IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 17, 2019, accepted August 1, 2019, date of publication August 13, 2019, date of current version August 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935021

An Improved Depth Image Based Virtual View Synthesis Method for Interactive 3D Video

SHIPING ZHU, (Member, IEEE), HAO XU, AND LINA YAN

Department of Measurement Control and Information Technology, School of Instrumentation Science and Optoelectronics Engineering, Beihang University, Beijing 100191, China

Corresponding author: Shiping Zhu (spzhu@163.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61375025, Grant 61075011, and Grant 60675018, and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars from the State Education Ministry of China.

ABSTRACT Depth image-based rendering (DIBR) is one of the main techniques in interactive 3D video systems. But a critical problem of DIBR is that how to eliminate the holes appearing in virtual view images. In this paper, an improved depth image based virtual view synthesis algorithm is proposed. In depth map pre-processing, only local areas of the depth map with sudden changes are pre-processed by the asymmetric filter. After this, we combine the forward and inverse 3D-warping to synthesize the virtual view. Firstly, it warps the left image to the virtual view and then only reversely warps the region of the holes in the virtual view to the right view to get the lost information of holes. Compared with the traditional method which have to warp the two whole images, computation cost can be reduced by 30% efficiently. Finally, we employ an improved image inpainting based on depth information to fill hole regions. We have improved the method in two portions: 1) depth information is added to the priority computation and searching the best matching patch and 2) as the filling proceeds, the confidence term of the newly inpainted pixels is updated with the mean square error (MSE) of pixels in the target matching patch and the source matching patch. Experimental results show significant objective and subjective gains of the proposed method in comparison to the reference methods.

INDEX TERMS DIBR, 3D warping, hole filling, depth map.

I. INTRODUCTION

With the development of computer vision and multimedia technology, interactive 3D video has become the main direction of multimedia technology research. Furthermore, it allows users to interactively choose the viewpoint and synthesize a new view dynamically. Therefore, it has received much attention in the broadcast research community as a promising technology for three-dimensional television (3D TV) systems [1]–[4]. However, transmitting the video in every view will lead to the drastic increase in the volume of data, which accounts for the need of using view synthesis techniques [5]. In general, virtual view synthesis algorithms can be separated into two major classes, model-based rendering (MBR) and image-based rendering (IBR) [6]. Among all the algorithms, DIBR has become the main approach in interactive 3D video systems due to its low bandwidth cost as well as the arbitrary rendering viewpoint [7]–[10]. The main technique of DIBR can be realized by the three-dimensional image transform technique (3D image warping) [11]. However, the most significant problem in DIBR is how to deal with holes appearing in the virtual images which are mainly caused by the occlusion between objects. To address the problems, many algorithms have been proposed in recent years.

One solution would be to rely on more complex multi-dimensional data representations, like Layered Depth Image (LDI) algorithms as represented in [12], [13]. In this instance, several layers of depth image are needed to provide sufficient information. Although this method can reduce the craze efficiently, it also will increase the computing complexity. In [14], Muller improved this method and increased constraint conditions when choosing level constraints. However, it is more computationally demanding and requires more bandwidth for transmission.

Basis on above, some researchers proposed the technology of processing depth maps as described in [15]–[18] to reduce the holes in the virtual viewpoint. Experimental results show that this added pre-processing step can reduce the initial

The associate editor coordinating the review of this article and approving it for publication was Lei Wei.

errors of the depth map and the warped holes in the virtual image at some degree. However, the traditional pre-filter may lead to depth map blurring both the holes regions and non-hole regions. Besides, they will also bring about geometric distortion and texture artifacts.

Furthermore, some image inpainting algorithms are used in the post-processing of the warped image. In this instance, holes in the virtual image are filled with the neighboring pixels of the image. Ho *et al.* [19] used the points around the background to fill the holes of the obtained images. But this algorithm is limited to fill small holes. Criminisi *et al.* [20] proposed an image inpainting algorithm which combined the partial differential equation with the parameters of texture synthesis. This method adopted the concept of sampling proposed by Efros and Leung [21] and acquired good effects in the holes filling step. However, background textures could not be propagated to the hole regions since the hole regions lying in the foreground sometimes is filled before hole regions lying in the background and foreground pixels may be used for prediction. Annoying artifacts are created in synthesized views. Besides, as filling proceeds in the hole, the priorities of newly filled pixels become smaller even to zero. But in Criminisi's image inpainting, it simply takes the newly fixed and source pixels with the same confidence. There are also studies to reduce the coding distortion [22], [23].

In this paper, we propose an improved depth image based virtual view synthesis algorithm to deal with the occlusions in virtual images. The main contributes are: 1) Depth maps are pre-processed by our proposed local asymmetric depth filter instead of traditional filter, which can avoid blurring the non-hole region and bringing the geometric distortion. 2) We combine the forward and inverse 3D-warping instead of the traditional warping two images and blending them. In the inverse 3D-warping step, we only warp the holes region in the virtual image to the right image instead of warping the whole image which can reduce the computation cost a lot. 3) Compared with Criminisi's image inpainting method, we add depth information to the priority computation and matching patch equation. Besides, we also update the computation of the confidence term which is proportional to the mean square error (MSE) of pixels in the target matching patch and the source matching patch.

The remainder of this paper is organized as follows: Section 2 gives a detailed description of the proposed virtual view synthesis algorithm. The experimental results are analyzed and evaluated in detail in Section 3. Finally, we conclude our work and outline the future research directions in Section 4.

II. PROPOSED DIBR BASED VIRTUAL VIEW RENDERING ALGORITHM

To improve the synthesized view quality, we propose a novel algorithm to generate the new virtual view image. The block of the overall system is shown in Fig. 1. In the following, these steps will be discussed in detailed.

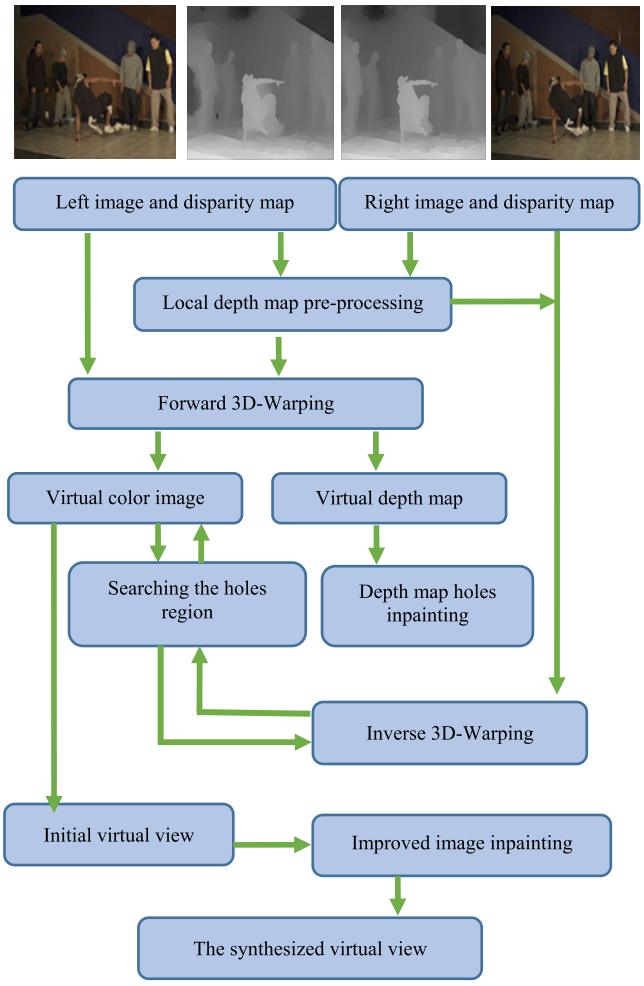

FIGURE 1. The block diagram of the proposed algorithm.

A. LOCAL DEPTH MAP PRE-PROCESSING

Generally, the holes often occur in the mutation regions between foreground and background. So it is reasonable to filter such regions with sudden changes on depth value to reduce the holes while keeping other regions as their original values. Therefore, we propose a local depth map filter to refine the depth map without bring any blur and geometric distortion.

The process of detecting the mutation regions consists of two processes: edge extraction and mutation regions chosen. Firstly, we implement edge detection using morphological operations whose speed is 100 times faster than Canny edge detection. This operation is defined as:

$$\mathrm{M} = (\mathrm{I}_1 \oplus s) - \mathrm{I}_1, \tag{1}$$

where $\oplus$ denotes image dilation; I_1 denotes the binary depth map. s denotes the structural element of dilation which is given by the following equation:

$$s = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0, \end{bmatrix} \tag{2}$$

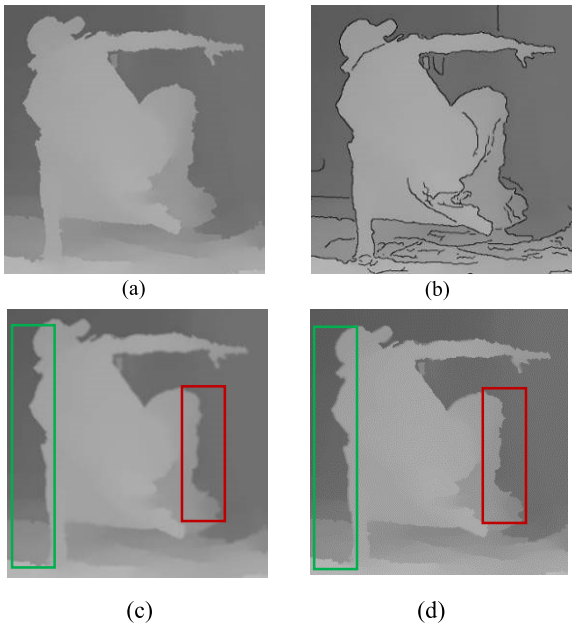

FIGURE 2. The process of local depth map filter. (a) A close up of partial depth map of "Breakdancers". (b) Depth map after edges being extracted. (c) Depth map after the whole edge regions being filtered. (d) Depth map with the mutation pixels only near the right edges being the being filtered.

The edge map of the left depth mapping is detected as Fig. 2(b). In the depth pre-processing, we only filter the mutation regions in the background and ensure the accuracy of the depth value in the foreground. The holes regions need to be filtered are associated with the parallax distance value between the reference view and virtual view. The size of the holes region being filtered is associated with the neighboring depth value as defined:

$$\Delta D = \frac{h}{b \times f} \times \frac{1}{(\frac{1}{255Z_n} - \frac{1}{255Z_f})}, \tag{3}$$

where ΔD is the difference of depth value between the neighboring pixels in the horizontal direction. h is the size of the warped holes in the horizontal direction. b and f represent the distance of baseline and focal length. Z_n and Z_f represent the nearest and farthest distance from the camera of the scene.

After labeling the holes regions, we apply an asymmetric Gaussian filter to filter the transition regions to avoid depth distortion. In this paper, the vertical and horizontal filter strength are set as 12 and 4, respectively. The window size of the corresponding filter is three times of the strength. The image after filtering is shown as Fig. 2(c).

Moreover, since holes usually exist at the regions where the depth values are from lower to higher in the left reference view, and the regions where the depth values are from higher to lower in the right reference view, we can only filter these partial edge regions that may result in holes in 3D warping.

Fig. 2 shows the results of filtering the edge regions of objects. Fig. 2(c) is the depth map with respect to that the whole edge regions are filtered. By contrast, Fig. 2(d) shows the result with respect to local filtering in which only outer pixels near the left edges are filtered as shown in the red and green rectangular. From Fig. 2, we can see that mutation pixels of depth map, which may become hole regions after warping, will be filtered by our proposed algorithm without bring any blur to other original regions.

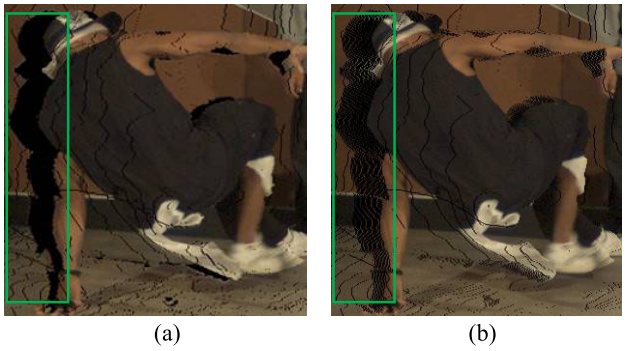

FIGURE 3. Results of the warped image. (a) Without depth map pre-processing. (b) Proposed local depth map pre-processing.

Fig. 3 shows the comparison of the depth map before and after being filtered using our proposed pre-processing algorithm. From Fig. 3(b), we can see that the number of holes in the warped image reduced at some degree. But the overall effect is not improved obviously and need further processing.

B. COMBINATION OF FORWARD AND INVERSE 3D-WARPING

Traditionally, left and right images adopt the forward 3D warping to synthesize the virtual view images. Then the warped images are merged to fill the big holes. However, most of the pixels in the left and right image are the same and only a small part of big holes need to be filled. Warping the whole two images will cost a lot of time and we can reduce the computation cost by reducing the warping data. Hence, we apply the combination of the forward and inverse 3D-warping algorithm to the left and right images respectively. In this paper, we treat the left viewpoint as the reference and the right viewpoint as the assistance. The detailed steps are as the following.

Step 1: Forward 3D-warping. Left image and its corresponding depth map are warped to synthesis the virtual image using forward 3D-warping function. The warping function can be expressed as follows:

$$s_L \mathbf{m_L} = \mathbf{K_L}[\mathbf{R_L}|\mathbf{t_L}]\mathbf{M}, \tag{4}$$

$$z_V \mathbf{m_V} = \mathbf{K_V}\mathbf{R_V}\mathbf{M} + \mathbf{K_V}\mathbf{t_V}, \tag{5}$$

$$z_v \mathbf{m_v} = \mathbf{K_V}\mathbf{R_V}(\mathbf{K_L}\mathbf{R_L})^{-1}(s_L \mathbf{m_L} - \mathbf{K_L}\mathbf{t_L}) + \mathbf{K_V}\mathbf{t_V}, \tag{6}$$

where m_L and m_V are coordinates of the reference image and the synthesized virtual view; s_L and z_V are scale parameter equaled to the depth value in the world space. M is the world coordinates. Matrix K_L and K_V represent the intrinsic parameters of reference camera and virtual camera; R_L and t_L represent external parameters of the reference camera; R_V and t_V represent external parameters of the virtual camera.

We assume the left camera coordinate overlaps with the world coordinate. Thus, the parameters are set as $R = I_{3\times3}$, $t = 0_{3\times1}$ and $s_L = z_V$. So, equation (6) can be simplified as:

$$z_v\mathbf{m_v} = \mathbf{K_V}\mathbf{R_V}\mathbf{K_L^{-1}}s_L\mathbf{m_L} + \mathbf{K_V}\mathbf{t_V}, \tag{7}$$

Through the 3D-warping function, we can get the warped depth map and color image as shown in Fig. 4(a) and Fig. 5(a) respectively.

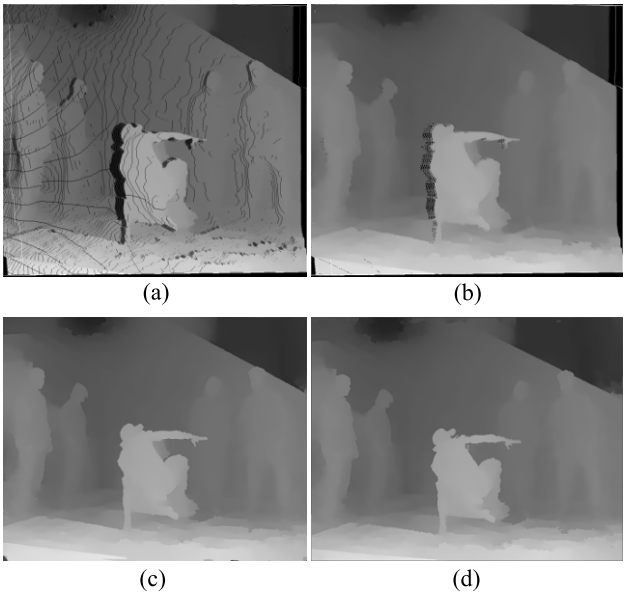

FIGURE 4. Results of the warped virtual depth map. (a) Initial warped depth map. (b) Depth map after small cracks filling. (c) Large holes filling. (d) The truth depth map.

Step 2: Filling the synthesized depth map. From Fig. 4(a), we can see the virtual depth map will contain many cracks and holes because of occlusion. Since depth map can be considered as a grey-scale image without texture, it is much easier to be filled. For small cracks, they are filled with liner interpolation of the neighboring pixels. The image after cracks being eliminated is shown in Fig. 4(b).

When the hole occurs in a flat area, the holes region should be filled with the most common disparity of its neighborhood. On the contrary, in the mutation area, the holes preferred to be filled with the neighboring disparity with lower depth value. Based on this, the large holes regions in the depth map are filled by histogram voting method. Centered around each hole, we select an $N \times N$ window N. Firstly, we calculate the number of the pixels in different value of the window N. The values are classified into B bins. If the window could not contain the full hole, its size is increased by ΔN. We use the variance σ^2 to determine the holes existing in the mutation area or flat area. The disparity value I to fill the holes is set as:

$$I = \arg\min(Cost(b_i)), \quad 1 \le i \le B, \tag{8}$$

$$Cost(b_i) = \mu\sigma^2 b_i + \frac{1}{n(b_i)}, \tag{9}$$

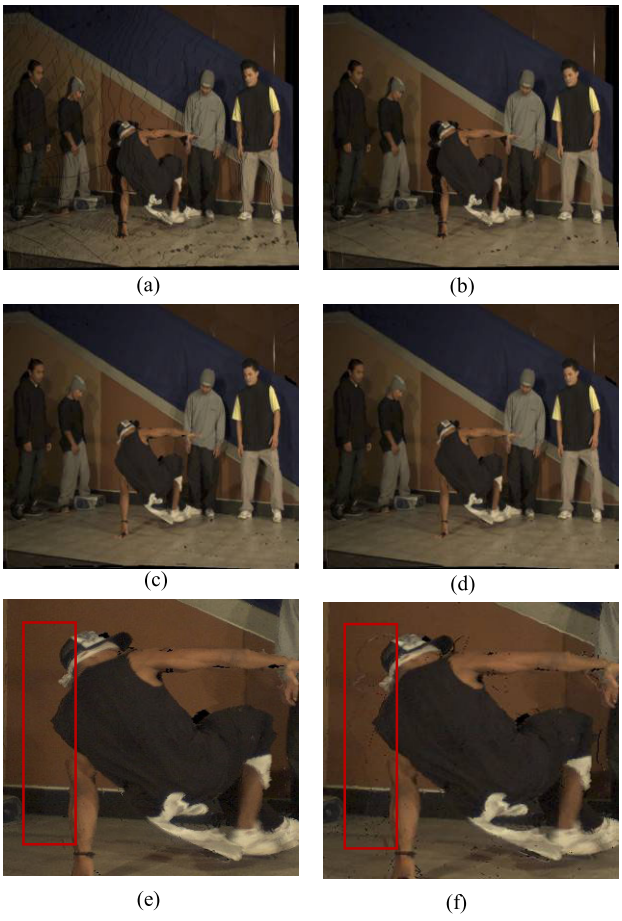

FIGURE 5. Detailed results of the combination of the forward and the inverse 3D warping. (a) Forward 3D warping. (b) Cracks filling. (c) Proposed FI-DIBR algorithm. (d) Traditional 3D warping. (e) and (f) are magnified portions of (c) and (d).

where $Cost(b_i)$ is the cost function of b_i in the histogram; $n(b_i)$ is the number of pixels in b_i; μ is a regulation parameter. When the variance σ^2 is high, the cost is mainly depended on the first term of equation (9). On the contrary, when the σ^2 is low, the holes exists in a flat region and should be inpainted with the most common disparity. The cost is mainly depended on the second term $n(b_i)$. The balance is adjusted by the parameter μ. In this paper, we used $N = 31$, $\Delta N = 12$, $B = 10$ and $\mu = 1000$ respectively.

The result is shown in Fig. 4(c). Fig. 4(d) is the real depth map. We can see the holes in the depth map can be filled efficiently and we also compute the PSNR of the virtual inpainted depth map which is 30.49dB.

Step 3: Inverse 3D-warping. To fill the holes and cracks in the virtual color image warped by step 1, we adopt an inverse 3D-warping. Firstly, we fill the small cracks by liner interpolation and the result is shown in Fig. 5(b). Furthermore, to prevent the artifacts occurring in the boundary, we expand the holes boundary in the virtual images with 3×3 rectangular. Thus, the regions where may occur artifacts are wiped as a hole which will be filled with the information in the right image.

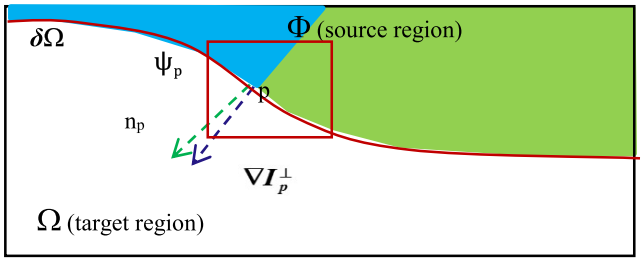

FIGURE 6. The process of image inpainting algorithm.

After the holes being enlarged, we should fill them with the inverse 3D-warping method. The kernel idea of the method is that starting from the virtual view, for each hole pixel in the virtual view, we can find a corresponding point in the 3D world according (9). Then the calculated 3D point is projected onto the right image by employing (10).

$$\mathbf{M} = z_V \mathbf{R_V}^{-1}\mathbf{K_V}^{-1}\mathbf{m_V} - \mathbf{R_V}^{-1}\mathbf{t_V}, \tag{10}$$

$$s_R\mathbf{m_R} = \mathbf{K_R}[\mathbf{R_R}|\mathbf{t_R}]\mathbf{M}, \tag{11}$$

where s_R is scale parameter equaled to the depth value in the world space; m_R is the coordinate of the right image; Matrix K_R represents the intrinsic parameters of right camera; R_R and t_R represent external parameters of the right camera.

In this way, we can find each hole pixel in the virtual view m_V corresponding to the pixel in the right assistant viewpoint m_R and obtain the color information of hole pixel m_V. Fig. 5(c) shows the holes after being filled with the proposed algorithm. From Fig. 5(e) and (f), we can see our method can eliminate the artifacts efficiently and get a satisfying effect as shown in the red rectangular.

C. IMPROVED IMAGE INPAINTING

Although the conventional Criminisi's image inpainting algorithm can be applied to the disocclusion inpainting, a significant disadvantage is that some holes may be wrongly filled with the information in the foreground. To make it more appropriate, we propose a depth map based image inpainting by giving higher priority to background over foreground. We make some improvements in the following two aspects: Filling order estimation and patch matching principles.

1) FILLING ORDER ESTIMATION

Firstly, we should compute the priority of the patches with the center pixel on the boundary of the holes region. Considering an input image I, and a missing region Ω as shown in Fig. 6. The source area is defined as: $\Phi = I - \Omega$. I is the input image. The boundary of the hole area is denoted as $\delta\Omega$. Pixel p belongs to the boundary namely $p \in \delta\Omega$. Ψ_p denotes the target matching block with the center of p.

In Criminisi's image inpainting algorithm, the priority of the p is defined as:

$$P(p) = C(p)D(p), \tag{12}$$

$$C(p) = \frac{\sum_{q\in\Psi_p\cap\Phi} C(q)}{|\Psi_p|}, \tag{13}$$

$$D(p) = \frac{\left|\nabla I_p^{\perp}\cdot n_p\right|}{\alpha'}, \tag{14}$$

where $C(p)$ and $D(p)$ represent the confidence term and data term respectively. $|\Psi_p|$ denotes the area of Ψ_p. n_p denotes the unit vector which is perpendicular to $\delta\Omega$ at pixel p. $\nabla^{\perp} = (\partial y, -\partial x)$ denotes the isophote direction operator. α' is the normalization coefficient and its value can be set as 255 for grey image. $C(q)$ is the last iteration of $C(p)$. The value of $C(p)$ depends on the number of source pixels contained the matching block. More source pixels included, higher the credibility item is. $D(p)$ assigns higher priority to the points on the edge of the delay line. In the initial conditions, we set $C(p)$ as follows:

$$C(p) = \begin{cases} 0 & \text{for } \forall p \in \Omega \\ 1 & \text{for } \forall p \in \Phi \end{cases} \tag{15}$$

a: IMPROVED PRIORITY COMPUTATION

As filling proceeds in the hole, the priorities of pixels in the new holes boundary need to be updated timely. In Criminisi's image inpainting, it simply takes the newly fixed and source pixels with the same confidence value. However, since there are some errors between in the newly fixed pixel and the real pixel in the location, the error will be larger along with the propagation of the inpainting patch. So the confidence term $C(p)$ of the newly fixed pixel gradually becomes smaller or even to zero. No matter how large the data term $D(p)$ is, the priority $P(p)$ will likely remain zero and could not determine the order of patch filling. Therefore, equation (12) is modified to sum form as follows:

$$P(p) = \alpha \times C(p) + \beta \times D(p) + \gamma \times Z(p) \tag{16}$$

where $Z(p)$ denotes the depth term which ensures the patch with lower depth value has higher priority. α, β and γ are the default weights which are set as 0.5, 0.3 and 0.2, respectively. $Z(p)$ is defined as:

$$Z(p) = \frac{\sum_{i\in\Phi\cap\psi_p} d_{\max} - Z(i)}{d_{\max} \times N_1} \tag{17}$$

where $d_{\max}$ denotes the maximal depth value in the depth map; $Z(i)$ is the depth value of source pixels in ψ_p. N_1 is the number of the source pixels contained the matching block.

b: CONFIDENCE TERM UPDATED WITH IMPROVED ALGORITHM

Along with the image inpainting, the confidence term of the newly filled pixels becomes smaller. However, they will be used as the new source pixels in the next inpainting step. Thus, in order to prevent the error of the inpainted pixels propagating, the confidence term is updated as:

$$C(p) = \frac{\sum_{q\in\Psi_p\cap\Phi} C(q) \times e^{-MSE}}{|\Psi_p|}, \tag{18}$$

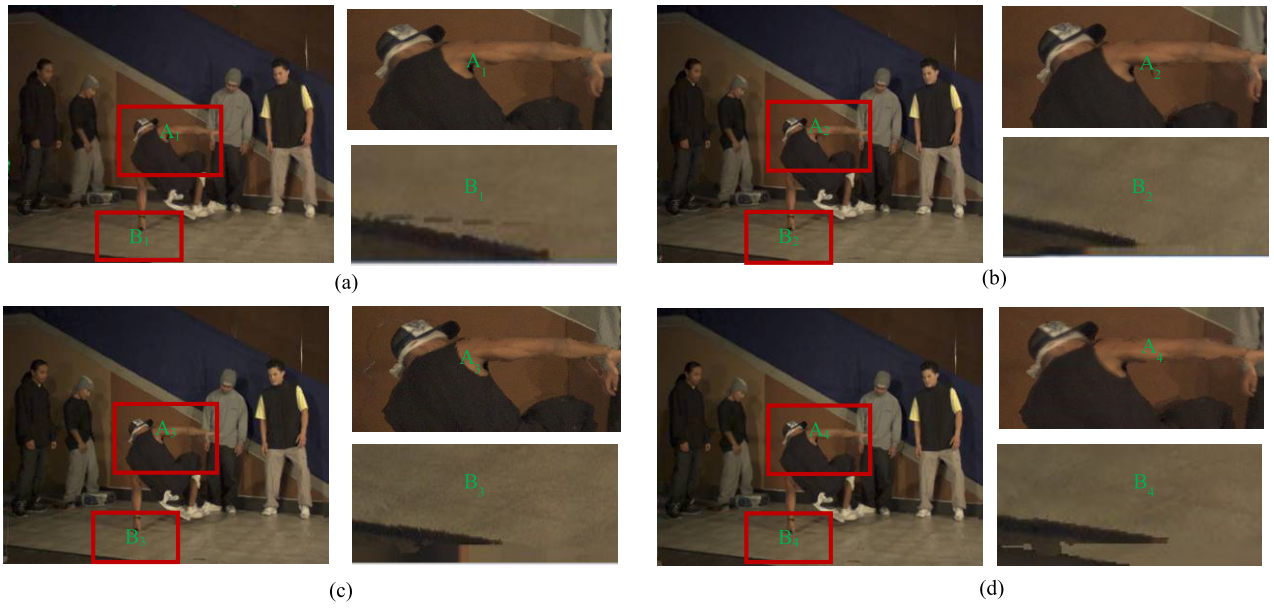

FIGURE 7. Performance comparison of the four algorithms of "Breakdancers". (a) Proposed algorithm. (b) Method of [24]. (c) Method of [25]. (d) Image inpainting with method [20].

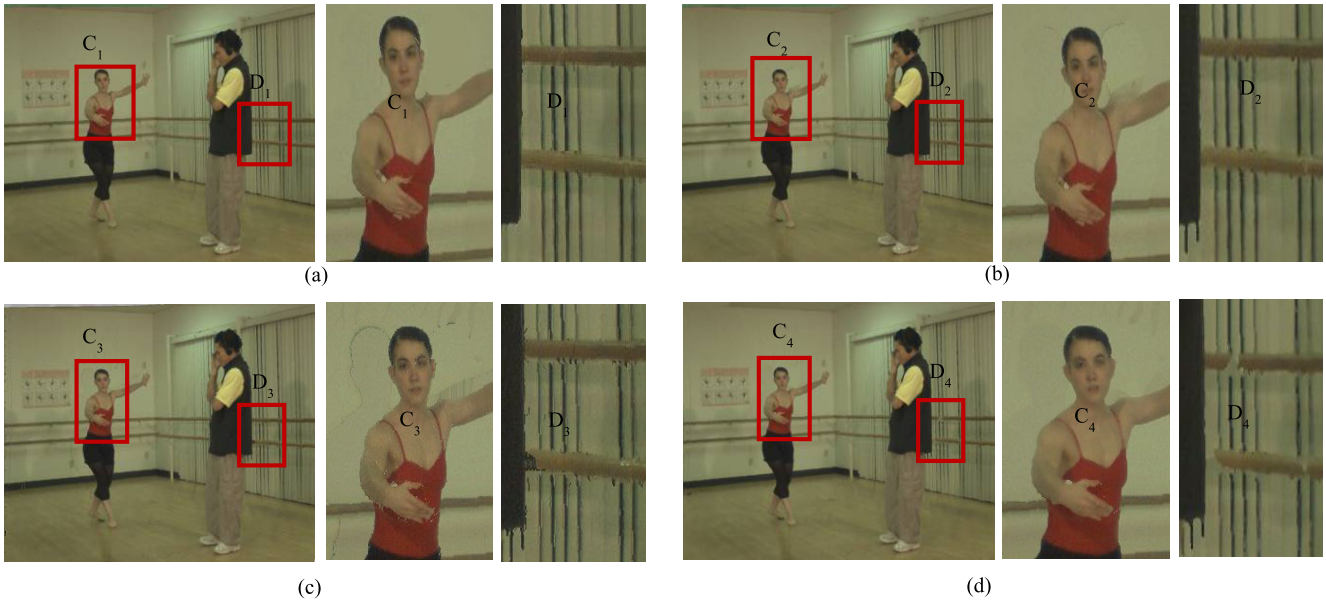

FIGURE 8. Performance comparison of the four algorithms of "Ballet". (a) Proposed algorithm. (b) Method of [24]. (c) Method of [25]. (d) Image inpainting with method [20].

$$MSE = \frac{\sum_{i \in \psi_p \cap \Omega} (y_i - x_i)^2}{N_2}, \tag{19}$$

where y_i denotes the source pixel value in the last target patch $\Psi_{\hat{p}}$; x_i is the pixel value in the last best matching patch block $\Psi_{\hat{q}}$. N_2 is the number of the source pixels of ψ_p. MSE is the mean square error (MSE) of pixels in the target matching patch and the source matching patch. The larger the MSE is, the smaller confidence term of the pixel is.

2) IMPROVED PATCH MATCHING

After priority computation, the patch $\Psi_{\hat{p}}$ with the highest priority is selected to be filled at first. We apply block matching algorithm to find the most similar matching block $\Psi_{\hat{q}}$ from the source area Φ to fill the holes. It is important to choose the best matching candidates in the background and restrict the search to the same depth level. Thus, depth information is introduced in this equation in order to find the best matching patch with the smallest matching cost.

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} \{SSD_{RGB}\left(\Psi_{\hat{p}}, \Psi_q\right) + SSD_D\left(Z_{\hat{p}}, Z_q\right)\} \tag{20}$$

where $Z_{\hat{p}}$ denotes the depth information of pixels in the best matching patch; Z_q is the corresponding depth value of pixels in the target matching patch; SSD represents the sum

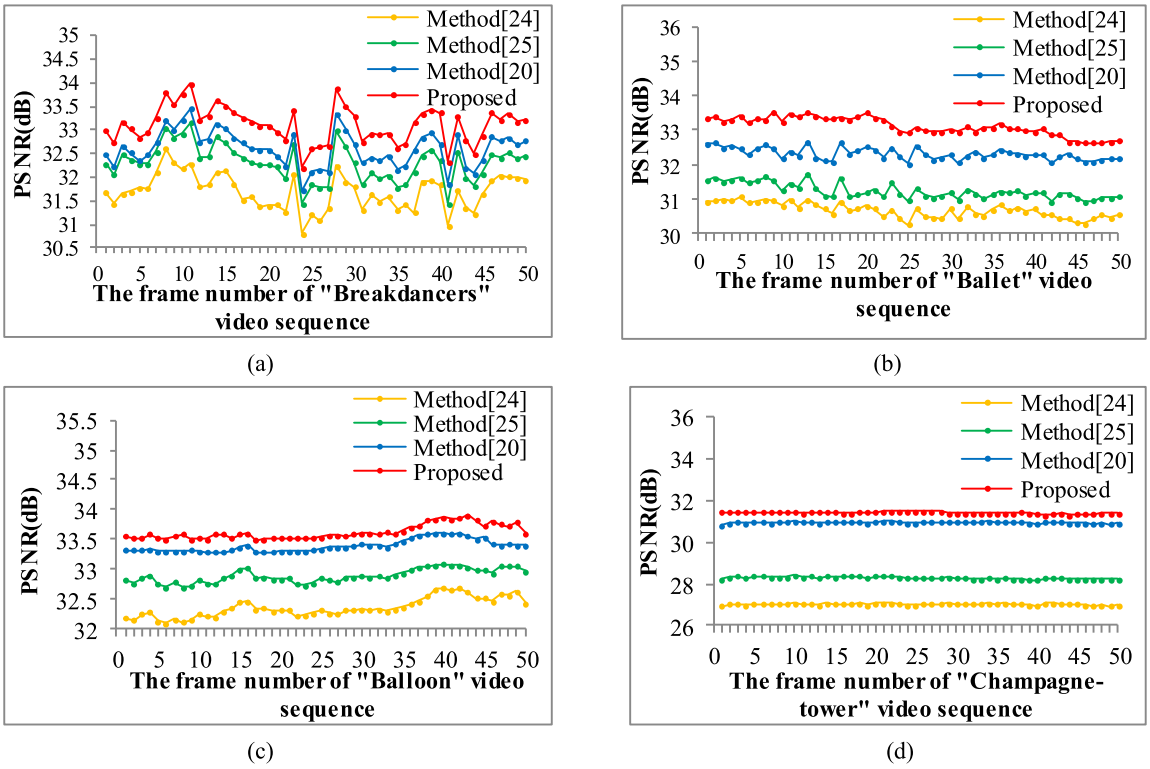

FIGURE 9. Peak signal noise ratio (PSNR) curves of four test images.

TABLE 1. The comparison of the average objective quality of virtual view images.

Video sequence	PSNR (dB)				SSIM			
	Proposed	VSRS[24]	Inverse 3D-warping[25]	Method[20]	Proposed	VSRS[24]	Inverse 3D-warping[25]	Method[20]
Breakdancers	**33.11**	31.32	32.30	32.60	**0.93**	0.90	0.90	0.91
Ballet	**33.12**	30.69	31.24	32.31	**0.94**	0.92	0.92	0.93
Balloon	**33.61**	32.35	32.89	33.38	**0.97**	0.95	0.95	0.96
Champagne-tower	**31.40**	27.01	28.28	30.93	**0.97**	0.94	0.95	0.97
Average	**32.81**	30.34	31.18	32.31	**0.95**	0.93	0.93	0.94

of squared difference (SSD) function between the matching patches.

Compared with Criminisi's method, depth information is added to the similarity function which can ensure the depth consistence between the candidate and target patches. By considering the depth information, the best matched patch has a very similar property as the patch to be filled with both the texture and depth information. After getting the best matching block, we copy the color information of $\Psi_{\hat{q}}$ to the target block $\Psi_{\hat{p}}$. Repeating the above process, we can complete filling the holes area.

III. EXPERIMENTS AND ANALYSIS

A. EVALUATION OF VIDEO SEQUENCE USING OUR ALGORITHM

We run the proposed algorithm on an Intel Core (TM) i7 CPU with 3.60GHz under MATLAB 2014 environment. The performance of the proposed method is evaluated on 3D video "Breakdancers" and "Ballet", "Balloon" and "Champagne-tower". The filling patch size is set as 5×5 and the search window size is set as 45×45 empirically. We test the one frame of video "Breakdancers" and "Ballet" to see the visual image quality. In all simulations, we treat left viewpoint as the reference image and right viewpoint as the assistant image. The compared experimental results of our proposed method, the VSRS [24] virtual view synthesis of the method proposed in [25] and image inpainting with [20] are shown in Fig. 7 and Fig. 8, respectively. The main differences of images generated by these four algorithms are marked with red rectangles and magnified for further comparison.

There are some obvious wrongly inpainted patches in Fig. 7(b), (c) and (d). This is because the methods of VSRS [24] and Criminisi *et al.* [20] do not take the depth information into consideration. Besides, all of the three methods of [24], [25] and [20] simply take the newly inpatinted pixels have the same confidence term, which leads the holes

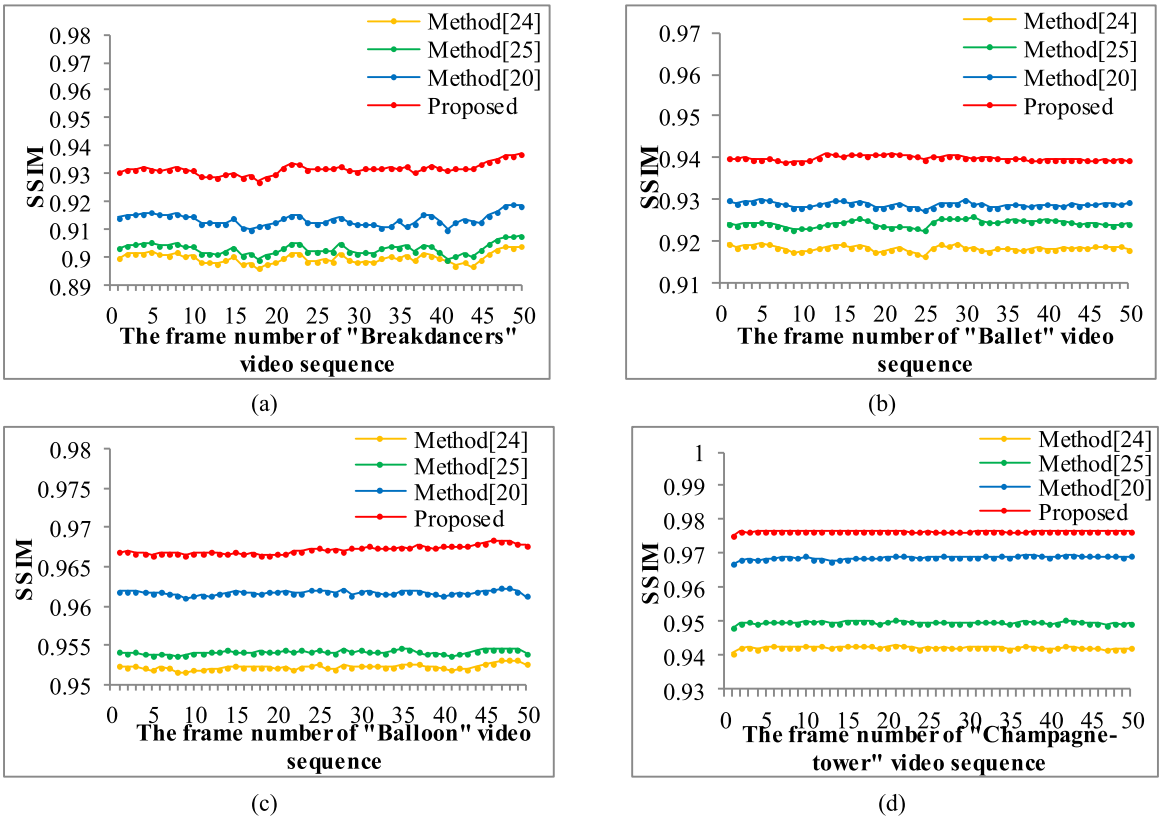

FIGURE 10. Structural similarity (SSIM) curves of four test images.

filled with wrong pixels. Furthermore, there are artifacts at the edge of foreground objects, which can be found in A3 of Fig. 7(c). This is because the algorithm proposed in [25] simply warped the pixels with inverse 3D warping function and fill the warped holes without eliminating the pixels where may occur artifacts. Virtual view generated by our proposed method is shown in Fig. 7(a). Through subjective comparison, it can be seen that the proposed method performs well as expected.

As can be seen from Fig. 8, obvious artifacts appear in C2 and C3, which degrades the quality of the performance seriously. Besides, There are many matching errors in Fig. 8 (c) and (d), especially in the D3 and D4. In comparison with these three algorithms, the proposed algorithm can synthesize more natural images with satisfying visual quality.

To evaluate the performance of the algorithm more objectively, the PSNR [26] and SSIM [27] performance of the proposed method is reported. The comparison results of the four video sequences using different algorithms are shown in Fig. 9 and Fig. 10. Especially, method [20] represents the images synthesized by our proposed algorithm except inpainting the images with the traditional Criminisi's algorithm.

As shown in Fig. 9 and Fig. 10, we can see that virtual view images using our proposed algorithm have a higher peak signal noise ratio and structural similarity than the compared algorithms. Specially, compared with method [20], we can see our improved image inpainting algorithm have a significant improvement in the image quality.

Furthermore, we compute the average PSNR and SSIM of per frame as shown in Table 1. From Table 1, we can see the proposed algorithm has increased the average SSIM by 0.01-0.02 and PSNR by 0.5-2.37dB. Meanwhile, we also have achieved a great progress than our previous algorithm, which is discussed in [28].

In addition, to test the high efficiency of combining the forward and inverse 3D-warping method, we also compute the execution time of the four compared algorithms on the four test video sequences. Fig. 11 shows the average execution time of per frame. The average running time of the pre frame of four test video sequences "Breakdancers", "Ballet", "Balloon", "Champagne-tower" is: 2.9s, 2.38s, 2.6s and 3.8s. Compared with method [24] our algorithm has reduced the running time about 30%. This is because our proposed algorithm only warping one image and disocclusion region instead of warping the two whole images. Method [25] has the highest complexity because it uses inverse 3D warping to warp the two images which costs more time. Compared with Criminisi's method [20], our proposed method costs a little more time. This is because the proposed method's priority computation is more complex than that of traditional Criminisi's.

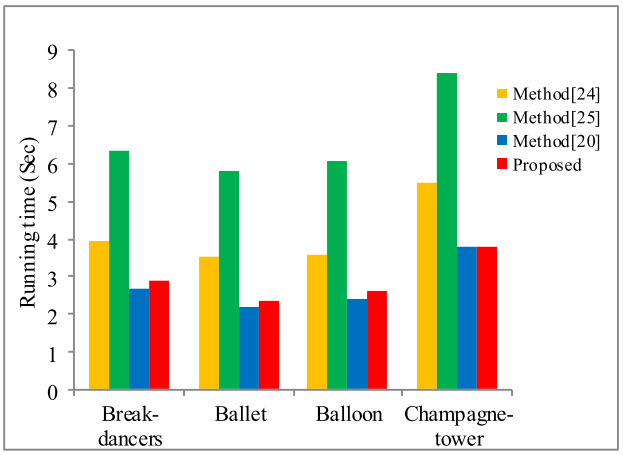

FIGURE 11. The average running time of per frame.

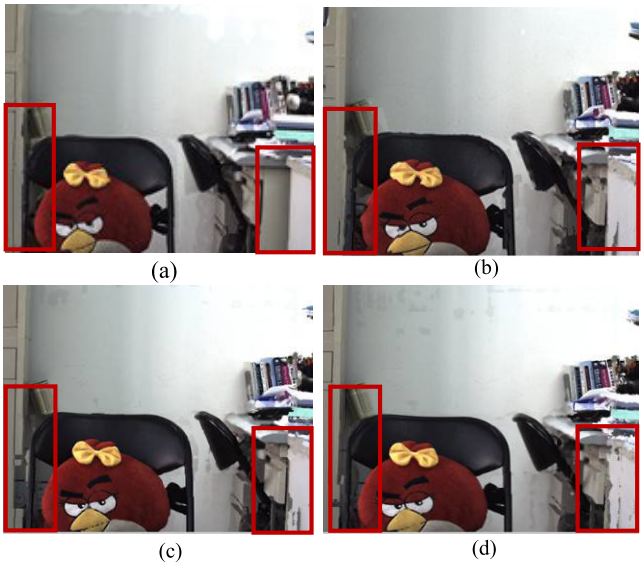

FIGURE 12. Comparison of different algorithms on image "Birds". (a) Proposed algorithm. (b) VSRS method [24]. (c) Method [25]. (d) Criminisi's method [20].

B. EVALUATION OF REAL SCENE IMAGES USING OUR ALGORITHM

In additional, we also synthesize the virtual view images with the real scene images taken by the binocular cameras of our laboratory. We take the real scene image pairs named "Birds" with resolution of 500×378. The corresponding disparity maps are obtained using our stereo matching algorithm proposed in [29]. Based on the images and the disparity maps, we can synthesize the virtual view images at any position between the two cameras. The synthesized virtual view images are shown in Fig. 12.

From Fig. 12, we can see the virtual image in Fig. 12(a) is better than that of (b), (c) and (d). There are many error matching pixels in the red rectangular. However, there still exist some artifacts and holes in the boundary of the images. This phenomenon is mainly due to the low accuracy of the obtained disparity map. Therefore, we can enhance the quality of the virtual view image by optimizing the disparity maps.

IV. CONCLUSION AND FUTURE WORK

This paper presents an improved DIBR based virtual view synthesis algorithm. The proposed algorithm efficiently solved the geometric distortion by using the local depth map filter instead of the traditional filter. Besides, we also reduced the algorithm cost by combining the forward and inverse 3D warping, which only warp one view and the disocclusion region. And it is important for the real time DIBR system. Finally, in image inpainting process, we improved the priority computation by adding depth information and update the confidence term of the newly inpainted pixels, which improved inpainting results efficiently. Experimental results show that our algorithm has an outstanding performance than the other algorithms in both the standard video sequences and real scene images. In recent years, there are also some new studies [30], [31] on virtual view synthesis. In the future work, we will investigate how to further improve the quality of the virtual view image.

REFERENCES

[1] A. Smolic, "3D video and free viewpoint video—From capture to display," *Pattern Recognit.*, vol. 44, no. 9, pp. 1958–1968, 2011.
[2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.
[3] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 191–199, Jun. 2005.
[4] S. Zinger, L. Do, and P. H. N. de With, "Recent developments in free-viewpoint interpolation for 3DTV," *3D Res.*, vol. 3, no. 1, p. 4, 2012.
[5] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 65–72, 2009.
[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H. N. De With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, 2009.
[7] L. Gao, H. Chen, C. Liu, and W. Zhao, "A newly virtual view generation method based on depth image," in *Proc. IEEE 11th Int. Conf. Signal Process.*, Beijing, China, Oct. 2012, pp. 1088–1091.
[8] J. Konrad, M. Wang, and P. Ishwar, "2D-to-3D image conversion by learning depth from examples," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Providence, RI, USA, Jun. 2012, pp. 16–22.
[9] L. Wang, C. Hou, J. Lei, and W. Yan, "View generation with DIBR for 3D display system," *Multimedia Tools Appl.*, vol. 74, no. 21, pp. 9529–9545, 2015.
[10] I. Ahn and C. Kim, "A novel depth-based virtual view synthesis method for free viewpoint video," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 614–626, Dec. 2013.
[11] L. Mcmillan, "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, Dept. Comput. Sci., UNC, Chapel Hill, NC, USA, 1997.
[12] L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004.
[13] Y. Mao, G. Cheung, and Y. Ji, "On Constructing *z*-dimensional DIBR-synthesized images," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1453–1468, Aug. 2016.
[14] K. Muller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3D video systems," *EURASIP J. Image Video Process.*, vol. 2008, Dec. 2009, Art. no. 438148.
[15] W. J. Tam, G. Alain, L. Zhang, T. Martin, and R. Renaud, "Smoothing depth maps for improved steroscopic image quality," *Proc. SPIE*, vol. 5599, pp. 162–172, Oct. 2004.

[16] W.-Y. Chen, Y.-L. Chang, S.-F. Lin, L.-F. Ding, and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1314–1317.
[17] Y. K. Park, K. Jung, Y. Oh, S. Lee, J. K. Kim, G. Lee, H. Lee, K. Yun, N. Hur, and J. Kim, "Depth-image-based rendering for 3DTV service over T-DMB," *Signal Process. Image*, vol. 24, nos. 1–2, pp. 122–136, 2009.
[18] W. Liu, D. Zhang, M. Cui, and J. Ding, "An enhanced depth map based rendering method with directional depth filter and image inpainting," *Vis. Comput.*, vol. 32, no. 5, pp. 579–589, 2016.
[19] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," in *Proc. Picture Coding Symp.*, Chicago, IL, USA, May 2009, pp. 1–4.
[20] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
[21] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Corfu, Greece, Sep. 1999, pp. 1033–1038.
[22] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3-D video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 485–497, Apr. 2011.
[23] H. Yuan, J. Liu, Z. Li, and W. Liu, "Coding distortion elimination of virtual view synthesis for 3D video system: Theoretical analyses and implementation," *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 558–568, Dec. 2012.
[24] M. Gotfryd, K. Wegner, and M. Domański, *View Synthesis Software and Assessment of Its Performance*, document ISO/IEC JTC1/SC29/WG11 MPEG/M 15672, Hannover, Germany, 2008.
[25] C. Cheng, J. Liu, H. Yuan, X. Yang, and W. Liu, "A DIBR method based on inverse mapping and depth-aided image inpainting," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Beijing, China, Jul. 2013, pp. 518–522.
[26] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Istanbul, Turkey, Aug. 2010, pp. 2366–2369.
[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[28] S. Zhu, Z. Li, and Y. Yu, "Virtual view synthesis using stereo vision based on the sum of absolute difference," *Comput. Electr. Eng*, vol. 40, no. 8, pp. 236–246, 2014.
[29] S. Zhu, R. Gao, and Z. Li, "Stereo matching algorithm with guided filter and modified dynamic programming," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 199–216, 2015.
[30] D. M. M. Rahaman and M. Paul, "Virtual view synthesis for free viewpoint video and multiview video compression using Gaussian mixture modelling," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1190–1201, Mar. 2018.
[31] S. Fachada, D. Bonatto, A. Schenkel, and G. Lafruit, "Depth image based view synthesis with multiple reference views for virtual reality," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (3DTV-CON)*, Helsinki, Finland, Jun. 2018, pp. 1–4.

SHIPING ZHU (M'05) received the B.Sc. and M.Sc. degrees from the Xi'an University of Technology, Xi'an, China, in 1991 and 1994, respectively, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 1997.

From 1997 to 1999, he was a Postdoctoral Fellow with Beihang University, Beijing, China. From 2000 to 2002, he was a Postdoctoral Fellow with the Brain and Cognition Research Center, Université Paul Sabatier, Toulouse, France. From 2002 to 2004, he was a Postdoctoral Fellow with the Department of Computer Science and the Department of Electrical and Computer Engineering, Université de Sherbrooke, Sherbrooke, QC, Canada. Since 2005, he has been an Associate Professor with Beihang University. He has authored or coauthored more than 80 journal and conference papers. He is the holder of 50 China invention patents. His current research interests include image processing and video coding, computer vision, machine vision for 3-D measurement, etc.

HAO XU received the B.Sc. degree in measurement and control technology and instrumentation from Beihang University, Beijing, China in 2017, where he is currently pursuing the master's degree. His current research interest includes stereo matching.

LINA YAN was born in Hebei, China. She received the B.Sc. degree in measuring and testing technologies and instruments from the Hebei University of Science and Technology, Hebei, China, in 2014, and the M.Sc. degree in instrumentation science and technology from Beihang University, Beijing, China, in 2017. Her current research interests include image processing, computer vision, and virtual reality.