


View synthesis with 3D object segmentation-based asynchronous blending and boundary misalignment rectification

Jing Liu^{1,2}  · Chunpeng Li¹ · Xuefeng Fan^{1,2} · Zhaoqi Wang¹ · Min Shi³ · Jie Yang³

Published online: 18 April 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Numerous depth image-based rendering algorithms have been proposed to synthesize the virtual view for the free viewpoint television. However, inaccuracies in the depth map cause visual artifacts in the virtual view. In this paper, we propose a novel virtual view synthesis framework to create the virtual view of the scene. Here, we incorporate a trilateral depth filter with local texture information, spatial proximity, and color similarity to remove the ghost contours by rectifying the misalignment between the depth map and its associated color image. To further enhance the quality of the synthesized virtual views, we partition the scene into different 3D object segments based on the color image and depth map. Each 3D object segment is warped and blended independently to avoid mixing the pixels belonging to different parts of the scene. The evaluation results indicate that the proposed method significantly improves the quality of the synthesized virtual view compared with other methods and are qualitatively very similar to the ground truth. In addition, it also performs well in real-world scenes.

Keywords 3D object segmentation · Trilateral depth filter · Virtual view synthesis · Boundary misalignment

1 Introduction

Recently, the free viewpoint television (FTV) [1] has attracted considerable attention because of its wide applications, like virtual reality and immersive telecommunication. The virtual view synthesis with multi-view plus depth is one of the most active research areas in FTV. Because of greatly reducing the number of reference views and saving storage space and transmission bandwidth, the depth-based image rendering (DIBR) [2], which renders arbitrary views using neighboring color images and their associated depth maps, has become one of the most efficient methods. However, DIBR suffers from the accuracy of depth map, which usually causes visual artifacts in the synthesized virtual view. Furthermore, the occluded areas in the reference images become visible in the virtual view owing to the change of viewpoint. Additionally, another problem facing DIBR is the how to remove ghost contours caused by the boundary misalignment between the depth map and its corresponding color image. So high-quality virtual view synthesis technique remains an open research topic. The virtual synthesis reference software (VSRS) [3] has been released which is based on bi-directional DIBR. Müller et al. [4] proposed the layered method with image regions marked as reliable and unreliable areas to address the depth discontinuities. Solh and AlRegib [5] proposed the adaptive hierarchical hole-filling approaches to solve the disocclusion problem in the virtual view. Although the existing algorithms have made significant progress, ghost contours and error pixels caused by combining the pixels belonging to different parts of the scene still exist.

Here, we propose a novel virtual view synthesis framework that views the scene as a set of 3D objects. We partition the reference color images and depth maps into different 3D object segments where each segment can be viewed as the projection of a 3D object. Our method uses 3D object seg-

✉ Jing Liu
liujing01@ict.ac.cn

¹ The Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road Zhongguancun, Haidian District, Beijing 100190, China

² University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing 100049, China

³ School of Control and Computer Engineering North China Electric Power University, No.2 Beinong Road, Beijing 102206, China

mentation as the major step. Each pixel is assigned to a unique 3D object segment according to its image coordinates and depth value. The major contributions are as follows:

- I: A bilateral depth filter (TDF) is used to significantly remove ghost contours by rectifying the boundary misalignment between the depth map and color image. TDF ensures that the boundaries of the depth map are adjusted to align with the color edges in the edge-transitional regions of the color image. It reassigned the depth values around the boundaries of the depth map based on the similarity measurement. We incorporate the local texture information, spatial proximity, and color similarity into the TDF as constraints to obtain a reliable similarity matching result. The texture variance and gradient is considered as a 24-dimensional feature vector to represent the difference in local texture structure between matching pixels and enhance the robustness to the noise. Spatial proximity and color similarity information enhances the *self-similar* assumption that neighboring pixels should have similar color and depth values.
- II: A 3D object segmentation-based asynchronous blending strategy is used to avoid the visual ambiguities caused by the inaccuracies in the depth map. The segmentation algorithm [6] is first used to divide the captured scene into approximately regions. Then we regularize each generated region as a 3D object, and partition the reference color images and depth maps into different segments to regard each segment as the projection of a 3D object. The process of 3D object segmentation is advantageous to our asynchronous blending strategy is several ways. We separately warp each 3D object segment from the farthest to the nearest pixels to prevent the background pixels from covering the foreground. Furthermore, we blend each 3D object segment with its corresponding 3D object segment in other reference images to avoid mixing pixels from different physical parts of the scene.

The remainder of this paper is organized as follows: The details of our method are discussed in Sect. 2. While the evaluation results are presented in Sect. 3. Section 4 gives some conclusions for the future work. Note that for notation clarity, in this paper, we focus only on synthesizing the virtual view from stereoscopic views with their associated depth maps. However, our method can easily be generalized to handle virtual view synthesis from multi-view video plus depth.

2 Proposed method

Our method consists of three phases. The pre-processing is the first part, boundary misalignment rectification and 3D object segmentation-based asynchronous blending are the

second and third parts. The input is the depth maps (D_L and D_R) and texture images (I_L and I_R) of the left and right views, and the output is the synthesized virtual view. In the pre-processing, because initial depth maps usually contain noises, we improve the quality of D_L and D_R using the median filter. The boundary misalignment rectification and 3D object segmentation-based asynchronous blending are the main contributions and will be discussed later.

2.1 Boundary misalignment rectification

The depth map and color image have complementary characteristics. The depth map boundary is usually very sharp [7], but the color image usually has an edge-transitional region where the foreground and background colors are mixed. Thus, color intensities smoothly change over the edge transitional region while the depth variations is sharp at the object boundary. The misalignment can be denoted as that the boundary between foreground and background of depth map is not aligned with the edge between foreground and background in the color transitional region that mixed color of foreground and background (see Fig. 1). As a result, the depth map usually assigns incorrect depth values to pixels located in the color edge-transitional regions. Ghost contours in the synthesized virtual view are highly affected by the boundary misalignment. There are two reasons for this. First, foreground color pixels located in the edge-transitional region may be mistakenly assigned to background depth values and then deemed to be background pixels during the 3D warping and hole filling. Second, the background information from neighboring views is used to fill the holes that are typically considered as part of the background of the scene. The same artifacts can occur when background pixels located in the edge-transitional region are mistakenly assigned foreground depth values (Fig. 2a–c).

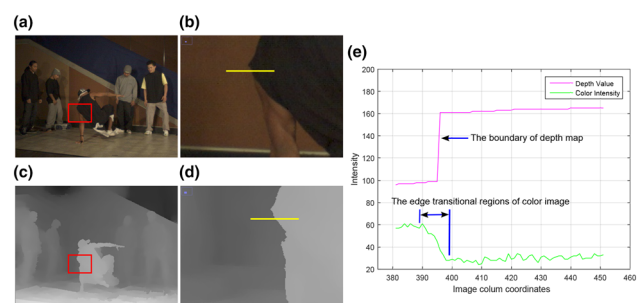


Fig. 1 Basic characteristics of depth map and *color image*. **a**, **c** are the close-up image of the first *color image* frame and its associated depth image of view 4 of the Breakdancers video sequence [2]. **c** The *color* intensities and depth values along the *horizontal yellow line* in **a**, **b**, which spans between the coordinates (460, 380) and (460, 450). The depth map boundary is located in the middle of the *color* edge-transitional region. It is clear that the depth map boundary is not aligned with *color image* edge correctly

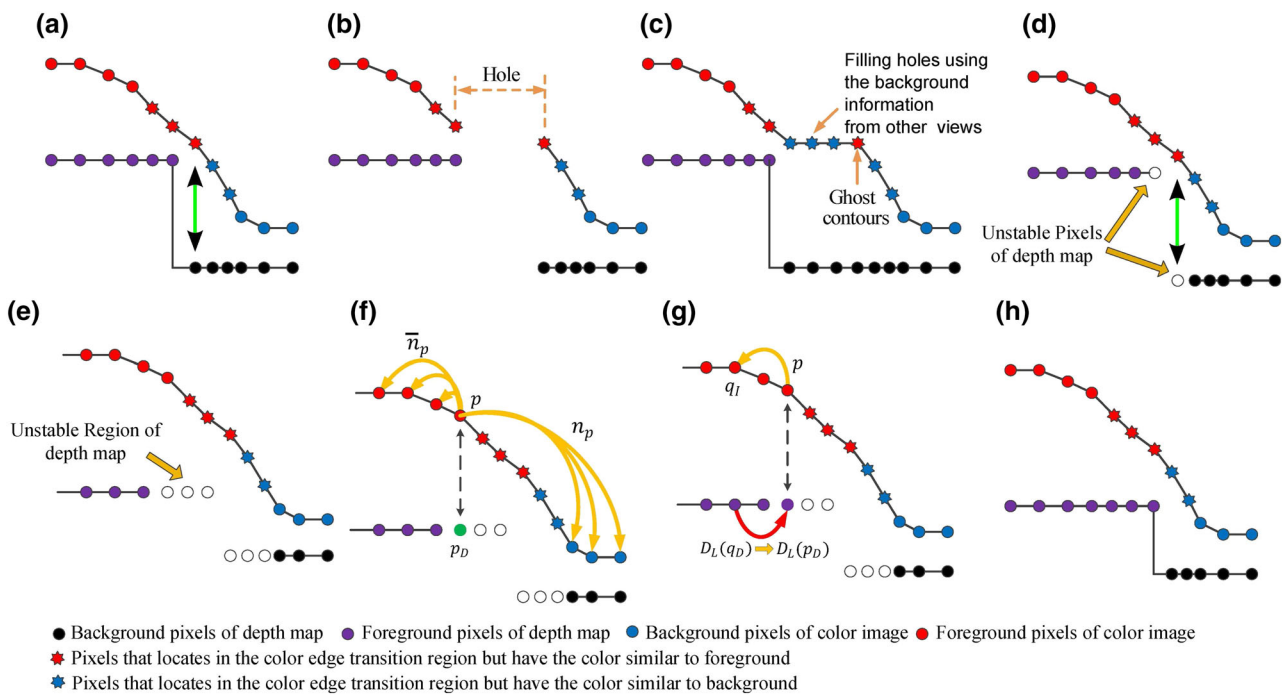


Fig. 2 The conceptual flow algorithm of the boundary misalignment rectification. **a** Pixel in the *color map* mistakenly assigned to a background depth value. **b** Hole in depth map after 3D image warping, which is assumed to be background. **c** Example of ghost contour. The foreground pixel is mistakenly assigned the background depth value and regarded as the background pixel. When the hole is filled using the background information from neighboring views, the ghost contours appear.

d Unstable pixels. **e** Unstable region. **f** Searching p 's most similar pixel along the direction of n_p and \bar{n}_p . **g** The winner-take-all optimization strategy is used to search for the best matching pixel (q_I) of p . And the corresponding depth value (q_D) of q_I is assigned as p 's depth value (p_D). **h** Repeating the optimization iteration until the depth values of all unstable pixels are assigned. It is clear that the misalignment has been removed

To address this problem, we propose a trilateral depth filter (TDF) to rectify the depth boundary to align with the color edge in the edge-transitional region. We take the color image (I_L) and depth map (D_L) of the left reference view as an example to illustrate the boundary misalignment rectification process. First, we determine the boundaries between the foreground and background using the method described in [8]. We classify the depth pixels along sharp boundaries as unstable depth pixels (i.e., they are likely to correspond to erroneous depth values). Unstable pixels may not completely cover the entire misalignment regions because they are located within the color edge-transitional region (where the color changes smoothly). In that case, foreground (background) color pixels located in the edge-transitional region are mistakenly assigned to background (foreground) depth values and may be regarded as background (foreground) pixels causing annoying ghost contours. To enhance the quality of the synthesized virtual view, we use the dilation filter to extend the width of the unstable regions by five pixels. Let \bar{S}_D^L be the regions composed of unstable pixels in D_L , and the stable regions consisting of pixels with known depth values be $S_D^L = D_L - \bar{S}_D^L$. Then, the boundaries between stable and unstable regions are defined as $\delta\bar{S}_D^L$ (see Figs. 2d, e, 3).

During the process of the boundary misalignment rectification, the depth values within unstable regions are

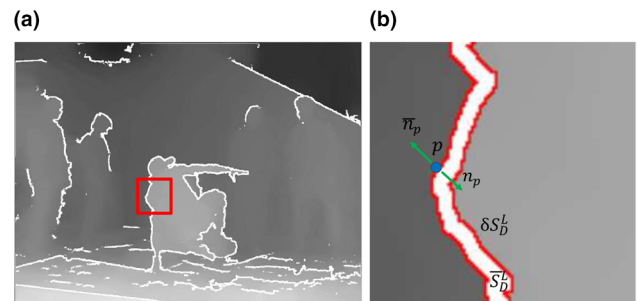


Fig. 3 Unstable regions. **a** Unstable region map. All unstable pixels are marked in white. **b** Is the close-up image of red rectangles in **a**. p is a pixel that locates at $\delta\bar{S}_D^L$. n_p is the unit vector orthogonal to $\delta\bar{S}_D^L$ at p . \bar{n}_p is the reverse vector of n_p , which points to the opposite direction of n_p

reassigned. As shown in Fig. 2f, For each unstable pixel p_D ($p_D \in \delta\bar{S}_D^L$) in depth map D_L ($p_D \in D_L$), we define p_I as the corresponding pixel of p_D in the color reference image I_L ($p_I \in I_L$) (i.e., p_I and p_D share the same pixel coordinates). To reassign the depth value of p_D , we use the TDF to search p_I 's most similar pixel q_I along the direction of n_p and \bar{n}_p in the color image (I_L) according to the texture similarity. This is a simple but very effective way to yield more reliable similarity measurements between two pixels.

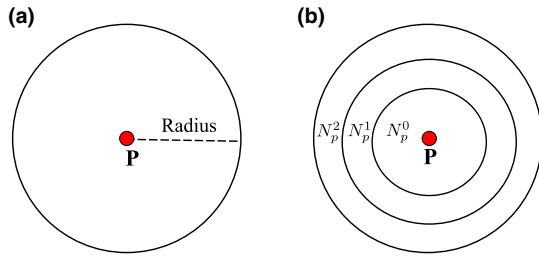


Fig. 4 Surrounding neighborhood patch, N_p , for: **a** pixel p ; and **b** its corresponding sub-regions

For clarity, only stable pixels in S_D^L are considered as matching candidates. The TDF-based similar measure function is denoted as

$$C(p, p_i) = 3 - e^{-\frac{I(p, p_i)}{\gamma_I}} - e^{-\frac{\xi(p, p_i)}{\gamma_\xi}} - e^{-\frac{\rho(p, p_i)}{\gamma_\rho}} \quad (1)$$

$I(p, p_i)$ measures the color similarity of p and p_i . It is denoted as the color difference in the RGB color spaces. We use the texture variance and gradient as cues to represent the similarity of the local texture structure of matching pixels in the color image. It also enhances the robustness to the noise. We define a neighborhood patch N_p (with a radius of 15) centered at p (see Fig. 4). We evenly divide N_p into three annular subregions because the annular spatial histogram is translation and rotation invariant. We compute the normalized intensity eight-bin gray histogram $\Psi_p^i = \{\psi_p^{(i,j)}, i = 0, 1, 2, j = 0 \dots 7\}$ of each subregion N_p^i ($i = 0, 1, 2$) to represent the annular distribution density of N_p as a 24-dimensional feature vector.

$$\xi(p, p_i) = \sum_{i=0}^2 \sum_{j=0}^7 \Psi(\psi_p^{(i,j)}, \psi_{p_i}^{(i,j)}) \quad (2)$$

$$\Psi(\psi_p^{(i,j)}, \psi_{p_i}^{(i,j)}) = \begin{cases} 1 & |\psi_p^{(i,j)} - \psi_{p_i}^{(i,j)}| \geq T_\Psi \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$\xi(p, p_i)$ signifies the local texture similarity between p and p_i (Eq. 2) using the Hamming distance (Eq. 3) between the annular distribution densities of p and p_i . $\xi(p, p_i)$ will be a small value when p and p_i are located at the similar texture regions. $\rho(p, p_i)$ signifies the image coordinate distance between p and p_i . It ensures that two spatially near pixels have a larger support weight. The winner-take-all optimization strategy is used to search for the best matching pixel that gives the minimum matching cost:

$$q_I = \min_{p_i \in \Phi_p} C(p, p_i) \quad (4)$$

Φ_p is a $1 \times N$ patch centered at p and comprises pixels with stable depth values along the direction of n_p and \bar{n}_p in I_L .

The corresponding depth value $D_L(q_I)$ of q_I is assigned as p 's depth value ($D_L(p_D)$) (see Fig. 2g). Notice that in our method, each unstable pixel p is viewed as the stable one if it is reassigned a valid depth value. We repeat the process to extend every boundary pixel belonging to the foreground to background unstable regions until they cover all of the similar color pixels in the color edge-transitional region. Similarly, we extend every boundary pixel belonging to the background to foreground unstable regions until they cover all of the similar color pixels in the color edge-transitional region. The iterations continue until all the unstable pixels are assigned their final corrected depth values (Fig. 2h).

2.2 3D object segmentation-based asynchronous blending

We use the 3D object segmentation-based asynchronous blending to avoid warping the background to the foreground region and mixing the pixels belonging to different physical parts of the scene. We regard the captured scene as a set of 3D objects and assign every pixel to a unique 3D object according to its image coordinates and corresponding depth value. Here, the method described in [6] is used to divide the scene into approximately regions by combining color and depth maps. We regularize each region as a 3D object pertaining to a unique physical part of the scene.

By applying the segmentation algorithm [6] to the depth map of the left view, we can divide the depth map (D_L) and color image (I_L) into different segments. Each segment of $D_L(I_L)$ signifies the 3D object segment that is the unique projection of the 3D object O_i in $D_L(I_L)$ (see Fig. 5g). Meanwhile, we must match the 3D object segments from the left and the right views to each other. Thus, we project each pixel $p \in D_L$ from D_L to D_R using the 3D warping function (Eqs. 5, 6):

$$P_w = R_L^{-1} \cdot \left(Z_L \cdot F_L^{-1} \cdot \begin{pmatrix} u_L \\ v_L \\ 1 \end{pmatrix} - T_L \right) \quad (5)$$

$$Z_R \cdot \begin{pmatrix} u_R \\ v_R \\ 1 \end{pmatrix} = F_R \cdot (R_R \cdot P_w + T_R), \quad (6)$$

where P_w is the 3D space coordinate in the world coordinate system, F is the camera intrinsic parameter matrix, R and T are the rotation and translation vectors, respectively. (u, v) is the image pixel coordinate in the depth map, and Z is the depth value of the pixel located at (u, v) . The subscripts L and R denote the left and right views, respectively. The 3D warping is not a one-to-one mapping function. In the case where multiple pixels map to the same location, we choose the candidate pixel that is closer to the camera with the Z -buffer principle. As shown in Fig. 5e, f, after the 3D warping,

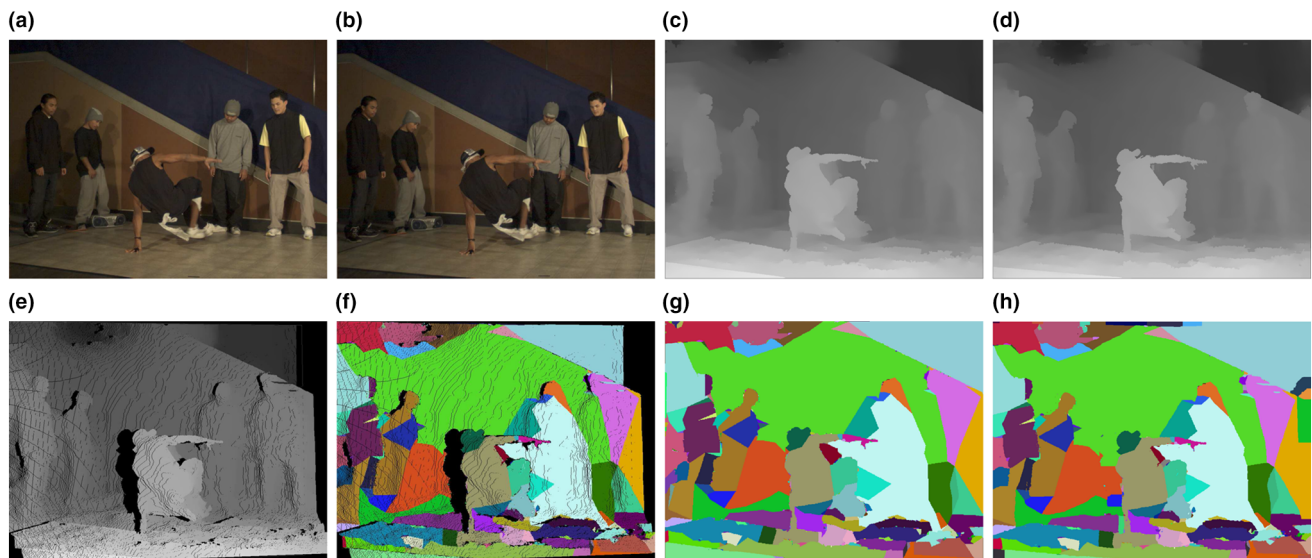


Fig. 5 The results of 3D object segmentation. **a** The *left color image* (I_L). **b** The *right color image* (I_R). **c**, **d** are the corresponding depth maps (D_L and D_R) of I_L and I_R , respectively. **e**, **f** are the initial depth map and the initial 3D object segments map which are projected from *left to right* using the 3D warping function. Unreliable pixels without matching pixels from the *left view* are marked in *black*. **g**, **h** are the 3D

object segmentation results of the *left and right* views. Each *color region* represents a 3D object segment that is viewed as the unique projection from a 3D object to the *left and right* views. The *same color regions* in **(g)** and **(h)** means that they are the matching segments projected from the same 3D object to different views

we assign the 3D object segment label of pixel p in the left view to its matching pixel q in the right view, meaning that they belong to the same physical part of the scene.

However, because we capture the image from different viewpoints, some pixels in the right view are not observed in the left view. Thus, there exist some pixels without matching pixels from the left view (Fig. 5f). Let the pixels in the right view without 3D object segment label be unreliable pixels (\bar{R}) and the others be the reliable pixels (R). We formulate the 3D object segment labelling of unreliable pixels as an optimization problem through the energy function:

$$E(\bar{R}) = E_d(\bar{R}) + E_s(\bar{R}) \quad (7)$$

Equation 7 is a function of the 3D object segment labels. It is minimized using the α -expansion algorithm [9] to obtain the optimal 3D object segment label assignment. The data term E_d measures the likelihood of a particular label hypothesis based on a matching score. It is denoted as the difference between the depth value of pixel $p \in \bar{R}$ and the average depth value D^i of the 3D object segment (O_i) containing p :

$$E_d = \sum_{O_i \in O} \sum_{p \in \bar{R}} \lambda_d \cdot \min\{|D_R(p) - D^i|, \lambda_T^d\}, \quad (8)$$

where λ_d and λ_T^d are constant positive values. $D_R(p)$ is the depth value of p in the depth map. E_d can encourage the assignment of most likely 3D object segment label and ensure that the depth values of each 3D object segment have a

compact distribution that strengthens coherent depth assignments. The smoothness term penalizes neighboring pixels belonging to different 3D object segments:

$$E_s = \sum_{p \in \bar{R}} \sum_{p_i \in N_p^s} \lambda_s \cdot T[O(p), O(p_i)], \quad (9)$$

where $O(p)$ and $O(p_i)$ are the label values of the 3D object segments containing p and p_i , respectively. λ_s is a constant penalty value. N_p^s is a 4 neighborhood system of p in image coordinate. $T[O(p), O(p_i)]$ is the Potts model function that is equal to 0 ($O(p) = O(p_i)$) and 1 ($O(p) \neq O(p_i)$). The aim of the smoothness term is to smooth the depth variation within each 3D object segment. It encourages the *self-similar* assumption that neighboring pixels should come from the same physical part of the scene. We optimize the energy minimization by the standard α -expansion algorithm using pixels as nodes. The label set is composed of 3D object segment labels. During the optimization, we only focus on unreliable pixels. After the optimization, the 3D object segments of all unreliable pixels in the right reference view are obtained, we divide the depth map (D_R) and color image (I_R) into different 3D object segments. Each segment can be viewed as the unique projection of the 3D object in $D_R(I_R)$. We thus match the segments between the left and right views to each other (see Fig. 5h).

Based on the result of the 3D object segmentation, the asynchronous blending strategy applies 3D warping and blending to each 3D object segment independently. It avoids

mixing the pixels belonging to different physical parts of the scene. During the warping, small holes appear in the warped image because of the change in the viewpoint. We fill these holes using a median filter. Given the segmented and warped color and depth images of the left and right reference views, we take the i -th 3D object segment as an example to perform the blending method proposed. Let $D_V^i(p)$ and $I_V^i(p)$ be the depth and color values located at pixel p in the synthesized virtual view, respectively. $D_{vL}^i(p)$ ($I_{vL}^i(p)$) and $I_{vL}^i(p)$ ($I_{vR}^i(p)$) are the depth and color values, respectively, of pixel p belonging to the i -th 3D object segment in the synthesized virtual view projected from the left (right) reference to the virtual view. There are four cases for blending each 3D object segment:

- Case I: The projection from the right reference view to the virtual view is invalid ($D_{vR}^i(p) = 0$) while the projection from the left reference view to the virtual view is valid ($D_{vL}^i(p) \neq 0$) (green points in Fig. 6). $D_{vL}^i(p)$ and $I_{vL}^i(p)$ determine the depth and color values of p in the synthesized virtual view.
- Case II: The projection from the left reference view to the virtual view is invalid ($D_{vL}^i(p) = 0$) while the projection from the right reference view to the virtual view is valid ($D_{vR}^i(p) \neq 0$) (red points in Fig. 6). $D_{vR}^i(p)$ and $I_{vR}^i(p)$ determine the depth and color values of p in the synthesized virtual view.
- Case III: The projections from both reference views to the virtual view are valid ($D_{vL}^i(p) \neq 0, D_{vR}^i(p) \neq 0$), but the difference between $D_{vL}^i(p)$ and $D_{vR}^i(p)$ is larger than

a threshold ($|D_{vL}^i(p) - D_{vR}^i(p)| > T_B$). Based on the Z-buffer principle, if $D_{vL}(p) \leq D_{vR}(p)$, D_{vL} and $I_{vL}^i(p)$ determine the depth and color values of p in the synthesized virtual view (pink points in Fig. 6). On the contrary, $D_{vR}^i(p)$ and $I_{vR}^i(p)$ determine the depth and color values.

- Case IV: The projections from both reference views to virtual view are valid ($D_{vL}^i(p) \neq 0, D_{vR}^i(p) \neq 0$) and the difference between $D_{vL}^i(p)$ and $D_{vR}^i(p)$ is less than a threshold ($|D_{vL}^i(p) - D_{vR}^i(p)| \leq T_B$) (blue points in Fig. 6). By using this blending function, we can consider the baseline spacing of the reference views. The information from the closer camera plays more important role for us and we thus allocate a higher weight to the warped pixel that is closer to the virtual view. $\zeta \cdot D_{vL}^i(p) + (1 - \zeta) \cdot D_{vR}^i(p)$ and $\zeta \cdot I_{vL}^i(p) + (1 - \zeta) \cdot I_{vR}^i(p)$ determine the depth and color values of p in the synthesized virtual view. T_V, T_L , and T_R are the translation vectors of the left reference view, the right reference view, and the virtual view, respectively.

$$\zeta = \frac{|T_v - T_L|}{|T_v - T_L| + |T_v - T_R|} \tag{10}$$

At this stage, we combine all of the blended 3D object segments to obtain the final blended virtual view with the same resolution as the reference views. Because some points are not observed from either the left or the right reference views, some small holes appear in the final blended image. We use the median filter technique to fill these small holes.

3 Experimental results

In this section, a series of quantitative and qualitative evaluations were performed to verify the effectiveness of the proposed method. All experiments were conducted using the Breakdancers and Ballet video sequences that are generated and distributed by the Interactive Visual Group at Microsoft Research [2]. These video sequences were captured by arranging eight cameras along a one-dimensional arc spanning about 20 degrees from one end to the other. Each sequence contains 100 frames of 1024×768 images captured at 15 fps. For the experimental results below, view 3 and view 5 were selected as the left and right views to generate view 4, where the raw data from the view 4 were used as the ground truth. All parameters are presented in Table 1.

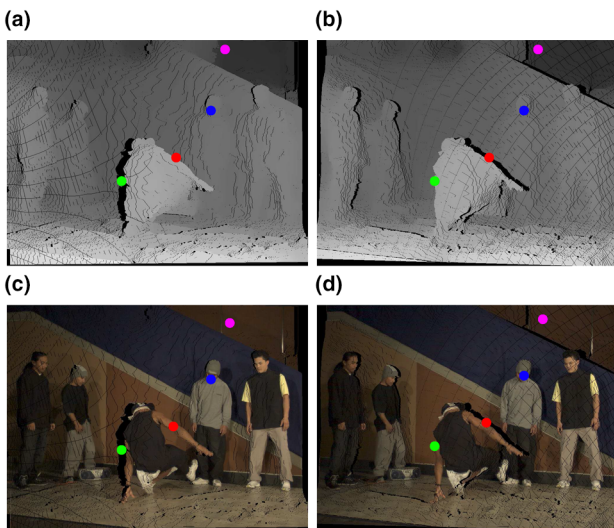


Fig. 6 Four cases for blending each 3D object segment. **a, c** are depth and color images projected from the right reference view. **b, d** are depth and color images projected from the left reference view. Black is the regions without valid values. The brighter value in **a** and **b** means the closer distance

Table 1 Parameter settings for all experiments

| γ_H | γ_ξ | γ_ρ | T_ψ | λ_s | λ_d | λ_T^d | T_B |
|------------|--------------|---------------|----------|-------------|-------------|---------------|-------|
| 45 | 27 | 35 | 0.1 | 100 | 17 | 5 | 3 |

3.1 Quantitative evaluation with existing algorithms

First, we compare the performance of the proposed method with those of the state-of-the-art view synthesis methods. In order to confirm the accuracy of the proposed method, the peak signal-to-noise ratio (PSNR) [10] and the structural similarity (SSIM) [11] index are used as the evaluation metrics. The PSNR is an evaluation function measuring the quality of the synthesized virtual view; a higher value is better than a lower one. The SSIM is used to measure the similarity between two images, a value of 0 means no similarity, while a value of 1 means exact similarity.

As shown in Fig. 7, we randomly selected the synthesized result of the sixth frame of the ballet sequence as an example of the improvement that our algorithm yields. First, we used the virtual synthesis reference software (VSRS) [3] with the traditional depth image-based rendering technique to yield a suitable synthesized result. However, because of the misalignment between the depth map and its associated color image, significant ghost contours occur at boundaries between the foreground and background, and some dark foreground pixels are mistakenly added to lighter background areas (see the green rectangle regions in Fig. 7a). Wolinski et al. [12] used the inter-view consistency-based algorithm to inpaint the disoccluded areas before projection. While it enhances the inter-view consistency and effectively retains object boundaries, it disregards the relationship between each physical part of the scene. This often leads to the warping of

background pixels to foreground regions or the combination of pixels belonging to different physical parts of the scene (see the red rectangle regions in Fig. 7b). The segmentation-based view synthesis algorithm proposed by Loghman and Kim [8] overcomes some of the problems caused by blending pixels from different physical parts of the scene, but it requires the user to specify the number of segmented images and uses the average of thresholds obtained from the multi-level thresholding algorithm to segment the left and right reference views. Because it does not consider the 3D spatial structure of the scene, if the scene is complex and contains many physical parts located at different depth levels, the corresponding segments of the left and right reference views may not suitably match. Thus, their results are degraded by the boundary misalignment between the depth map and color image, often leading to the corona-artificial or ghost contours (see the yellow rectangle regions in Fig. 7c). Solh and AlRegib [5] generated the synthesized view using hierarchical hole filling to enhance the efficiency. But their method only works for a narrow baseline distance. Serious distortions appear as the baseline distance between reference views gradually increases (see the blue rectangle regions in Fig. 7d). As shown in the pink rectangle regions in Fig. 7e, Fukushima et al. [13] improved a blur transfer type of depth image-based rendering. Their method solves some visual artifact problems by performing post-filtering at the virtual image plane assuming that the depth value should vary smoothly inside a region of similar color. However, this method suffers from ghost

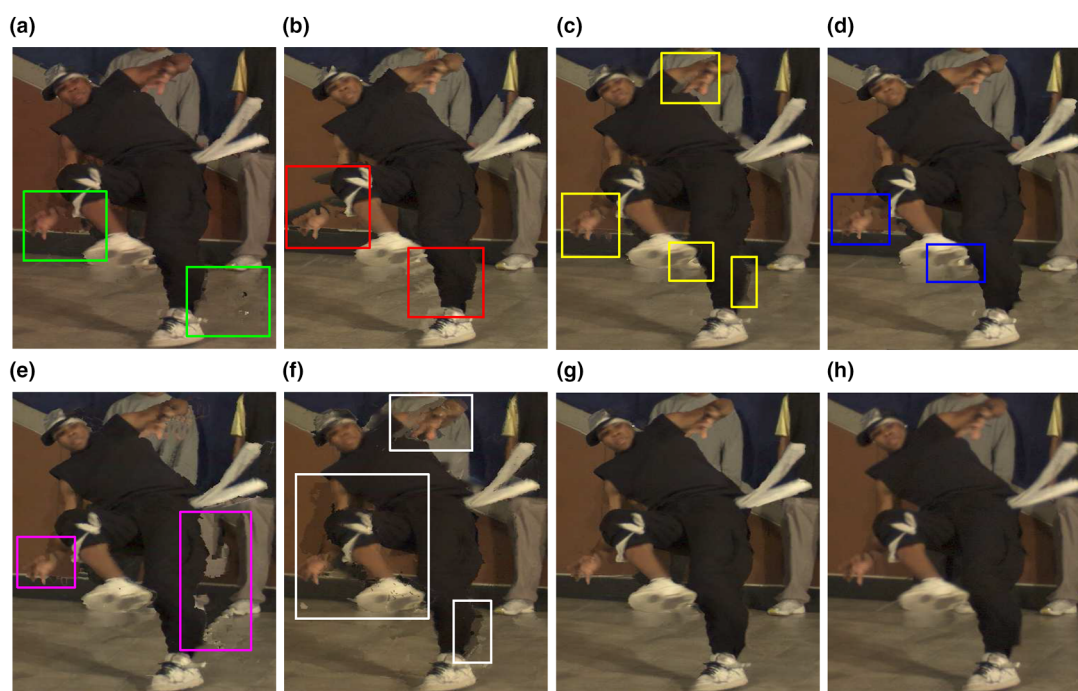


Fig. 7 Visual artifacts in the synthesized virtual view using the enlarged 41th color frame of view 4 of the Breakdancers video sequence [2]. **a** VSRS [3], **b** Wolinski et al. [12], **c** Loghman and

Kim [8], **d** Solh and AlRegib [5], **e** Fukushima et al. [13], **f** Ahn and Kim [14]. **g** Ours. **h** The ground truth. The raw data of 41 color frame from the view 4

contours. These artifacts are primarily caused by the boundary misalignment between the sharp depth map boundaries and their corresponding texture edges in the color images. Ahn and Kim [14] filled the disocclusions using patch-based texture synthesis by considering the robust structure tensor and a new confidence term to enhance the robustness to noise. They chose the best-matching patch in the background regions according to the patch distance measure. Because their method does not consider the connectivity and 3D spatial construction of objects, it produces visual artifacts in the hole area (see the white rectangle regions in Fig. 7f). In the proposed method, we rectify the boundary misalignment to eliminate the ghost contours. Furthermore, we separate the scene into different 3D object segments according to the depth maps and then apply an asynchronous blending strategy to each 3D object segment for the 3D warping and blending to synthesize the virtual view. The proposed method avoids mixing pixels belonging to different physical parts of the scene (Fig. 7g). The results indicate that the proposed method significantly improves the quality of the synthesized virtual view compared with other methods and is qualitatively very similar to the ground truth. Figure 8 shows the PSNR and SSIM comparison of the proposed method with those of

other state-of-art algorithms over 100 synthesized datasets of virtual view 4 for each Microsoft sequence. Meanwhile, we compute the mean and variance of the PSNR and SSIM values in Table 2 over the 100 frames of the synthesized virtual views for each sequence. Based on the above datasets, we can confirm that the mean of our method is superior to that of the other state-of-art methods [3, 5, 8, 12–14], meaning that our results are quantitatively closer to the ground truth, while the lower variance means that our method is temporally stable for a video input.

3.2 Quantitative evaluation with each component

In this section, we evaluate the performance of each part of our proposed scheme. In each experiment, we omit one part of our method and retain the remaining parts. We analyze the average PSNR and SSIM values over 100 frames of synthesized virtual view 4 for Microsoft datasets. First, we omit the boundary misalignment rectification, meaning that the depth map obtained from stereo matching or depth sensors may not correctly align with its corresponding color image. This typically causes distracting ghost contours in the synthesized view (red rectangles in Fig. 9a, b). The

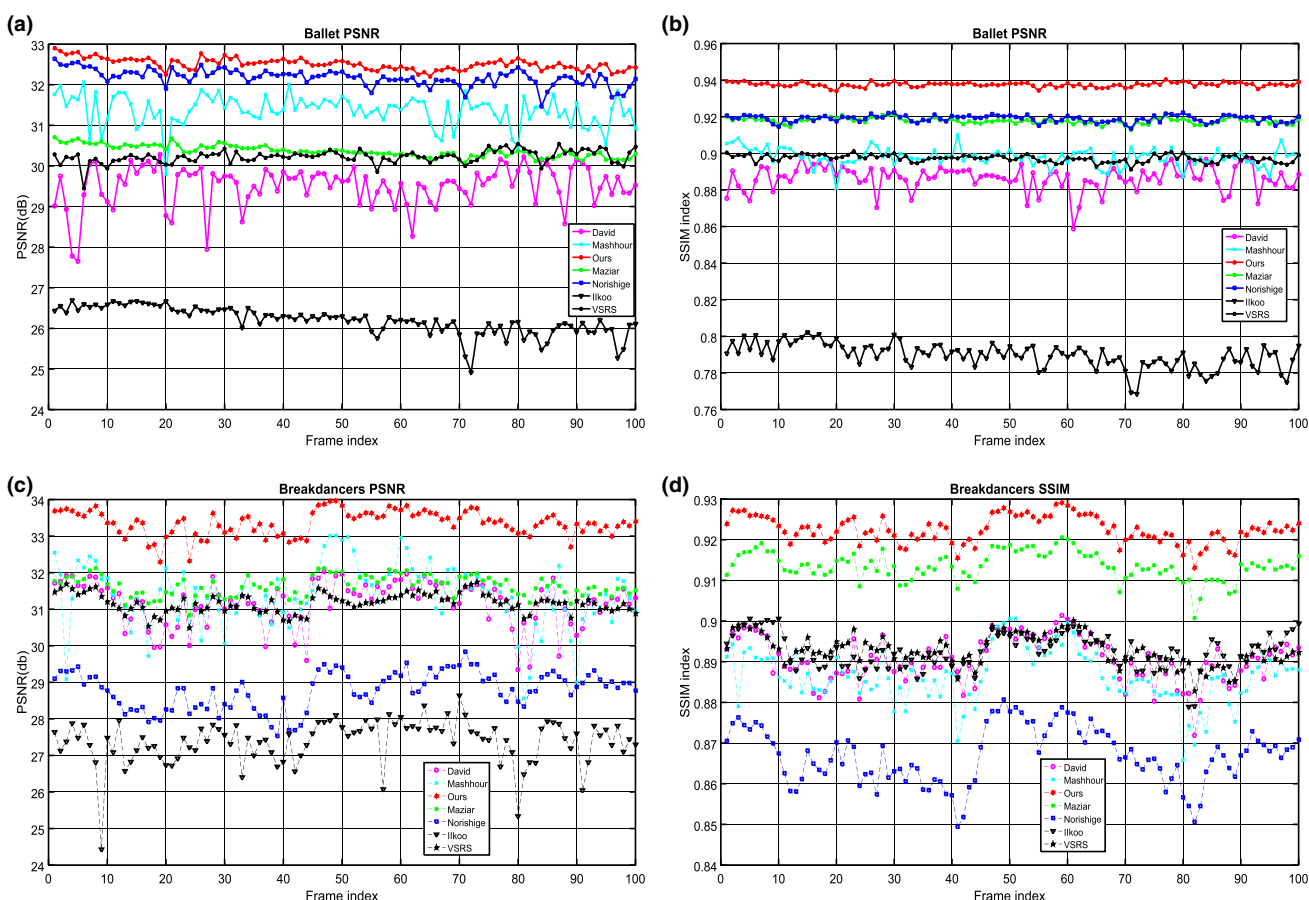


Fig. 8 The PSNR and SSIM distributions obtained from our method with those of other state-of-art algorithms over 100 synthesized datasets of virtual view 4 for the Ballet and Breakdancers video sequence

Table 2 Comparison results on Microsoft datasets by selecting view 3 and view 5 as reference views to generate view 4

| | Breakdancers | | | | Ballet | | | |
|-----------------------|--------------|----------|--------|----------|--------|----------|--------|----------|
| | PSNR | | SSIM | | PSNR | | SSIM | |
| | Mean | Variance | Mean | Variance | Mean | variance | Mean | Variance |
| Solh and AlRegib [5] | 30.82 | 0.8755 | 0.8801 | 0.0061 | 31.57 | 0.3752 | 0.8935 | 0.0048 |
| Fukushima et al. [13] | 30.53 | 0.5045 | 0.8652 | 0.0070 | 32.18 | 0.1995 | 0.9189 | 0.0018 |
| Wolinski et al. [12] | 31.16 | 0.6218 | 0.8906 | 0.0057 | 29.48 | 0.5078 | 0.8868 | 0.0069 |
| Loghman and Kim [8] | 31.64 | 0.2726 | 0.9135 | 0.0035 | 30.36 | 0.1412 | 0.9177 | 0.0016 |
| Ahn and Kim [14] | 27.38 | 0.6045 | 0.8924 | 0.0044 | 26.20 | 0.3323 | 0.7896 | 0.0068 |
| VSRS [3] | 31.17 | 0.2618 | 0.8929 | 0.0037 | 30.23 | 0.1567 | 0.8972 | 0.0020 |
| Ours | 33.33 | 0.2388 | 0.9235 | 0.0025 | 32.52 | 0.1374 | 0.9375 | 0.0012 |

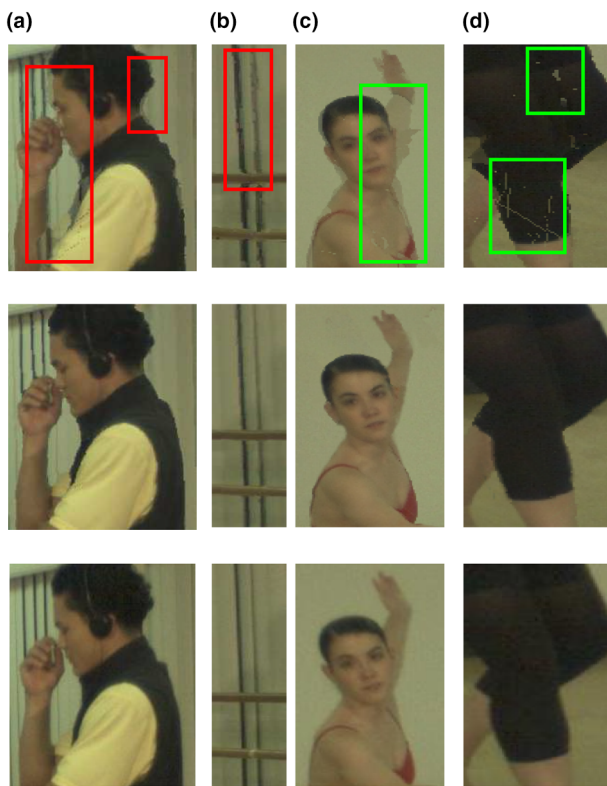


Fig. 9 Taking the first synthesized view frame of view 4 of the Ballet sequence as the example to illustrate the evaluations when turning off some term. **a, b** Results without the boundary misalignment rectification. **c, d** Results without the 3D object segmentation-based asynchronous blending strategy. For each column from up to down is the result with one term turned off, our method with all terms turned on and the ground truth

average PSNR and SSIM values over 100 frames decreased sharply to 30.90 dB and 0.8806 index, respectively, for the Breakdancers sequence, and 29.49 dB and 0.8411 index, respectively, for the Ballet sequence. Next, we omit the 3D object segmentation-based asynchronous blending strategy. In this case, we see that the conventional DIBR technique is easily affected by noise and yields some regions contain-

ing a mixture of foreground and background pixels (green rectangles in Fig. 9c, d). The average PSNR and SSIM values over 100 image frames decrease sharply to 32.94 dB and 0.9142 index, respectively, for the Breakdancers sequence, and 31.60 dB and 0.9035 index, respectively, for the Ballet sequence. It is thus clear that our method obtains the highest average PSNR and SSIM values when all terms are applied.

3.3 Quantitative evaluation with various baseline distances

To investigate the robustness of the proposed method, we use it to synthesize the virtual view for varying baseline distances between the two reference views. We conduct three evaluations with different baseline distances for comparison. In case I, view 3 and view 5 are taken as the reference views to synthesize virtual view 4; the baseline distance between them is 7.69. In Case II, view 2 and view 6 are taken as the reference views to synthesize virtual view 4; the baseline distance between them is 15.20. In Case III, view 0 and view 7 are taken as the reference views to synthesize virtual view 4; the baseline distance between them is 23.13. We compute the PSNR and SSIM values in Table 3 by averaging the results of the proposed method compared with those of the existing state-of-the-art methods [3, 5, 8, 12–14] over 100 frames for each baseline distance. We can see that our method achieves higher average PSNR and SSIM values, outperforming the next best state-of-the-art method, which means that our method generates higher quality results that are robust to various baseline distances.

4 Conclusion

In this paper, we proposed a view synthesis framework to obtain a precise virtual view estimation of a scene. This method was implemented on a PC with Core i5-2500 3.30 GHZ CPU and 4 GB RAM. It took approximately

Table 3 Average values of 100 synthesized results under different baseline distances

| | Breakdancers | | | | | | Ballet | | | | | |
|-----------------------|--------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|
| | 7.69 | | 15.20 | | 23.13 | | 7.69 | | 15.20 | | 23.13 | |
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Solh and AlRegib [5] | 31.38 | 0.8864 | 28.48 | 0.8510 | 27.21 | 0.8146 | 31.48 | 0.8979 | 27.32 | 0.7842 | 26.26 | 0.7138 |
| Fukushima et al. [13] | 28.81 | 0.8674 | 30.52 | 0.8652 | 28.83 | 0.8674 | 32.17 | 0.9189 | 28.65 | 0.8439 | 22.73 | 0.6647 |
| Wolinski et al. [12] | 31.18 | 0.8916 | 29.98 | 0.8654 | 28.04 | 0.8270 | 29.48 | 0.8864 | 25.66 | 0.7550 | 24.67 | 0.6975 |
| Loghman and Kim [8] | 31.64 | 0.9135 | 31.31 | 0.8970 | 29.44 | 0.8595 | 30.36 | 0.9177 | 29.12 | 0.8629 | 23.14 | 0.6647 |
| Ahn and Kim [14] | 27.37 | 0.8924 | 28.37 | 0.8931 | 27.53 | 0.7967 | 26.19 | 0.7896 | 23.66 | 0.6467 | 19.72 | 0.5596 |
| VSRS [3] | 31.17 | 0.8929 | 29.88 | 0.8649 | 27.15 | 0.8213 | 30.27 | 0.8972 | 28.43 | 0.8235 | 23.24 | 0.6790 |
| Ours | 33.33 | 0.9235 | 32.62 | 0.9047 | 30.10 | 0.8691 | 32.52 | 0.9375 | 29.39 | 0.8816 | 27.49 | 0.8100 |

Maximum value in each column is bold

0.3 s to synthesize each virtual view on the Middlebury data. Our major contributions are the boundary misalignment rectification and 3D object segmentation-based asynchronous blending strategy. The evaluation results show that our method can obtain satisfy results. In the further, we intend to transform our method to a parallel GPU implementation.

Acknowledgments This work is supported and funded by the National Natural Science Foundation of China (No. 61300131), the National Key Technology Research and Development Program of China (No. 2013BAK03B07).

References

- Liu, J., Li, C., Mei, F., Wang, Z.: 3D entity-based stereo matching with ground control points and joint second-order smoothness prior. *Vis. Comput.* **31**(9), 1253–1269 (2015)
- Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. In: *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 600–608. ACM (2004)
- Software for view synthesis. <http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/mpeg2/VS.htm>
- Müller, K., Smolic, A., Dix, K., Kauff, P., Wiegand, T.: Reliability-based generation and view synthesis in layered depth video. In: *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pp. 34–39. IEEE (2008)
- Solh, M., AlRegib, G.: Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video. *Sel. Topics Signal Process. IEEE J.* **6**(5), 495–504 (2012)
- Dahan, M.J., Chen, N., Shamir, A., Cohen-Or, D.: Combining color and depth for enhanced image segmentation and retargeting. *Vis. Comput.* **28**(12), 1181–1193 (2012)
- Liu, M.Y., Tuzel, O., Taguchi, Y.: Joint geodesic upsampling of depth images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 169–176 (2013)
- Loghman, M., Kim, J.: Segmentation-based view synthesis for multi-view video plus depth. *Multimed. Tools Appl.* **74**(5), 1611–1625 (2015)
- Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *Pattern Anal. Mach. Intell. IEEE Trans.* **23**(11), 1222–1239 (2001)
- Peak signal-to-noise ratio. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *Image Process. IEEE Trans.* **13**(4), 600–612 (2004)
- Wolinski, D., Le Meur, O., Gautier, J.: 3D view synthesis with inter-view consistency. In: *Proceedings of the 21st ACM International Conference on Multimedia*, pp. 669–672. ACM (2013)
- Fukushima, N., Kodera, N., Ishibashi, Y., Tanimoto, M.: Comparison between blur transfer and blur re-generation in depth image based rendering. In: *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2014, pp. 1–4. IEEE (2014)
- Ahn, I., Kim, C.: A novel depth-based virtual view synthesis method for free viewpoint video. *Broadcast. IEEE Trans.* **59**(4), 614–626 (2013)



member of China Computer Federation.



Jing Liu was born in 1985 in the Hebei Province, China. He is a Ph. D. student at the Virtual Reality Laboratory, Institute of Computing Technology, Chinese Academy of Sciences. He received his B.S. degree in Communications Engineering in 2008 and M.S. degree in Measurement Technology and Automation Device in 2011 from the Hebei University. His research interests include computer vision, image processing, and virtual reality. He is a member of China Computer Federation.

Chungpeng Li was born in the 1980 in Henan Province, China. He received his Ph.D. in 2008 and is now an associate researcher at the Virtual Reality Lab of Institute of Computing Technology, Chinese Academy of Sciences. His main research interests are virtual reality and computer graphics. He is a member of China Computer Federation.



Xuefeng Fan was born in 1990 in the Hubei Province, China. Currently, he is a research assistant and developer at the Virtual Reality Lab of Institute of Computing Technology, Chinese Academy of Sciences. He received his B.S. degree in computer science and technology in 2013 from the Huazhong University of Science and Technology. Image processing and computer vision are his major research fields.



Min Shi was born in 1975 in the Shanxi Province, China. She received her Ph.D. in 2012 and is now an associate professor at the North China Electric Power University. Her main research interests are virtual reality and computer graphics



Zhaoqi Wang was born in 1966 in the Hunan Province, China. He is a researcher and a director of Ph.D. students of the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include virtual reality and computer graphics. He is a senior member of the China Computer Federation.



Jie Yang was born in 1991 in Hebei Province, China. She is an M.S. student at the school of control and computer engineering of the North China Electric Power University. She received her B.S. degree in software engineering in 2014 from the Hebei University