electronics

MDPI

Article

Fast Hole Filling for View Synthesis in Free Viewpoint Video

Hui-Yu Huang * and Shao-Yu Huang

Department of Computer Science and Information Engineering, National Formosa University, Yunlin 632, Taiwan; 10663114@gm.nfu.edu.tw

* Correspondence: hyhuang@nfu.edu.tw

Received: 21 April 2020; Accepted: 26 May 2020; Published: 29 May 2020

Abstract: The recent emergence of three-dimensional (3D) movies and 3D television (TV) indicates an increasing interest in 3D content. Stereoscopic displays have enabled visual experiences to be enhanced, allowing the world to be viewed in 3D. Virtual view synthesis is the key technology to present 3D content, and depth image-based rendering (DIBR) is a classic virtual view synthesis method. With a texture image and its corresponding depth map, a virtual view can be generated using the DIBR technique. The depth and camera parameters are used to project the entire pixel in the image to the 3D world coordinate system. The results in the world coordinates are then reprojected into the virtual view, based on 3D warping. However, these projections will result in cracks (holes). Hence, we herein propose a new method of DIBR for free viewpoint videos to solve the hole problem due to these projection processes. First, the depth map is preprocessed to reduce the number of holes, which does not produce large-scale geometric distortions; subsequently, improved 3D warping projection is performed collectively to create the virtual view. A median filter is used to filter the hole regions in the virtual view, followed by 3D inverse warping blending to remove the holes. Next, brightness adjustment and adaptive image blending are performed. Finally, the synthesized virtual view is obtained using the inpainting method. Experimental results verify that our proposed method can produce a pleasant visibility of the synthetized virtual view, maintain a high peak signal-to-noise ratio (PSNR) value, and efficiently decrease execution time compared with state-of-the-art methods.

Keywords: depth image-based rendering (DIRB); free viewpoint TV; 3D warping; view synthesis; image blending; image inpainting

1. Introduction

Owing to the current technological era and the rapid development of digitalization, trends in three-dimensional (3D) videos and imaging technology are developing. The recent emergence of 3D movies and 3DTV indicates increasing interest in 3D content. Although we live in a 3D world, most TV and computer screens provide only two-dimensional (2D) images. Various improvements in 3D technology have garnered more interest in multiview video (MVV) applications such as 3DTV, which offers depth perception without requiring special glasses. Development based on stereoscopic displays that capture, transmit, and display two separate video streams has been used to develop the 3DTV system [1].

Free-viewpoint television (FTV) is regarded as the ultimate 3DTV; it consists of an infinite number of virtual views and provides viewers the flexibility to freely select a viewpoint for viewing 3D scenes [2,3]. A practical method to obtain a free viewpoint video (FVV) is to create virtual views from multiview images. However, multiview video transmission requires a large bandwidth, thus limiting its use [4]. An alternative approach to solve this problem is to generate virtual views using a single reference texture and its coordinate depth map [1]. Depth-based image rendering (DIBR) is a core

technology to generate a virtual view synthesis [1–9]. The base concept of DIBR involves the projection of two reference texture views onto a selected virtual view. The neighboring side views (left-side and right-side reference views) with the reference depth maps are backprojected separately to the points on a 2D image into the 3D world coordinates; subsequently, the resulting points in the 3D world coordinates system are reprojected into the 2D virtual image. A virtual view is synthesized by two virtual views warped from the neighboring side views. This process is known as 3D warping [10]. However, the backprojection and reprojection processes generate cracks (holes), ghosts, and disocclusion regions. These cracks (holes) occur owing to sampling in the x-and y-directions of the reference images and inaccurate depth values. Disocclusion regions refer to areas that are nonexistent in the reference view, but become visible from the virtual view, or those that are visible in the reference view, but become invisible in the virtual view. The disocclusion regions are located in the background [6]. The edges of the disocclusion regions may contain artifacts called ghosts, which occur on depth discontinuities [11]. Hence, overcoming holes, ghosts, and disocclusions in the DIBR algorithm based on 3D warping is a major challenge in virtual view synthesis. Several state-of-the-art DIBR algorithms have been proposed to improve the rendering quality by reducing the foregoing challenges [5,6,8,11–16].

Owing to the drawback of DIBR, filling disocclusion regions is challenging in view synthesis, because those regions are typically large. Hence, an exemplar-based inpainting algorithm by Criminisi et al. [17] is typically used to fill the large holes and disoccluded regions. This exemplar-based inpainting algorithm by [17] computes the priority for a pixel at a hole boundary according to the confidence term and data term; furthermore, it searches for the optimal patch from the entire source image, in which the patch with the best priority region is selected and pixels within the selected patch are copied to fill the holes. Algorithms such as those reported in [4,12,13] are based on the propagation of the neighborhood information (or rely on the depth information) to overcome holes, ghosts, and disocclusions. Daribo and Saito [12] added depth information to the priority computation involved in the exemplar-based inpainting algorithm by Criminisi et al. Subsequently, Gautier et al. [13] extended Criminisi's algorithm. They defined the data term using a 3D structure tensor of the Di Zenzo matrix and added the depth information in the best patch calculation module. However, the Di Zenzo matrix could reflect only the strong gradients. Therefore, it will produce blurred results under diffusion during the inpainting. Oh et al. [15] used a histogram matching strategy before blending, such that the side views have similar color conditions with the blended virtual view; additionally, they used the fast matching method (FMM) [18] to achieve hill inpainting. However, this method cannot entirely remove color discontinuities between the unoccluded and disocclusion regions. With and Zinger [16] preprocessed a depth map using a low-pass filter and used a bilateral filter to smooth the boundaries between foregrounds (FGs) and backgrounds (BGs), such that holes and disoccluded regions could be reduced. However, if the depth map after filtering is extremely smooth, the geometric distortion and the rendering view quality will be degraded.

In the exemplar-based inpainting method [17] for hole filling, FG textures are prone to be mapped to disoccluded regions using the original inpainting method [17]. This is a mistake because missing information is derived from the BG region. To solve this problem, Muddala et al. [11] used layered depth images warped from 3D warping to segment the BG and FG and then extracted the boundary between them based on a depth threshold; the combined depth information and the hole boundary belonged to the BG and hole filling was achieved using the inpainting algorithm. The inpainting algorithm of [11] modified the filling priority order, including the depth information to favor BG pixels. However, a different order does not guarantee that the FG information will not be propagated to holes in the virtual view. Several algorithms [4,11,13,19,20] are based on changing the filling priority order.

Cheng et al. [21] proposed a DIBR method based on inverse warping and a depth value to remove holes and inpainting, based on the exemplar-based method [17]. Using the dilation process to avoid ghosts that appear on the edges of disocclusions, Yang et al. [8] proposed a statistical method for removing color discontinuities; in this method, the brightness of one side view is used as a base to adjust that of another side view. Subsequently, holes were filled by the depth-assistance asymmetric

dilation inpainting method. Zarb and Debono [22] improved Yang et al.'s [8] method, that used depth data to extract the BG regions; they used the brightness of the BG regions as a base to adjust the brightness. Finally, by utilizing depth information to determine the hole nature and to subsequently select the appropriate neighboring pixels, hole filling was interpolated using the method in [6].

In addition, several approaches [23–26] proposed hole filling methods based on BG construction from a succession of frames and using the available information in a temporal domain. Su et al. [23] used the relation between frames to increase the accuracy of repairing information by the moving behavior and texture similarity. In [24], Sehmeing and Jiang used neighboring pixels from a spatial-temporal window to determine the BG candidates for each pixel to fill in the hole region. Disocclusion filling involves a patch-based inpainting method, but uses superpixels instead of the square of image patches; using superpixels can reduce the number of entities. Although this method can be used to obtain the BGs of the hole regions, it often requires substantial time consumption when camera motion occurs. Lie et al. [25] proposed generating a key-frame sprite BG that is removed from the FG to fill a disocclusion/hole. Luo and Zhu [26] proposed creating a BG video based on the foreground removal and removed positions filling to remove the FGs and then to refill FGs positions from both a 2D image. Its corresponding BG depth created based on the FGs extracted and removed in the depth map by computed 3D warping; subsequently, a BG video and its BG depth map are generated before 3D warping and are used to eliminate holes in the synthesized video. However, the approaches in [25,26] rely heavily on the correct classification of FG regions, which is difficult, especially when several depth layers exist. Oliveira et al. [27] proposed an improved patch-based inpainting that involved the depth term and the BG term in the priority computation to select the best patch. The patch was selected using an erosion process to remove the FG information and to fill the holes. However, the FG–BG classification could not distinguish precisely; it could not remove ghosts on the edges between the FG and BG. Many other approaches fill all holes in a single step using all valid image contents [28,29].

In this study, we focused on the challenges of DIBR to fill the holes/disocclusions in virtual view synthesis. This proposed method includes DIBR with improved 3D warping, filtering, 3D inverse warping, brightness adjustment, blending views, and hole filling to achieve our purpose.

The remainder of this study is organized as follows. Section 2 presents the related techniques. The proposed method is described in Section 3. In Section 4, we present the experimental results that verify the performance of the proposed method. Finally, Section 5 concludes this study.

2. Related Techniques

In this section, we briefly describe the methods applicable to our proposed approach.

2.1. Camera Geometry Model

For 3D warping in DIBR, the camera geometry model is a fundamental framework in generated 3D vision. A general camera is modeled as a pinhole, as shown in Figure 1. C is the camera center and I is the view plane in the camera's coordinates. The orthogonal axis with a line including C is called the optical axis (Z) and its intersection with I is the principal point (p). $m_k = (u, v)$ is the coordinates of M in the I plane, which is the projection of W expressed as $w = (x, y, z)$ in the world coordinates. A full perspective projection is represented by mapping from 3D to 2D coordinates by a linear transformation in homogenous coordinates [30–32]. Let $\widetilde{m} = [u\ v\ 1]^T$ and $\widetilde{w} = [x\ y\ z\ 1]^T$ be homogenous coordinates for M and W, respectively. According to the transformation matrix $\widetilde{\mathbf{P}}$ in perspective transformation, the 3D coordinates are transformed to 2D coordinates, expressed as

$$k\widetilde{m} = \widetilde{\mathbf{P}}\widetilde{w} \tag{1}$$

where k is a scale factor called the perceptive depth. k becomes the true orthogonal distance of the point from the local plane to the camera. According to [33], k is expressed as

$$k = \frac{1}{\frac{d_z}{255}\left(\left(\frac{1}{MinZ}\right) - \left(\frac{1}{MaxZ}\right)\right) + \left(\frac{1}{MaxZ}\right)}, \tag{2}$$

where d_z is the depth value in the depth map; $MinZ$ and $MaxZ$ denote the nearest and the farthest depth values in the 3D real world, respectively. $\widetilde{\mathbf{P}}$ is decomposed into the product

$$\widetilde{\mathbf{P}} = \mathbf{A}[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}, \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{3\times3}$ is the camera's intrinsic parameter matrix, formed by

$$\mathbf{A} = \begin{bmatrix} -k_u f & \gamma & u_0 \\ 0 & -k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

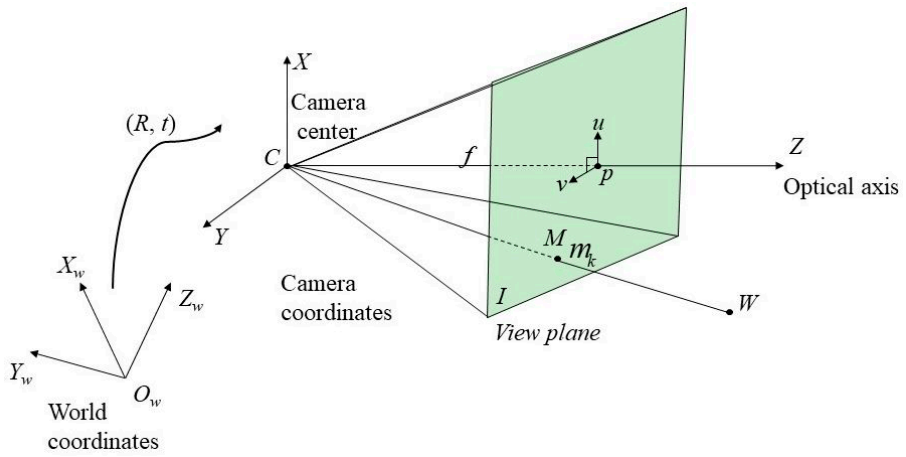


Figure 1. Pinhole camera geometry model. C is the camera center and p is a principal point in the view plane. f is the local length in millimeters.

Matrix $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times4}$ indicates the camera's extrinsic parameters matrix that contains a 3×3 rotation matrix R and the vector 3×1 translation t between the world plane and the camera plane. k_u and k_v are the numbers of pixels per unit distance in (u, v) coordinates; γ is the skew factor; (u_0, v_0) are the coordinates of the principal point p given by the intersection of the optical axis and view plane, as shown in Figure 1.

2.2. Traditional 3D Warping

A typical DIBR virtual view synthesis composed of left-side and right-side cameras using the camera parameter, texture image, and depth images, is illustrated in Figure 2. Moreover, 3D warping is the core method to generate a depth-based virtual view synthesis by transferring two side views based on DIBR.

In 3D warping [12], first, a 2D pixel coordinate (u, v) in the reference view is backprojected into 3D world coordinates; $d(u, v)$ denotes the depth value in the depth map, corresponding to the coordinate (u, v) in the reference view. Next, a second projection is performed from the 3D world coordinates to the target camera at pixel location (U, V), and so on for each pixel location, as shown in Figure 3.

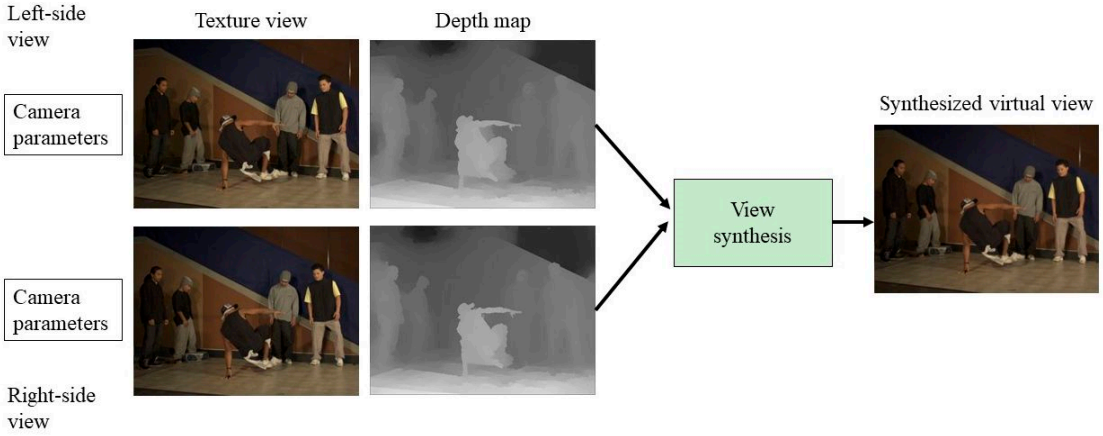


Figure 2. Profile of depth image-based rendering (DIBR) virtual view synthesis.

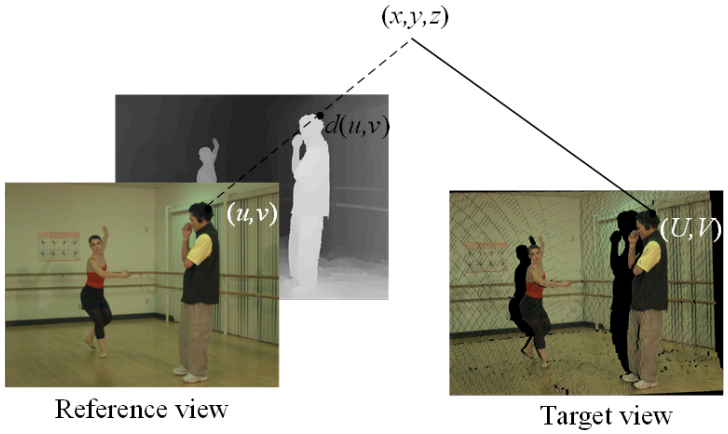


Figure 3. Profile of traditional 3D warping $d(u,v)$ denotes the depth value in the depth map corresponding to the coordinates (u,v).

Generally, 3D warping can be divided into two steps: a backprojection of the reference view into a 3D world coordinates system, followed by a projection of the backprojected 3D scene into the targeted view based on (5) and (6) [12,30]. To perform these operations, the dimensions of the camera intrinsic matrix, rotation matrix, and translation vector are required. $\mathbf{K}_r$, $\mathbf{R}_r$, and $\mathbf{t}_r$ denote the 3×3 camera intrinsic matrix, the 3×3 rotation matrix, and the 3×1 translation vector of the reference view, respectively. The 3D world backprojected point (x, y, z) is expressed in nonhomogeneous coordinates as

$$(x, y, z)^T = \mathbf{R}_r^{-1}\mathbf{K}_r^{-1}(u, v, 1)^T d(u, v) - \mathbf{R}_r^{-1}\mathbf{t}_r. \tag{5}$$

By the target camera quantities $\mathbf{K}_t$, $\mathbf{R}_t$, and $\mathbf{t}_t$, the backprojected 3D world point $(x, y, z, 1)$ is then mapped into the targeted 2D view coordinates $(U, V, 1)$ in homogeneous coordinates as

$$(u'_t, v'_t, w'_t)^T = \mathbf{K}_t\mathbf{R}_t(x, y, z)^T + \mathbf{K}_t\mathbf{t}_t. \tag{6}$$

The homogeneous system is converted into the pixel location, denoted in integer coordinates $(U, V, 1) = \left(\frac{u'_t}{w'_t}, \frac{v'_t}{w'_t}, 1\right)$ in the virtual view.

3. Proposed Method

In this section, we describe the proposed method, including depth map preprocessing, improving 3D warping, filtering, 3D inverse warping, brightness adjustment, adaptive blending of virtual views,

and hole filling. The flowchart of the proposed system is illustrated in Figure 4. The details of the procedures are described in the following subsections.

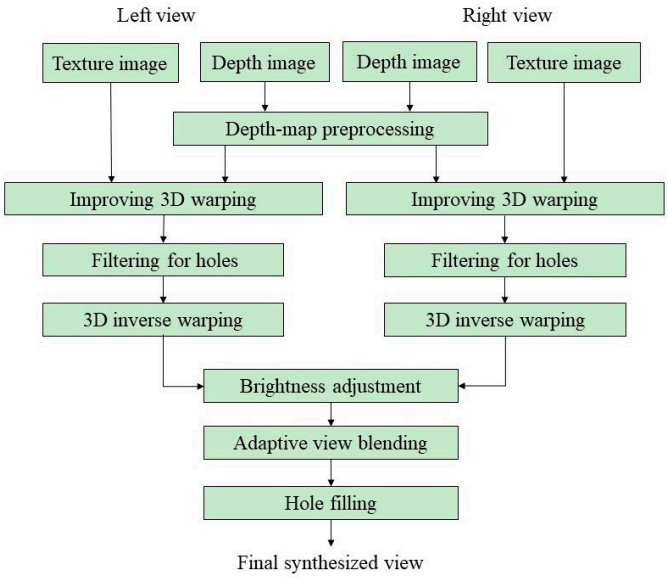
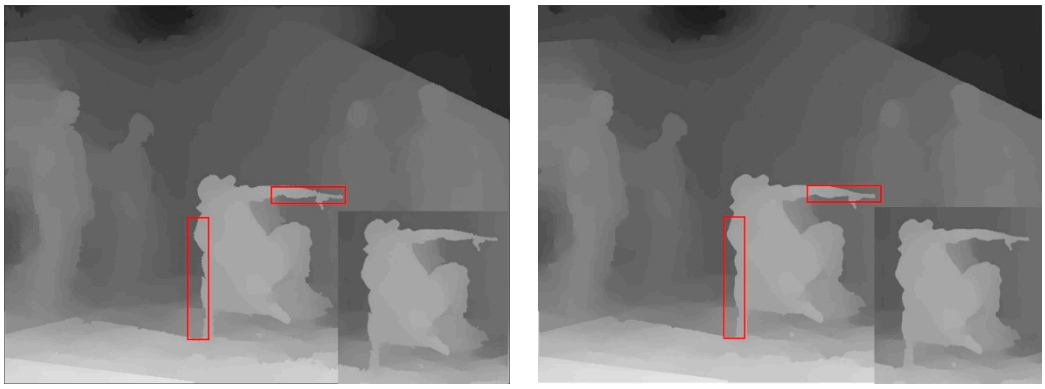


Figure 4. Flowchart of the proposed system.

3.1. Depth Map Preprocessing

Generally, the depth map can be generated using a depth camera system, computer graphics methods, or mathematical calculation using a depth estimation algorithm. Although depth estimated from a depth camera system is the most popular approach, a depth camera is costly and the images generated by computer graphics computation cannot illustrate real scenes. Hence, calculating depth values mathematically is currently adopted. However, the depth values estimated by mathematical calculations tend to produce erroneous values in certain regions in the image, or yield inconsistent values in various regions across spatial or temporal neighbors. Hence, we first performed a closing morphologic procedure twice and then used a median filter of size 5×5 to preprocess the initial depth map. Based on these strategies, the initial depth map can retain the native information well and efficiently reduce hole generation in the subsequent 3D warping procedure. Figure 5 shows the results of the depth map preprocessing.



(**a**) (**b**)

Figure 5. Example of depth map preprocessing for Breakdancers. (**a**) Initial depth map and (**b**) processed result of (**a**).

3.2. Improving 3D Warping Process in DIBR

Traditional depth-based 3D warping typically generates erroneous points, such as those illustrated as black points in Figure 6. This is because many pixels of the texture image are produced after performing a backprojection from the 2D reference view to the 3D world coordinates system, and the reprojection of backprojection from the 3D world coordinates system to a 2D virtual view is projected to the same location in the virtual view, because of round-off coordinates. These erroneous points (called holes) will seriously affect the visual quality of video sequences and cause inconsistent color values, presented as false black-contours in the synthesized view. To reduce the generation of holes, we propose a novel strategy to improve the traditional 3D warping; our proposed novel 3D warping process can significantly reduce the number of holes in each sequence.

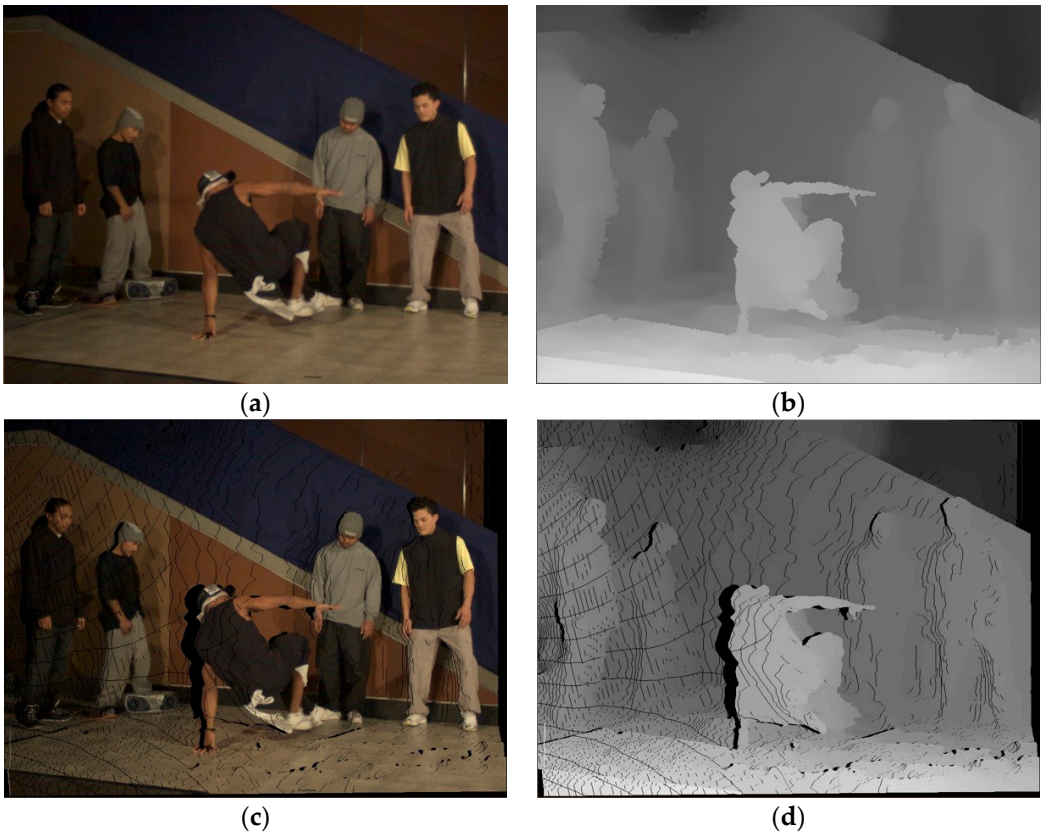


Figure 6. Example of traditional 3D warping. (**a**) Texture image, (**b**) depth map, (**c**) 3D warped texture image with erroneous points, corresponding to (**a**), and (**d**) 3D warped depth map with erroneous points, corresponding to (**b**).

As mentioned above, the 3D warping process requires the camera's intrinsic and extrinsic matrices provided by [33] to transform a 2D reference view plane to 3D world coordinates, and from 3D world coordinates to a 2D virtual view plane. First, let (s,t) be the pixel coordinates in the 2D reference view plane; they are then backprojected into the 3D world coordinates $\left(W_x, W_y, W_z\right)$ using (5). Equation (5) is rewritten as

$$\begin{pmatrix} W_x \\ W_y \\ W_z \end{pmatrix} = \mathbf{R}_{ref}^{-1}\mathbf{K}_{ref}^{-1}\begin{pmatrix} s \\ t \\ 1 \end{pmatrix} d(s,t) - \mathbf{R}_{ref}^{-1}\mathbf{t}_{ref}, \tag{7}$$

where $\mathbf{R}_{ref}$, $\mathbf{K}_{ref}$, and $\mathbf{t}_{ref}$ denote the rotation matrix of size 3×3, the camera intrinsic matrix of size 3×3, and the translation matrix of size 3×1, respectively. $d(s,t)$ denotes the depth value corresponding

to the coordinates (s,t). Next, we perform the second projection from the 3D world coordinates $\left(W_x, W_y, W_z\right)$ to the 2D virtual view coordinates (U_r, V_r) using (8) and (9).

$$\begin{pmatrix} V_u \\ V_v \\ V_w \end{pmatrix} = \mathbf{K}_{vir}\mathbf{R}_{vir}\begin{pmatrix} W_x \\ W_y \\ W_z \end{pmatrix} + \mathbf{K}_{vir}\mathbf{t}_{vir} \tag{8}$$

$$U_r = round\left(\frac{V_u}{V_w}\right), V_r = round\left(\frac{V_v}{V_w}\right) \tag{9}$$

where $\mathbf{K}_{vir}$, $\mathbf{R}_{vir}$, $\mathbf{t}_{vir}$ denote the camera intrinsic matrix of size 3×3, the rotation matrix of size 3×3, and the translation matrix of size 3×1 in the virtual view. Owing to the round-off errors of the coordinates by (9), this operation will result in hole generation. This is because several points in the 2D reference view are projected into the same location and they lose the reliable texture data (color information) for filling. These points that are projected into the same location using the traditional 3D warping tend to have overlapping values, or erroneous values, or vain values, thus, degrading the quality of video sequences. Hence, we propose a novel method to improve the traditional 3D warping process further. The improved strategy is described in the following.

First, given a pixel coordinates (s,t) in the 2D reference view plane (I) followed by calculations using (7)–(9), we decide the reliable mapping coordinates according to the following condition, to change the coordinates in the virtual view. The decision condition is expressed as

$$(T_u, T_v) = \underset{(s,t)\in I}{\text{argmin}}\, d(s,t), \text{ if } s = U_r,\ t = V_r. \tag{10}$$

When several points that belong to the I are projected into the same location in the virtual view, the point with the smallest depth value will be selected as the virtual view coordinates. According to the depth map characteristics, the depth value of a pixel in the depth map, represented by an 8-bit grayscale in the range [0, 255], defines a distance along the Z axis. The "0" represents the most distant depth value in the depth map, while the "255" represents the closest depth value. Hence, the larger the depth value, the nearer the camera is. In other words, the FG has the larger depth value. In contrast to the FG, the BG represents the smaller depth value. Hence, we selected the smallest depth value in (10), because holes are typically generated from the FG region. Hence, the advantages of selecting the smallest depth value are as follows: (1) the number of holes after projection can be reduced, and (2) the reprojection processes and the BG region in the 2D reference view can finish efficiently prior to the projection process, to obtain the corresponding coordinates in the virtual view. Next, after computing (10), if $(U_r, V_r) \neq (T_u, T_v)$, the traditional 3D warping method will be adopted to achieve the projection. If $(U_r, V_r) = (T_u, T_v)$, we will further modify the virtual coordinates (T_v, T_v) according to the relative depth value, color information of (T_v, T_v), and (U_r, V_r). The modified method is expressed as

$$\left(T_{u_{update}}, T_{v_{update}}\right) = \begin{cases} (T_u, T_v), & \text{if } d(T_u, T_v) > d(U_r, V_r), \\ \text{argmin}(\|c_1(T_u, T_v)\|, \|c_2(U_r, V_r)\|), & \text{if } d(T_u, T_v) = d(U_r, V_r), \\ NULL & \text{otherwise,} \end{cases} \tag{11}$$

where $c_1(T_u, T_v) = (r_{c1}, g_{c1}, b_{c1})$, $c_2(U_r, V_r) = (r_{c2}, g_{c2}, b_{c2})$, (r, g, b) denote the red, green, and blue components of the color space, respectively; $NULL$ denotes the hole; $\|\bullet\|$ denotes the norm. Finally, based on (11), we can obtain the updated projected coordinates $\left(T_{u_{update}}, T_{v_{update}}\right)$ in the virtual view. The

updated coordinates $\left(T_{u_{update}}, T_{v_{update}}\right)$ will be backprojected into the 3D world coordinates to obtain the update coordinates $\left(W_{x'}, W_{y'}, W_{z'}\right)$ using (12).

$$\begin{pmatrix} W_{x'} \\ W_{y'} \\ W_{z'} \end{pmatrix} = \mathbf{R}_{ref}^{-1}\mathbf{K}_{ref}^{-1}\begin{pmatrix} T_{u_{update}} \\ T_{v_{update}} \\ 1 \end{pmatrix} d\left(T_{u_{\text{update}}}, T_{v_{update}}\right) - \mathbf{R}_{ref}^{-1}\mathbf{t}_{ref}. \tag{12}$$

The updated 3D world point $\left(W_{x'}, W_{y'}, W_{z'}\right)$ is substituted into (8), and (8) is rewritten as

$$\begin{pmatrix} T_{u'_{\text{update}}} \\ T_{v'_{\text{update}}} \\ 1 \end{pmatrix} = \mathbf{K}_{vir}\mathbf{R}_{vir}\begin{pmatrix} W_{x'} \\ W_{y'} \\ W_{z'} \end{pmatrix} + \mathbf{K}_{vir}\mathbf{t}_{vir}, \tag{13}$$

where $\left(T_{u'_{update}}, T_{v'_{update}}\right)$ is the final virtual view coordinates corresponding to the pixel coordinates (s, t) in the 2D reference view. We repeat the procedures above until all pixels in the 2D reference view plane have undergone the projection operation into the 2D virtual view plane. Figure 7 illustrates the profile of our proposed improved 3D warping process in DIBR.

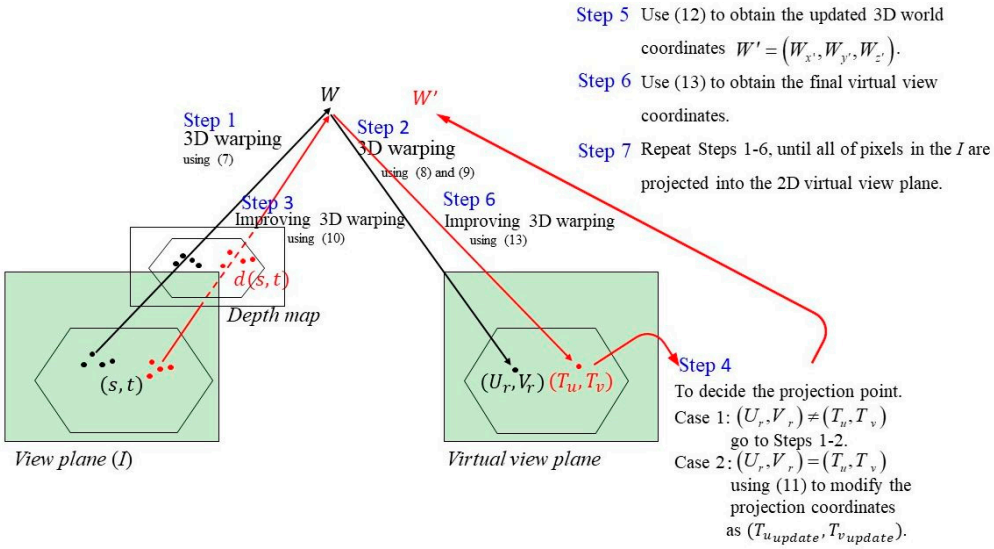


Figure 7. Profile of the improving 3D warping process in DIBR.

A summary of how the 3D warping procedures are improved is as follows.

Input: A series of reference views, the corresponding depth maps, and the camera's parameters provided by [33].

Step (1) Given a pixel location (s, t) in the 2D reference view plane (I), perform a backprojection from the 2D reference view plane to the 3D world coordinates $\left(W_x, W_y, W_z\right)$ using (7).

Step (2) Find the 2D virtual view coordinates (U_r, V_r) using (8) and (9).

Step (3) Verify whether the points in I are mapped into the same location in the virtual view plane. If yes, the projected virtual point will be changed using (10).

Step (4) Decide the changed virtual point in the virtual view plane according to the result of Step 3; If $(U_r, V_r) \neq (T_u, T_v)$, proceed to Steps 1–2. If $(U_r, V_r) = (T_u, T_v)$, the changed virtual point will be further updated using (11), and proceed to Step 5.

Step (5) Compute the updated 3D world coordinates using (12).

Step (6) Use the updated 3D world point to obtain the final virtual view coordinates based on (13).

Step (7) Repeat Steps 1–6, until all pixels in I have been processed.

The two side views (left and right) shown in Figure 4 are used individually for improving the 3D warping process, to obtain the corresponding virtual views.

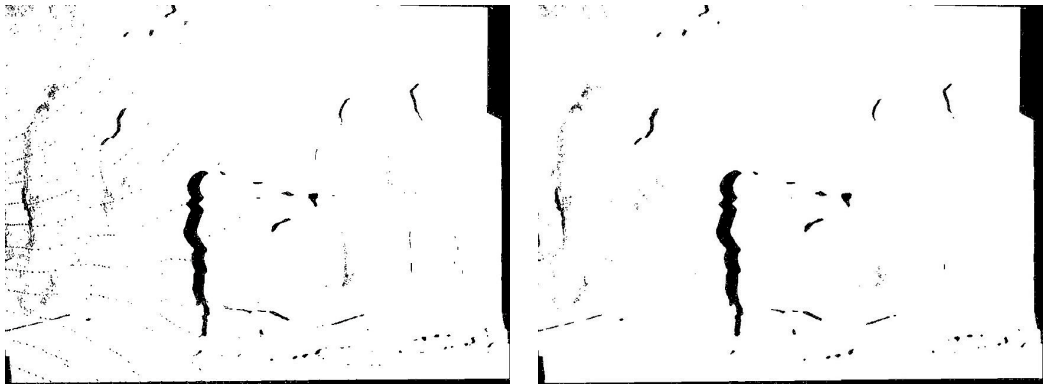
3.3. Filtering for Holes

Owing to the drawback of the depth map, the depth map has been preprocessed, as described in the previous section. After improving the 3D warping process using the preprocessed depth map, holes still appear in the virtual view. If a median filter is used directly in a depth map with holes using the improved 3D warping process, the resulting image will be extremely smooth and vague. To avoid this, we used a median filter of window size 3×3 in the holes, as expressed in (14).

$$Holes_{i,j} = median\left(Y_{i,j}\right), Y_{i,j} \in \text{Hole region}, \tag{14}$$

where $Y_{i,j}$ denotes the depth value at the coordinates (i, j) as a center point around a 3×3 window size.

This process not only modifies the smooth and vague images, but also decreases the computational time of the subsequent procedures. If this procedure is removed, it will cause a poor image quality and increase the execution time of the system. Figure 8 illustrates the binary results of filtering the entire depth map and filtering the holes.



(**a**) (**b**)

Figure 8. Filtering results. (**a**) Filtering the entire depth map, (**b**) filtering the holes.

3.4. 3D Inverse Warping Process

The DIBR technique is transferred from the neighboring view to the virtual view, which is known as forward mapping. However, in forward mapping, cracks might be transmitted into the virtual view, owing to the round-off errors and inaccurate depth values. To circumvent the usual problems associated with the forward mapping rendering techniques (cracks), an inverse warping rendering technique can be used to correct the cracks. This technique is simple, and accurately resamples synthetic pixels [32]. Hence, we adopted the inverse warping rendering technique to remove the usual problems associated with rendering, using forward mapping and further improved the view quality.

First, for each pixel in the virtual view plane and the corresponding depth map, (R_v, K_v) corresponds to the rotation parameter and the intrinsic parameter of the virtual camera; a 3D world point $\left(W_{2x}, W_{2y}, W_{2z}\right)$ is calculated using backprojection, expressed as

$$\begin{pmatrix} W_{2x} \\ W_{2y} \\ W_{2z} \end{pmatrix} = \mathbf{C}_v + \lambda \mathbf{R}_v^{-1} \mathbf{K}_v^{-1} \mathbf{d}_v, \tag{15}$$

where λ is defined as

$$\lambda = \frac{W_{2z} - C_{vz}}{r_3} \text{ with } \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \mathbf{R}_v^{-1}\mathbf{K}_v^{-1}\mathbf{d}_v \text{ and } \mathbf{C}_v = \begin{pmatrix} C_{vx} \\ C_{vy} \\ C_{vz} \end{pmatrix}. \tag{16}$$

Here, the depth value W_{2z} is defined by the pixel value at coordinate $\mathbf{d}_v$ in the depth map of the virtual view; in our system, we used the k value obtained using (2) instead of λ. Next, the calculated 3D world point $\left(W_{2x}, W_{2y}, W_{2z}\right)$ is projected into the reference texture view plane using (17).

$$\lambda p_1 = \mathbf{K}_{ref}\mathbf{R}_{ref}\begin{pmatrix} W_{2x} \\ W_{2y} \\ W_{2z} \end{pmatrix} + \mathbf{K}_{ref}\mathbf{t}_{ref}, \tag{17}$$

where $p_1 = (s_{r'}, t_{r'}, 1)^T$ refers to the coordinates of a pixel in the reference view plane. $\mathbf{K}_{ref}$ refers to the camera intrinsic parameter of the reference view; $\mathbf{R}_{ref}$ and $\mathbf{t}_{ref}$ represent the camera extrinsic parameters of the reference view. The color of a pixel in the virtual view plane can be interpolated from the pixels surrounding p_1 in the reference view plane. Based on (17), the possible holes in the virtual view plane can be found in the corresponding pixel values in the reference view plane, and can be padded based on inverse warping blending. Figure 9 illustrates the results of 3D inverse warping rendering.

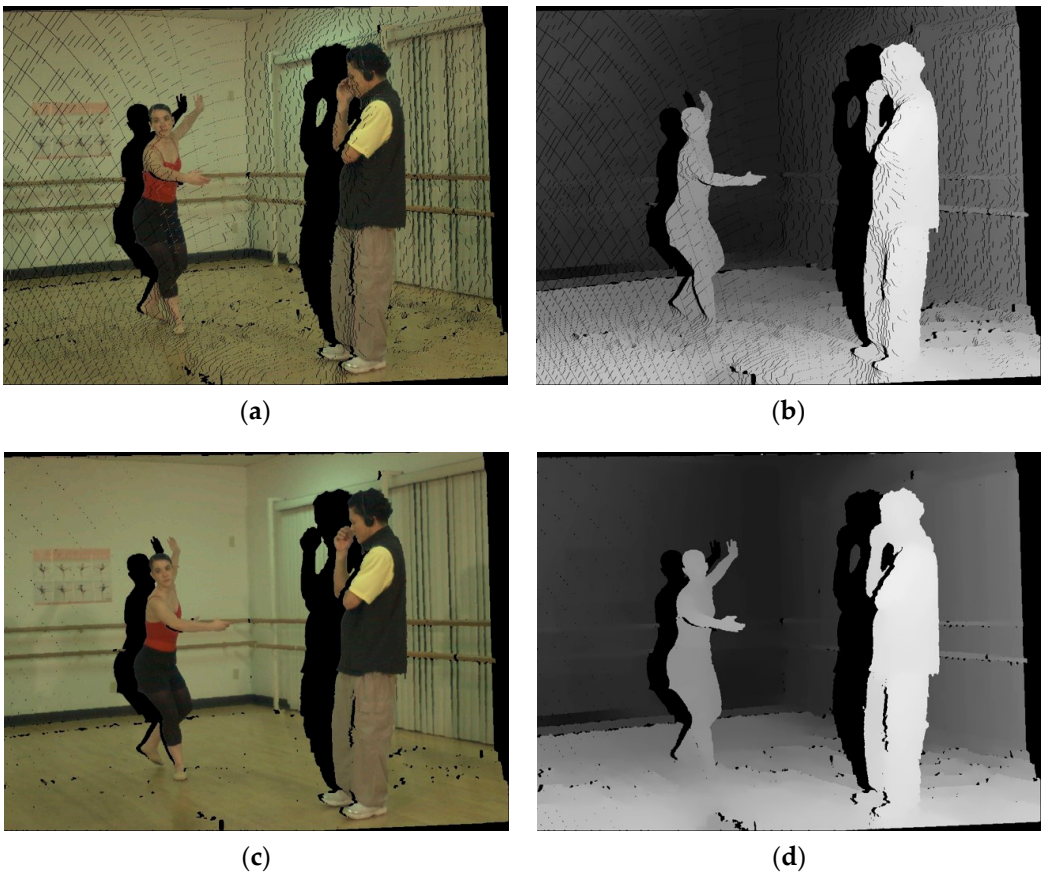


Figure 9. Results of 3D inverse warping rendering (left view). (**a**) and (**b**) the virtual view and the corresponding depth map. (**c**) and (**d**) Results of 3D inverse warping rendering corresponding to (**a**) and (**b**), respectively.

3.5. Brightness Adjustment

After performing the procedures above, we obtained two virtual views corresponding to two side views (left side and right side). The disocclusions are not visible in the main reference view (left-side view), but they are visible in the virtual view. Hence, we can recover these disocclusions by the reference view on the other side of the virtual view, which is called the auxiliary reference view. That is, the auxiliary reference view is only used to fill the disocclusions. Blending refers to combining two virtual views warped from two side views into a synthesized virtual view. Additionally, the brightness of two reference views is typically different; therefore, two virtual views will display inconsistent brightness. Hence, if we directly synthesize these two virtual views without brightness adjustment, the synthesized view will display inconsistent brightness and discontinuous colors. Hence, we used a strategy based on the method in [8] for brightness adjustment, to improve these problems.

First, two side reference views (left-side view, right-side view) are warped to the corresponding virtual views based on DIBR. The left-side reference view is called the main view; the other side reference view is called the auxiliary view. V_{main} and V_{aux} represent the main virtual view warped from the main view and the auxiliary virtual view warped from the auxiliary view, respectively. To obtain virtual views with the same number of holes warped from the main and auxiliary views, we created two virtual views: V_{main} involved with V_{aux} is referred as V'_{main}; V_{aux} involved with V_{main} is referred as V'_{aux}, as shown in Figure 10. Next, the brightness of the auxiliary virtual view is adjusted based on the brightness of the main view. The V_{aux}, V'_{main}, and V'_{aux} are converted into the hue, saturation, value (HSV) color space. The component V represents the brightness. We then calculate the brightness ratio for each pixel of the nonhole regions in V'_{main} and V'_{aux}. The ratio of the brightness component is computed as follows:

$$B_r(i,j) = \frac{V'_{main}(i,j)}{V'_{aux}(i,j)},\ (i,j) \in \text{non-hole regions}, \tag{18}$$

$$B_{rMean} = \frac{1}{n}\sum B_r(i,j), \tag{19}$$

where n denotes the number of nonhole pixels; (i,j) represents the coordinates of a pixel belonging to nonhole regions; B_{rMean} is the mean ratio used for adjusting the brightness of the auxiliary virtual view in the HSV color space. Finally, the adjusted auxiliary virtual view is converted into the red, green, blue (RGB) color space. Thus, the brightness of the virtual view warped from the auxiliary view has been adjusted to match that of the main view, as shown in Figure 11. Figure 11 illustrates the results of the brightness adjustment. From Figure 11, after adjusting the brightness, the head surrounding is improved and the ghost is reduced. Hence, in order to reduce the ghost generation and maintain the vision quality for viewing, brightness adjustment is a necessary step.

(**a**) (**b**)

Figure 10. Virtual views with the same holes warped from the main and auxiliary views. (**a**) V'_{main} and (**b**) V'_{aux}.

(**a**) (**b**)

Figure 11. Example of brightness adjustment. (**a**) Nonadjusted and (**b**) adjusted.

3.6. Adaptive Blending of Virtual Views

After performing the brightness adjustment, two virtual views warped from the left-side and the right-side views individually can represent the consistent brightness based on the brightness of the main reference view. Subsequently, to obtain a synthesized virtual view, two virtual views are blended into the synthesized virtual view. In most studies, the simplest view blending using a weighted sum of two images is performed to synthesize 3D warped views to a virtual view, expressed as

$$I_v(u,v) = (1-\alpha)I_L(u,v) + \alpha I_R(u,v), \tag{20}$$

where I_L and I_R are the left-side and right-side warped reference texture views corresponding to the coordinates (u,v), respectively; I_v is the blended view; α is the position-dependent interpolation parameter. However, this common blending can yield inconsistent pixel values, because of inaccurate camera parameters, inconsistent depth values, or the warping of round-off errors. These inconsistent pixels from both views can result in warped images, producing artifacts or jagged edges. To avoid this problem, we adopted adaptive blending to perform view blending based on the method in [22]. Herein, the main virtual view is defined as a virtual view warped from the left-side view, and an auxiliary virtual view is the warped view from the right-side view. The adaptive blending is expressed as

$$I_v(u,v) = \alpha I_A(u,v) + (1-\alpha)I_B(u,v), \tag{21}$$

where I_A and I_B denote the main virtual view and the auxiliary virtual view after brightness adjustment, respectively. α is 1 for a nonhole region and 0 for the hole pixels in I_A depending on the selected viewpoint. Based on (21), most regions of the blended view are from the main virtual view, and some holes in the blended view are filled from the auxiliary virtual view.

3.7. Hole Filling

Most of the holes are filled after adaptive blending is performed. However, a few holes still exist in the blended virtual view. The remaining holes in the synthesized view are the disocclusion regions due to the reprojected process; they are generally located in the BG areas and occur at the FG boundaries, or appear because of inaccurate depth. To fill the remaining holes, an exemplar-based inpainting method [17] is typically adopted. However, this exemplar-based method is highly time-consuming, because it is a global-based search strategy that searches for the best patch to repair. In the 3DTV FVV scheme, computational time is a serious problem. Hence, we used a sample method combined with the FMM algorithm [18] to fill the holes quickly, by modifying the boundary condition in the inpainting. Briefly, The FMM procedures are as follows: the region Ω to be inpainted and its boundary $\delta\Omega$ are defined; the pixel p belonging to Ω would be inpainted by its neighboring region $B_\varepsilon(p)$ of size ε, as

shown in Figure 12a. Herein, we set ε to 3. Assume that a gray value image is considered, and color images are the following extension. First, to compute an approximation $I_q(p)$ of the image in point p, given the image $I(q)$ and its gradient value $\nabla I(q)$ of q point (Figure 12b).

$$I_q(p) = I(q) + \nabla I(q)(p - q) \tag{22}$$

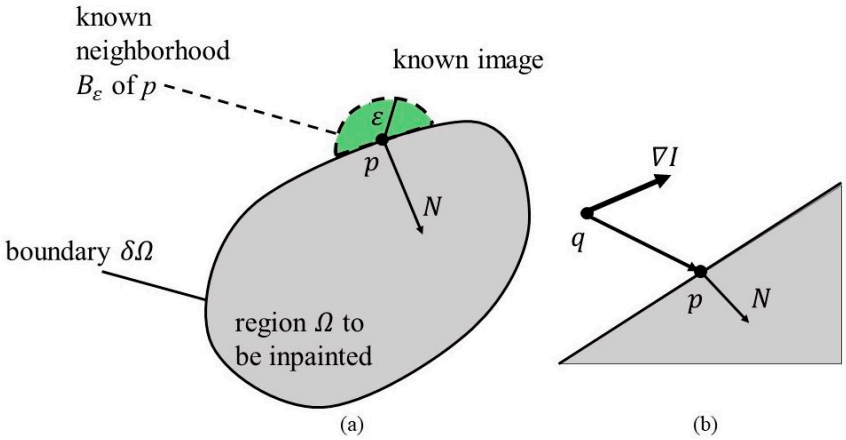


Figure 12. Profile of fast matching method (FMM) inpainting method [18]. (**a**) An example of region Ω and its boundary $\delta\Omega$ for inpainting and (**b**) gradient value $\nabla I(q)$ of q point corresponding to point p.

Next, inpaint point p as a function of all points q in $B_\varepsilon(p)$, by summing the estimates of all points q, weighted by a weighting function $w(p,q)$.

$$I(p) = \frac{\sum_{q \in B_\varepsilon(p)} w(p,q)[I(q) + \nabla I(q)(p - q)]}{\sum_{q \in B_{\varepsilon(p)}} w(p,q)}. \tag{23}$$

The weighting function is defined as a product of three factors: $w(p,q) = dir(p,q) \cdot dst(p,q) \cdot lev(p,q)$.

$$\begin{aligned} dir(p,q) &= \frac{p - q}{\|p - q\|} \cdot N(p), \\ dst(p,q) &= \frac{d_0^2}{\|p - q\|^2}, \\ lev(p,q) &= \frac{T_0}{1 + \left|T(p) - T(q)\right|}. \end{aligned} \tag{24}$$

The directional component $dir(p,q)$ ensures that the contribution of the pixels is close to the normal direction $N = \nabla$ (gradient). The geometric distance component $dst(p,q)$ decreases the contribution of the pixels geometrically farther from p. The level set distance component $lev(p,q)$ ensures that pixels close to the contour through p contribute more than the farther pixels; d_0 and T_0 are the interpixel distances, to be set as 1. $T(p)$ is the distance between the point p and the initial inpainting boundary $\delta\Omega$.

Additionally, the inpainting concept should be changed to be applied to hole filling in view synthesis, because the boundary of the holes may belong to both the FG and BG. Hence, we modified the boundary information of the hole in the inpainting based on the method presented in [15]. We changed the boundary region with the FG on the hole to the corresponding BG region located on the opposite side. To distinguish between the FG and BG, we adopted the relative magnitude of the depth values between two points that were horizontally opposite to each other on the hole boundary. The boundary point with the larger depth value that was regarded as belonging to the FG will be replaced with the boundary point with the smaller depth value that was regarded as belonging to the BG. The

interval of the boundary region was the same as ε and set as 3 pixels. Figure 13 shows an example of the boundary region replacement for the hole. The replacement mechanism is represented by

$$p_{fg} \in \delta\Omega_{fg} \rightarrow p_{bg} \in \delta\Omega_{bg}, \tag{25}$$

$$B_\varepsilon(fg) \rightarrow B_\varepsilon(bg),$$

where fg and bg denote the FG and the BG, respectively. Figure 14 shows the inpainting result. Thus, the improved boundary strategy on hole filling will efficiently reduce the disturbance of the FG under inpainting and accelerate the inpainting computational time.

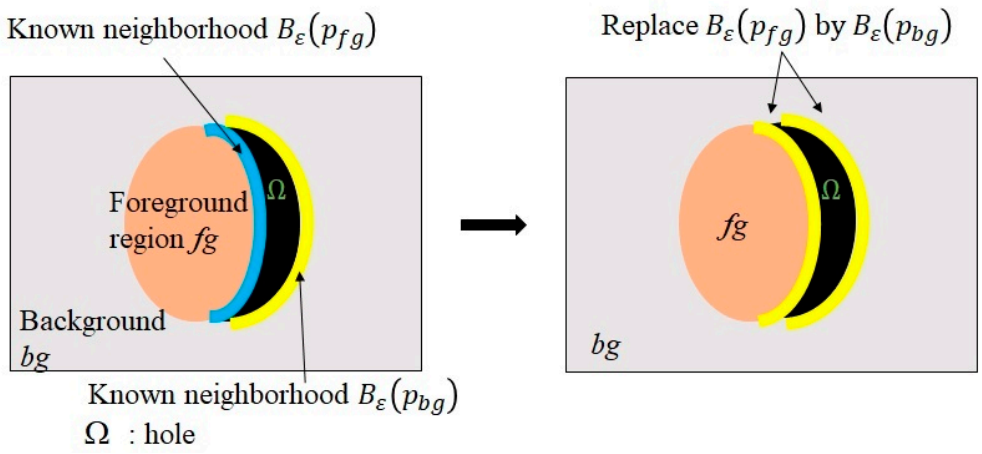


Figure 13. Example of boundary region replacement for hole.

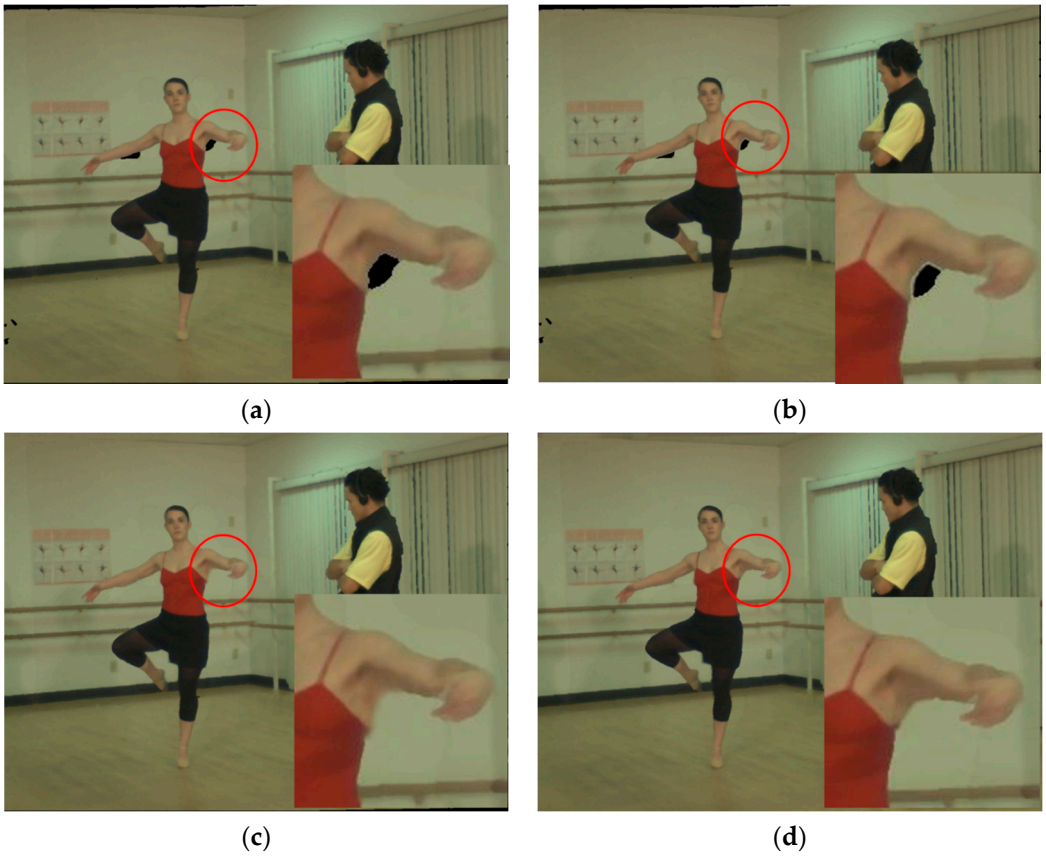


Figure 14. Inpainting results. (**a**) View with holes, (**b**) boundary replacement, (**c**) inpainting result of (**a**), and (**d**) proposed inpainting result corresponding to (**b**).

4. Experimental Results and Discussion

To verify the performance of the proposed hole filling method for 3D virtual view synthesis, the experimental results are compared with those of Chang et al. [21], Cho et al. [19], Luo and Zhu [26], and Oliveira et al. [27]

4.1. Experimental Setup and Datasets

The experimental data were "Ballet" and "Breakdancers" video sequences of size 1024 × 768 derived from Microsoft research [33], and each video sequence included 100 images. Two neighboring side views (left-side view and right side view) were used to render a virtual view. In this study, the virtual view was at the Camera 4 location; thus, Cameras 3 and 5 that were placed at the left and right were used as the two side reference views. The experiments were implemented in Microsoft Visual Studio C++2017, on an Intel ® core i7-4790@ 3.6 GHz computer with 16 GB of RAM.

4.2. Performance Evaluation

For performance evaluation, we used the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [34], to demonstrate our proposed method. The PSNR and mean square error (MSE) are expressed as

$$PSNR = 10 \log \frac{S_{\max}^2}{MSE}, \tag{26}$$

$$MSE = \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \left| f(i,j) - f'(i,j) \right|^2, \tag{27}$$

where $f(i,j)$ and $f'(i,j)$ denote the original image (ground truth) for Camera 4 and the synthesized virtual view corresponding to the coordinates (i,j), respectively. w and h denote the width and height of the image, respectively. $S_{\max}$ is 255 for a gray-level image. The higher the PSNR value, the closer the synthesized view and ground truth are to each other. Similarly, the SSIM value is closer to 1, and both the synthesized view and ground truth are more similar to each other.

4.3. Results

The experimental results are presented and discussed in the following sections.

4.3.1. Comparison of Traditional 3D Warping and Improved 3D Warping

According to our proposed method, we improved the traditional 3D warping method to decrease the number of error matching points. That is, the hole numbers can be reduced efficiently. We computed the number of holes using traditional 3D warping and improved 3D warping to yield the performance shown in Table 1.

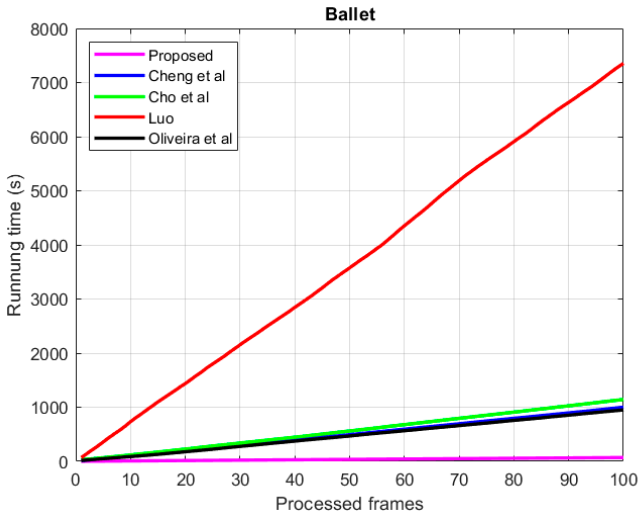
Table 1. Average number of holes (pixel points) for one image compared with traditional 3D warping and improved 3D warping.

Video	Traditional 3D Warping	Improved 3D Warping	Reduced Rate per Image (%)
Ballet	126,605	111,501	1.9
Breakdancers	74,105	56,762	2.2

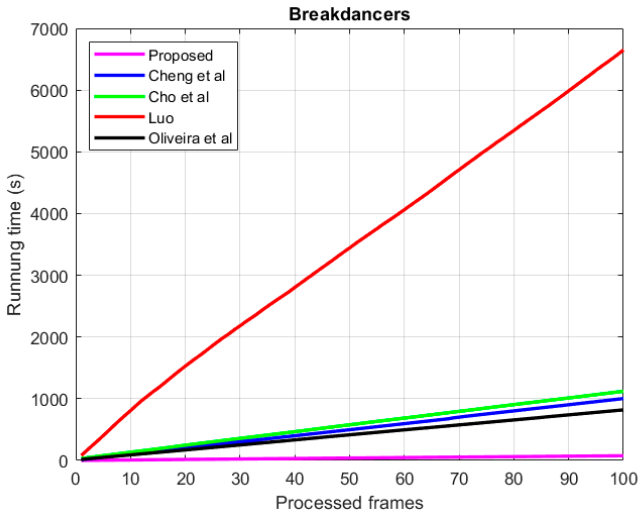
From Table 1, it is clear that the improved 3D warping method decreased the number of hole pixels by more than 1.9% in both Ballet and Breakdancers. That is, the fewer the number of holes, the shorter the execution time.

4.3.2. Comparison of Execution Time

In this section, the execution times are illustrated for both datasets. The execution times in seconds are illustrated in Table 2. Figure 15 shows the cumulative execution times corresponding to the number of frames compared with our proposed method and the methods of Chang et al. [21], Cho et al. [19], Luo and Zhu [26], and Oliveira et al. [27] As shown in Table 2, the execution speed of our proposed approach is faster than that of the other methods. The average execution time of our proposed method for Ballet is only 0.69 s per frame. In contrast to our proposed method, the other methods require more time on both datasets, especially that of Luo [20]. The hole repair strategy in the synthesized virtual view for the compared methods is to improve the exemplar-based inpainting method [17]. The inpainting method by [17] is highly time-consuming, because a global-based method is used to search for the matching patch. Hence, the methods in [19,21,27] require at least 9.49 s. In Luo's method, FG objects are first removed and then holes are repaired based on [17]. The average execution time for Luo's method exceeds 66.5 s, which is significantly higher than our proposed method.



(**a**)



(**b**)

Figure 15. Cumulative execution times corresponding to the number of frames compared with our proposed method, and the methods of Chang et al. [21], Cho et al. [19], Luo and Zhu [26], and Oliveira et al. [27] (**a**) Ballet video and (**b**) Breakdancers video.

Table 2. Execution time in seconds.

Method	Ballet		Breakdancers	
	Average	Total	Average	Total
Our proposed	0.69	69.79	0.76	76.01
Chang et al. [21]	10	1000.33	10.02	1002.96
Cho et al. [19]	11.42	1142.81	11.20	1120.82
Luo [26]	73.54	7354.75	66.50	6650.17
Oliveira et al. [27]	9.49	95,049.79	8.19	819.27

4.3.3. Evaluation of Synthesized Virtual View

In our experiments, we used the average PSNR value (Ave_PSNR) and SSIM value (Ave_SSIM) to evaluate the synthesized virtual view obtained by our proposed method. Table 3 shows the average PSNR and SSIM values. Figures 16 and 17 illustrate the detailed results for the PSNR value and SSIM value per frame, respectively. From Table 3, it is clear that the average PSNR value for both the Ballet and Breakdancers datasets is superior to those of other methods. For the Ballet dataset, the SSIM of our proposed method was slightly lower than that of Cho et al.'s [19]. This was because the hole filling based on Criminisi et al.'s method [17] increased the BG term into the priority computation to prevent FG propagation; thus, the BG regions were inpainted prior to the FG regions or boundary regions. The average SSIM value increased only by 0.013. However, in the Breakdancers dataset, the average SSIM value was less than that of our proposed method. In addition, the average SSIM value obtained from Chang et al.'s [21] and our proposed method on the Ballet dataset was almost the same. However, the execution time of our proposed method, as shown in Table 2, was 14 times faster than that of Chang et al.'s [21] and Cho et al.'s [19] method. For the Breakdancers dataset, the average SSIM value obtained by our proposed method was higher than those using other methods.

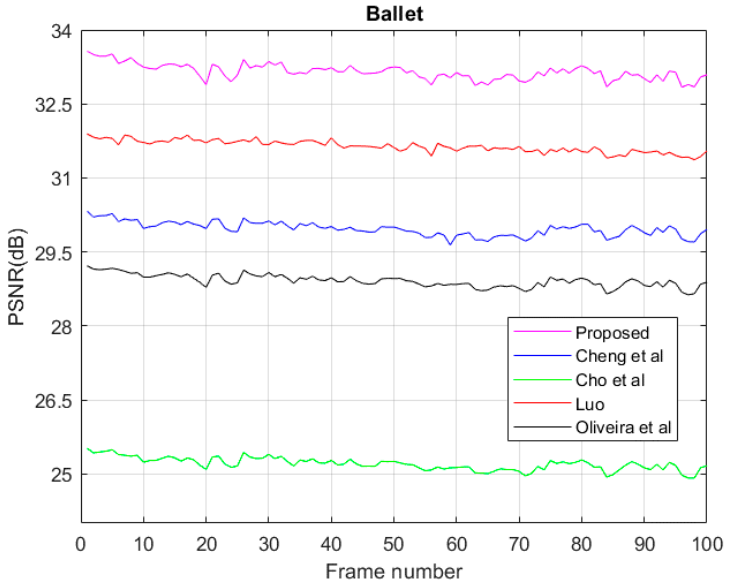
Table 3. Comparative results.

Method	Ballet		Breakdancers	
	Ave_PSNR (dB)	Ave_SSIM	Ave_PSNR (dB)	Ave_SSIM
Proposed	33.10	0.921	34.49	0.913
Chang et al. [21]	29.97	0.920	31.73	0.896
Cho et al. [19]	25.20	0.934	27.17	0.885
Luo [26]	31.64	0.831	30.8	0.817
Oliveira et al. [27]	28.91	0.846	29.24	0.843

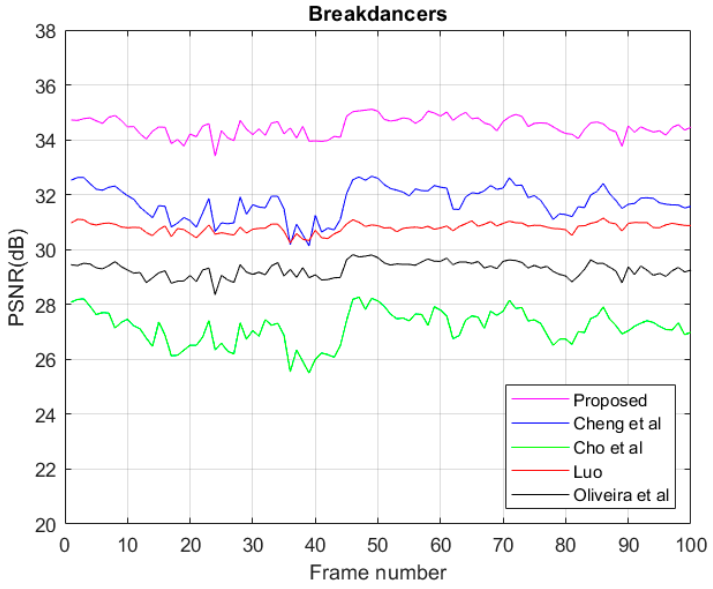
In addition, comparing the datasets Ballet and Breakdancers for our proposed method, as shown in Table 3, the average PSNR value on Breakdancers was greater than the average PSNR value on Ballet. This was because the camera parameters and depth map quality of the different datasets provided by Microsoft research were distinctly different. These factors affected the performance evaluation. From Table 3, we verified that the camera parameters and depth map quality on Breakdancers were better than those on Ballet.

Figures 18 and 19 show the visual quality of the synthesized virtual view compared with those of Chang et al.'s [21], Cho et al.'s [19], Luo and Zhu's [26], and Oliveira et al.'s [27]. As shown in Figures 18 and 19, the hole filling for the synthesized virtual view using our proposed method presented better quality. Figure 20 shows an enlarged synthesized virtual view. As shown in Figure 20, the quality of the synthesized virtual view was much better than those obtained using other methods, especially in the boundary region and motion region. For the head surrounding areas of Figure 20, our proposed method could remove ghost shadow generations more efficiently than the other methods using traditional 3D warping. This is because the traditional 3D warping method does not involve depth map preprocessing and lightness adjustment, such as the methods in [19,21,26,27]. For the axilla region on the Ballet dataset, the hole filling by our proposed method was close to the ground truth

when compared with [27]. The holes filling results using the methods in [19,21,26] may cause blurred or unfinished images. For the foot region on the Breakdancers dataset, our proposed method can match the foot region more exactly. The methods in [19,21,26] presented motion blurring that did not fit the focus.



(**a**)



(**b**)

Figure 16. Comparative results of peak signal-to-noise ratio (PSNR) value by frame. (**a**) Ballet dataset and (**b**) Breakdancers dataset.

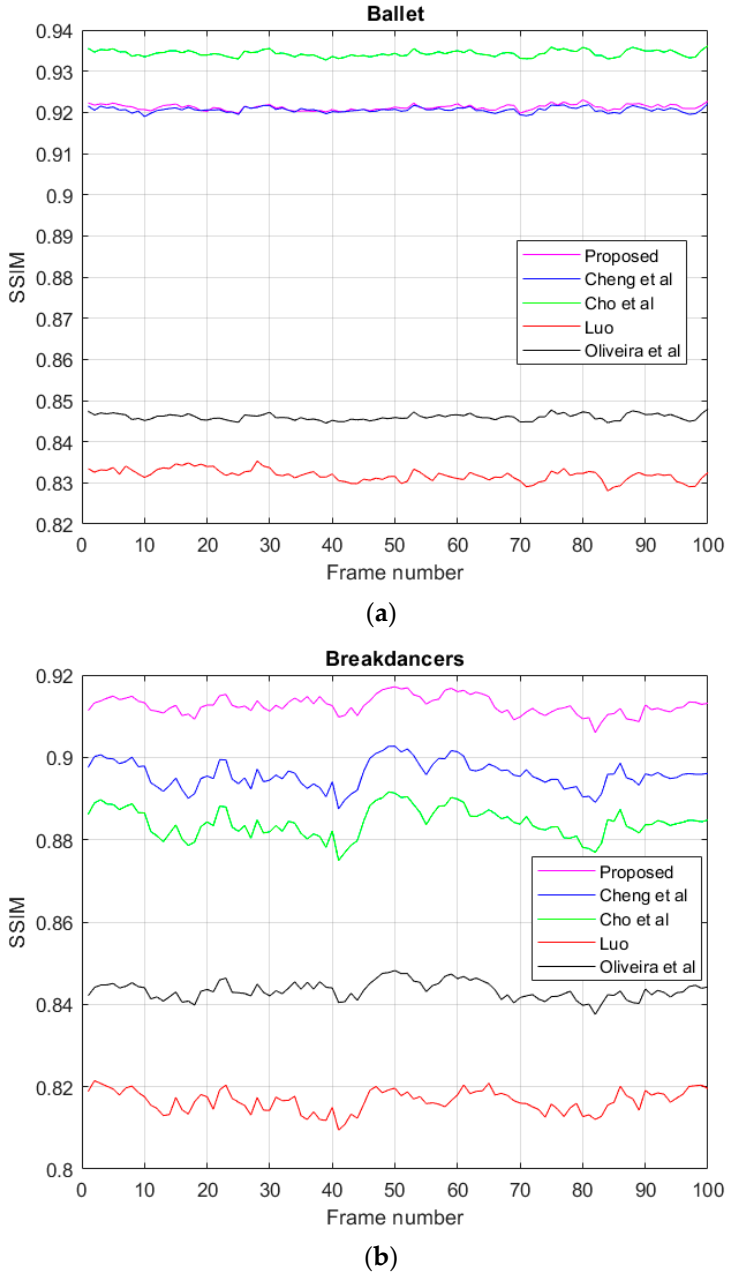


Figure 17. Comparative results of structural similarity (SSIM) value per frame. (**a**) Ballet dataset and (**b**) Breakdancers dataset.

4.3.4. Discussions

In summary, our proposed method achieved better results, regardless of the objective evaluation or visual quality representation, as shown in Tables 1–3 and Figures 18–20. Additionally, the higher the PSNR and SSIM values, the better was the synthesized virtual view. Table 3 illustrates the results. The average PSNR and SSIM values of our method were better than those of other methods, except the average SSIM on the Ballet dataset for the method of Cho et al. In addition, the improved 3D warping method in the DIBR algorithm proposed by us can decrease the number of holes by 1.9% per frame, thus improving the execution time better than the methods of Cheng et al., Cho et al., Luo and Zhu, and Oliveira et al., as shown in Tables 1 and 2. Regarding the visual quality of the synthesized virtual view, the representation of the synthesized virtual view was good; it allows the human eyes to view

the synthesized virtual view comfortably, as shown in Figures 18–20. Evidently, our proposed method could yield a high-quality synthesized virtual view and accelerate the execution time significantly.

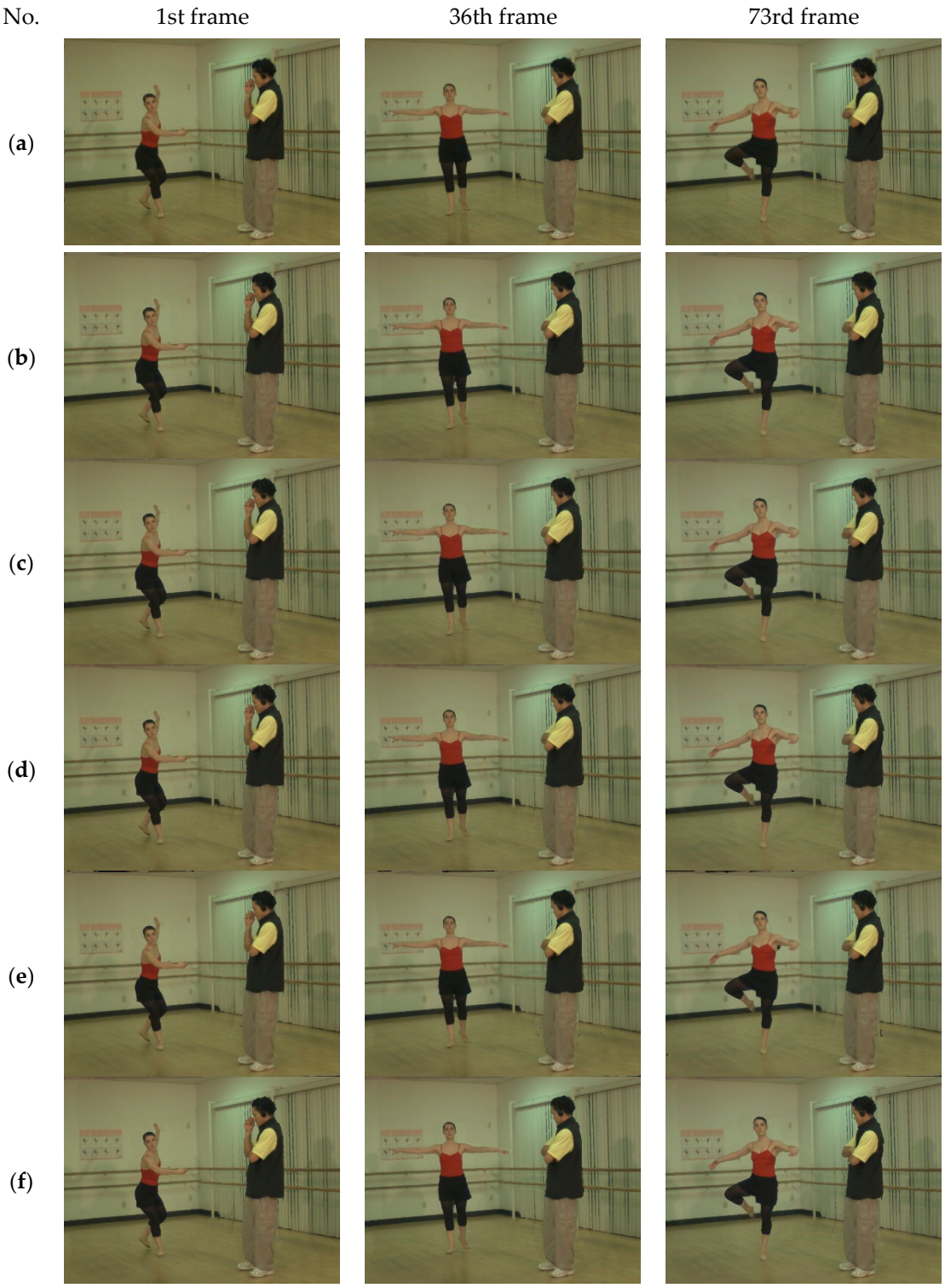
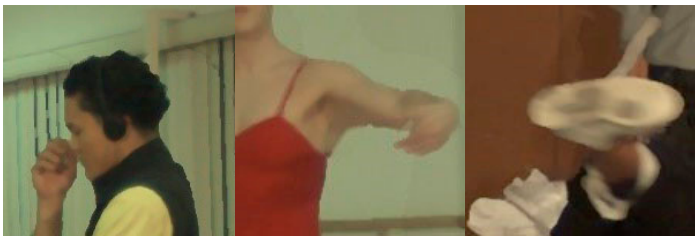


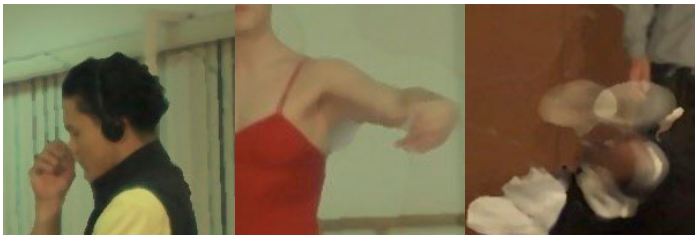
Figure 18. Synthesized virtual view for 1st, 36th, and 73rd frames on Ballet dataset. (**a**) Ground truth of Camera 4 of (**b**) Our proposed method. (**c**) Chang et al.'s method [21]. (**d**) Cho et al.'s method [19]. (**e**) Luo and Zhu's method [26]. (**f**) Oliveira et al.'s method [27].

Figure 19. Synthesized virtual view for 1st, 36th, and 73rd frames on Breakdancers dataset. (**a**) Ground truth of Camera 4 of (**b**) Our proposed method. (**c**) Chang et al.'s method [21]. (**d**) Cho et al.'s method [19]. (**e**) Luo and Zhu's method [26]. (**f**) Oliveira et al.'s method [27].

(**a**)

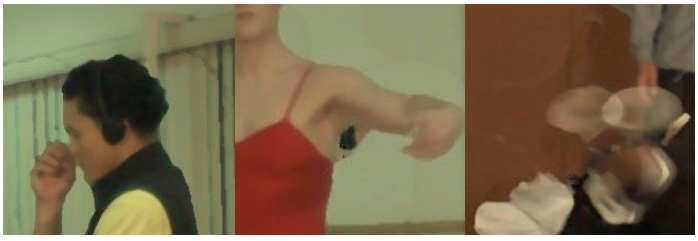


(**b**)

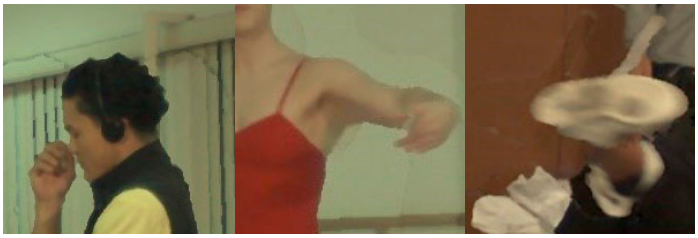


(**c**)

(**d**)



(**e**)



(**f**)

Figure 20. Enlarged images corresponding to the 1st and 73rd frames of Ballet and the 36th frame of Breakdancers. (**a**)Ground truth of Camera 4 of (**b**) our proposed method, (**c**) Cheng et al.'s method [21], (**d**) Cho et al.'s method [19], (**e**) Luo and Zhu's method [26], and (**f**) Oliveira et al.'s method [27].

5. Conclusions

In this study, we proposed a hole filling method in a 3D virtual view synthetic system based on DIBR. In DIBR, the most important criteria are the quality of the rendered image and the execution speed. This approach improved the core technology of 3D warping, used median filtering on the hole areas after warping, and adopted the depth information and BG information to complete the optimization of hole filling. Combined with novel methods, including depth map preprocessing, inverse mapping, brightness correction, and adaptive hybrid blending, our experimental results demonstrated that our proposed approach could achieve better visibility and faster execution time than other methods.

Although our proposed approach could improve the rendering quality more quickly and effectively, some problems still exist. For example, if the resolution of the images is over 2 K or 4 K, the execution speed becomes a serious issue. In addition to the camera resolution, the internal and external parameters of the camera must be effectively controlled and corrected to achieve the adaptability of the DIBR method. The adjustment of camera parameters and image resolution will be performed in future work.

Author Contributions: For corresponding author. H.-Y.H. is major to design the framework of the system and theorem base and writing. Author 2: S.-Y.H. is to code and implement the experimental results. All authors have read and agree to the published version of the manuscript.

Funding: This research was partly funded by the Ministry of Science of Technology, Taiwan, under grand MOST 108-2635-E-150-001.

Conflicts of Interest: The authors declare that they have no competing interests in this work.

References

1. Fehn, C. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In Proceedings of the SPIE 5291, Stereoscopic Displays and Virtual Reality Systems XI, San Jose, CA, USA, 21 May 2004; pp. 93–104.
2. Li, W.; Zhou, J.; Li, B.; Sezau, M.I. Virtual view specification and synthesis for free viewpoint television. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 533–546.
3. Smolic, A. 3D video and free viewpoint video-from capture to display. *Pattern Recognit.* **2011**, *44*, 1958–1968. [CrossRef]
4. Ahn, I.; Kim, C. A novel depth-based virtual view synthesis method for free viewpoint video. *IEEE Trans. Broadcast.* **2013**, *59*, 614–626.
5. Mori, Y.; Fukusgima, N.; Fujii, T.; Tanimoto, M. View generation with 3D warping using depth information for FTV. *Signal Process. Image Commun.* **2009**, *24*, 65–72.
6. Zinger, S.; Do, L.; DeWith, P.H.N. Free-viewpoint depth image based rendering. *J. Vis. Commun. Image Represent.* **2010**, *21*, 533–541. [CrossRef]
7. Tanimoto, M. FTV: Free-viewpoint television. *APSIPA Trans. Signal Inf. Process.* **2012**, *1*. [CrossRef]
8. Yang, X.; Liu, J.; Sun, J.; Li, X.; Liu, W.; Gao, Y. DIBR based view synthesis for free-viewpoint television. In Proceedings of the 3DTV Conference True Vision-Capture, Transmission and Display of 3D video (3DTV-CON), Antalya, Turkey, 16–18 May 2011.
9. Feng, Y.M.; Li, D.X.; Luo, K.; Zhang, M. Asymmetric bidirectional view synthesis for free viewpoint and three-dimensional video. *IEEE Trans. Consum. Electron.* **2009**, *55*, 2349–2355. [CrossRef]
10. Mark, W.R.; McMillan, L.; Bishop, G. Post-rendering 3D warping. In Proceedings of the 1997 Symposium on Interactive 3D graphics, Providence, RI, USA, 27–30 April 1997; pp. 7–16.
11. Muddala, S.M.; Sjostrom, M.; Olsson, R. Virtual view synthesis using layered depth image generation and depth-based inpainting for filling disocclusions and translucent disocclusions. *J. Vis. Commun. Image Represent.* **2016**, *38*, 351–366. [CrossRef]
12. Daribo, I.; Saito, H. A novel inpainting-based layered depth video for 3DTV. *IEEE Trans. Broadcast.* **2011**, *57*, 533–541. [CrossRef]

13. Gautier, J.; LeMeur, O.; Guillemot, C. Depth-Based image completion for view synthesis. In Proceedings of the 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D video (3DTV-CON), Antalya, Turkey, 16–18 May 2011.
14. Koppel, M.; Muller, K.; Wiegand, T. Filling disocclusions in extrapolated virtual views using hybrid texture synthesis. *IEEE Trans. Broadcast.* **2016**, *62*, 457–469.
15. Oh, K.J.; Yea, S.; Vetro, A.; Ho, Y.S. Virtual view synthesis method and self-evaluation metrics for free viewpoint television and 3D video. *Int. J. Imaging Syst. Technol.* **2010**, *20*, 378–390.
16. Zinge, S. Free-viewpoint rendering algorithm for 3D TV. In Proceedings of the 2nd International Workshop of Advances in Communication, Boppard, Germany, 13–15 May 2009; pp. 19–23.
17. Criminisi, A.; Perez, P.; Toyama, K. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **2004**, *13*, 1200–1212. [CrossRef] [PubMed]
18. Telea, A. An image inpainting technique based on the Fast Marching Method. *J. Graph. Tools* **2004**, *9*, 25–36. [CrossRef]
19. Cho, J.H.; Song, W.; Choi, H.; Kim, T. Hole filling method for depth image based rendering based on boundary decision. *IEEE Signal Process. Lett.* **2017**, *24*, 329–333. [CrossRef]
20. Oliveira, A.; Fickel, G.; Walter, M.; Jung, C. Selective hole-filling for depth-image based rendering. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 1186–1190.
21. Cheng, C.; Liu, J.; Yuan, H.; Yang, X.; Liu, W. A DIBR method based on inverse mapping and depth-aided image inpainting. In Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 518–522.
22. Zarb, T.; Debono, C.J. Depth-based image processing for 3D video rendering applications. In Proceedings of the IWSSIP 2014 Proceedings, Dubrovnik, Croatia, 12–15 May 2014; pp. 12–15.
23. Su, C.L.; Wu, J.H.; Chen, K.P. Interframe hole filling for DIBR in 3D videos. In Proceedings of the 2015 IEEE International Conference on Consumer Electronics-Taiwan, Taipei, Taiwan, 6–8 June 2015; pp. 386–387.
24. Schmeing, M.; Jiang, X. Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting. *IEEE Trans. Multimed.* **2015**, *17*, 2160–2173.
25. Lie, W.N.; Hsieh, C.Y.; Lin, G.S. Key-frame-based background sprite generation for hole filling in depth image-based rendering. *IEEE Tran. Multimed.* **2018**, *20*, 1075–1087.
26. Luo, G.; Zhu, Y. Foreground removal approach for hole filling in 3D video and FVV synthesis. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 2118–2131.
27. Oliveira, A.; Fickel, G.; Walter, M.; Jung, C. An artifact-type aware DIBR method for view synthesis. *IEEE Signal Process. Lett.* **2018**, *25*, 1705–1709. [CrossRef]
28. Luo, G.; Zhu, Y.; Guo, B. Fast MRF-based hole filling for view synthesis. *IEEE Signal Process. Lett.* **2018**, *25*, 75–79. [CrossRef]
29. Qiao, Y.; Jiao, L.; Yang, S.; Hou, B.; Feng, J. Color correction and depth-based hierarchical hole filling in Free viewpoint generation. *IEEE Trans. Broadcast.* **2019**, *65*, 294–307.
30. McMillan, L. An Image-Based Approach to Three-Dimensional Computer Graphics. Ph.D. Thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1997.
31. Mariottini, G.L.; Prattichizzo, D. EGT for multiple view geometry and visual servoing: Robotics and vision with pinhole and panoramic cameras. *IEEE Robot. Autom. Mag.* **2005**, *12*, 26–39. [CrossRef]
32. Morvan, Y. Acquisition, Compression and Rendering of Depth and Texture for Mult-View Video. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2009.
33. Zitnick, C.L.; Kang, S.B.; Uyttendalel, M.; Winder, S.; Szeliski, R. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. (TOG)* **2004**, *23*, 600–608. [CrossRef]
34. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]