \end{cases} \quad (6)$$

$$occ_L(u,v) = \begin{cases} 1 & (D_L(u,v) = 0) \\ 0 & (D_L(u,v) > 0) \end{cases} \quad (7)$$

$$occ_R(u,v) = \begin{cases} 1 & (D_R(u,v) = 0) \\ 0 & (D_R(u,v) > 0) \end{cases}$$

$$\alpha = \frac{|t - t_L|}{|t - t_L| + |t - t_R|} \quad (8)$$

In (6), $C(u,v)$ means the pixel value at (u,v) virtual plane. C_L and C_R mean the images generated by left and right reference views. occ is the occlusion map defined in (7). $D_L(u,v)$ and $D_R(u,v)$ mean the left and right pixel depth values at (u,v) depth image. While depth value equals to zero, it means this pixel stay in disocclusion region which corresponding to the black blank regions in Fig. 6. These pixels can't be used for blending. α is a weighting coefficient defined in (8). t_L, t_R and t are the translation vectors of left camera, the right camera and the virtual camera, respectively. The blending result is shown in Fig. 6.

4.5 Disocclusion inpainting

As we can see from Fig. 6(c), after blending two warped images into virtual image, the synthesized image still contains disocclusions. These regions either occur due to erroneous depth values, or are areas that become visible in the virtual view, while being occluded in both original views. Most disocclusions inpainting methods are based on texture extrapolation. These methods may get good results when disocclusions only surrounded by foreground or background. If the disocclusions are surrounded by both foreground and background, the inpainted regions often contain a certain



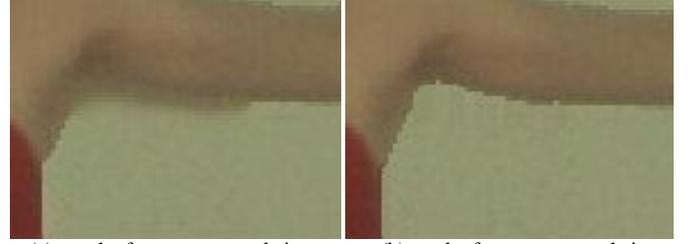
(a) image warped from left reference view

(b) image warped from right reference view



(c) the synthesized image

Figure 6. alpha blending result



(a) result of texture extrapolation inpainting without depth information

(b) result of texture extrapolation inpainting with depth information

Figure 7. the inpainting result

amount of blurs. In the experiment, we found that the disocclusions are always background and not part of foreground. On the basis of this fact, we use texture extrapolation based on depth information to inpaint the disocclusions, specified by

$$\forall_{u,v} \in O, C(u,v) = \frac{\sum_{i=1}^N d_i^{-2} * f_i * C_i}{\sum_{i=1}^N d_i^{-2} * f_i} \quad (9)$$

$$\text{with } f_i = \begin{cases} 0 & , \text{if } \max D_j - D_i > T \quad (j = 1, \dots, N) \\ 1 & , \text{else} \end{cases}$$

Here, O is the disoccluded region, parameter $N = 8$, d is the distance of current pixel to the edge of the disoccluded region and C_i is the texture value of edge pixel in direction i . f_i reflects C_i staying in foreground or background. D_i denotes the depth value. T is predefined threshold. In this summation, only those texture values belonging to the background are used. The advantage of this method is that no blur exists between foreground and background. The drawback is that the inpainted region may become a low-frequency patch, when the disocclusion region is large. Fig. 7 shows the comparison between traditional texture extrapolation and our method.

V. RESULTS

5.1 Image dataset

To evaluate the performance of proposed method, the well-known multi-views sequence of “Ballet” and “Breakdancer” are used. The camera setup of the test sequences consists of eight reference cameras positioned along an arc, spanning about 30° from one end to the other. Both of the two sequences include 100 images captured from every camera. The depth maps were computed from stereo and also included for each camera along with the calibration parameters. The captured images have a resolution 1024×768 . We select camera 4 as the virtual view and compare the generated images with the reference images.

5.2 Experimental results

Both subjective and objective evaluations are applied to evaluate the performance of proposed method. For objective evaluation, the synthesized images are compared with the reference images in the virtual view based on the Peak signal-to-noise ratio (PSNR) and structural similarity metric (SSIM). For subjective evaluation, we present some synthesized images and compare with the reference images.

To compare the object performance of our proposed method with those in Mori et al. [5] and Luat et al. [6], average PSNR results calculated over 100 images are illustrated in Fig. 8 with changing distance between the left and right reference cameras. The distances are calculated by $|t - t_L| + |t - t_R|$ which defined in (8). To compare the performance in detail, Fig.9 (a) and (b) show detailed PSNR results of 100 images synthesized by camera 3 and camera 5. The average PSNR of these images are corresponding to the first points of proposed method in Fig. 8(a) and Fig. 8(b), respectively. In order to evaluate our method more comprehensively, we also present our SSIM results in Table 1. Because Mori et al. and Luat et al. didn’t measure their results on SSIM, we only present our results.

In order to evaluate subjective performance of our method, we present some synthesized images in Fig. 10. Compared with the reference images, our results obtain good visual quality.

5.3 Discussion

As can be seen, for the two sequences, our approach has significantly outperformed the other two and generates much higher PSNR values than them. The large gains are caused by the fact that we successfully remove the contours in foreground and background and we preserve the foreground boundary information at the same time. Our crack-filling method only focuses on the crack regions instead of using filtering which may break the original warped information of non-crack regions. Our disocclusion inpainting method is based on depth information and reduces blurs in the synthesized images.

From experimental results, we can see that the quality of synthesized images have large difference between the two sequence. Through the study of dataset, we find that the scene of “Breakdancer” is more complex than “Ballet” and the depth maps are calculated using stereo vision algorithm, so the depth maps of “Breakdancer” have lower accuracy. The rendering

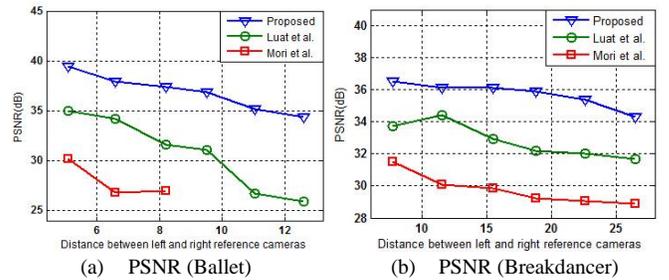


Figure 8. PSNR results with different distances (mm) between left and right reference cameras. Each point of proposed method is an average result of 100 synthesized images.

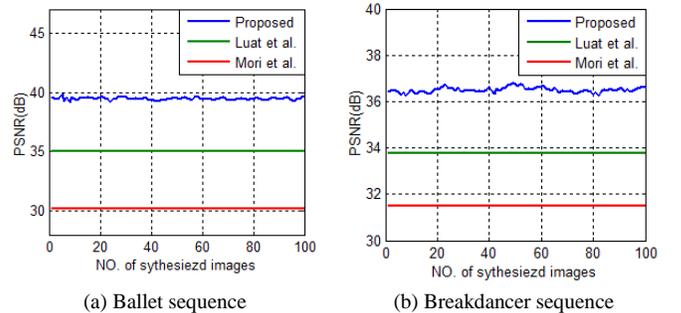


Figure 9. PSNR results of 100 images synthesized by camera 3 and camera 5. Lines from top to bottom are results of proposed, Luat et al. and Mori et al. respectively.

TABLE I. SSIM RESULTS WITH DIFFERENT DISTANCES BETWEEN LEFT AND RIGHT REFERENCE CAMERAS. EACH RESULT IS AN AVERAGE OF 100 SYNTHESIZED IMAGES.

Sequence	Distance between left and right view					
	7.7	11.5	15.5	18.8	22.6	26.5
Ballet	0.95	0.94	0.93	0.91	0.87	0.84
Breakdancer	0.92	0.92	0.91	0.89	0.88	0.85

method based on DIBR is sensitive to the accuracy of depth maps.

We also notice that with the increase of distance between left and right reference cameras, the quality of synthesized images become worse. When the baseline between reference cameras becomes larger, there will be more computing errors and larger occluded areas.

Through the above analysis, the qualities of synthesized images are sensitive to the depth map accuracy and length of baseline between reference cameras.

VI. CONCLUSION AND FUTURE WORK

The view synthesized using methods based on DIBR always contains various artifacts, where contour, crack and disocclusion artifacts are typical cases which have veritably degraded the quality of the synthesized images. In this paper, we propose effective methods to remove such artifacts. Based on depth map edge detection and color consistency correction, contour artifacts are successfully removed from both foreground and background whilst preserving the foreground boundary information. For filling cracks, depth difference is applied to detect crack and non-crack pixels. The proposed



Figure 10. Comparison synthesized images (left) with reference images (right) in the view of camera4.

method for filling cracks is proved to be effective. With the disocclusion regions, we applied a method of texture extrapolation with depth information. Experimental results on two well-known sequences have demonstrated the promising results of the proposed approaches. According to quantitative assessment using PSNR, the proposed algorithm outperforms two state-of-the-art approaches. Good results also are obtained in SSIM and visual quality. To get good synthesized result in

large baseline and faster rendering speed, future work will focus on large baseline and real-time free-viewpoint system.

VII. ACKNOWLEDGMENT

This work was supported by Importance Industry Chain Projects of Science and Technology Coordination Innovation Engineering, Shaanxi, China (2015KTZDGY04-01).

REFERENCES

- [1] M. Domański, A. Dziembowski, A. Kuehn, 2014. Experiments on acquisition and processing of video for Free-viewpoint television, *IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, Budapest, Hungary, pp.1-4.
- [2] M. Tanimoto, M. P. Tehrani, and T. Fujii, 2011. Free-viewpoint TV, *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 67-76.
- [3] Y. Cai, R. Wang, and T. Cui, 2013. Intermediate view syn-thesis based on edge detecting, *IEEE International Conference on Image Processing*, Paris, France, pp. 3172-3175.
- [4] V. Paradiso, M. Lucenteforte, M. Grangetto, 2012. A novel interpolation method for 3D view synthesis, *IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, Zurich, Switzerland, pp. 1-4.
- [5] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, 2009. View generation with 3D warping using depth information for FTV, *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 65–72.
- [6] L. Do, G. Bravo, S. Zinger, 2012. GPU-accelerated real-time free-viewpoint DIBR for 3DTV. *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 633-640.
- [7] H. C. Shin, G. Lee, N. Hur. 2014. View interpolation using a simple block matching and guided image filtering. *IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, Budapest, Hungary, pp. 1-4.
- [8] A. Smolic. 2011. 3D video and free viewpoint video—From capture to display. *Pattern recognition*, vol. 44, no. 9, pp. 1958-1968.
- [9] C. L. Zitnik, S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, 2004. High-quality video view interpolation using a layered representation, *ACM SIGGRAPH*, Los Angeles, USA, pp. 600–608.
- [10] K. Oh, S. Yea, Y. Ho, 2009. Hole-Filling Method Using Depth Based Inpainting For View Synthesis in Free Viewpoint Television (FTV) and 3D Video, *IEEE Picture Coding Symposium*, pp. 1-4.
- [11] A. Smolic, K. Mller, K. Dix, P. Merkle, P. Kauff, T. Wiegand, et al. 2008. Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems, *IEEE Image Processing*, pp. 2448-2451.
- [12] S. Zinger, L. Do, and P. H. N. de With, 2010. Free-viewpoint depth image based rendering, *Journal of Visual Communication & Image Representation*, pp. 533-541.
- [13] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, T. Wiegand, et al. 2008. View Synthesis for Advanced 3D Video Systems, *EURASIP Journal on Image and Video Processing*, pp. 1-12.
- [14] A. Telea, 2004. An image inpainting technique based on the fast marching method, *Journal of Graphics Tools*, vol. 9,no. 1, pp. 25-36.
- [15] P. Ndjiki-Nya, M. Koppel, D. Doshkov, 2011. Depth image-based rendering with advanced texture synthesis for 3-D video. *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453-465.
- [16] W. Chen, Y. Chang, S. Lin, 2005. Efficient depth image based rendering with edge dependent depth filter and interpolation. *IEEE International Conference on Multimedia and Expo*. Amsterdam, The Netherlands, pp. 1314-1317.
- [17] X. Xu, L. M. Po, K. W. Cheung, 2013. Watershed based depth map misalignment correction and foreground bi-ased dilation for DIBR view synthesis, *IEEE International Conference on Image Processing*, Paris, France, pp. 3152-3156.