EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Depth-image-based rendering with spatial and temporal texture synthesis for 3DTV

Ming Xi[1,2], Liang-Hao Wang[1,2*], Qing-Qing Yang[1,2], Dong-Xiao Li[1,2] and Ming Zhang[1,2]

Abstract

A depth-image-based rendering (DIBR) method with spatial and temporal texture synthesis is presented in this article. Theoretically, the DIBR algorithm can be used to generate arbitrary virtual views of the same scene in a three-dimensional television system. But the disoccluded area, which is occluded in the original views and becomes visible in the virtual views, makes it very difficult to obtain high image quality in the extrapolated views. The proposed view synthesis method combines the temporally stationary scene information extracted from the input video and spatial texture in the current frame to fill the disoccluded areas in the virtual views. Firstly, the current texture image and a stationary scene image, which is extracted from the input video, are warped to the same virtual perspective position by the DIBR method. Then, the two virtual images are merged together to reduce the hole regions and maintain the temporal consistency of these areas. Finally, an oriented exemplar-based inpainting method is utilized to eliminate the remaining holes. Experimental results are shown to demonstrate the performance and advantage of the proposed method compared with other view synthesis methods.

Keywords: Virtual view synthesis, Three-dimensional television (3DTV), Depth-Image-Based Rendering (DIBR), Stationary scene extraction, Inpainting

1 Introduction

Year 2010 is considered to be the year of breakthrough for 3D video and 3D industry [1]. Numerous 3D films are produced and released to the market. Stereo movies provide people stereo perceptions by showing two slightly different images of the same scene. Consumers can have immersive feelings by watching them in theaters with stereo eyeglasses. Disks and players of 3D Blu-ray standard have entered the home entertainment. The prosperity of 3D industry gives an important opportunity for three-dimensional television (3DTV) system, which is believed to be the next generation of television broadcasting after high-definition television. The concept of 3DTV system is defined by European project ATTEST [2] and developed by Morvan et al. [3] and Kubota et al. [4]. To improve the depth perception of users, autostereoscopic display technology without any need of additional glasses is preferred in the display part of 3DTV. Autostereoscopic displays can provide comfortable stereo parallax and smooth motion disparity by displaying multiview images of the same scene simultaneously. A simple approach is to capture, compress, and transmit multiple views directly. The current multiview video coding standard [5,6] with high compression efficiency, which exploits the spatial correlations of the neighboring views, is used to encode and decode the multiple video streams, generally more than eight views. But the transmission bandwidth cost remains a challenging and unresolved problem. Meanwhile, it is commonly suggested that the future 3DTV systems should have completely decoupled capture and display operations [7]. A proper abstract intermediate representation of the captured data, video plus depth format, is proposed by Fehn [8] to achieve such a decoupled operation with an acceptable increment of bandwidth. The depth-image-based rendering (DIBR) [2] algorithm will be used to render multiple perspective views from the video plus depth data according to the requirement of autostereoscopic displays. Thus, the DIBR method has attracted much attention, and become a key technology of the 3DTV system [1].

The video plus depth data format consists of one texture color image and its corresponding perpixel dense depth

*Correspondence: wang_sunsky@163.com
[1] Institute of Information and Communication Engineering, Zhejiang University, Hangzhou, 310027, P.R. China
[2] Zhejiang Provincial Key Laboratory of Information Network Technology, Zhejiang University, Hangzhou, 310027, P.R. China

map. Theoretically, being provided with the intrinsic and extrinsic parameters of the virtual views, the DIBR algorithm can be used to synthesize any virtual perspective views from the video plus depth data. But there exists three problems [2], which are visibility, resampling, and disocclusion. Multiple pixels of the reference view may fall into the same position in the virtual image plane, which will cause the visibility problem. A Z-buffer algorithm [9] can solve this problem by recording the Z values and choosing the nearest pixel to the virtual camera plane. The phenomenon of an integer pixel position in the reference view image being projected to a subpixel position in the virtual view is called resampling problem, which can be coped with upsampling procedure or backwards warping with interpolation. The remaining disocclusion problem is the fact that some parts of the captured scene, which are occluded in the original views, become visible in the virtual views. It is caused by the lack of scene information occluded by the foreground objects in the original view position. As the distance from virtual view to reference view increases, the disoccluded area becomes larger, as shown in Figure 1.

The disocclusion problem is considered to be the most significant and difficult one of the DIBR algorithm. It is well handled in the interpolation operation [10-13], but will become severe in the extrapolation situation, where the missing image information needs to be reconstructed by appropriate algorithms. Lots of algorithms have been developed to solve this problem, which can be divided into three categories.

The first is layered-depth-image (LDI) [14,15], which can achieve excellent rendering results by providing sufficient information of the scene. LDI data are composed of a number of color layers and their corresponding depth layers, which contain not only the texture and depth information of visible scene from the front view, but also that of the occluded regions. It is very simple to obtain high-quality multiview images from LDI data. However, the procedure of creating LDI is computationally complex and quite time-consuming. The transmission bandwidth of LDI data also increases drastically with the number of layers. A simplified data format of LDI, which is called the “Declipse” format [16], is proposed by Philips Corporation. The “Declipse” format data consist of foreground layer and background layer. It presents the advantage to improve the rendering quality with a quite small overhead in terms of complexity and bitrate.

The second approach is called depth image preprocessing. To reduce the disoccluded areas in virtual views, low pass filter is applied to smooth the depth image. Fehn [2] uses a suitable Gaussian filter preprocessing the depth image to eliminate the disocclusions with the cost of slightly geometric distortions. An asymmetric smoothing method is proposed by Zhang and Tam [17]. By enlarging the standard deviation and window size of Gaussian filter in vertical direction, the vertical structure distortion is reduced. The filtering effect is to smooth the sharp discontinuities in the depth image, thus reducing the hole areas near object boundaries. A consequence of these algorithms is that the whole depth map has been modified, which will severely blur the distance between scene objects in different depth layers. To cope with depth loss, different kinds of oriented filters [18-21] are designed with the same principle, i.e., smoothing the sharp edge in the depth image locally and keeping the depth of the other regions unchanged. The oriented filters can improve the image quality of the virtual views, but still induce geometric distortion. Although the depth image preprocessing methods can be used to handle the disoccluded regions in the virtual views of small baseline, obvious geometric distortions will occur when the baseline is getting larger.

The third approach to filling the disoccluded areas is image completing techniques. This approach can be further classified into statistical-based methods, partial differential equations (PDE)-based methods, and exemplar-based methods. Statistical-based methods [22-25] have good performance in pure texture synthesis applications, but fail to complete natural images with complex structure. PDE-based methods [26-29], which are also called image inpainting methods, propagate linear structures into the disoccluded areas smoothly via diffusion. The diffusion process is simulated by the PDE of physical heat flow. Inpainting methods are suitable for removing small image artifacts, such as speckles, scratches, and overlaid texts. When the disocclusion is getting larger, the diffusion process will over-smooth the image and cause

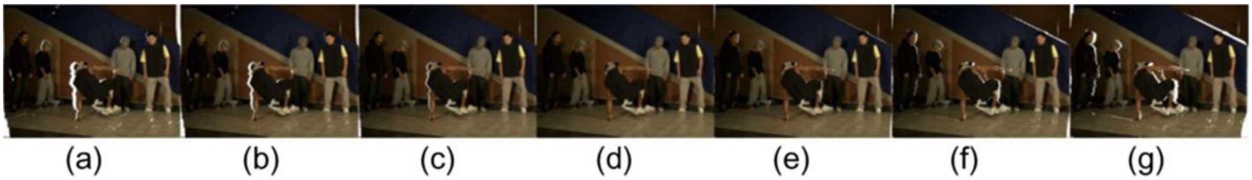

Figure 1 DIBR results for frame 0 of the “Breakdancers” sequence. (d) Reference view. **(a, b, c)** Virtual views on the left side. **(e, f, g)** Virtual views on the right side. White color (R = 255, G = 255, B = 255) is used to represent the hole pixels. The larger the distance between the virtual view and reference view is, the bigger the disoccluded area becomes.

visible blurring artifacts. Exemplar-based methods [30,31] fill the hole regions by copying patches with the similar texture from the known neighborhood of the image. Criminisi et al. [30] use the exemplar-based method to remove objects from images. Komodakis and Tziritas [31] propose an efficient belief propagation method to obtain global optimization. Exemplar-based methods have been used for the case of video completion in [32,33]. Multiple frames are provided as the searching source of best match patch by Cheng et al. [34] to achieve temporal continuity. Exemplar-based methods have been the most powerful techniques for dealing with large disoccluded regions. Schmeing and Jiang [35] first obtain the background information with a computed background model. But their approach cannot handle the uncovered areas caused by static foreground objects. For each virtual view, Ndjiki-Nya et al. [36] use a background sprite to update the texture and depth information of disoccluded areas. There are two major drawbacks of this method. One is the valuable background information of disocclusions, which cannot be reused during the generation of other virtual views. The other is the memory cost increases with the number of virtual views.

In this article, a new virtual view generation method with spatial and temporal texture synthesis is proposed. The structure information of the captured scene in the temporal domain is taken into account by maintaining an accumulated sprite of stationary scene. An oriented exemplar-based inpainting algorithm is applied to restore the rest disoccluded areas with background texture.

The remainder of this article is organized as follows. In Section 2, a brief description of the algorithm framework is given. The details of each processing modules are demonstrated in Sections 3, 4, 5, and 6. Experimental results are compared with state-of-the-art methods in Section 7. The conclusions and future works can be found in Section 8.

2 System overview

The framework of proposed DIBR method with spatial and temporal texture synthesis is shown in Figure 2. The proposed method is divided into four main stages, i.e., stationary scene extraction, backward DIBR, merging operation, and oriented exemplar-based inpainting.

In the first stage, a sprite of stationary scene is maintained throughout the view synthesis process, which stores the temporally accumulated structure and depth information of stationary image part. The Structural SIMilarity index (SSIM) [37] is utilized to distinguish the stationary scene from the moving foreground objects by combining the input depth images. For stationary scene, the SSIM index between adjacent frames is large, so the image part, which is stationary in both adjacent frames, can be extracted by using the SSIM index values. But there still are some stationary scenes, which cannot be distinguished due to the occlusions of moving foreground objects. By considering the spatial relationship provided by the input depth maps, the texture information of these occluded stationary scenes can also be obtained. In the demonstration of our algorithm, the camera of input view is supposed to be still for simplicity. If the camera is moving, an additional camera tracking module needs to be inserted before stationary scene extraction stage to compensate the global motions, which is beyond the discussion in this article.

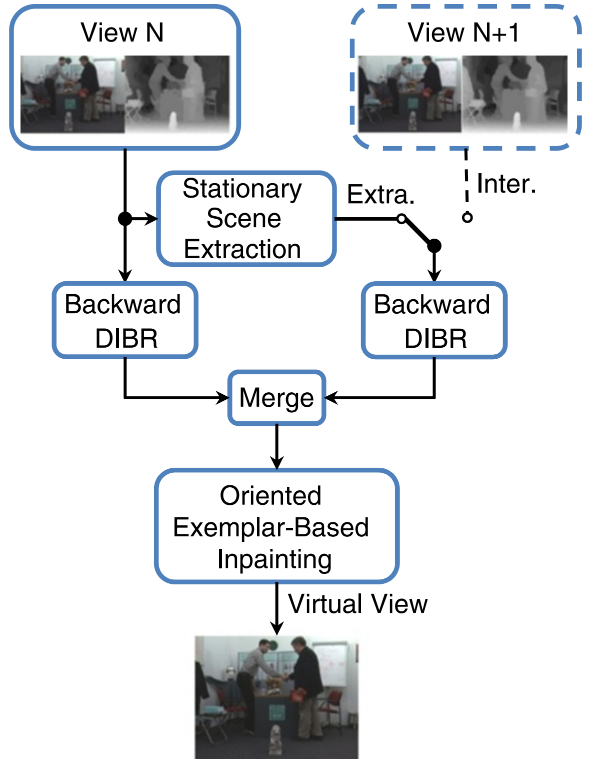

Figure 2 Flow diagram of the proposed virtual view synthesis method. First, the sprite of stationary scene is initialized with the first image frame of the input data. Next, each frame is compared with the sprite of stationary scene to extract more structure information. Then both the current frame and sprite of stationary scene are rendered to the same virtual view and merged together. Finally, the remaining disoccluded areas are filled by proposed oriented exemplar-based inpainting.

In the second stage, current frame and stationary scene sprite are warped to the same virtual perspective view by a backward DIBR method to tackle the visibility problem and resampling problem.

The proposed algorithm merges these two virtual images obtained from the second stage together with the third stage. The merging operation needs to be done very carefully, because the foreground objects in virtual views may still exist inner hole pixels. The merging operation can take use of most of the scene information provided by the sprite of stationary scene.

After the merging operation, there still exists a few blank regions without pixel values. In the final stage, oriented exemplar-based inpainting approach is applied to fill the remaining holes by searching best matching exemplar with background texture. Current virtual image is used as the searching source of best matching patch. The filling order of the inpainting method is steered from background structures to foreground objects.

Note that the proposed method only uses the sequence of color images and depth images from one captured view as the input data. If image data of another view are also provided, the switch in the framework can directly be shifted from the extrapolation mode to the interpolation mode without any changes of the framework.

3 Stationary scene extraction

The DIBR algorithm warps the original view to the virtual view position by projecting current pixels to points in real 3D space and re-projecting the 3D points to virtual image plane. Large disocclusions will appear in the discontinuous edges of depth map, which is the transition place between foreground and background in texture image. The background image part occluded by foreground objects should be visible in the virtual views. But the occluded background information is lost during the procedure of recording a 3D scene by a 2D image. To solve this problem, the proposed stationary scene extraction module tries to recover the lost background structure from video sequences. For a video captured by a fixed camera or a short cut of video, the image consists of moving foreground objects and stationary background. The occluded background information in current image frame may appear in frames at other moments. If the information can effectively be used, the filling effect of disoccluded areas will be more convincing.

Stationary scene extraction algorithm keeps a global sprite throughout the view generation process to accumulate structure and depth information of stationary scene in temporal direction. The global sprite of stationary scene is composed of two components: one is the texture image of stationary scene, denoted as C_{SS}, the other is the depth map of stationary scene, denoted as M_{SS}. C_{SS} and M_{SS} are, respectively, initialized with the first frame of the texture sequence and depth sequence of the original view. The initialization step is expressed as follows:

$$\begin{cases} C_{\mathrm{SS}}(p) = I_t(p) \\ M_{\mathrm{SS}}(p) = D_t(p) \end{cases}, \quad t = 0 \tag{1}$$

where $p : (i,j)$ corresponds to the pixel of column coordinate i and row coordinate j. I_t and D_t represent the color intensity frame and depth map frame of input original view at time t, respectively. D_t is represented as an 8-bits gray-scale image. The continuous depth range is quantized to 255 discrete depth values. The nearest object to the camera image sensor is assigned with 255 and the farthest object is assigned with 1. Pixels with depth value 0 are denoted as holes. The transform formula between discrete depth level and actual distance in real scene can be found in [12].

After the initialization, a temporary sprite of stationary scene, denoted as TC_{SS} and TM_{SS}, is obtained between each input image frame I_t and its previous frame I_{t-1} to extract the useful information of occluded background in I_t. For stationary scene, the SSIM index [37] between adjacent frames is large, so the image part, which is stationary in both adjacent frames, can be extracted by using the SSIM index values. For each pixel $p : (i,j)$, a structure similarity index p_{SSIM} defined in [37] is calculated between the corresponding square areas Φ_t^I and Φ_{t-1}^I of I_t and I_{t-1}, which take p as the center pixel and $L \times L$ as the window size. The SSIM p_{SSIM} is calculated as follows

$$p_{\mathrm{SSIM}} = \frac{(2\mu_{\Phi_t}\mu_{\Phi_{t-1}} + K_1)(2\sigma_{\Phi_{t(t-1)}} + K_2)}{(\mu_{\Phi_t}^2 + \mu_{\Phi_{t-1}}^2 + K_1)(\sigma_{\Phi_t}^2 + \sigma_{\Phi_{t-1}}^2 + K_2)} \tag{2}$$

where μ_{Φ_t}, $\mu_{\Phi_{t-1}}$ represent the luminance mean value of Φ_t^I and Φ_{t-1}^I, respectively. σ_{Φ_t} and $\sigma_{\Phi_{t-1}}$ represent the luminance standard deviation of Φ_t^I and Φ_{t-1}^I. $\sigma_{\Phi_{t(t-1)}}$ denotes the luminance correlation coefficient between Φ_t^I and Φ_{t-1}^I. K_1 and K_2 are constants. The value of K_1 and K_2 can be determined according to the research work in [37]. The expressions of mean, standard deviation, and correlation coefficient can also be found in [37].

Then an arbiter with threshold A is used to divide the pixels of input image frame I_t into stationary part I_s and rest part I_r. The classifier can be expressed as follows:

$$\begin{cases} p \in I_s, & p_{\mathrm{SSIM}} \geq A \\ p \in I_r, & p_{\mathrm{SSIM}} < A \end{cases}, \quad p : (i,j) \in I_t. \tag{3}$$

I_s contains the stationary pixels with high SSIM value, which can directly be used to update the same pixel positions in TC_{SS}. I_r are composed of three parts: the part with changed luminance P_{lc}, the relatively moving part P_{rm}, and the actually moving part P_{am}. P_{lc} represents the areas with similarly scene structure and different luminance which causes the decrease of SSIM value. P_{rm} is the region which is moving in I_{t-1} and stationary in I_t. P_{am} denotes the image part which is moving in I_t and stationary in I_{t-1}. As shown in Figure 3c, I_s between Figure 3a,b is marked as black, the actually moving part P_{am} is marked as red, the region with changed luminance P_{lc} is marked as green, and the relatively moving area P_{rm} is marked as blue. The first two kinds P_{lc} and P_{rm} can be also used to update TC_{SS} directly, whereas the third kind P_{am} needs to be excluded from I_t and the pixels in the same

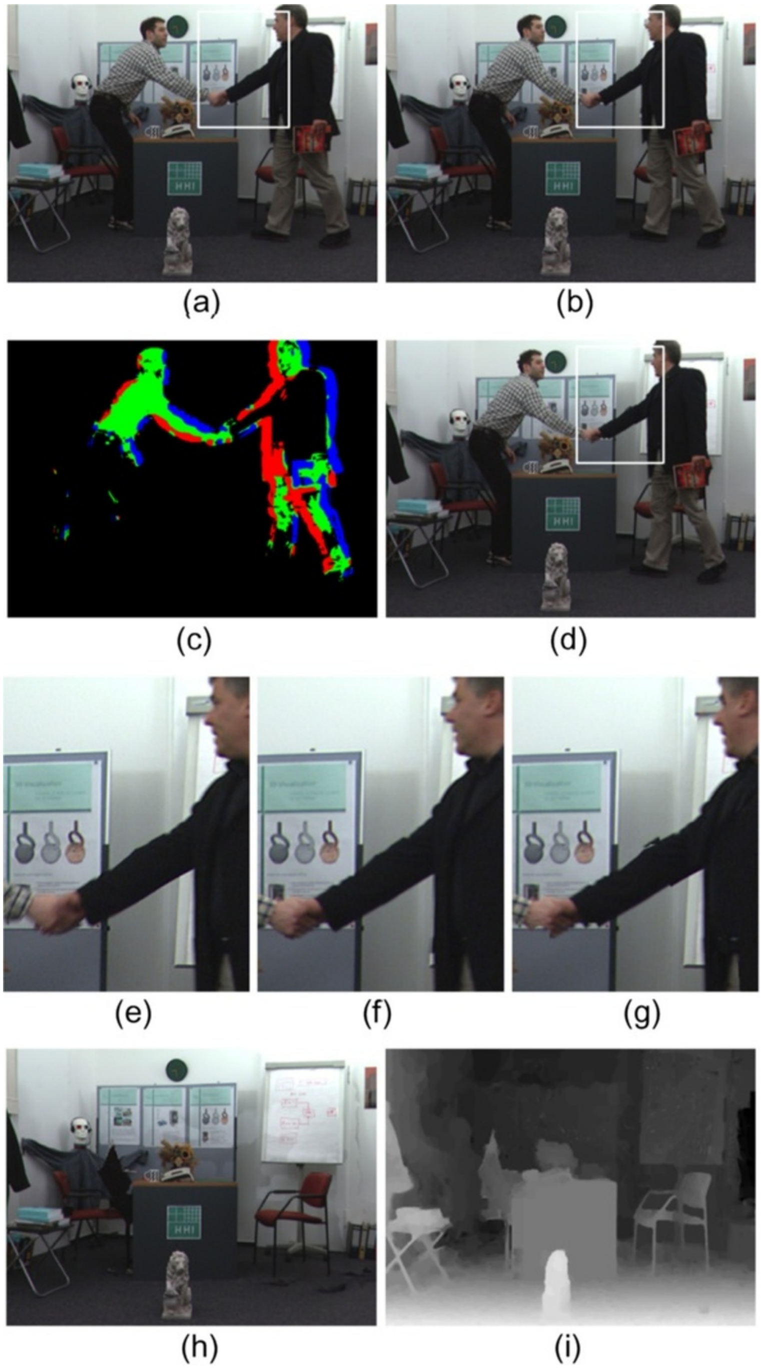

Figure 3 Stationary scene extraction result of the "Book arrival" sequence. (a) The 36th frame and **(b)** the 37th frame of "Book arrival" sequence from the 8th camera position. **(c)** The segment map of **(a)**. The stationary part I_s between **(a)** and **(b)** is marked as black. The actually moving part P_{am} is marked as red. The region with changed luminance P_{lc} is marked as green. The relatively moving area P_{rm} is marked as blue. **(d)** The texture image of the temporary stationary scene sprite for the 37th frame. **(e)** Magnified subsection in **(a)**. **(f)** Magnified subsection in **(b)**. **(g)** Magnified subsection in **(d)**. **(h)** The texture image of global stationary scene sprite for **(a)**. **(i)** The depth map of global stationary scene sprite for **(a)**.

regions of I_{t-1} will be used to update TC_{SS}. As shown in Figure 3e–g, the poster occluded by the men's hands in Figure 3e and the white board behind the man in Figure 3f are all preserved in Figure 3g. Provided with the corresponding depth map D_t and D_{t-1}, the three different image parts are defined as follows.

$$\begin{cases} p \in P_{\mathrm{lc}}, & \left|\mu_t^D - \mu_{t-1}^D\right| \leq T \\ p \in P_{\mathrm{rm}}, & \mu_t^D - \mu_{t-1}^D < -T\,, \quad p : (i,j) \in I_r \\ p \in P_{\mathrm{am}}, & \mu_t^D - \mu_{t-1}^D > T \end{cases} \tag{4}$$

where μ_t^D and μ_{t-1}^D, respectively, represent the average depth value of square areas in D_t and D_{t-1}. The square neighborhoods have the same window size $L \times L$ with SSIM computation in Equation (2) and take the coordinates of pixel p as center position. T is a constant threshold, which defines the acceptable range of depth fluctuation. $|\cdot|$ is the absolute function.

Then the information of stationary scene between two adjacent frames can be extracted by the following equation:

$$\begin{aligned} TC_{\mathrm{SS}}(p) &= \begin{cases} I_t(p), & p : (i,j) \in I_s \cup P_{\mathrm{lc}} \cup P_{\mathrm{rm}} \\ I_{t-1}(p), & p : (i,j) \in P_{\mathrm{am}} \end{cases} \\ TM_{\mathrm{SS}}(p) &= \begin{cases} D_t(p), & p : (i,j) \in I_s \cup P_{\mathrm{lc}} \cup P_{\mathrm{rm}} \\ D_{t-1}(p), & p : (i,j) \in P_{\mathrm{am}} \end{cases} \end{aligned} \tag{5}$$

Finally, the temporary sprite of stationary scene (TC_{SS} and TM_{SS}) is used to update the global sprite (C_{SS} and M_{SS}). The update operation is described as follows.

$$\begin{aligned} C_{\mathrm{ss}}(p) &= \begin{cases} TC_{\mathrm{SS}}(p), \mu_{\mathrm{TM}}^p - \mu_M^p \leq T \\ C_{\mathrm{SS}}(p), \quad \text{otherwise} \end{cases} \quad p : (i,j) \in C_{SS} \\ M_{\mathrm{ss}}(p) &= \begin{cases} TM_{\mathrm{SS}}(p), \mu_{\mathrm{TM}}^p - \mu_M^p \leq T \\ M_{\mathrm{SS}}(p), \quad \text{otherwise} \end{cases} \quad p : (i,j) \in M_{\mathrm{SS}} \end{aligned} \tag{6}$$

where μ_{TM}^p and μ_M^p, respectively, represent the average depth value of square areas in TM_{SS} and M_{SS}. The square neighborhoods have the same window size $L \times L$ with SSIM computation and take the coordinates of pixel p as center position. T is the same constant threshold defined in Equation (4). Figure 3d shows TC_{SS} of Figure 3b. Figure 3h,i are C_{SS} and M_{SS} of Figure 3b, respectively. Almost all the texture and depth information of stationary scene are restored in Figure 3h,i.

So far, the appeared background information in past frames is stored in C_{SS} and M_{SS}, which can be used to partly solve the disocclusion problem of virtual view synthesis algorithm.

4 Backward DIBR

The backward DIBR method, which shares the same idea with the inverse warping method in [13], can efficiently eliminate the small cracks in virtual view caused by resampling problem in traditional DIBR process [2]. In general, the backward DIBR method can be divided into two steps: warping the depth map of the reference view to the virtual view position and generating the texture image of the virtual view.

In the backward DIBR method, D_t, is warped to virtual perspective position. A two-pixel-wide region around background–foreground transitions is marked as unreliable pixels. During the rendering process of depth map, the unreliable pixels will be skipped, because their depth values are inaccurate. There are four registers in each pixel $q\ :\ (u,v)$ of virtual view, which are used to store the depth and distance of four nearest pixels projected from the reference image. The four registers of pixel q only store rendered pixels from reference image whose distance to q is less than one pixel either in horizontal or vertical direction. VD_t, the depth map of virtual view, is calculated as follows

$$VD_t(q) = \begin{cases} \displaystyle\sum_{k=1}^{N(q)} \lambda_k D_k, N(q) > 0 \text{ and } N(q) \leq 4 \\ 0, \quad N(q) = 0 \end{cases} \tag{7}$$

where $N(q)$ denotes the numbers of pixels warped to q, which satisfy the condition mentioned above. If $N(q)$ is larger than 4, we sort the warped pixels by its depth value in large to small order and store the first four pixels with larger depth. D_k is the depth value of stored pixel. $N(q) = 0$ means there is no pixel that is projected to pixel q. λ_k represents the normalized weight factor with the combination of distance and depth, which is defined as

$$\lambda_k = \frac{\rho_k \omega_k}{\sum_{m=1}^{N(q)} \rho_m \omega_m}, \quad \sum_{k=1}^{N(q)} \lambda_k = 1 \tag{8}$$

where the weight factor of distance ω_k is expressed as Equation (9). (U_k, V_k) is the projected position of warped pixel in virtual image plane.

$$\omega_k = \frac{1}{\sqrt{(U_k - u)^2 + (V_k - v)^2}} \tag{9}$$

The weight factor of depth ρ_k is expressed as

$$\rho_k = \begin{cases} 1, & D_k \geq \mu_{ND} \\ 0, & D_k < \mu_{ND} \end{cases} \tag{10}$$

where μ_{ND} is the average depth value of all the stored warped pixels in pixel q.

The non-hole pixel (u,v) in VD_t is reprojected to position (X_{uv}, Y_{uv}) in image plane of original view to get the texture image of virtual view by interpolation operation. The texture image of virtual view VI_t is calculated by

$$VI_t(q) = \begin{cases} \dfrac{\sum\limits_{n=1}^{4} \theta_n I_n}{\sum\limits_{n=1}^{4} \theta_n}, & VD_t(q) > 0 \\ \text{hole}, & VD_t(q) = 0 \end{cases} \tag{11}$$

where 'hole' flag means there is no warped pixel from the reference image. We set the hole pixels with a white color (R = 255, G = 255, B = 255). I_n represents the color value of pixel (x_n, y_n) whose distance to (X_{uv}, Y_{uv}) is less than one pixel either in horizontal or vertical direction. θ_n is the weight factor of distance, which is expressed as

$$\theta_n = \frac{1}{\sqrt{(X_{uv} - x_n)^2 + (Y_{uv} - y_n)^2}}. \tag{12}$$

The virtual depth map VM_t projected from M_{SS} and the virtual texture image VC_t projected from C_{SS} can be obtained by the same backward DIBR method. Two results of our backward DIBR algorithm are given in Figure 4e,f.

5 Merging operation

To efficiently use the structure information in C_{SS}, the two virtual texture images (VI_t and VC_t) need to be merged together. The merged virtual image and its depth map are denoted as MI_t and MD_t, respectively. The virtual view image VI_t is dominated in the merging process. Available background information in VC_t is used to fill the blank areas in VI_t. There may be holes in both foreground and background due to the inaccuracy of depth map, as shown in Figure 4e. We do the merging operation carefully to avoid filling holes in foreground with background structures.

First, an estimated depth value D_E^q is obtained for each hole pixel $q : (u, v)$ in VI_t. As mentioned in Section 3, the hole regions of virtual view are lacking of background information. When q locates between background and foreground, we choose the small depth value of background scene as estimation and the average depth otherwise. The estimation is defined as

$$D_E^q = \begin{cases} \dfrac{\mu_D^{q_L} + \mu_D^{q_R}}{2}, \left|\mu_D^{q_L} - \mu_D^{q_R}\right| \le T \\ \mu_D^{q_R}, \quad \mu_D^{q_L} - \mu_D^{q_R} > T \\ \mu_D^{q_L}, \quad \mu_D^{q_L} - \mu_D^{q_R} < -T \end{cases} ; \quad q \text{ is hole} \tag{13}$$

where q_L and q_R represent the first left and first right non-hole pixel in horizontal column, respectively. $\mu_D^{q_L}$ and $\mu_D^{q_R}$ represent the average depth of the $K \times K$ windows which take q_L and q_R as the center pixels in VD_t. T is the same constant defined in Equation (4).

Then the merging operation is executed as follows.

$$MI_t(q) = \begin{cases} VI_t(q), & VI_t(q) \text{is non-hole} \\ VC_t(q), & VI_t(q) \text{ is hole and } VC_t(q) \text{ is non-hole and } \left|VM_t(q) - D_E^q\right| \le F \\ \text{hole}, & \text{otherwise} \end{cases} \tag{14}$$

where non-hole flag means there exists a meaningful value in this pixel position. The second condition in Equation (14) defines the situation, i.e., the pixel q is hole in VI_t, but meaningful pixel with available background texture in VC_t. This condition ensures that the holes in foreground objects will not be filled with the accumulated background information in VC_t. F represents the acceptable range of depth fluctuation in merging operation. In Figure 4g, the available texture of stationary background scene in Figure 4f is merged with the virtual image (Figure 4e) rendered from original view and the hole areas in foreground objects are reserved. The corresponding depth value of each non-hole pixel in merged virtual view MI_t is stored in MD_t, and the depth value of each hole pixel is set to zero.

6 Oriented exemplar-based inpainting

The merging operation can solve the disocclusion problem partly, because the useful background information in C_{SS} and M_{SS} is limited. There still exist hole areas in the merged virtual view MI_t, which are divided into two kinds: the foreground holes caused by inaccurate depth map and the blank areas caused by occlusion in original view. The image part with known pixels is defined by Λ, and the remaining hole area is denoted as Γ. The border of hole area Γ is defined as $\partial\Gamma$, as shown in Figure 5a.

To restore the missing information of the remaining hole areas, we propose an oriented exemplar-based inpainting algorithm based on the previous work of Criminisi et al. [30]. They determine the filling order of hole pixel $h \in \partial\Gamma$ by assigning each hole pixel a priority $P(h)$. The hole pixel with the highest priority is first filled with the best match patch in Λ. The priority is the product of the confidence term $C(h)$ and the data term $D(h)$. The confidence term enforces to fill hole with large support set of known pixels first, while the data term ensures the continuous propagation of linear structure into hole regions. Noticing the fact that most remaining holes are due to a lack of scene information of the stationary background, we improve their algorithm in two ways. One is filling the border pixel in $\partial\Gamma$ which is adjacent to background area, first. The other is choosing the texture of known background area to restore the disoccluded regions. The improvements are implemented by considering depth cue in the

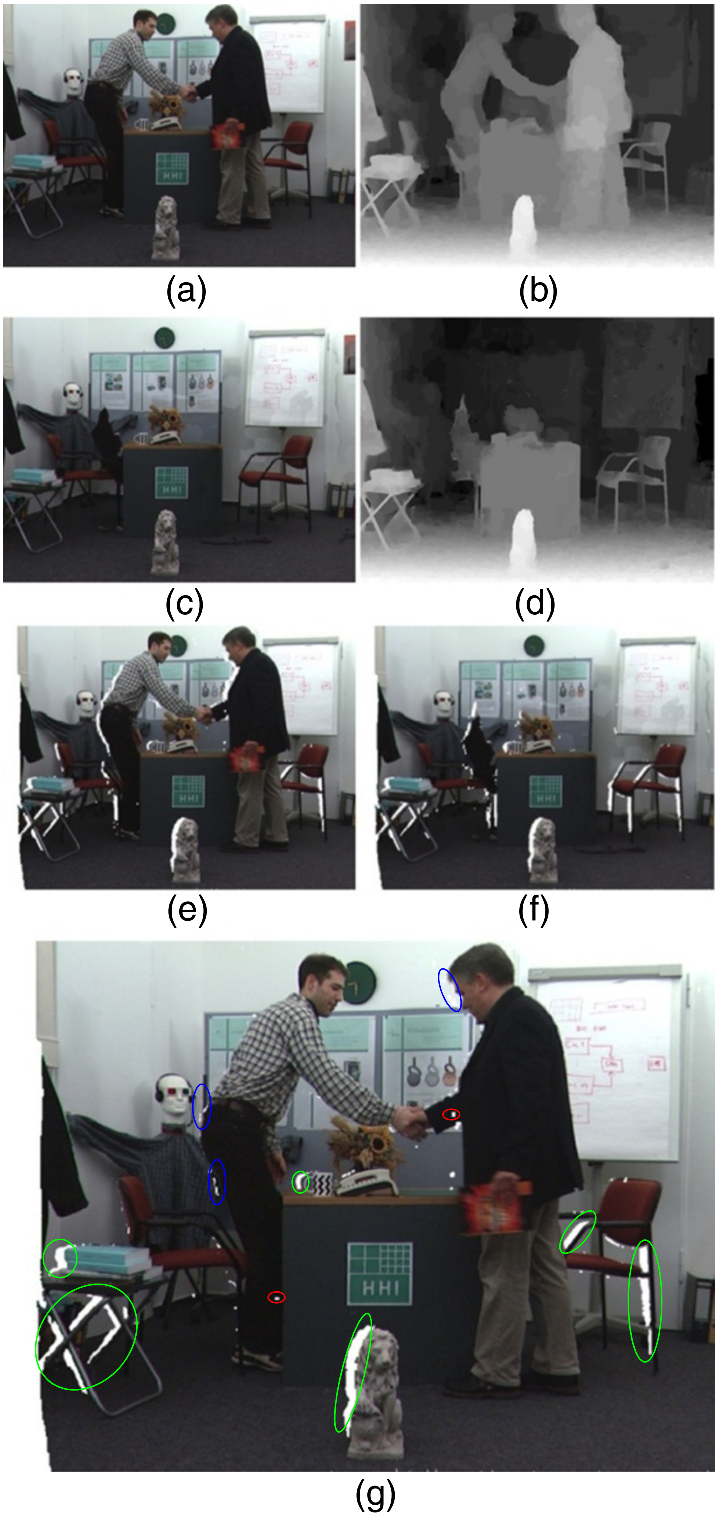

Figure 4 Backward DIBR results and merged virtual image of "Book arrival" sequence. (a) The 50th frame of "Book arrival" sequence from the 8th camera position. **(b)** The corresponding depth map of **(a)**. **(c)** The texture image of the global stationary scene sprite for **(a)**. **(d)** The depth map of the global stationary scene sprite for **(a)**. Our backward DIBR method results from the 8th camera position to the 10th camera position with disoccluded areas marked as white color: **(e)** generated from **(a, b, f)** generated from **(c, d)**. **(g)** Proposed merging approach result of **(e, f)**. The blank areas in red color circles are inner holes in foreground objects. The hole regions in green circles are disocclusion caused by stationary foreground objects. The hole regions in blue color circles are caused by the inaccuracy of depth map.

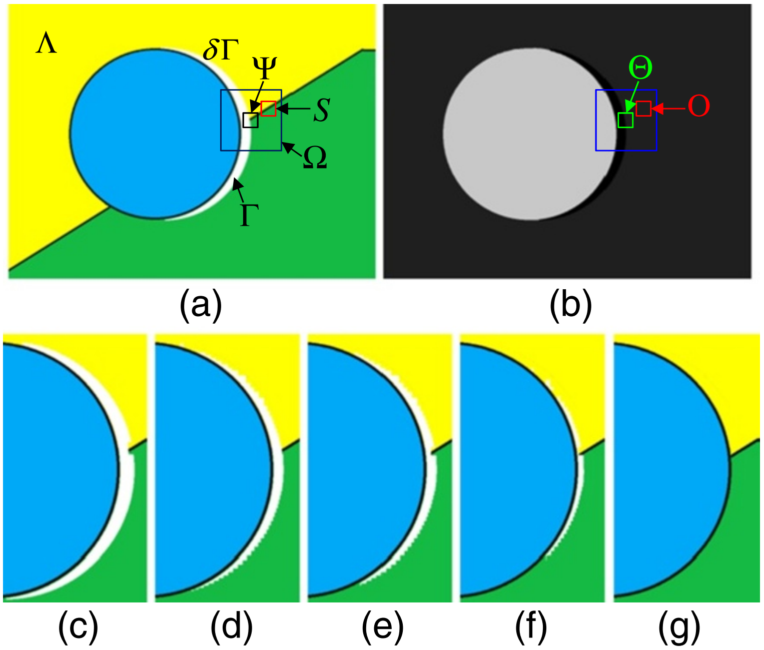

Figure 5 Effect of proposed oriented exemplar-based inpainting algorithm on a synthetic image. (a) A synthetic image of virtual view with disoccluded areas, which are represented by white color. **(b)** The corresponding depth map of **(a)**. The black color indicates the hole areas in depth map. **(c–g)** The magnified result of proposed oriented exemplar-based inpainting algorithm after N iterations: **(c)** $N = 2$, **(d)** $N = 70$, **(e)** $N = 77$, **(f)** $N = 145$, and **(g)** $N = 196$. The patch size of the oriented exemplar-based inpainting algorithm is set to 15 × 15, and the search window size is 15 × 15.

calculation of the priority term and the energy function, both of which are used for the best exemplar searching procedure.

The modified priority term is defined as

$$P(h) = C(h)D(h) + de(h), \quad h \in \delta\Gamma \tag{15}$$

where $de(h)$ represents the depth term. The definition of $C(h)$ and $D(h)$ is the same as Criminisi's approach, and their expressions can be found in [30]. The depth term is expressed as follows.

$$de(h) = \begin{cases} Q, & h \text{ near to } BG \\ 0, & h \text{ near to } FG \end{cases}, \quad h \in \delta\Gamma \tag{16}$$

where BG and FG represent the background areas and foreground objects, respectively. Q is a constant, which should be no less than the maximum of the product of $C(h)$ and $D(h)$. We set $Q = 256$ in our framework. The new priority term will steer the filling order from background to foreground and keep the advantage of linear structure propagation.

Let r denote the pixel with maximum priority in $\partial\Gamma$. The $J \times J$ samples patch, which takes r as center, is defined as Ψ. A square area around r with $W \times W$ samples is defined to be the searching area Ω. Then the oriented exemplar-based inpainting algorithm needs to search for the best match patch S in Ω, which has the most similar texture with Ψ. The center of S is denoted as s. The corresponding depth areas of Ψ and S are represented by Θ and O, respectively.

The energy function combining the depth cue is expressed as follows.

$$E = \sum_{m \in \Psi_k} \|\Psi(m) - S(m)\|^2 + \beta \sum_{m \in \Psi_k} \|\Theta(m) - \text{O}(m)\|^2 + \gamma \left|\mu_\Theta^k - \mu_\text{O}^u\right|^2 \tag{17}$$

where Ψ_k denotes the position set of known pixels in the filling target patch Ψ. The position set of hole pixels in Ψ is represented by $\Psi_u : \Psi_u = \Psi - \Psi_k$. $\Psi(m)$ and $S(m)$ denote the pixel value of pixel position m in Ψ and S, respectively. $\Theta(m)$ and $\text{O}(m)$ represent the depth value of pixel position m in Θ and O, respectively. β is a constant, which is the weighting factor for the depth values of corresponding pixels with Ψ_k in Θ. μ_Θ^k represents the average depth value of the corresponding pixels with Ψ_k in Θ. μ_O^u represents the average depth value of the corresponding pixels with Ψ_u in O. μ_Θ^k and μ_O^u are defined as

$$\mu_\Theta^k = \sum_{m \in \Psi_k} \Theta(m)/\left|\Psi_k\right|, \qquad \mu_\text{O}^u = \sum_{m \in \Psi_u} \text{O}(m)/\left|\Psi_u\right| \tag{18}$$

where $|\Psi_u|$ denotes the area of Ψ_u. γ is the penalizing factor for the candidate patches with foreground texture. γ is an adaptive parameter related to the area of Ψ_k, denoted as $|\Psi_k|$. Then γ is calculated as

$$\gamma = \begin{cases} 0, & \mu_{\mathrm{O}}^{u} - \mu_{\Theta}^{k} \leq T \\ 10\,|\Psi_k|\,, & \text{otherwise} \end{cases} \tag{19}$$

where T is a constant as defined in Equation (4).

The best match block in the searching area Ω is obtained by minimizing the energy cost function (17). The first term in energy function (17) represents the texture difference between the known pixels in target patch Ψ and the corresponding pixels in match patch S. In our approach, only the luminance component is considered. The second term in (17) indicates the depth similarity, which has lower importance than the first texture term. The third term is a penalization term. If there exist pixels of foreground objects in the corresponding area of Ψ_u in S, the penalization term will become larger. The likelihood of selecting patches with foreground pixels is greatly reduced by adding the penalization term. According to the definition of the energy function, the patches of the background scene, which contain similar texture and depth structure with the target block, will be selected to restore the missing information of the disoccluded image areas. We applied our oriented exemplar-based inpainting method to synthesize the missing texture information of disoccluded area in Figure 5a. The blank region is filled from background scene to foreground objects, and the linear structure is propagated into the hole in an appropriate way (see Figure 5c–g).

7 Methods

To evaluate the performance of the proposed method, we compare our approach with other methods, including the MPEG view synthesis reference software (VSRS, version 3.5) [38], the depth-based inpainting method in [29], and the Asymmetric Gaussian filtering method of Zhang and Tam [17].

Our experiments are carried out on three test sequences: “Book arrival”, “Breakdancers”, and “Ballet”. These sequences have 100 frames and a resolution of 1024 × 768 samples. Multiple video plus depth data from different camera views are available. “Book arrival” sequence is captured by a parallel camera array and the others are obtained by a toed-in camera array. The baseline between two adjacent cameras is approximately 6.5 cm for “Book arrival” sequence and 20 cm for the other two sequences.

The parameter values used in our proposed algorithm is summarized in Table 1. The optimized parameters are used for MEPG method (VSRS 3.5). For Asymmetric Gaussian filtering method, we utilize strong smoothing parameters to eliminate the disoccluded areas caused by large camera baseline. We set the horizontal and vertical standard deviations of the Gaussian kernel to 20 and 60, respectively. The filter window sizes are set to 61 samples horizontally and 193 samples vertically. In the experiments, the Asymmetric Gaussian filtering method and the depth-based inpainting method employ the backward DIBR approach proposed in Section 4 to handle the visibility and resampling problems, just the same as our proposed method.

Table 1 Parameter values used in proposed method

Parameter	*L*	*A*	*T*	*K*	*F*	*J*	*W*	*β*
Value	19	0.7	3	5	8	9	15	0.5

7.1 Subjective evaluation

The view synthesis results of these three test video sequences are shown in Figures 6, 7, and 8. All of the four presented approaches can handle the visibility and resampling problems and fill the disoccluded areas in virtual view. Our proposed algorithm has the best subjective effects compared to the others three methods.

The Asymmetric Gaussian filtering method causes noticeable geometric distortions. The vertical structure is curved in Figures 6c and 7c. The foreground objects become fat, as shown in Figures 6k and 7g,k. This method will slightly shift the object away from its correct position (see Figure 6g), which will reduce the disparity between reference image and virtual image and decrease the 3D feelings. For the purpose of autostereoscopic display, although the visual quality of Figure 6g is still pleasant, the depth perception of the scene is distorted due to these shifts. The distorted stereo display will make people fill uncomfortable and arouse visual fatigues. The depth-based inpainting method can restore the blank areas with color of background pixels, but induce severe blurring artifacts (see Figures 6l, 7h,l, and 8h) and some color bleeding defects (see Figures 7h,l and 8h). The filling results are very uncomfortable for visual experience. The VSRS method will lead to significant horizontal structure artifacts (as shown in Figures 6i,m, 7i,m, and 8i,m) and decrease the visual quality greatly.

The proposed approach utilizes the accumulated information of stationary scene to fill the disoccluded areas and achieves convincing effect, as shown in Figures 6n, 7j, and 8j. The missing structure of blank regions is restored with the true background structure. Even for the disoccluded areas caused by stationary foreground objects, our proposed method can obtain plausible filling results. As shown in Figures 6j and 7n, the hole areas are filled with the texture of background scene without losing the sharpness compared to Figure 6h,l. Figure 8l gives better visual effect than Figure 8n. Because the man’s leg is very close

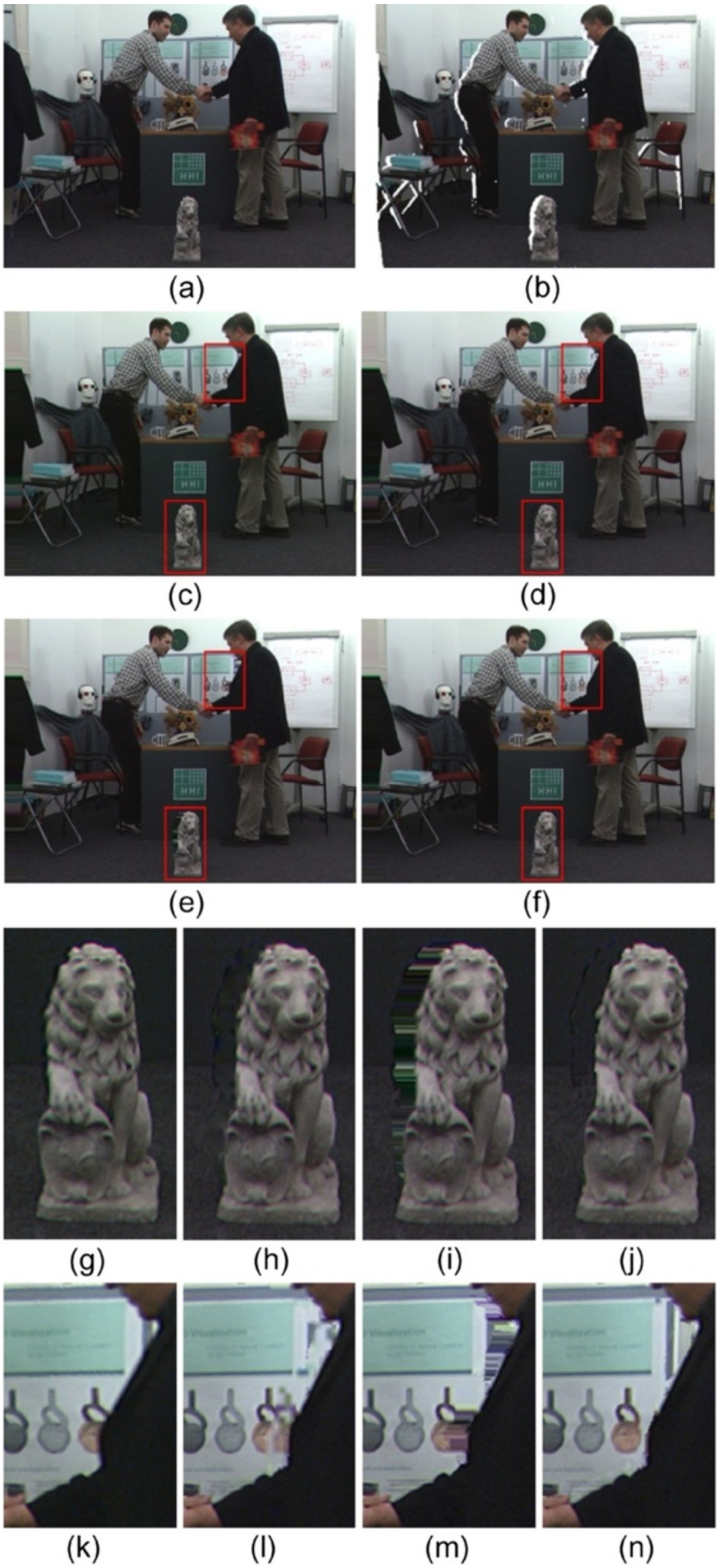

Figure 6 DIBR results for the "Book arrival" sequence from the 8th camera to the 10th camera. (a) Reference image, which is the 50th frame of the 10th camera position. **(b)** Rendered image of backward DIBR approach with hole areas, which are marked as white color. **(c)** Result of Asymmetric Gaussian filtering method. **(d)** Result of depth-based inpainting algorithm. **(e)** Result of VSRS. **(f)** Result of proposed approach. **(g, k)** Magnified subsection in **(c)**. **(h, l)** Magnified subsection in **(d)**. **(i, m)** Magnified subsection in **(e)**. **(j, n)** Magnified subsection in **(f)**.

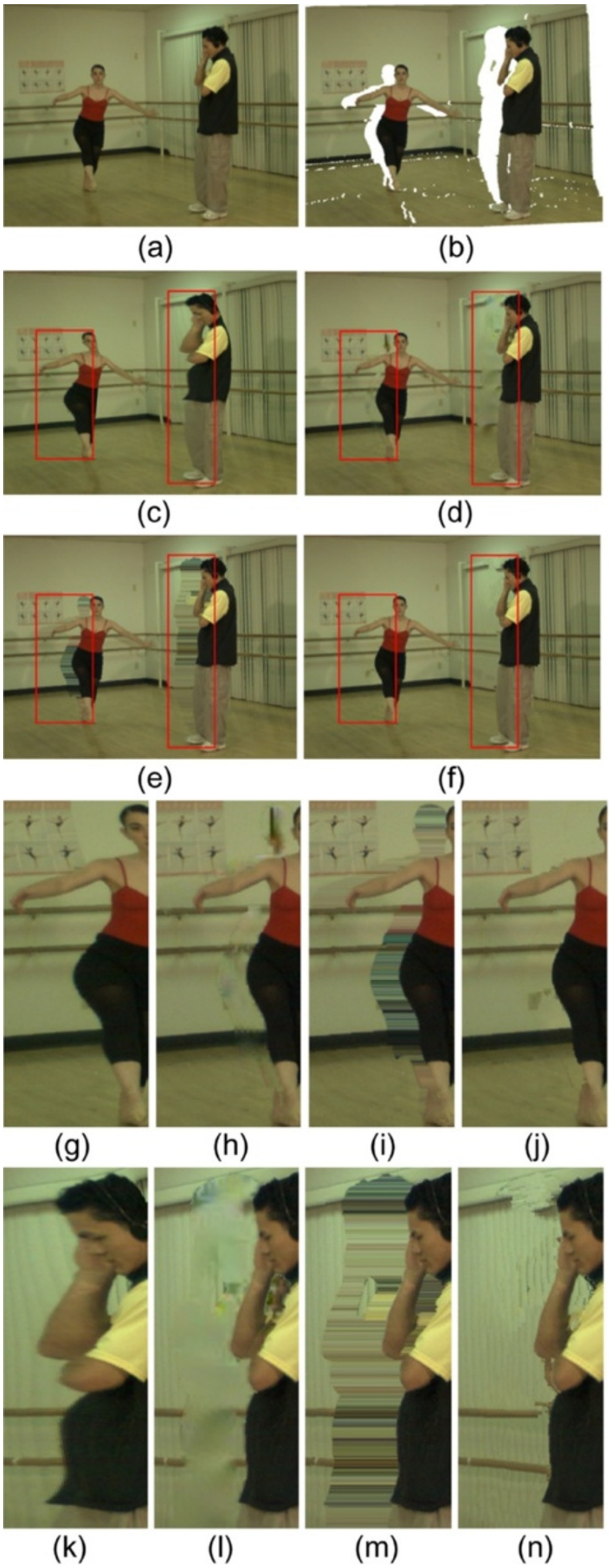

Figure 7 DIBR results for the "Ballet" sequence from the 3rd camera to the 4th camera. (a) Reference image, which is the 4th frame of the 4th camera position. **(b)** Rendered image of backward DIBR approach with hole areas, which are marked as white color. **(c)** Result of Asymmetric Gaussian filtering method. **(d)** Result of depth-based inpainting algorithm. **(e)** Result of VSRS. **(f)** Result of proposed approach. **(g, k)** Magnified subsection in **(c)**. **(h, l)** Magnified subsection in **(d)**. **(i, m)** Magnified subsection in **(e)**. **(j, n)** Magnified subsection in **(f)**.

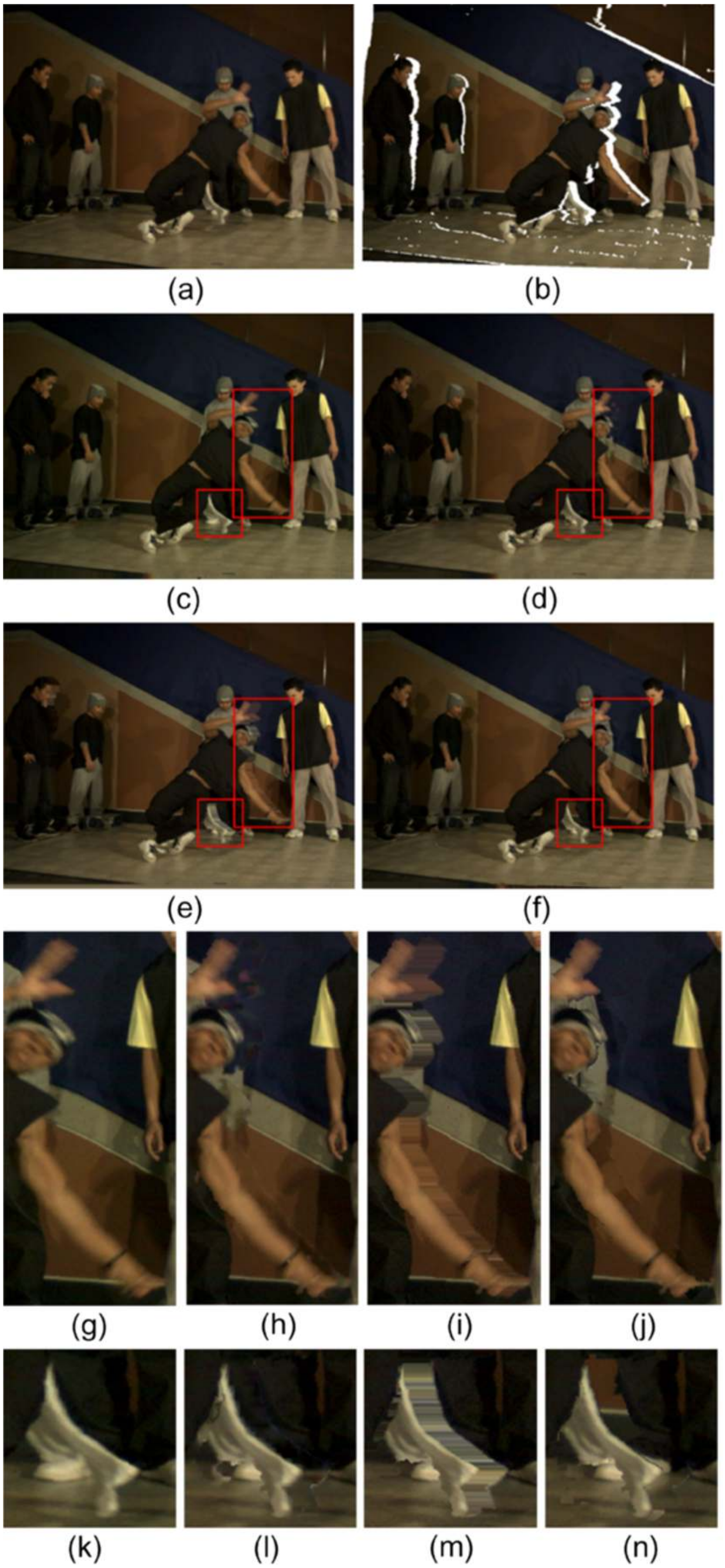

Figure 8 DIBR results for the "Breakdancers" sequence from the 5th camera to the 4th camera. (a) Reference image, which is the 69th frame of the 4th camera position. **(b)** Rendered image of backward DIBR approach with hole areas, which are marked as white color. **(c)** Result of Asymmetric Gaussian filtering method. **(d)** Result of depth-based inpainting algorithm. **(e)** Result of VSRS. **(f)** Result of proposed approach. **(g, k)** Magnified subsection in **(c)**. **(h, l)** Magnified subsection in **(d)**. **(i, m)** Magnified subsection in **(e)**. **(j, n)** Magnified subsection in **(f)**.

to the wall in Figure 8b, it is difficult to distinct the leg from the wall. In Figure 8n, our approach wrongly fill the hole with texture of the wall. Another important advantage of our approach is the temporary texture consistency of the filled disoccluded regions. For disoccluded areas caused by moving foreground objects, the missing texture is recovered from other frames. The true texture information in other frames is extracted and used to restore the hole areas. To demonstrate the consistency in temporal direction, a series of magnified virtual image subsection for "Ballet" sequence is shown in Figure 9. The disoccluded regions around the woman of adjacent frames are restored by the same true background structure, then the texture of filled image areas maintains consistent in time direction.

7.2 Objective comparison

We adopt peak-signal-to-noise ratio (PSNR) and SSIM [37] to compare the performance of proposed approach with the other three methods.

For every test case of each sequence, the PSNR and SSIM values are calculated for the whole image region of every virtual image frame. The mean values of PSNR

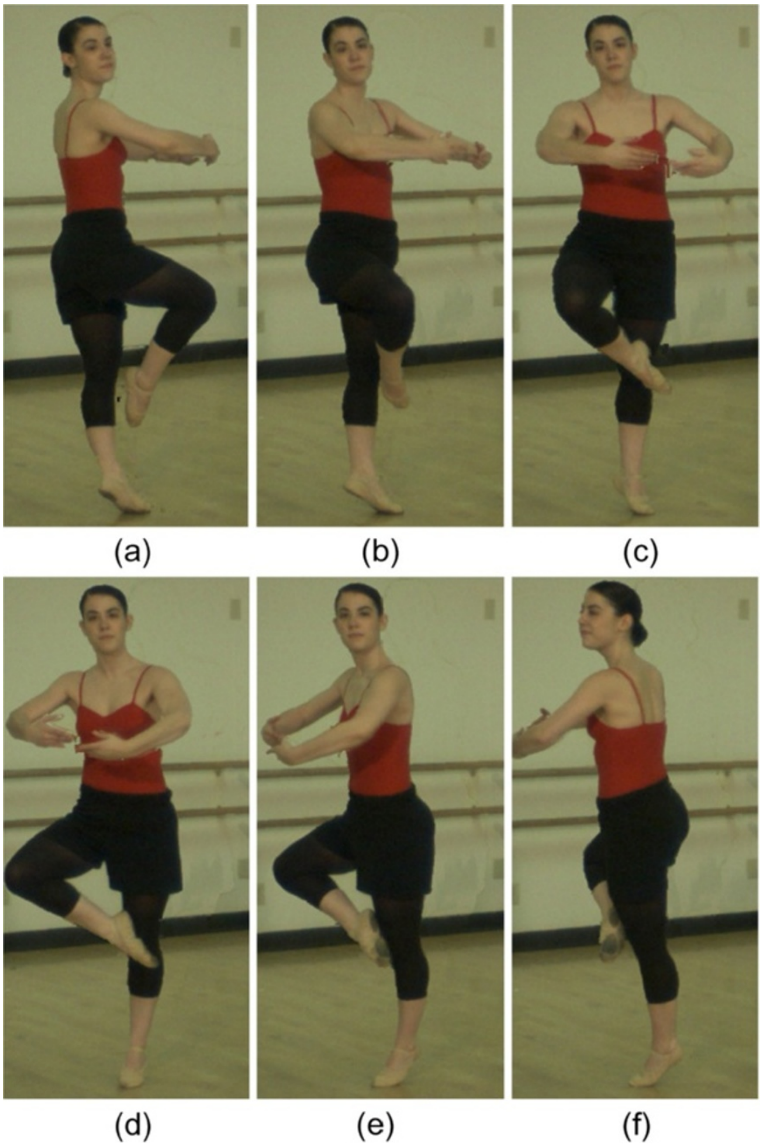

Figure 9 Temporal texture consistency in the disoccluded regions of the proposed algorithm. The DIBR results of our proposed approach for "Ballet" sequence from the 5th camera to the 4th camera. The enlarged same image regions from **(a)** the 56th frame, **(b)** the 57th frame, **(c)** the 58th frame, **(d)** the 59th frame, **(e)** the 60th frame, and **(f)** the 61st frame.

Table 2 PSNR and SSIM results

Seq.	Camera	PSNR (dB)				SSIM			
		Prop.	VSRS 3.5 [38]	Depth-based inpainting [29]	Asym. filter [17]	Prop.	VSRS 3.5 [38]	Depth-based inpainting [29]	Asym. filter [17]
Book arrival	8 → 9	**32.91**	32.85	32.69	28.85	**0.9814**	0.9803	0.9798	0.9500
	10 → 9	**32.12**	31.55	32.03	28.24	**0.9817**	0.9766	0.9811	0.9465
	8 → 10	**29.74**	29.50	29.53	28.85	**0.9672**	0.9647	0.9645	0.8876
	10 → 8	**28.92**	28.57	28.79	25.00	**0.9684**	0.9628	0.9660	0.8909
Break dancers	4→3	**31.91**	30.13	31.45	26.56	**0.9470**	0.9323	0.9440	0.8832
	3→4	**31.94**	29.67	31.78	26.70	**0.9518**	0.9357	0.9500	0.8848
	5→4	**32.58**	28.74	32.06	27.16	**0.9532**	0.9340	0.9503	0.8867
	5→6	**32.47**	31.51	32.12	26.59	**0.9503**	0.9441	0.9482	0.8876
Ballet	3→4	**30.10**	28.20	29.63	22.58	**0.9388**	0.9111	0.9288	0.8280
	5→4	**31.91**	26.87	31.85	23.28	**0.9436**	0.9151	0.9403	0.8304
	5→3	**27.74**	24.41	25.92	20.17	**0.8884**	0.8575	0.8820	0.7517
	3→5	**27.38**	25.98	27.10	20.82	**0.8799**	0.8447	0.8683	0.7489

This table shows the PSNR and SSIM values of four view synthesis methods. The best results are highlighted with boldface type.

and SSIM for each test case are stored in Table 2 and the best results are highlighted with boldface type. The "Camera" column indicates camera configuration of virtual view generation, i.e., "8→9" means synthesizing virtual view of the 9th camera's perspective position from the 8th camera.

From Table 2, we can observe that among these four methods the proposed framework has the best PSNR and SSIM performance for both the parallel and toed-in camera configuration. The Asymmetric Gaussian filtering method gets the lowest PSNR and SSIM values due to the geometric distortion. For the four test cases of "Book arrival" sequence, the baselines between the virtual view and reference view are small (6.5–13 cm). Because the holes around image boundary occupy great percentage of the whole disocclusions (see Figure 6b), the PSNR and SSIM gains of our proposed framework are small, i.e., 0.09–0.22 dB for PSNR and 0.0006–0.0027 for SSIM compared to depth-based inpainting method. For the test cases of "Breakdancers" and "Ballet" with large baseline (20–40 cm), our proposed approach obtains larger PSNR and SSIM gains compared to depth-based inpainting method, i.e., 0.16–1.82 dB for PSNR and 0.0018–0.0116 for SSIM. There are two important reasons for the improvements of PSNR and SSIM in our proposed framework. One is the available structure information from the stationary scene sprite; the other is the oriented exemplar-based inpainting process with reasonable filling orders. Figure 10 shows the PSNR and SSIM curves for two test cases. One is the virtual view of "Ballet" sequence, which is generated from the 3nd camera to the 4th camera. The other is the virtual view of "Breakdancers" sequence, which is generated from the 5th camera to the 4th camera.

Figure 11 gives the PSNR curves for a local area of "Book arrival" sequence. The concerned local area is the same subsection shown in Figure 6n. From the 1st frame to the 31st frame, the local area only covers background objects, so the performance is very close for these three algorithms. From the 32nd frame to the 99th frame, the local area contains not only background objects but also foreground objects. Then the disoccluded regions appear in the concerned local area due to the discontinuity of the depth. With the proposed stationary scene extraction algorithm, the true texture information of the background objects is utilized to recover the disoccluded regions. The temporal consistency of texture and structure is maintained for these frames using our algorithm. Compared to the VSRS and the depth-based inpainting algorithm, the fluctuation of the PSNR values is much smaller for the proposed method (as shown in Figure 11), which means that the temporal consistency of the rendered sequence is improved. It is obvious that the PSNR value drops at the 32nd frame and the 51st frame due to the sudden depth change in the input sequence. To obtain a more consistent rendered sequence, a temporal filtering procedure for the input depth sequence is beneficial.

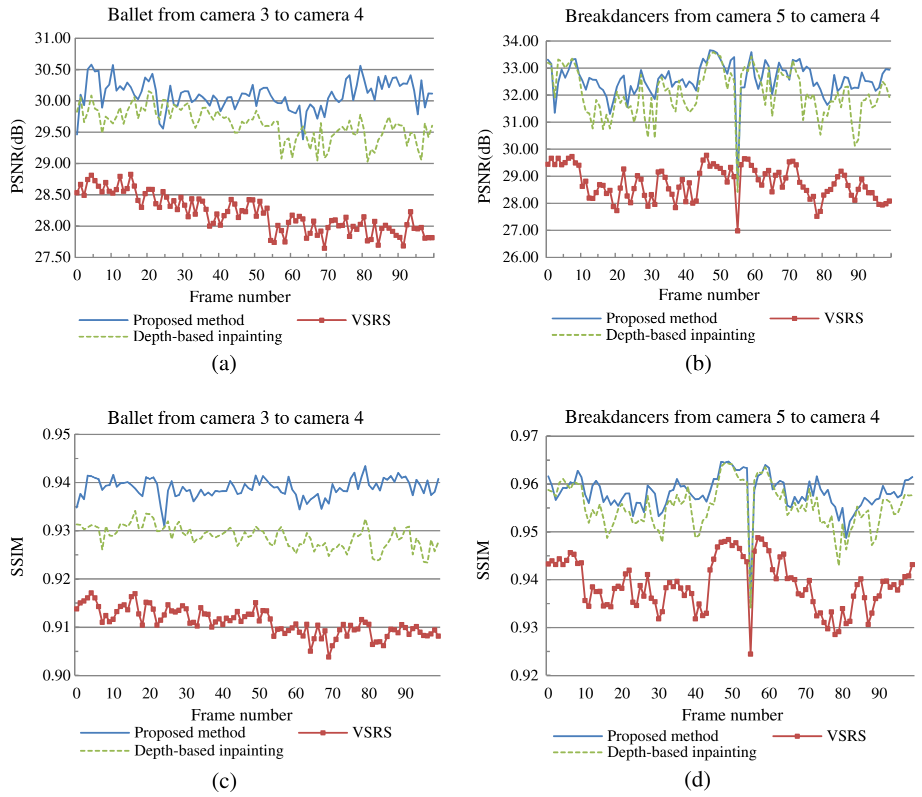

Figure 10 PSNR and SSIM curves for “Ballet” and “Breakdancers” sequences. (a) PSNR curve of “Ballet” from the 3rd camera to the 4th camera. **(b)** PSNR curve of “Breakdancers” from the 5th camera to the 4th camera. **(c)** SSIM curve of “Ballet” from the 3rd camera to the 4th camera. **(d)** SSIM curve of “Breakdancers” from the 5th camera to the 4th camera.

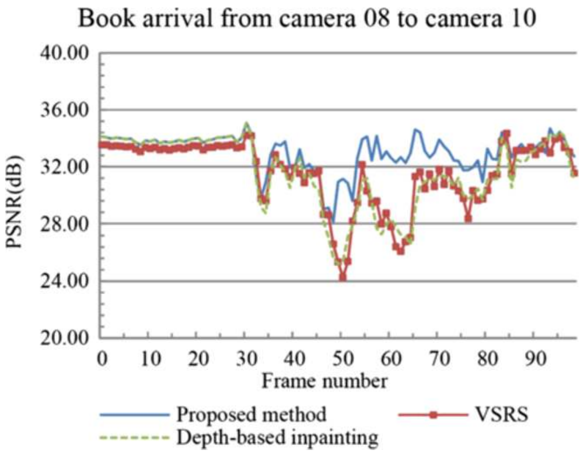

Figure 11 The PSNR curves for a local area of “Book arrival” sequence. The concerned local area is the same subsection shown in Figure 6n. The virtual view is the 10th camera position generated from the 8th camera position.

7.3 Execution time

We implement these four algorithms in C language on a workstation of DELL Corporation and evaluate the run-time costs, as summarized in Table 3. The execution time of each step in proposed framework is given in Table 4.

Table 3 Execution time comparison

Seq.	Runtime (s/frame)			
	Prop.	VSRS 3.5 [38]	Depth-based inpainting [29]	Asym. filter [17]
Book arrival	12.53	1.50	136.93	2.11
Break dancers	11.89	3.92	138.74	1.65
Ballet	31.02	5.32	190.78	3.00

Table 4 Execution time of proposed framework

Seq.	Runtime (s/frame)			
	BG extraction	Backward DIBR + Merge	Oriented inpainting	Total
Book arrival	3.01	3.50	5.85	12.53
Break dancers	3.01	2.89	5.99	11.89
Ballet	2.84	3.99	24.19	31.02

The workstation is equipped with an Intel 2.93-GHz Xeon quad-core CPU and 4-GB DDR2 RAM.

The runtime costs of Asymmetric Gaussian filtering and MPEG method are within 10 seconds per frame. The depth-based inpainting algorithm spends more than 2 min due to the time-consuming iteration operation. The proposed approach takes about 20 s to generate virtual view for each frame. The oriented exemplar-based inpainting process takes most of the time cost for our approach, about 50–80%, as shown in Table 4. The execution time of the oriented exemplar-based inpainting algorithm is depended on the size of disoccluded areas, the image patch size, and the size of searching window. For "Ballet" sequence, because the area of hole regions is larger than the other two test sequences (cf. Figures 7b, 6b, and 8b), the runtime cost increases about 2 times. The additional time cost is acceptable for the improvement in the objective and subjective qualities of virtual view image.

8 Conclusion and future work

This article presents a novel DIBR method combined with spatial and temporal texture synthesis. By maintaining a sprite of stationary scene of the original sequence, the useful structure information can be adopted to restore the missing texture of disocclusions in virtual view images. The remaining disoccluded areas are restored by proposed oriented exemplar-based inpainting approach. The oriented exemplar-based inpainting method fills the rest hole areas from background to foreground and propagates the structure and texture into the blank regions in an appropriate way. Combining these two algorithms, the proposed DIBR method solved the disocclusion problem well and achieved the spatial and temporal consistency. These features make the proposed approach very suitable for extrapolation of virtual view synthesis. Meanwhile, the proposed framework has the flexibility of shifting to the interpolation operation. Theoretical analysis and experimental results show that the proposed method outperforms state-of-the-art view synthesis methods. The increase of runtime cost is moderate and acceptable. Our future work will focus on the research of camera tracking and motion compensation to extend our proposed method to the situation with moving cameras.

Abbreviations

3DTV: three-dimensional television; DIBR: depth-image-based rendering; LDI: layered-depth-image; PDE: partial differential equations; PSNR: peak-signal-to-noise ratio; SSIM: structural similarity index; VSRS: view synthesis reference software.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

Ming Xi would like to thank Yin Zhao for his discussion and suggestion about the backward DIBR algorithm. Ming Xi also would like to thank Menno Wildeboer and Masayuki Tanimoto for their kindly help with the implementations. The authors would like to thank the Interactive Visual Media Group at Microsoft Research and the Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute for providing the "Breakdancers", "Ballet", and "Book arrival" sequences, respectively. This study was supported in part by the National Natural Science Foundation of China (Grant nos. 60802013, 61072081, 61271338), the National High Technology Research and Development Program (863) of China (Grant no. 2012AA011505), the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant no. 2009ZX01033-001-007), Key Science and Technology Innovation Team of Zhejiang Province, China (Grant no. 2009R50003) and China Postdoctoral Science Foundation (Grant no. 20110491804, 2012T50545).

Received: 27 July 2012 Accepted: 23 November 2012
Published: 11 February 2013

References

1. A Smolic, P Kauff, S Knorr, A Hornung, M Kunter, M Muller, M Lang, Three-dimensional video postproduction and processing. Proc. IEEE. **99**(4), 607–625 (2011)
2. C Fehn, in *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV (San Jose, CA, USA, 2004), pp. 93–104
3. Y Morvan, D Farin, PH de With, System architecture for free-viewpoint video and 3D-TV. IEEE Trans. Consum. Electron. **54**(2), 925–932 (2008)
4. A Kubota, A Smolic, M Magnor, M Tanimoto, T Chen, C Zhang, Multiview imaging and 3DTV. IEEE Signal Process. Mag. **24**(6), 10–21 (2007)
5. A Smolic, K Mueller, N Stefanoski, J Ostermann, A Gotchev, G Akar, G Triantafyllidis, A Koz, Coding algorithms for 3DTV—a survey. IEEE Trans. Circuits Syst. Video Technol. **17**(11), 1606–1621 (2007)
6. P Merkle, A Smolic, K Muller, T Wiegand, Efficient prediction structures for multiview video coding. IEEE Trans. Circuits Syst. Video Technol. **17**(11), 1461–1473 (2007)
7. L Onural, T Sikora, Introduction to the special section on 3DTV. IEEE Trans. Circuits Syst. Video Technol. **17**(11), 1566–1567 (2007)
8. C Fehn, in *Proceedings of the Visualization, Imaging, and Image Processing*, vol. 3. A 3D-TV approach using depth-image-based rendering (DIBR) (ACTA Press, Benalmadena, Spain, 2003), pp. 482–487
9. N Greene, M Kass, G Miller, in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. Hierarchical Z-buffer visibility (ACM Press, CA, USA, 1993), pp. 231–238
10. C Zitnick, S Kang, M Uyttendaele, S Winder, R Szeliski, High-quality video view interpolation using a layered representation. ACM Trans. Graph. (TOG). **23**(3), 600–608 (2004)
11. A Smolic, K Muller, K Dix, P Merkle, P Kauff, T Wiegand, in *15th IEEE International Conference on Image Processing*. Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems. (San Diego, CA, USA, 12–15 October 2008) pp. 2448–2451
12. Y Mori, N Fukushima, T Yendo, T Fujii, M Tanimoto, View generation with 3D warping using depth information for FTV. Signal Process.: Image Commun. **24**(1–2), 65–72 (2009)
13. S Zinger, L Do, et al., Free-viewpoint depth image based rendering. J. Visu. Commun. Image Represent. **21**(5–6), 533–541 (2010)
14. J Shade, S Gortler, L He, R Szeliski, in *in Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. Layered depth images (ACM, Orlando, FL, USA, 1998), pp. 231–242
15. S Yoon, Y Ho, Multiple color and depth video coding using a hierarchical representation. IEEE Trans. Circuits Syst. Video Technol. **17**(11), 1450–1460 (2007)
16. Comparative study and recommendations. http://www.3d4you.eu/
17. L Zhang, W Tam, Stereoscopic image generation based on depth images for 3D TV. IEEE Trans. Broadcast. **51**(2), 191–199 (2005)
18. W Chen, Y Chang, S Lin, L Ding, L Chen, in *IEEE International Conference on Multimedia and Expo*. Efficient depth image based rendering with edge dependent depth filter and interpolation. (Amsterdam, Netherlands, 6 July 2005) pp. 1314–1317
19. I Daribo, C Tillier, B Pesquet-Popescu, in *IEEE 9th Workshop on Multimedia Signal Processing*. Distance dependent depth filtering in 3D warping for 3DTV. (Chania, Crete, Greece, 1–3 October 2007) pp. 312–315

20. W Wang, L Huo, W Zeng, Q Huang, W Gao, in *IEEE International Symposium on Intelligent Signal Processing and Communication Systems*. Depth image segmentation for improved virtual view image quality in 3-DTV. (Xiamen, China, 28 November–1 December 2007) pp. 300–303
21. L Wang, X Huang, M Xi, D Li, M Zhang, An asymmetric edge adaptive filter for depth generation and hole filling in 3DTV. IEEE Trans. Broadcast. **56**(3), 425–431 (2010)
22. D Heeger, J Bergen, in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. Pyramid-based texture analysis/synthesis (ACM, Los Angeles, CA, USA, 1995), pp. 229–238
23. Bonet De J, in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. Multiresolution sampling procedure for analysis and synthesis of texture images (ACM Press/Addison-Wesley Publishing Co., Los Angeles, CA, USA, 1997), pp. 361–368
24. J Portilla, E Simoncelli, A parametric texture model based on joint statistics of complex wavelet coefficients. Int. J. Comput. Vis. **40**, 49–70 (2000)
25. G Doretto, A Chiuso, Y Wu, S Soatto, Dynamic textures. Int. J. Comput. Vis. **51**(2), 91–109 (2003)
26. M Bertalmio, G Sapiro, V Caselles, C Ballester, in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. Image inpainting (ACM Press/Addison-Wesley Publishing Co., New Orleans, LA, USA, 2000), pp. 417–424
27. M Bertalmio, L Vese, G Sapiro, S Osher, Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. **12**(8), 882–889 (2003)
28. T Chan, J Shen, Nontexture inpainting by curvature-driven diffusions. J. Vis. Commun. Image Represent. **12**(4), 436–449 (2001)
29. K Oh, S Yea, Y Ho, in *IEEE Proceedings of Picture Coding Symposium*. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video. (Chicago, IL, USA, 6–8 May 2009) pp. 1–4
30. A Criminisi, P Pérez, K Toyama, Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
31. N Komodakis, G Tziritas, Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Trans. Image Process. **16**(11), 2649–2661 (2007)
32. K Patwardhan, G Sapiro, M Bertalmío, Video inpainting under constrained camera motion. IEEE Trans. Image Process. **16**(2), 545–553 (2007)
33. T Shih, N Tang, J Hwang, Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. IEEE Trans. Circuits Syst. Video Technol. **19**(3), 347–360 (2009)
34. C Cheng, S Lin, S Lai, Spatio-temporally consistent novel view synthesis algorithm from video-plus-depth sequences for autostereoscopic displays. IEEE Trans. Broadcast. **57**(2), 523 (2011)
35. M Schmeing, X Jiang, in *IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. Depth image based rendering: a faithful approach for the disocclusion problem (Tampere, Finland, 7), pp. 1–4
36. P Ndjiki-Nya, M Koppel, D Doshkov, H Lakshman, P Merkle, K Muller, T Wiegand, Depth image-based rendering with advanced texture synthesis for 3-D video. IEEE Trans. Multimed. **13**(3), 453–465 (2011)
37. Z Wang, A Bovik, H Sheikh, E Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
38. M Tanimoto, T Fujii, K Suzuki, View synthesis algorithm in view synthesis reference software 2.0 (VSRS2. 0) ISO/IEC JTC1/SC29/WG11 (2008)

doi:10.1186/1687-5281-2013-9
Cite this article as: Xi *et al.*: Depth-image-based rendering with spatial and temporal texture synthesis for 3DTV. *EURASIP Journal on Image and Video Processing* 2013 **2013**:9.