Depth Map Driven Hole Filling Algorithm Exploiting Temporal Correlation Information

Chao Yao, Tammam Tillo, *Senior Member, IEEE,* Yao Zhao, *Senior Member, IEEE,* Jimin Xiao, Huihui Bai, and Chunyu Lin

***Abstract*—The depth-image-based-rendering is a key technique to realize free viewpoint television. However, one critical problem in these systems is filling the disocclusion due to the 3-D warping process. This paper exploits the temporal correlation of texture and depth information to generate a background reference image. This is then used to fill the holes associated with the dynamic parts of the scene, whereas for static parts the traditional inpainting method is used. To generate the background reference image, the Gaussian mixture model is employed on the texture information, whereas, depth maps information are used to detect moving objects so as to enhance the background reference image. The proposed holes filling approach is particularly useful for the single-view-plus-depth format, where, contrary to the multi-view-plus-depth format, only information of one view could be used for this task. The experimental results show that objective and subjective gains can be achieved, and the gain ranges from 1 to 3 dB over the inpainting method.**

***Index Terms*—Depth-image-based-rendering, view synthesis, Gaussian mixture model, foreground depth correlation, 3-D video, hole-filling.**

I. Introduction

3-D Video has emerged with many 3-D movies entering the mass market. In general, for 3-D video, large amount of views are required for providing 3-D depth perception [1], [2]. However, dealing with large number of views is a challenge for the acquisition and transmission process [3], [4]. Consequently, 3-D data format with limited number of views and capability to generate high quality 3-D videos is used to support 3-D video generation.

Depth-Image-Based-Rendering (DIBR) [5] is a technique for generating virtual views with limited 3-D data, which relies on the texture frames and the per-pixel-associated depth maps. DIBR exploits the 3-D geometry of the scene to generate arbitrary virtual views, which are close to the original real view. The 3-D geometry information includes depth maps of the original scene as well as the camera parameters, such as the extrinsic and intrinsic camera parameters. By using the depth information, each pixel in the original view can be projected into the virtual view [6]. In the virtual view generating process, the horizontally rectified camera setup is simplified to the disparities computation between the original view and the virtual view in the horizontal direction.

A critical problem in the DIBR system is how to deal with the hole regions after 3-D projection [7]. The holes could be classified into two categories based on the reason that cause them. One category of these holes is related to the disoccluded regions, in this category holes are generated because the occluded region in the original view become visible in the virtual view after 3-D projection, this happens due to the different depth levels between the foreground objects and the background. This type of hole is demonstrated in Fig. 1 (c), and marked in red. The second category of holes is related to the inaccurate depth values, which in some cases will lead to discontinuities of these values for objects that should not have the discontinuities; this category of holes is marked in yellow in Fig. 1 (c).

Generally, there are two basic solutions to address the disocclusion problem. One type of solutions preprocesses the depth maps before DIBR, aiming to reduce the disparity along the boundary between the foreground and background, so that no disocclusion appears in the virtual view. Based on this rule, *L. Zhang* proposed to smooth the whole original depth map using symmetric Gaussian filter in [8]. Later, in [9], an asymmetric filter was proposed to reduce the vertical edge artifacts compared with the symmetric filter. But in both [8] and [9], the lowpass filters would lead to geometrically distorted foreground objects, especially for the case of *large baseline* with large disparity between the original and the virtual views. With the intention to get rid of the geometry distortion, *Chen et al.* proposed an edge-dependent Gaussian filter to smooth the depth map while the depth on the boundary between the foreground and the background can be persevered [10]. Whereas in [11] *Lee et al.* utilized a lowpass adaptive Gaussian filter on the boundary of the objects and corrected the depth value on the horizontal and vertical direction to avoid distorting the objects. *Cheng et al.* [12] preprocessed the depth maps with a bilateral filter and filled the uncovered areas in the corresponding

Manuscript received November 16, 2012; revised November 7, 2013; accepted March 21, 2014. Date of publication May 16, 2014; date of current version June 4, 2014. This work was supported in part by the 973 Program under Grant 2012CB316400, in part by the National Natural Science Funds for Distinguished Young Scholar under Grant 61025013, in part by the National Natural Science Foundation of China under Grant 61210006, Grant 61202240, Grant 60972085, and Grant 61272051, and in part by the Program for Changjiang Scholars and Innovative Research Team in University.

C. Yao, H. Bai, and C. Lin are with the Institute of Information Science, Beijing Jiaotong University, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China (e-mail: yaochao1986@gmail.com; hhbai@bjtu.edu.cn; cylin@bjtu.edu.cn).

Y. Zhao is with the Institute of Information Science, Beijing Jiaotong University, and also with the State Key Laboratory of Rail Traffic Control and Safety, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn).

T. Tillo and J. Xiao are with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: tammam.tillo@xjtlu.edu.cn; jimin.xiao08@student.xjtlu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2014.2321671

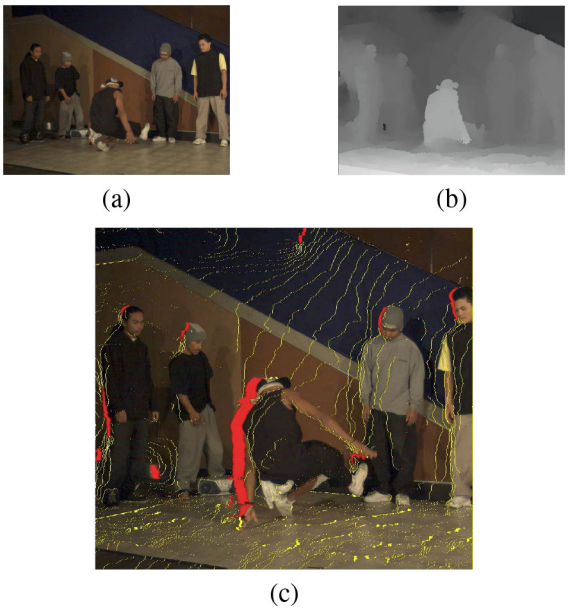

Fig. 1. DIBR results for the second frame of the Breakdancer sequence in camera 1. (a) Original texture. (b) Original depth map. (c) Left warped image, where the red regions represent the disocclusion, and the yellow regions are the holes due to inaccurate depth values.

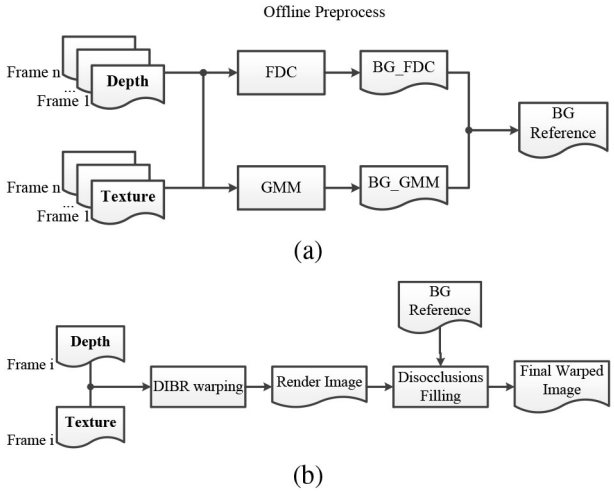

Fig. 2. Proposed framework (BG stands for background). (a) Offline preprocessing step for BG generation. (b) DIBR system with BG for disocclusion filling.

texture image using available background texture information. However, for these approaches, the typical artifacts can still be observed in case of large baseline. Another type of solutions is view synthesis [13] using the Multi-View-plus-Depth (MVD) format. These approaches exploit the fact that the invisible background part in the left view may be visible in the right view [14], [15], and then the disoccluded region in the virtual view warped from the left view can be filled with the background information from the right view. But this type of methods has strict requirement on the alignment between the left view and right view [14], otherwise the artifacts such as the ghost-image would appear in the virtual views. Besides, compared with the Single-View-plus-Depth (SVD) format, much higher bitrate is required for the MVD format, which is not applicable for limited bandwidth applications. Both types of solutions can not generate satisfactory image quality for the transition regions between the foreground and background, especially for the case of large baseline. Furthermore, for these approaches the virtual view is rendered frame by frame, ignoring the temporal correlation of the disoccluded regions, which will cause typical flickering artifacts in the virtual view. Thereby, *Schmeing and Jiang* [16] proposed to firstly determine the background information in the time domain using a background subtraction method. As a tentative approach, this method highly depends on the performance of the foreground segmentation method, thus it is not applicable for complex scenes. *Chen* [17] proposed to recover the disocclusions in the virtual view using the known regions of temporal neighboring virtual frames by exploiting the motion vector of H.264/AVC bit stream. However, the gain is limited due to lack of accurate motion vector. In [18] and [19], a background sprite is generated by the original texture and synthesized images from the temporally previous frames for disocclusion filling, but the temporal consistency of the the synthesized images need further investigation as described in [20]. In fact, the depth temporal inconsistency will affect the performance of the background update, and that is one of the most serious problem for the disocclusion filling methods that rely on the depth maps. The temporal inconsistency of the depth map is related to the way that it is generated. The depth map sequence could be captured by depth cameras which are based on the principle of time-of-flight, in this case [21], because of the limited accuracy of time measurement and the high velocity of light pulse, the captured depth map is not consistent within the sequence. Whereas, when the depth maps are generated using the depth estimation software, which includes automatic and semi-automatic modes, the generated depth map is also not sufficiently accurate and robust.

In this paper, we propose a disooclusion filling approach based on the *temporal correlation* information for the Single-View-plus-Depth (SVD) format. In the proposed approach, the background information is obtained from both the texture and depth sequences, by using the Gaussian Mixture Model (GMM) and Foreground Depth Correlation (FDC). On one hand, by using GMM, a temporally stable background sprite can be acquired. On the other hand, the FDC method is used to identify the covered background regions in different frames by detecting the movement of foreground regions. Finally, the obtained background image is used for disocclusion filling in DIBR system.

The rest of this paper is organized as follows. The overall framework is presented in Section II. The background generation process is described in Section III, and the disocclusion filling approach is in Section IV. In Section V, the experimental results are presented. Finally, conclusions are given in Section VI.

II. Proposed Framework

The objective of the proposed approach is to fill the disocclusion by using a temporal stable background information for the SVD format, where only one texture sequence and its depth sequence are available. In this section, the framework

(a) (b)

Fig. 3. Example of BG reference frame by using GMM. (a) Original texture frame from *Outdoor* sequence. (b) BG reference frame obtained by using GMM.

of the proposed approach is outlined in Fig. 2. This approach is based on the observation that most of occluded regions belong to the background which are covered by the foreground objects, and these occluded regions might become visible in other frames, due to the foreground movement. Thus, in the proposed approach, we generate *a temporal stable background image* in offline mode, then this image is used to fill the disocclusion regions in the DIBR system. As shown in Fig. 2 (a), in the proposed approach, an *offline preprocessing step* is used to generate a background image, by using several consecutive texture frames. In this stage a stable background image can be generated with the Gaussian Mixture Model (GMM), where the regions covered by the moving foreground objects are replaced by the temporal "stable pixels". In most cases, the temporal stable pixels belong to the background, especially for the covered regions by the foreground objects with translational motion. An example of that is shown in Fig. 3, where the background information covered by the moving car, could be recovered by using GMM method. However, in some cases, this process may blur the moving regions, especially for foreground objects with reciprocal motion, such as the one seen in Fig. 4 (a). In this scene, the dancer is rotating, and consequently most of foreground information are mistakenly modeled as background by using GMM. Therefore, the movement of foreground objects will be detected in the Foreground Depth Correlation (FDC) stage to help recover the background information. With the combination of GMM and FDC, a background image can be obtained.

This background information can be used during disocclusion filling in DIBR system. Obviously, using GMM and FDC can only help recovering the background regions occluded by the moving objects. Therefore, in the proposed disocclusion filling approach, the disocclusion along the static foreground objects, will not be updated using the background infromation, but using the conventional inpainting method [22]. Moreover, for some small disocclusion or holes caused by discontinuity of depth value, the inpainting method will be used for these regions filling. The details of each step will be described in the following sections.

III. Background Generation

A. Background Generation With GMM

The Gaussian Mixture Model is a commonly used method to detect the moving objects [23], and in the computer vision field it has been widely applied to model the stable background. Different from the methods based on block matching, the GMM is performed at pixel level, where each pixel is modeled independently by a mixture of K Gaussian distributions (a common setting is $K = 3$) [24], [25]. The Gaussian mixture distribution with K components can be written as:

$$p(x_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta\left(x_t, \mu_{i,t}, \sigma_{i,t}^2\right) \tag{1}$$

where $p(x_t)$ indicates the probability density of pixel x_t, η is the Gaussian function with x_t representing the pixel value at time t, $\mu_{i,t}$ and $\sigma_{i,t}^2$ denote the mean and variance of pixel x_t, respectively, and $\omega_{i,t}$ is the ith Gaussian distribution's weight, with $\sum_i^K \omega_{i,t} = 1$.

The detailed process of GMM that generates the stable reference background is described as follows [26]:

1) Firstly, an empty set of models is initialized at the time instant t_0.
   - The mean value μ_{i,t_0} of the first Gaussian model is set equal to the pixel value of the current frame, and that of the other models is set to 0.
   - The variance value σ_{i,t_0} of all the K Gaussian models are set to a pre-defined large value, e.g., 30 in this paper.
   - The weight value of the first Gaussian model ω_{1,t_0} is set to 1, and that of other models is set to 0.
2) For the next frame at the time instant t_1, the current pixel is used to match with the K Gaussian models. Then, for each model i, the condition $|x_t - \mu_{i,t-1}| \leq 2.5\sigma_{i,t-1}$ will be examined.
   - If the condition is satisfied, the matching process will be stopped and all the parameters of the Gaussian models will be updated using the following role:
     - The mean value of the matched Gaussian model, i.e., the ith model becomes, $\mu_{i,t} = (1-\rho)\mu_{i,t-1} + \rho x_t$, where $\rho = \alpha \cdot \eta(x_t, \mu_{i,t}, \sigma_{i,t}^2)$, the α is the learning rate, which is set to 0.005 [26].
     - The variance value of the matched Gaussian model, $\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(x_t - \mu_{i,t})^2$.
     - The weight value of the matched Gaussian model, $\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha$.
     - The mean and the variance of the other Gaussian models remain unchanged, while the corresponding weight value is updated, $\omega_{i,t} = (1-\alpha)\omega_{i,t-1}$.
   - Whereas, if all of the Gaussian models fail to match the current pixel, then a new Gaussian model is introduced with $\mu = x_t$, high σ^2 (e.g., $\sigma = 30$) and a low weight value $\omega = 0.001$ by evicting the Gaussian model which has the smallest ω/σ value.
     - the mean and variance value of the other Gaussian models remain unchanged;
     - the weight value of K Gaussian models are normalized to $\sum_i^K \omega_{i,t} = 1$.
3) The remaining frames are processed by repeating the previous step (2).

(a) (b) (c)

Fig. 4. Background reference from GMM and FDC. (a) Reference background with GMM. (b) Occluded regions which are recovered by FDC. (c) Final background reference with both GMM and FDC.

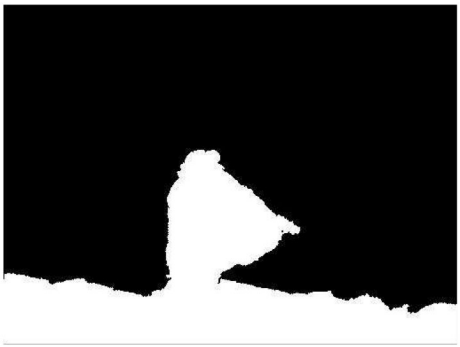

Fig. 5. Background–foreground segmentation using the second frame of the Breakdancer sequence in camera position 1; the white regions represent the foreground and the black regions represent the background.

Finally the K Gaussian models are sorted based on ω/σ, and the value of the background pixel at the time instant t is $\mu_{1,t}$ [23], this process will ensure the most stable pixels on time domain will be modeled as the background. One example of the temporal stable background reference generated by the GMM method is shown in Fig. 3 (b).

B. Background Update With FDC

In the process of background modeling in GMM, pixels with temporal stable intensity will be modeled as background pixels. However, if the motion of the foreground object is reciprocal motion (e.g., rotation movement), it happens that the foreground object occlude the background in most of frames. Thereby the GMM erroneously regards this foreground object as part of the stable background. In this case, the GMM will fail to recover the occluded background information. One example is shown in Fig. 4 (a), where the body of the ballet dancer is incorrectly regarded as background. In this case, the GMM method cannot generate a satisfactory background.

The problem mentioned in the above paragraph could be solved by exploiting the depth map information. In 3-D video data, the depth value describes the distance between the objects and the viewer, hence, depth value could be used to coarsely distinguish the foreground and background. To do this, in this work the k-means cluster will be used [27]. The k-means cluster, aims to partition all pixels into k clusters in which each pixels belongs to the cluster with the nearest mean. As shown in Fig. 5, by using k-means cluster ($k = 2$), the whole of depth map can be coarsely classified into two classes, where the white regions are the foreground and the black regions are the background. At this stage, the occluded background can be updated by detecting the change of the foreground regions among several consecutive depth maps. The initial foreground regions are firstly generated for the first frame of the sequence and it is denoted as Ω_1; an example of the initial foreground is shown in Fig. 6 (a) with white color. Then the same process is repeated for the following frame, the outcome of this is shown in Fig. 6 (b) and it is represented by Ω_2. Then comparing the differences between the first mask and the second mask, the moving regions between these two frames will be obtained [the white part in Fig. 6 (c)]. With the help of the texture of the second frame, some occluded regions in the first frame can be recovered by using the previously obtained mask, as shown in Fig. 6 (d). The same method will be applied to process the following frames, and this will allow recovering the majority of the occluded regions. Due to the fact that this process exploits the correlation of the foreground regions in the depth maps, we called this process as Foreground Depth Correlation (FDC) method.

It is worth mentioning that, if the foreground objects are close to the background, then the depth values of the foreground objects will be relatively similar with the background values. In this case, using the k-means to classify the depth values may not generate satisfactory results, as it is difficult to classify the foreground pixels due to the similar depth values. For example in Fig. 6 (f), some foreground objects are close to the background, these regions are mistakenly classified as background regions in the k-means ($k = 2$) method, those regions are marked with red circles. Nevertheless, in such situation, the depth difference between the foreground and background is very small. Therefore, after 3-D warping, the disocclusion region should be also very small [as shown in Fig. 6 (g), the green regions around the foreground objects]. To fill these holes, the conventional hole-filling method, such as inpainting, would be enough.

Finally, with the combination of GMM and FDC, a background image can be obtained. The details of the background image generation is described as Algorithm 1.

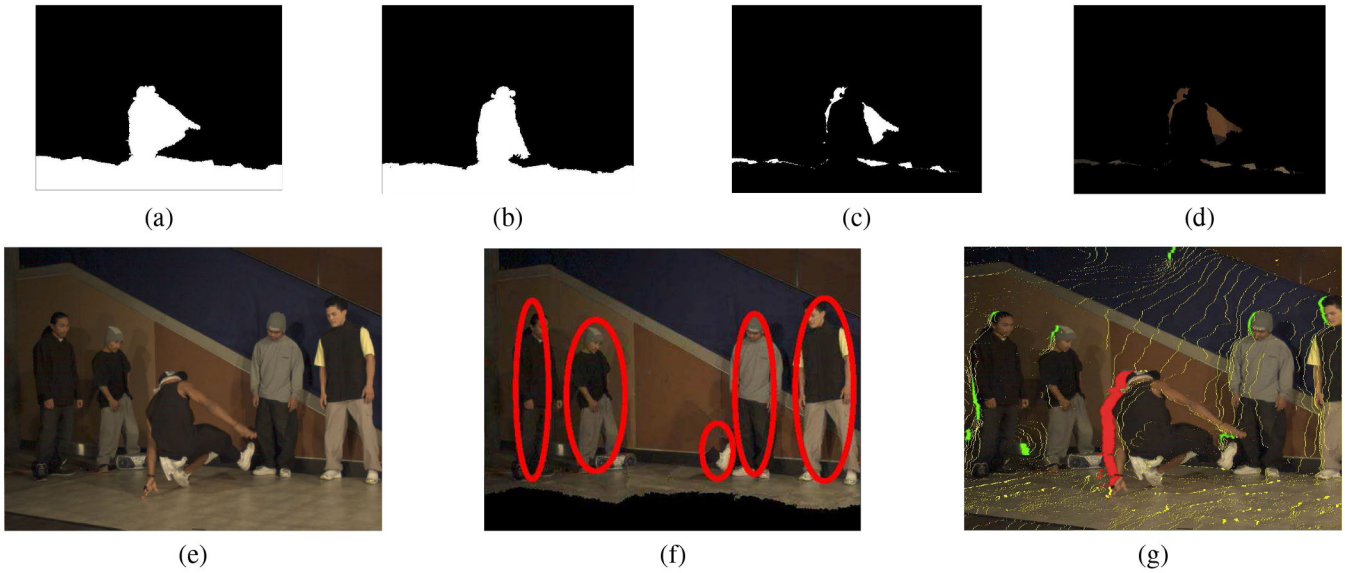

Fig. 6. FDC method used with the *Breakdancer* sequence. (a) Foreground region in first frame. (b) Foreground region in second frame. (c) Visible background region in second frame compared with first frame. (d) Recovered background region. (e) Original texture of the second frame in camera position 1. (f) Result of depth map classification based on *k*-means; the red circle denotes the foreground regions which are mistakenly regarded as the background regions. (g) Two types of disocclusion regions, the red regions happen in the foreground layer, the green regions are in the background layer.

Algorithm 1 Background Image Generation

GMM background, Ψ_g, generated using GMM, and the final background image is set $\Psi = \Psi_g$;
FDC background, Ψ_f, is initialized as the first texture frame in the sequence as $\Psi_f = T_1$;
foreground regions Ω_1 classified from the depth Frame 1;
$n = 2$;
for each frame **do**
  foreground Ω_n is classified from the depth map of Frame n, the texture Frame n is T_n;
  if (pixel $(i,j) \notin \Omega_n$) $\cap$ (pixel$(i,j) \in \Omega_1$) **then**
    Pixel (i,j) is a background pixel in the Frame n, and $\Psi_f(i,j) = T_n(i,j)$;
  end if
end for
for all pixels in the background image **do**
  if pixel $(i,j) \in \Omega_1$ **then**
    $\Psi(i,j) = \Psi_f(i,j)$;
  end if
end for
The final background reference is Ψ.

IV. Disocclusion Filling

It is important to select the proper region of the background image to fill the disocclusion regions in the rendered images. The right background pixels used to fill the disocclusion regions should be identified. Our idea stems from the fact that the disocclusion regions are almost along the transition areas between foreground and background regions, where the depth level is quite different between these two regions. We use the disocclusion detection method in the view synthesis reference software (VSRS 3.5) [28] to detect the depth discontinuity between the foreground and background regions. The method used to detect the disocclusion is summarized in the following:

$$DisocclusionMask(i,j) = \begin{cases} 1, \text{ if } D(i,j) - D(i-1,j) \\ \quad > \text{threshold, for the} \\ \quad \text{left warped view} \\ 2, \text{ if } D(i-1,j) - D(i,j) \\ \quad > \text{threshold, for the} \\ \quad \text{right warped view} \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

where $D(i,j)$ is the depth value at position (i,j), the threshold is pre-defined value, such as 3∼5. For holes locate in regions marked by 0 the inpainting method [22] will be used. In order to simplify the demonstration, an example of how to use the disocclusion detection method to fill the disocclusion regions in the rendered images, is illustrated graphically in Fig. 7 (the original view is warped to the left). Firstly for simplicity, we assume that the texture and the depth edges are both sharp and well-aligned, and the disparity difference between the adjacent background pixel and the foreground pixel is 3. So suppose that after 3-D warping the background pixels (in Fig. 7, pixels A–C) shift to the right by 21 pixels, whereas the foreground pixels shift to the right by 24 pixels (in Fig. 7, pixels H–J), this will lead to a 3-pixel width hole. Consequently, based on the disparity value of the pixel on either side of this line, the position of the disocclusion regions could be identified. In this case using (2) allows identifying the boundary line along the transition area between the background and foreground regions. This boundary line helps to determine the positions of the background pixels in the reference background frame (pixels A to C in Fig. 7). Finally, the corresponding occluded background information (i.e., pixels D to F) will be used to fill the disocclusion regions in the rendered image. As described in Section III-B, the small disocclusion region [e.g., the green

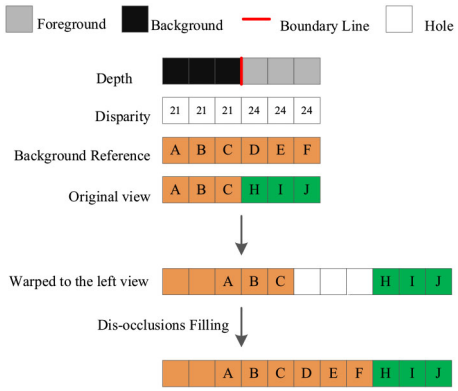

Fig. 7. The relationship between disocclusions and the background reference image.

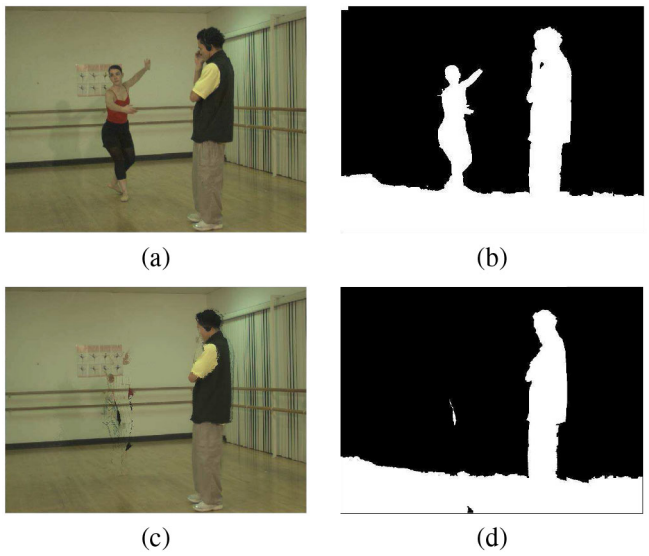

Fig. 8. Detecting static foreground region generation. (a) Original texture image of first frame from Ballet sequence in camera 1. (b) Foreground regions from depth frame 1. (c) Background reference image with GMM and FDC. (d) Static foreground regions after using FDC method.

regions in Fig. 6 (g)] will be filled by using inpainting method.

It is also worth mentioning that, in a general scene, some foreground objects are dynamic (moving), while some are static. So for the regions covered by the moving foreground objects, the recovered background information can be used to fill the disocclusion regions, while for regions covered by the static foreground objects, there is no reliable information in the obtained background image that could be used to fill the disocclusion. However, during the background generation, these static foreground objects will be modeled as background information. An example of this latter case is represented by the standing man in Fig. 8 (c), where the background information behind the standing man cannot be recovered. Therefore, the information in the obtained background image can not be used to fill the disocclusion around these regions. In the proposed approach, for these disoccluded regions, the inpainting method will be used. Whereas, in order to avoid mistakenly filling these disocclusion areas, the occluded regions by the static foreground objects need to be marked in the obtained background image. To this end, the FDC method is also used to mark these static foreground regions before DIBR. By comparing the current frame foreground regions with the previous frame, the changed regions can be recovered by the corresponding texture frame [as shown in Fig. 6 (a) $\sim$ (d)], as they are dynamic foreground regions, while for the unchanged regions, they will be considered as static foreground regions. An example is shown in Fig. 8 (d), where the static foreground regions are marked with white color.

V. Experimental Results

In this section, the experiments that validate the proposed approach will be described. The test sequences include: Microsoft data set [30] *Ballet* (1024 $\times$ 768, 50 frames), *Breakdancer* (1024 $\times$ 768, 50 frames), the baseline between two adjacent cameras is 20cm for these two sequences, whereas the MPEG-3DTV test sequence [31] *Mobile* (720 $\times$ 540, 50 frames) has baseline 5cm. For all, these sequences, there is no scene change. The depth maps of the test sequences are generated with the MPEG depth estimation reference software (DERS) [32] based on graph cuts, and the camera parameters are provided with the sequences.

The proposed method is compared with our previous GMM-based method [29] and the traditional inpaintng method [22]. In case of small baseline, the disocclusion regions in the virtual view will appear as cracks, so the inpainting method or the interpolation method is efficient enough to fill the disocclusion. Whereas in case of large baseline, the horizontal disparities between the original view and the virtual view become so large that the disocclusion regions become large and wide. Thus, the inpainting method will fail to fill these regions perfectly, e.g., in Fig. 9 (a) and (d), the regions along the foreground objects are blured after inpainting with the adjacent information. Whereas, for the *Ballet* sequence, the GMM-based method will fail to recover the occluded background because of the reciprocal motion of the dancing woman. Hence, after disocclusion filling, the double-image effect will reduce the vision quality of the rendered image, as shown in Fig. 9 (b) and (e). In Fig. 9 (c) and (f), we can appreciate the improved performance of the proposed method which could effectively suppress or eliminate the disocclusion in the virtual view. For the disocclusion regions along the moving foreground objects, the proposed method can recover the occluded background in the original view and fill the disocclusion regions by the background information. It is worth noticing that, some artifacts in the rendered view, e.g., the crack artifacts in Fig. 9 (c) and (f), are caused by the imperfect depth map information. The depth enhancement techniques, i.e., [14], could handle these artifacts, but in this work, these artifacts will be ignored so as to avoid lossing focus. In order to objectively assess the performance of the proposed approach, both PSNR and SSIM are used. In this paper, to avoid the effect of imperfect depth map which cause some artifacts over the whole image, we just focus on the quality of the disocclusion regions. Thus, the PSNR is evaluated only for the disocclusion regions in the

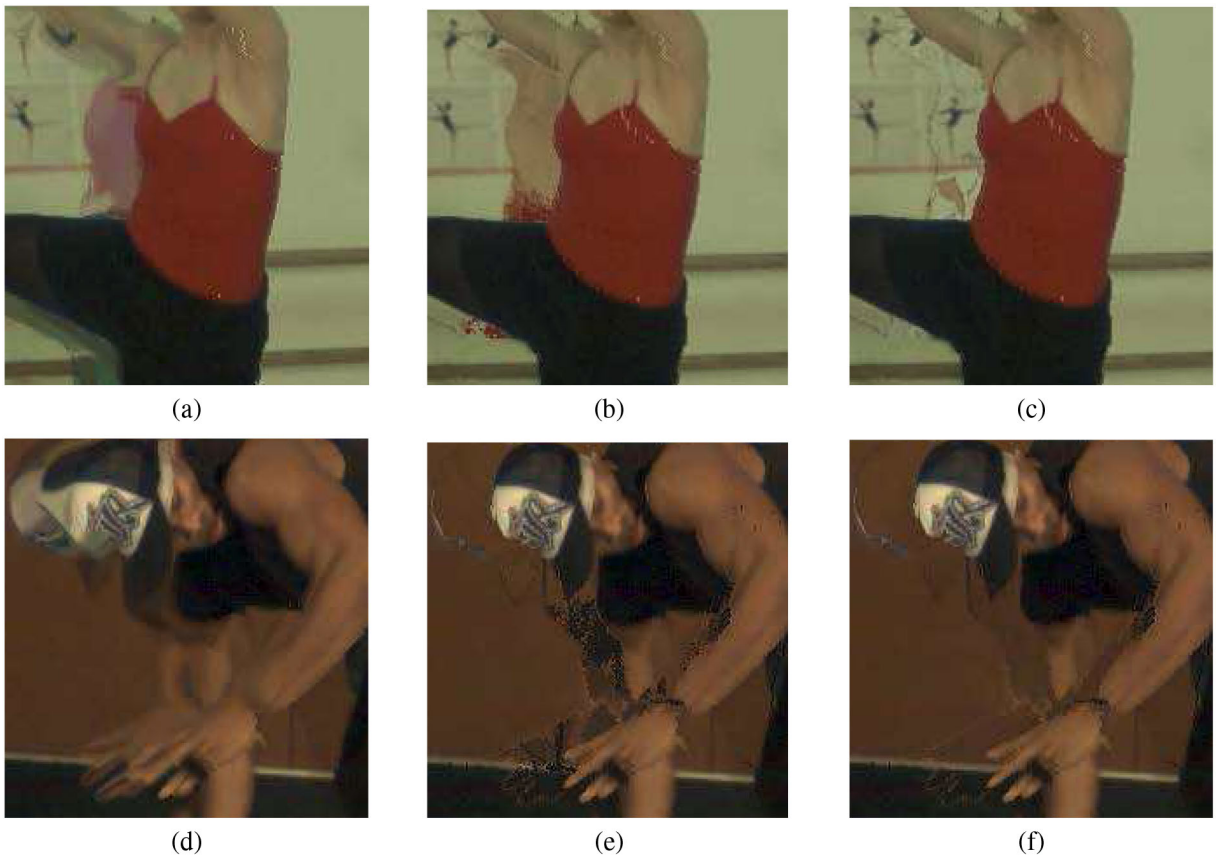

Fig. 9. Some subjective results: (a) and (d) inpainting method [22]; (b) and (e) GMM-based method [29]; (c) and (f) proposed method.

Fig. 10. Disocclusion regions along the moving foreground objects are marked in red: (a) Ballet; (b) Breakdancer; (c) Mobile.

TABLE I
PSNR AND SSIM COMPARISON

Sequence	Camera	Baseline(cm)	PSNR(dB)			SSIM		
			proposed	GMM-based [29]	inpainting [22]	proposed	GMM-based [29]	inpainting [22]
Ballet	01 → 02	20	**32.49**	31.87	31.19	**0.8822**	0.8817	0.8820
	01 → 03	40	**31.11**	30.04	29.37	**0.8822**	0.8732	0.8754
Breakdancer	01 → 02	20	**34.14**	33.75	33.70	**0.8696**	0.8692	0.8696
	01 → 03	40	**32.91**	32.42	31.82	**0.8457**	0.8449	0.8443
Mobile	06 → 05	5	**37.45**	37.09	35.40	**0.9903**	0.9894	0.9902
	06 → 04	10	**36.77**	36.08	33.14	**0.9891**	0.9877	0.9883

image. As shown in Fig. 10, the disocclusion regions marked with red color will be used for comparison. Other regions will not be assessed, since they will use the same inpainting technique [22] in all three approaches, these regions include the artifacts and the disocclusion regions along the static foreground objects. Whereas, to evaluate the geometry distortion of the rendered image, SSIM is evaluated using the whole image as it is not easy to apply it to an arbitrary region. The used reference image to evaluate the PSNR and SSIM is the original sequence at the same corresponding warped position, i.e., in the process of generating the virtual view by warping view 1 to view 2, the original view 2 is used as a

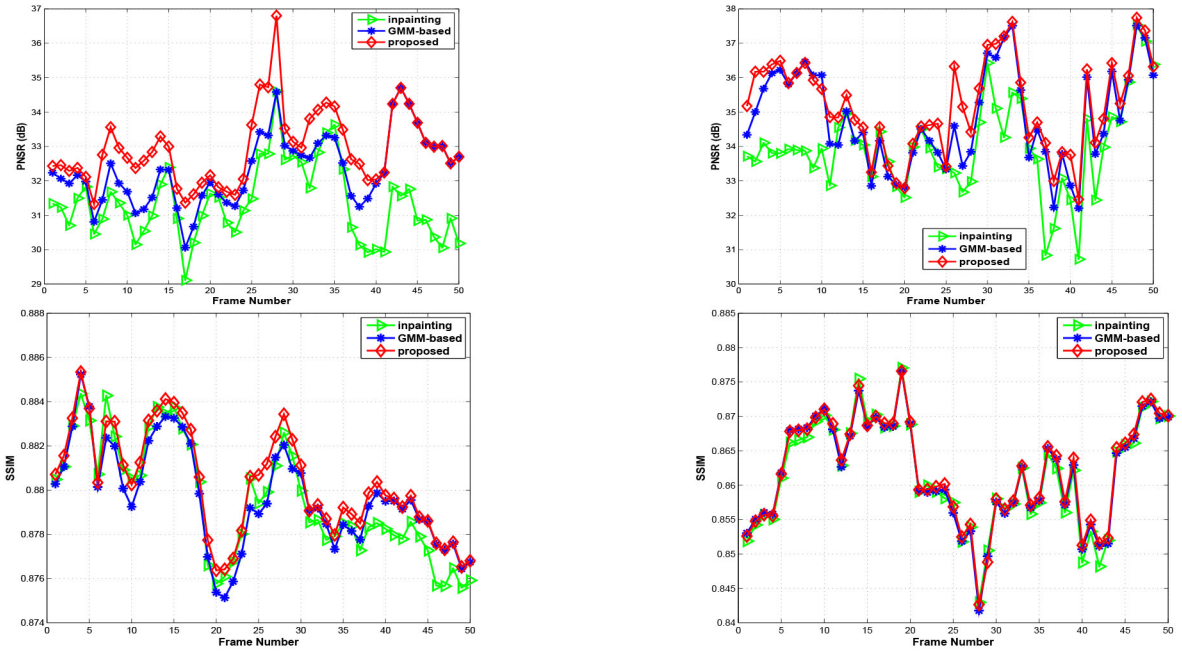

Fig. 11. Objective results for the sequence. (a) Ballet. (b) Breakdancer. The results are obtained with the proposed method, GMM-based method [29], and inpainting method [22].

TABLE II
PSNR AND SSIM RESULTS FOR BALLET AND BREAKDANCER SEQUENCE

Sequence	Frame Number	Camera Index	PSNR(dB)			SSIM		
			proposed	GMM-based [29]	inpainting [22]	proposed	GMM-based [29]	inpainting [22]
Ballet	4	01 → 02	**32.37**	32.16	31.49	**0.8853**	0.8852	0.8843
Ballet	28	01 → 02	**36.80**	34.58	34.59	**0.8834**	0.8820	0.8825
Breakdancer	4	01 → 02	**36.38**	36.12	33.80	0.8555	**0.8557**	0.8550
Breakdancer	26	01 → 02	**36.32**	34.59	33.23	**0.8524**	0.8518	0.8517

reference to evaluate the PSNR and SSIM for each rendered image.

The achieved frame by frame objective results are shown in Fig. 11. The red line with diamond marks represents the PSNR results of our proposed method, the blue line with star marks indicates the PSNR results of GMM-based method and that of the inpainting method is shown with green line and triangle marks. The PSNR curves show that the proposed method is better than both the GMM-based method and the inpainting method in most frames, and similarly the proposed method in aggregate is better than the other methods in terms of SSIM. To further evaluate the performance of the proposed method in case of small baseline and large baseline, more experiments are implemented in the following two scenarios:

- with the regular baseline of two adjacent cameras.
- with twice the regular baseline of two adjacent cameras.

The mean PSNR value and the mean SSIM value are evaluated for 50 frames for the two previously described scenarios. As shown in Table I, the objective PSNR results show that for the regular baseline, the proposed method is 0.4 ∼ 0.6 dB higher than the GMM-based method, and this later is 0.45 ∼ 2.2 dB higher than the inpainting method; in the case of twice the regular baseline, the proposed method is 0.7 ∼ 1.1 dB higher than the GMM-based method, and this later is 1.3 ∼ 4.0 dB higher than the inpainting method. It is worth noticing that the larger the baseline is the larger the obtained gains.

In addition to the objective measurements, Fig. 12 and Fig. 13 show some subjective results. Fig. 12 shows Frame 4 and 28 from *Ballet* sequence, whereas, Fig. 13 shows Frame 4 and 26 from *Breakdancer* sequence. The objective gain of the proposed method is large for Frame 28 (*Ballet*) and 26 (*Breakdancer*), whereas, it is low for Frame 4 for both sequences, the corresponding objective results are shown in Table II. Nevertheless, subjectively, the gains are consistent, and the proposed method has obviously better performance than the two other methods. In fact, as shown in Fig. 12 (a) and Fig. 13 (a), it could be noted that the inpainting method leads to some blur regions along the transition regions between the foreground and background objects; whereas GMM-based yields some double-image and blur regions [Fig. 12 (b) and Fig. 13 (b)], which are due to the reciprocal motion of the moving objects. It is worth indicating that although in the disocclusion regions, the proposed method may yield to some artifacts (due to the imperfect depth maps [14]), it recovers the occluded regions and maintains the shape of the foreground objects [Fig. 12 (c)

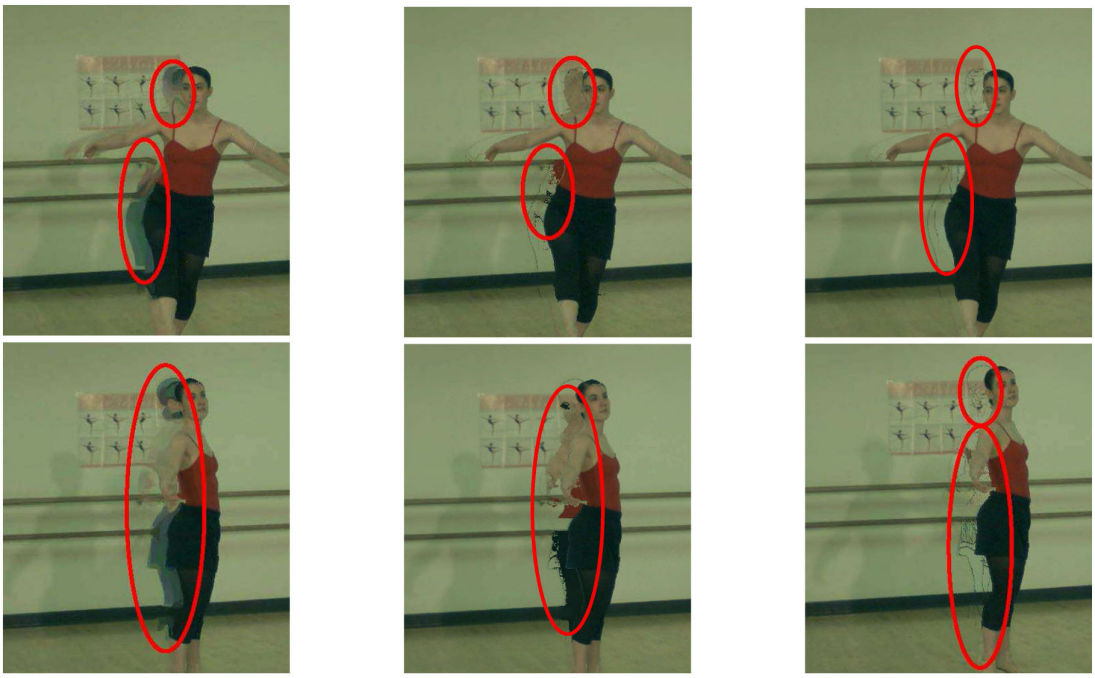

Fig. 12. Some subjective results: (a) inpainting method [22]; (b) GMM-based method [29]; (c) proposed method. The top images are Frame 4 and the bottom images are Frame 28.

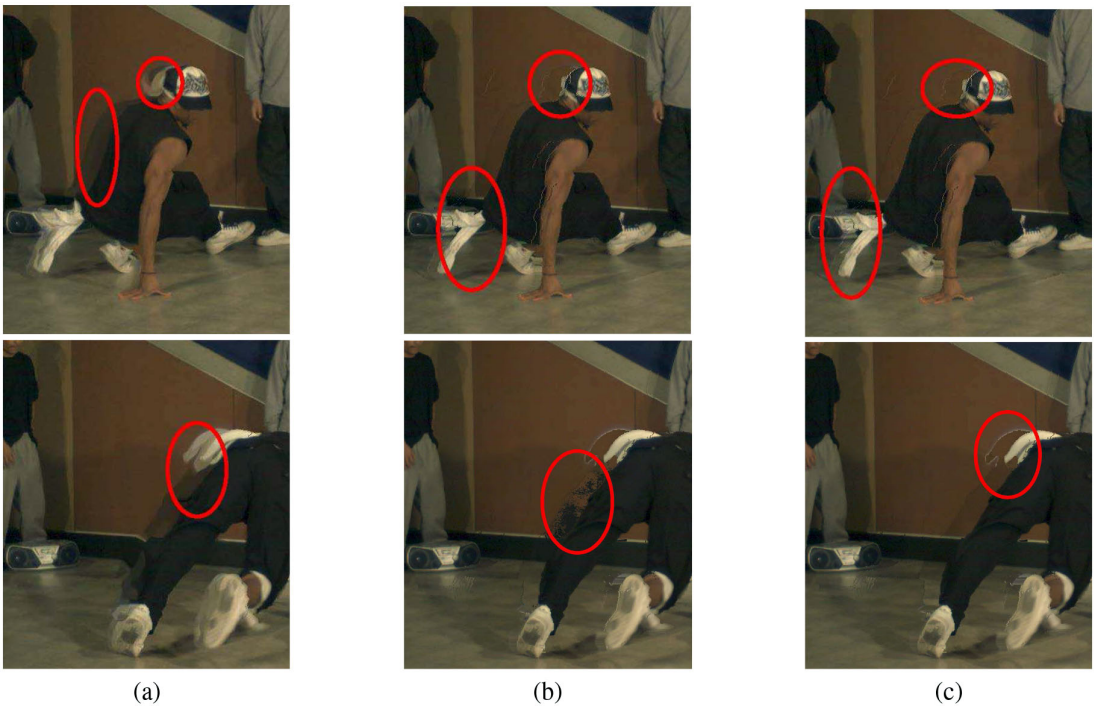

Fig. 13. Some subjective results: (a) inpainting method [22]; (b) GMM-based method [29]; (c) proposed method. The top images are Frame 4 and the bottom images are Frame 26.

and Fig. 13 (c)]. More experiment results are available for download at http://www.mmtlab.com/DGMMFDCDF.ashx.

VI. Conclusion and Future Work

In this paper, we discussed how to fill the disocclusion regions using the temporal correlation of texture and depth frames for SVD format, where a temporal stable background image is generated using a background update technique. In the proposed approach the disocclusion regions are filled using a combination of two approaches, the first one uses the GMM method to generate a background reference to fill the regions covered by moving objects; whereas, the second

exploits its depth maps information to detect the dynamic regions; finally, the inpainting technique is executed on the regions along the static objects. The reported results show that the proposed approach yields good subjective and objective results. Future research work will focus on wrapping the background using depth information in GMM. In addition, we will investigate how to generate a background image for complex scene with moving camera.

REFERENCES

[1] P. Benzie *et al*., "A survey of 3DTV displays: Techniques and technologies," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1647–1658, Nov. 2007.

[2] I. Ahn and C. Kim, "A novel depth-based virtual view synthesis method for free viewpoint video," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 614–626, Dec. 2013.

[3] L. Shen, P. An, Z. Liu, and Z. Zhang, "Low complexity depth coding assisted by coding information from color video," *IEEE Trans. Broadcast.*, vol. 60, no. 1, pp. 128–133, Mar. 2014.

[4] J. Xiao, T. Tillo, H. Yuan, and Y. Zhao, "Macroblock level bits allocation for depth maps in 3-D video coding," *J. Signal Process. Syst.*, vol. 74, no. 1, pp. 127–135, 2014.

[5] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Proc. SPIE Stereoscopic Displays Virtual Reality Syst. XI*, San Jose, CA, USA, May 2004, pp. 93–104.

[6] Y. Zhao, C. Zhu, Z. Chen, and L. Yu, "Depth no-synthesis-error model for view synthesis in 3-D video," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2221–2228, Aug. 2011.

[7] S. Il Lee, Y. J. Jung, H. Sohn, F. Speranza, and Y. M. Ro, "Effect of stimulus width on the perceived visual discomfort in viewing stereoscopic 3-D-TV," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 580–590, Dec. 2013.

[8] L. Zhang, W. J. Tam, and D. Wang, "Stereoscopic image generation based on depth images," in *Proc. IEEE ICIP*, 2004, pp. 2993–2996.

[9] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 191–199, Jun. 2005.

[10] W.-Y. Chen, Y.-L. Chang, S.-F. Lin, L.-F. Ding, and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *Proc. IEEE ICME*, Amsterdam, The Netherlands, 2005, pp. 1314–1317.

[11] P.-J. Lee *et al.*, "Nongeometric distortion smoothing approach for depth map preprocessing," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 246–254, Apr. 2011.

[12] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *Proc. 19th ICPR*, Tampa, FL, USA, 2008, pp. 1–4.

[13] M. Karsten *et al*., "View synthesis for advanced 3D video systems," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–11, 2009.

[14] Y. Zhao, C. Zhu, Z. Chen, D. Tian, and L. Yu, "Boundary artifact reduction in view synthesis of 3D video: From perspective of texture-depth alignment," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 510–522, Jun. 2011.

[15] C. Zhu, Y. Zhao, L. Yu, and T. Masayuki, *3D-TV System with Depth-Image-Based Rendering*. New York, NY, USA: Springer, 2013.

[16] M. Schmeing and X. Jiang, "Depth image based rendering: A faithful approach for the disocclusion problem," in *Proc. 3DTV-CON*, Tampere, Finland, 2010, pp. 1–4.

[17] K.-Y. Chen, P.-K. Tsung, P.-C. Lin, H.-J. Yang, and L.-G. Chen, "Hybrid motion/depth-oriented inpainting for virtual view synthesis in multiview applications," in *Proc. 3DTV-CON*, Tampere, Finland, 2010, pp. 1–4.

[18] M. Koppel *et al*., "Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering," in *Proc. 17th IEEE ICIP*, Hong Kong, China, 2010, pp. 1809–1812.

[19] P. Ndjiki-Nya *et al*., "Depth image-based rendering with advanced texture synthesis for 3-D video," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 453–465, Jun. 2011.

[20] E. Bosc *et al*., "Can 3D synthesized views be reliably assessed through usual subjective and objective evaluation protocols?" in *Proc. 18th IEEE ICIP*, Brussels, Belgium, 2011, pp. 2597–2600.

[21] Y. Zhao, L. Yu, Z. Chen, and C. Zhu, "Video quality assessment based on measuring perceptual noise from spatial and temporal perspectives," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1890–1902, Dec. 2011.

[22] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[23] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, Fort Collins, CO, USA, 1999, p. 252.

[24] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[25] M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. 19th ICPR*, Tampa, FL, USA, 2008, pp. 1–4.

[26] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*. New York, NY, USA: Springer, 2002, pp. 135–144.

[27] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[28] *MPEG-3DV View Synthesis Reference Software* [Online]. Available: ftp://ftp.merl.com/pub/avetro/3dv-cfp/software/VSRS software.zip

[29] C. Yao, Y. Zhao, and H. Bai, "View synthesis based on background update with Gaussian mixture model," in *Proc. Adv. PCM*, Singapore, 2012, pp. 651–660.

[30] *Sequence Microsoft Ballet and Breakdancers* [Online]. Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/

[31] F. Bruls, R. Gunnewiek, and P. Van De Walle, "Philips response to new call for 3DV test material: Arrive book & mobile," *ISO/IEC JTC1/SC29/WG11 Doc. M*, vol. 16420, 2009.

[32] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 2.0 (vsrs2. 0)," *ISO/IEC JTC1/SC29/WG11 M*, vol. 16090, 2009.

Chao Yao received the B.S. degree in computer science from Beijing Jiaotong University, Beijing, China, in 2009, where he is currently pursuing the Ph.D. degree in signal and information processing from the Institute of Information Science. His current research interests include video compression and processing, 3-D video coding, and 3-D computer vision.

Tammam Tillo (M'05–SM'12) was born in Damascus, Syria. He received the Engineer Diploma in electrical engineering from the University of Damascus, Damascus, Syria, and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Torino, Italy, in 1994 and 2005, respectively. In 2004, he served as a Visiting Researcher at Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, and from 2005 to 2008, he was a Post-Doctoral Researcher at the Image Processing Laboratory of Politecnico di Torino. For few months, he was an Invited Research Professor at the Digital Media Laboratory, SungKyunKwan University, Seoul, Korea. He joined Xi'an Jiaotong-Liverpool University (XJTLU), Jiangsu, China, in 2008. From 2010 to 2013, he was the Head of the Department of Electrical and Electronic Engineering at XJTLU University, and he was the Acting Head of the Department of Computer Science and Software Engineering from 2012 to 2013. He serves as an Expert Evaluator for several national-level research programs. His current research interests include the areas of robust transmission of multimedia data, image, and video compression, and hyperspectral image compression.

Yao Zhao (M'06–SM'12) received the B.S. degree from Fuzhou University, Fuzhou, China, and the M.E. degree from Southeast University, Nanjing, China, both in radio engineering, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1989, 1992, and 1996, respectively. In 1998, he was an Associate Professor at BJTU, where he became a Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow at the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the Director of the Institute of Information Science, BJTU. He is currently leading several national research projects with the 973 Program, the 863 Program, and the National Science Foundation of China. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He serves on the editorial boards of several international journals, including as an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS and IEEE SIGNAL PROCESSING LETTERS, an Area Editor of *Signal Processing: Image Communication* (Elsevier), and as an Associate Editor of *Circuits, System δ Signal Processing* (Springer). He was the recipient of the National Science Foundation of China for Distinguished Young Scholars Award in 2010.

Jimin Xiao was born in Suzhou, China. He received the B.S. and M.Eng. degrees in telecommunication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2007, respectively. Since 2009, he has been pursuing the Ph.D. degree from the University of Liverpool, Liverpool, U.K. From 2007 to 2009, he was a Software Engineer at Motorola (China) Electronics, Ltd., and later as a System Engineer at Realsil (Realtek) Semiconductor Corporation. His current research interests include the areas of video streaming, image and video compression, and 3-D video coding.

Huihui Bai received the Ph.D. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2008. She is currently an Associate Professor with the Institute of Information Science in BJTU. She has been engaged in research and development work in video coding technologies and standards such as HEVC, 3-D video compression, multiple description video coding, and distributed video coding. She is leading or participating in several research projects with the 973 Program, the 863 Program, National Natural Science Foundation of China, Beijing Natural Science Foundation, and Jiangsu Provincial Natural Science Foundation.

Chunyu Lin was born in LiaoNing Province, China. He received the Ph.D. degree from Beijing Jiaotong University (BJTU), Beijing, China, in 2011. He is currently a Lecturer with BJTU. From 2009 to 2010, he was a Visiting Researcher at the ICT Group of Delft University of Technology, Delft, The Netherlands. From 2011 to 2012, he was a Post-Doctoral Researcher at Gent University, Gent, Belgium. His current research interests include the areas of image/video compression and robust transmission, 2-D to 3-D conversion, stereo matching, and 3-D video coding.