Virtual View Synthesis Method and Self-Evaluation Metrics for Free Viewpoint Television and 3D Video

Kwan-Jung Oh,[1] Sehoon Yea,[2] Anthony Vetro,[2] Yo-Sung Ho[1]

[1] GIST, Department of Information and Communications, Gwangju, South Korea

[2] MERL, Multimedia Group, Cambridge, MA

Received 20 January 2009; accepted 24 May 2010

ABSTRACT: Virtual view synthesis is one of the most important techniques to realize free viewpoint television and three-dimensional (3D) video. In this article, we propose a view synthesis method to generate high-quality intermediate views in such applications and new evaluation metrics named as spatial peak signal-to-noise ratio and temporal peak signal-to-noise ratio to measure spatial and temporal consistency, respectively. The proposed view synthesis method consists of five major steps: depth preprocessing, depth-based 3D warping, depth-based histogram matching, base plus assistant view blending, and depth-based hole-filling. The efficiency of the proposed view synthesis method has been verified by evaluating the quality of synthesized images with various metrics such as peak signal-to-noise ratio, structural similarity, discrete cosine transform (DCT)-based video quality metric, and the newly proposed metrics. We have also confirmed that the synthesized images are objectively and subjectively natural. Int J Imaging Syst Technol, 20, 378–390, 2010; Published online in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/ima.20253

Key words: view synthesis; free viewpoint television; 3D video; image-based rendering; evaluation of virtual view

I. INTRODUCTION

Three-dimensional (3D) video provides users with a realistic 3D impression of the scene and is now considered a key technology that could spur the next wave of multimedia experiences such as 3D cinema, 3D broadcasting, 3D displays, and 3D mobile services (Isgro et al., 2004; Tanimoto, 2004; Smolic and Kauff, 2005; Tanimoto, 2006).

The key technical building blocks of the 3D processing chain are coding and rendering. The role of efficient coding becomes much more important for 3D systems due to the drastic increase in the volume of data. Some of the past research and standardization efforts to address this issue include MPEG-2 multiview video profile (Chen and Luthra, 1997), MPEG-4 multiple auxiliary component (Karim et al., 2005), and moving picture experts group (MPEG)/joint video team (JVT) multiview video coding (MVC) (ISO/IEC JTC1/SC29/WG11, 2003, 2005, 2007a; Shum et al., 2004; Smolic and McCutchen, 2004; Smolic et al., 2005). Recently, MPEG has initiated a work aimed specifically toward 3D video applications. While the previous MPEG/JVT standardization activities for MVC was focused on compression efficiency improvement for generic multiview coding scenarios, this activity will target a broader technical scope including issues such as depth estimation, coding, and rendering. One of the current underlying key design assumptions is the use of depth maps along with camera parameters for rendering intermediate views for either free viewpoint navigation or 3D displays.

On the other hand, given the ever increasing diversity in 3D services and displays, proper rendering of 3D views is indispensable. In other words, it becomes necessary to resample the views and resize each view depending on the number of views and resolutions required by the display, respectively. For applications such as free viewpoint television (FTV) (Shade et al., 1998; Zitnick et al., 2004; Gan et al., 2005) and the case when there are more views to be rendered at the display than are actually coded, resampling means generation of virtual views based on the actual views. The problem of generating an arbitrary view of a 3D scene has been heavily addressed in the area of computer graphics. Among the techniques for rendering, image-based rendering (IBR) techniques have received much attention lately for rendering real world scenes. These techniques use image rather than geometry as primitives for rendering virtual views and often are classified into three categories depending on how much geometric information is used (Chang et al., 1999): rendering without geometry, with explicit geometry, and with implicit geometry. Techniques such as plenoptic modeling (McMillan and Bishop, 1995), light-field rendering (Debevec et al., 1998), lumigraph (ISO/IEC JTC1/SC29/WG11, 2006), and ray space (ISO/IEC JTC1/SC29/WG11, 2007b,c) belong to the rendering without geometry. In this approach, the quality of view synthesis usually depends on the baseline distance and the synthesis quality increases with the number of available views within a restricted viewing angle. On the other hand, an IBR system with depth maps that uses techniques such as 3D warping and layered depth images (LDIs) belongs to the second category while view morphing and view interpolation as in Gortler et al. (1996), Levoy and Hanrahan (1996), Droese et al. (2004), Tanimoto (2005), (ISO/IEC JTC1/SC29/WG11, 2007d, and Chan et al. (2007) belong to the third category as they use the point correspondences. Obviously, the quality of view synthesis in these explicit/implicit geometry-based rendering approaches largely depends on the accuracy of the geometry information.

Correspondence to: Sehoon Yea; e-mail: yea@merl.com

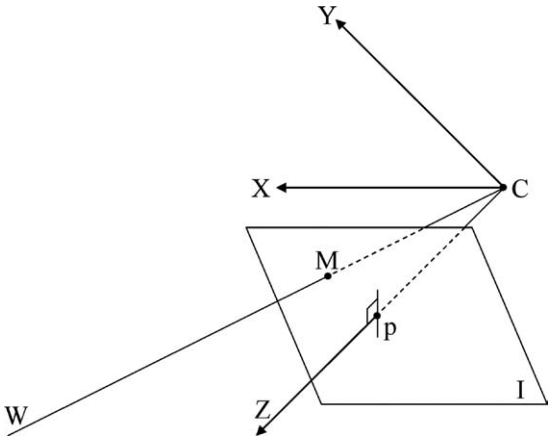

Figure 1. The pinhole camera model.

In this article, we propose a new view synthesis algorithm within the aforementioned scope of the FTV and 3D video activities (ISO/IEC JTC1/SC29/WG11, 2007e) and new evaluation metrics to measure the spatial and temporal consistencies of the synthesized views. The proposed view synthesis method consists of five major steps: depth preprocessing, depth-based 3D warping, depth-based histogram matching, base plus assistant view blending, and depth-based hole-filling. First, a preprocessing is performed on the acquired scene depth data to correct errors and enhance the spatial and temporal consistencies of depth values. Second, a depth-based 3D warping technique is adopted to avoid the discontinuity problem in the direct warping of textures caused by round-off errors. Third, a depth-based histogram matching algorithm is used to reduce the illumination difference between two reference views. Fourth, a base plus assistant view blending is introduced to blend two 3D warped reference images in a robust manner against the inaccuracy of the depth and camera parameters. Finally, a depth-based hole-filling technique is used to fill the remaining holes using a depth-based inpainting technique. The synthesized view is evaluated by peak signal-to-noise ratio (PSNR), structural SIMilarity (SSIM) (Wang et al., 2004), DCT-based video quality metric (VQM) (Xiao, 2000; MSU), and the newly proposed spatial PSNR (SPSNR) and temporal PSNR (TPSNR).

The rest of this article is organized as follows. In Section II, we describe the basics of view synthesis. We explain the details of the proposed view synthesis algorithm and the evaluation metrics in Sections III and IV, respectively. We then demonstrate and evaluate the performance of the proposed scheme in Section V, and conclude this article in Section VI.

II. BACKGROUND

This section briefly reviews the camera geometry model and the general idea of depth-based view synthesis.

A. Camera Geometry Model. A general pinhole camera is modeled by its optical center C and its image plane I. A 3D point W is projected into an image point M given by the intersection of I with the line containing C and W. The line containing C and orthogonal to I is called the optical axis (Z) and its intersection with I is the principal point (p). The distance between C and I is the focal length.

Let $w = [x\ y\ z]^{\mathrm{T}}$ be the coordinates of W in the world reference frame (fixed arbitrarily) and $m = [u\ v]^{\mathrm{T}}$ the coordinates of M in the image plane (pixels). The mapping from 3D coordinates to 2D coordinates is perspective projection, which is represented by a linear transformation in homogeneous coordinates. Let $\tilde{m} = [u\ v\ 1]^{\mathrm{T}}$ and $\tilde{w} = [x\ y\ z\ 1]^{\mathrm{T}}$ be the homogeneous coordinates of M and W, respectively; then, the perspective transformation is given by the matrix $\tilde{P}$:

$$\kappa\tilde{m} = \tilde{P}\tilde{w}, \tag{1}$$

where κ is a scale factor called projective depth. κ becomes the true orthogonal distance of the point from the focal plane of the camera. The camera is therefore modeled by its perspective projection matrix (henceforth simply camera matrix) $\tilde{P}$, which can be decomposed, using the QR factorization, into the product

$$\tilde{P} = A[R|t]. \tag{2}$$

The matrix A depends on the intrinsic parameters only and has the following form:

$$A = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\alpha_u = -fk_u$, $\alpha_v = -fk_v$ are the focal lengths in horizontal and vertical pixels, respectively (f is the focal length in millimeters, k_u and k_v are the effective number of pixels per millimeter along the u and v axes), (u_0, v_0) is the coordinate of the principal point given by the intersection of the optical axis with the retinal plane as shown in Figure 1, and γ is the skew factor that models nonorthogonal $u - v$ axes.

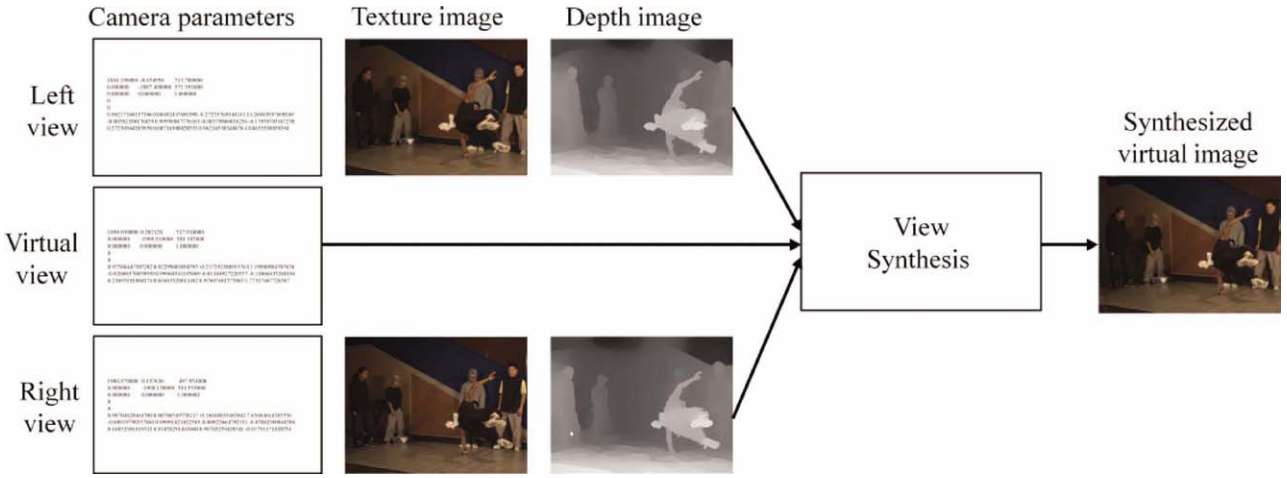

Figure 2. Depth-based virtual view synthesis. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

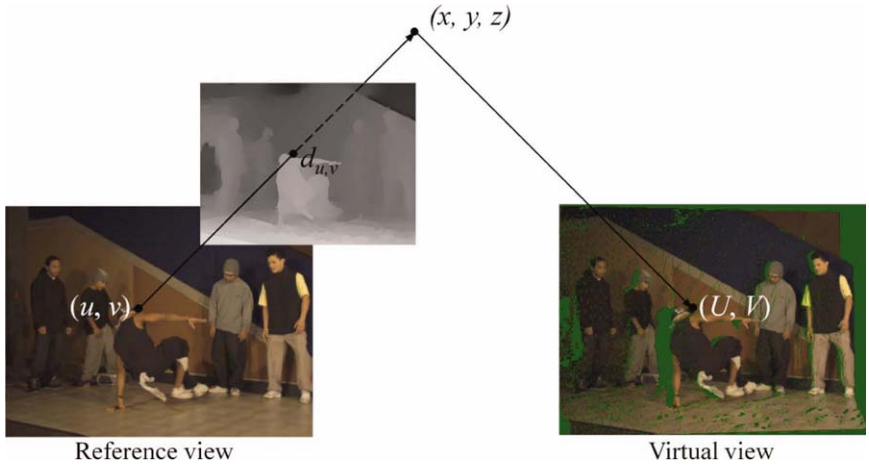

Figure 3. General concept of 3D warping. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The camera position and orientation (extrinsic parameters) are represented by the 3×3 rotation matrix R and the translation vector t, respectively, corresponding to the rigid transformation that brings the camera reference frame onto the world reference frame (Trucco and Verri, 1998; Fusiello, 2000; Hartley and Zisserman, 2000).

B. Depth-Based View Synthesis. The schematic diagram of a typical depth-based view synthesis system is shown in Figure 2. The goal of such a system is to synthesize a virtual view from its neighboring views using the camera parameters, texture images, and depth images.

The 3D image warping is the key technique in depth-based view synthesis. In 3D warping, pixels in the reference image are back projected to 3D spaces and reprojected onto the target viewpoint as shown in Figure 3.

Equations (4) and (5) represent the back projection and the reprojection processes, respectively.

$$(x, y, z)^T = R_{\text{ref}} A_{\text{ref}}^{-1} (u, v, 1)^T d_{u,v} + t_{\text{ref}} \tag{4}$$

$$(l, m, n)^T = A_{\text{vir}} R_{\text{vir}}^{-1} \{(x, y, z)^T - t_{\text{vir}}\} \tag{5}$$

where A, R, and t are camera parameters and d represents the depth value of a point in the 3D space that needs to be back-/reprojected. The coordinates (l, m, n) in (5) is normalized to $(l/n, m/n, 1)$ and then represented as an integer coordinate (U, V) in the virtual view.

III. PROPOSED VIEW SYNTHESIS ALGORITHM

The proposed view synthesis algorithm consists of five steps: depth preprocessing, depth-based 3D warping, depth-based histogram matching, base plus assistant view blending, and depth-based hole-filling. Figure 4 shows a diagram of the proposed view synthesis scheme and each subalgorithm will be detailed in the following subsections.

A. Depth Preprocessing. In general, the depth data can be obtained by a special depth camera system and computer graphics tools or mathematically calculated by depth estimation algorithms. Currently, depth estimation is the most popular approach and actively studied since the depth camera is too expensive and computer graphic images cannot represent real scenes.

However, mathematically calculated depth data tend to have erroneous values in certain regions in the image or have inconsistent values across spatial or temporal neighbors due to the local nature of depth estimation process. These problems associated with depth could lead to various visual artifacts in the synthesized images. To resolve these issues, we propose to preprocess the depth data. The proposed depth preprocessing consists of three steps: temporal filtering, initial error compensation, and spatial filtering. Basically, we apply a median filtering instead of averaging filter because averaging filter results in new pixel values which do not exist in the initial depth image, which degrades rendering quality.

As a first step, we apply a 1D median filter along the colocated pixels of consecutive depth image frames. It aims to reduce the temporal inconsistency of depth values belonging to the same object or background. In this article, we apply a median filter as follows:

$$Y_{i,j,t} = \begin{cases} \text{median}(J_{i,j,t}), & \text{for } \max(J_{i,j,t}) - \min(J_{i,j,t}) <= \gamma \\ X_{i,j,t}, & \text{otherwise} \end{cases} \tag{6}$$

where $X_{i,j,t}$ is the value of a pixel at the spatial location (i, j) at time t, $J_{i,j,t}$ is a set of pixels in a 3×3 window centered around the

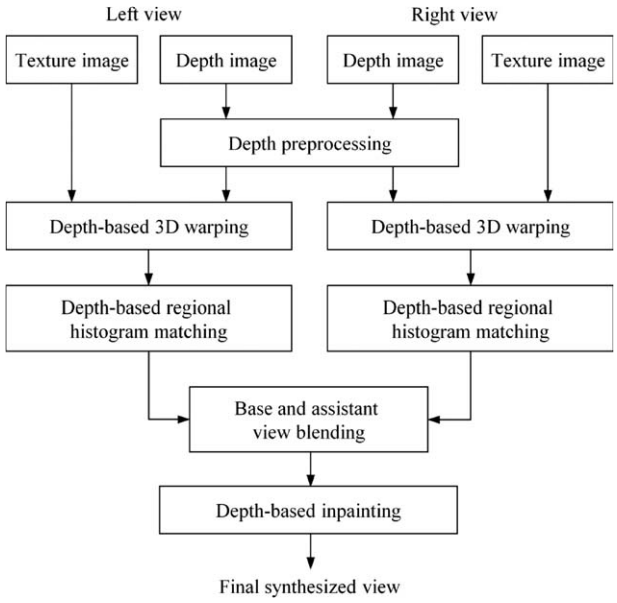

Figure 4. Diagram of the proposed view synthesis scheme.

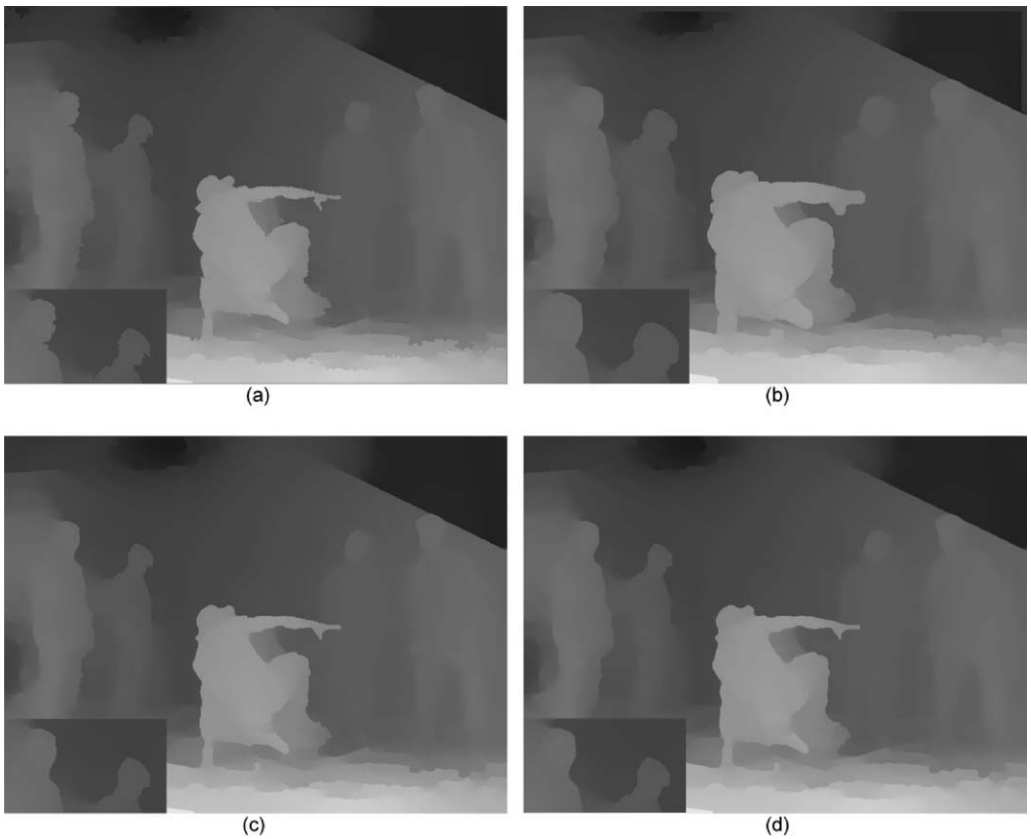

Figure 5. An example of depth preprocessing for "Breakdancers" sequence: (a) temporal filtered image, (b) dilated image, (c) eroded image, and (d) spatial filtered image.

spatio-temporal location (i, j, t), and γ is a threshold value to determine whether or not the filter will be applied.

The next step has to do with compensating for the initial error that is caused by an erroneous merge of foreground and background in the typical depth estimation process. Usually, it occurs when the foreground and the background have similar textures. The human eyes can easily distinguish them but it is often a difficult task for an automated algorithm. In this article, we correct the initial errors by using image dilation and erosion as in (7) and (8), respectively, (Bangham and Marshall, 1998). As the quality of a synthesized image will be worse in case the foreground has a background's depth value than the other way around, image dilation is conducted prior to image erosion in the proposed scheme.

$$A \oplus B(x, y) = \max_{(x,y)\in B} [A_B(x, y)] \tag{7}$$

$$A \ominus B(x, y) = \min_{(x,y)\in B} [A_B(x, y)] \tag{8}$$

where A represents the image and B is structuring element which operates on A. The A_B is a masked region with B and (x, y) is a pixel in the image A. In this article, we use a disk-shaped structuring element with disk radius set to five.

Figure 6. 3D warping with erroneous blanks: (a) depth image and (b) 3D warped texture image using (a). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

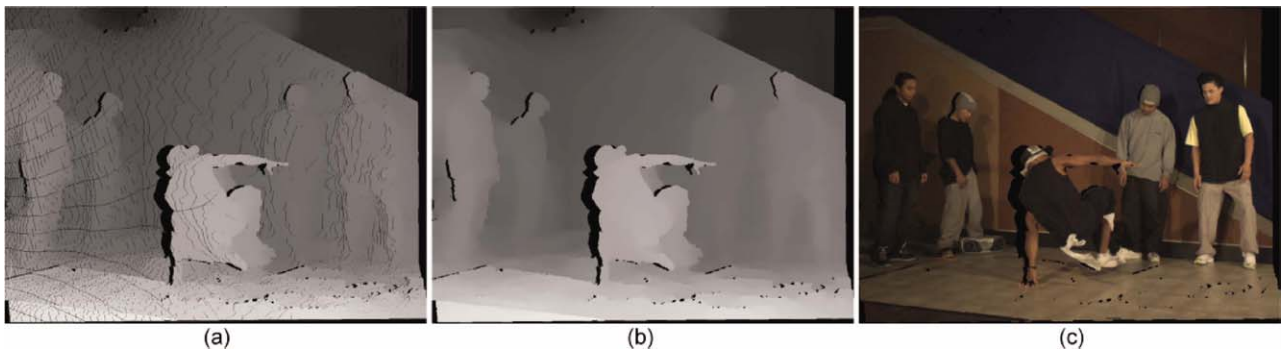

Figure 7. 3D warping without erroneous blanks: (a) 3D warped depth image, (b) median filtered depth for (a), and (c) 3D warped texture image using (b). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The final step has to do with smoothing outliers in an estimated depth image using a 2D median filter. It smoothes out the outlier of objects in a depth image and removes the unwanted noises. In this article, we use a 5×5 median filter for every pixel at (i, j) as follows:

$$Y_{i,j} = \text{median}(J_{i,j}) \tag{9}$$

where $J_{i,j}$ is a set of pixels in a 5×5 window centered around the location (i, j).

Figure 5 illustrates the result of each step of the proposed depth preprocessing for "Breakdancers" provided by MicroSoft Research (MSR). The effect of the proposed scheme is noticeable especially around the faces of the two men standing behind on the left side of the dancer as well as around the boundaries of the dancer on the floor. The proposed depth preprocessing method not only compensates for the initial depth errors efficiently but also recovers the spatial and temporal consistency (van den Branden Lambrecht and Verscheure, 1996). Hence, the preprocessed depth will lead to significantly improved objective and subjective qualities of the synthesized images.

B. Depth-Based 3D Warping. Most previous view synthesis algorithms warp the texture images using the corresponding depth maps. However, a direct 3D warping of texture images of neighboring views into the virtual image plane often causes false black-contours in the synthesized image as shown in Figure 6b. These contours are caused by round-off errors involved with the integer representation of the virtual view's coordinate as well as by spurious initial depth values.

However, once the depth image corresponding to the virtual view is obtained, we can use it to always find, by inverse warping, the proper texture values from its neighboring view without generating false black-contours in the synthesized view. To obtain the depth image corresponding to the virtual view, we first warp the depth values of the reference view. Note that the false black-contours appear in the warped depth image as shown in Figure 7a for the exactly same reason as with the texture warping. To remove these erroneous contours, we apply a median filtering (ISO/IEC JTC1/SC29/WG11, 2008a). Figure 7 illustrates the above procedures.

C. Depth-Based Histogram Matching. In case we have two reference views for the virtual view synthesis as shown in Figure 2, we can first synthesize two 3D warped views, i.e., one from each view. Before blending these two warped images, we apply a histogram matching to reduce the illumination and color differences between the two images which may cause inconsistency of the synthesized image. On the basis of previous histogram matching algorithm (Fecker et al., 2007), we modify the mapping condition considering the distributions of cumulative histograms and then apply this modified histogram matching regionally using depth-based segments.

The histograms of the two 3D warped images for reference views are analyzed and those 3D warped images are adjusted to have a similar distribution. The whole procedures of histogram matching are as follows. The first step is to modify the two 3D warped images to have same holes and then to apply a median filter for noise reduction as shown in Figure 8. By using the modified images instead of original 3D warped images, the accuracy of the histogram matching is improved.

The second step is to compute the histograms of the left image and the right image. Let $y_L[m, n]$ denote the amplitude of the left image. Then, its histogram is given as follows:

Figure 8. Image modification for histogram matching: (a) 3D warped view 3, (b) 3D warped view 5, (c) modified view 3, and (d) modified view 5. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

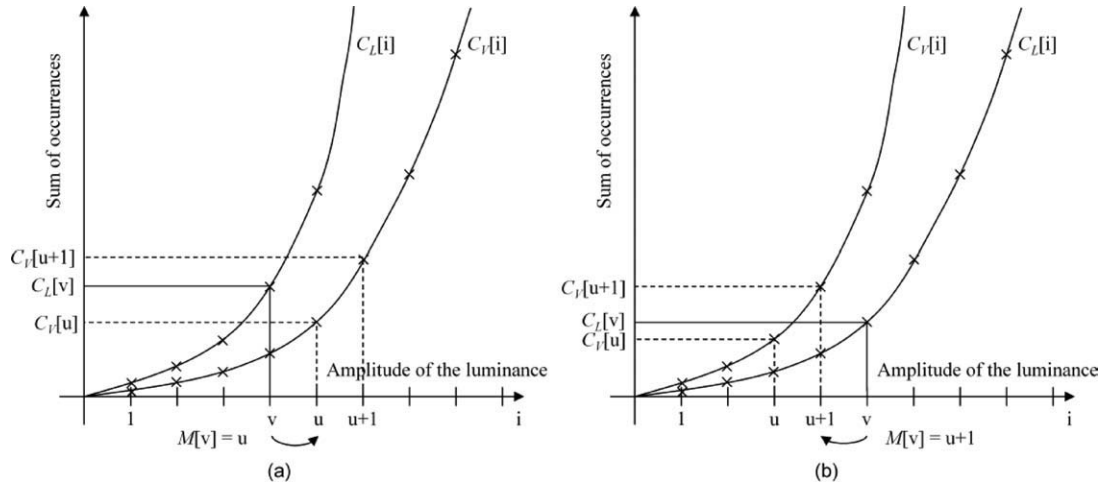

Figure 9. Mapping algorithm using cumulative histograms: (a) $C_V[v] <= C_L[v]$, (b) $C_V[v] > C_L[v]$.

$$h_L[v] = \frac{1}{w \cdot h} \sum_{m=0}^{h-1} \sum_{n=0}^{w-1} \delta[v, y_L[m,n]], \text{ with } \delta[a,b] = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

In (10), w denotes the image width and h is the image height. The value of v ranges from 0 to 255. The histogram matching is done by mapping the left and right images to a virtual image. Two steps are necessary to generate the mapping function M. First, the cumulative histogram $C_L[v]$ of the left image is created:

$$C_L[v] = \sum_{i=0}^{v} h_L[i] \tag{11}$$

The histogram $h_R[v]$ and cumulative histogram $C_R[v]$ of the right image are calculated in the same manner. Both the left and right images, which are already warped into the virtual view position, are median filtered and modified to have the same holes as shown in Figures 8c and 8d so that the two views have almost identical textures except for slight differences in their illuminations.

Based on the cumulative histograms, we make a cumulative histogram $C_V[v]$ for virtual:

$$C_V(v) = \alpha\, C_L(v) + (1 - \alpha) C_R(v) \tag{12}$$

where C_L and C_R are the cumulative histograms for left and right images. Generally, the weighting factor α is calculated based on the baseline distance as follows:

$$\alpha = \frac{|t_V - t_L|}{|t_V - t_L| + |t_V - t_R|} \tag{13}$$

where t is a translation vector for each view.

The mapping function between the left image and the virtual image is obtained by matching the number of occurrences in the reference image to that of occurrences in the virtual image as in (14) and as shown in Figure 9 as an example.

$$M[v] = \begin{cases} u, & \text{for } C_V[u] < C_L[v] \le C_V[u+1] \;\&\; C_V[v] <= C_L[v] \\ u+1, & \text{for } C_V[u] < C_L[v] \le C_V[u+1] \;\&\; C_V[v] > C_L[v] \end{cases} \tag{14}$$

The calculated mapping function is applied to the left image $y_L[m, n]$, resulting in the histogram-matched image $y_{HML}[m, n]$ as in (15). The histogram $y_{HMR}[m, n]$ of the right image is calculated in the same manner.

$$y_{HML}[m,n] = M[y_L[m,n]] \tag{15}$$

Figure 10. Rough region division by depth: (a) foreground region and (b) background region. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

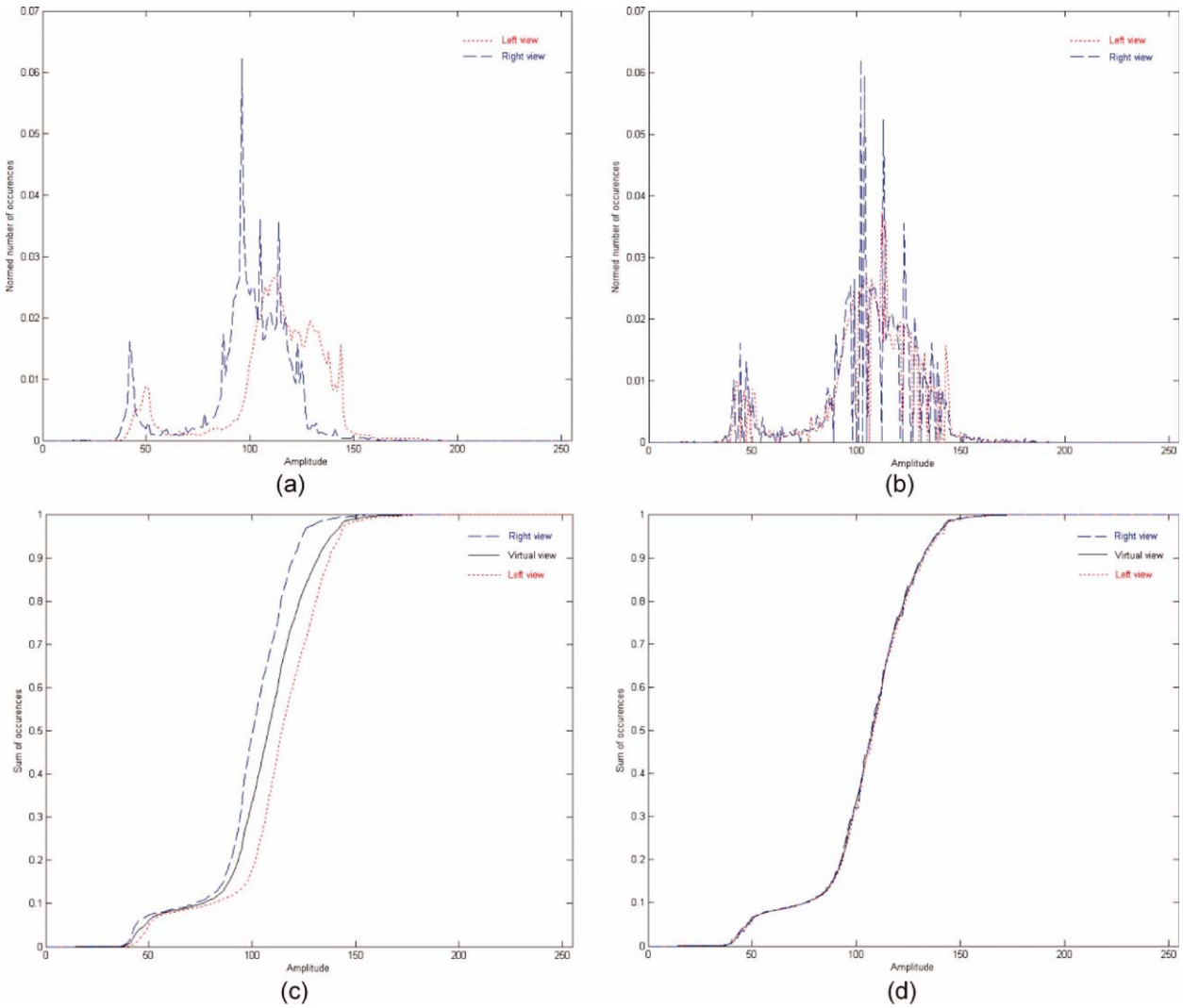

Figure 11. Histogram matching: (a) histograms, (b) histograms after histogram matching, (c) cumulative histograms, and (d) cumulative histograms after histogram matching. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In general, we assume that the difference of volume of light for each camera causes the illumination and color differences and differently affects each object and color component. By considering the above assumption, we apply the histogram matching regionally and the regions are divided using depth. Figure 10 shows an example of rough region division for the image in Figure 8d.

Figure 12. Hole extension: (a) before extension and (b) after extension. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Figure 13. View blending methods: (a) weighted sum method and (b) base and assistant method. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

While the previous histogram matching converts one view to the other to have a similar histogram, the proposed histogram matching modifies both the views to have similar histogram as that of the virtual view, which is defined by considering baseline distances. In addition, the proposed histogram matching maps the indices differently for the two cases in Figure 9.

Figure 11 shows an example for proposed histogram matching. In this case, histograms of the 3D warped left and right views have similar shapes but different distributions caused by illumination and color differences. By mapping these two reference view to have a similar cumulative histogram with that of the virtual view, we can reduce the illumination differences between two views. The proposed histogram matching is independently applied to each color component of RGB format.

D. Base Plus Assistant View Blending. The boundary errors around the big holes are usually caused by inaccuracy of the camera parameters and inaccurate boundary matching between texture images and depth images. To remove these visible errors, we extend the hole boundaries by using image dilation as shown in Figure 12. These extended holes can be filled by the other 3D warped view and we expect more natural synthesized view by removing this kind of errors.

The next step is view blending to combine 3D warped views to the virtual view and the simplest way would be taking a weighted sum of the two images as below:

$$I_{\mathrm{V}}(u,v) = \alpha I_{\mathrm{L}}(u,v) + (1-\alpha) I_{\mathrm{R}}(u,v) \tag{16}$$

where I_{L} and I_{R} are the 3D warped reference texture images and I_{V} is an image to be blended. Generally, the weighting factor α is calculated based on the baseline distance as in (13).

However, a drawback of this method is that inconsistent (due to, for e.g., camera parameters, inconsistent depth values, etc.) pixel values from both views can contribute to the warped image and often leads to an unnaturalness such as double edge artifacts and smoothing as shown in Figure 13. To avoid such a problem, we define a base view and an assistant view for view blending. The base view is the main reference view from which most of the pixel values are warped, and the assistant view is used as a supplementary reference view for hole-filling. Then (16) can be rewritten as (17), where α is 1 for nonhole regions and 0 for hole regions in the 3D warped base view. In other words, most regions of the blended view came from the base view and some remaining holes are filled from the assistant view. We choose a closer view from the virtual view as the base view.

$$I_{\mathrm{V}}(u,v) = \alpha\, I_{\mathrm{B}}(u,v) + (1-\alpha) I_{\mathrm{A}}(u,v) \tag{17}$$

where I_{B} is the base view and I_{A} is the assistant view.

E. Hole-Filling Using Depth-Based In-Painting. The last step of the proposed view synthesis is depth-based hole-filling. Even though view blending efficiently fills up most disoccluded regions, some holes still remain. In general, these remaining holes are caused by still remaining disocclusion regions and wrong depth value. Disocclusion regions are defined as areas that cannot be seen in the reference image but exist in the synthesized one. Many existing hole-filling methods use image interpolation or in-painting techniques and fill up the remaining holes using neighboring pixels solely based on geometrical distance. However, observe that it make more sense to fill up the holes using the background pixels rather than the foreground ones as the disoccluded area usually belongs to the background by definition. Therefore, we propose a hole-filling algorithm that prefers the background pixels over the foreground ones in addition to considering the existing in-painting technique.

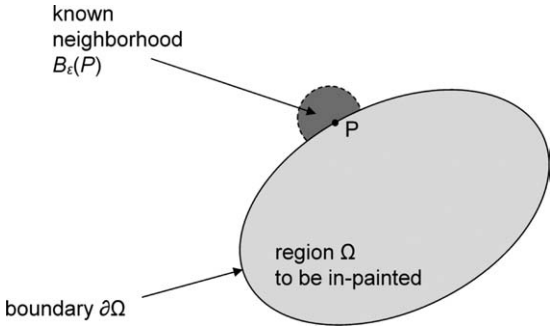

Figure 14. General in-painting circumstance.

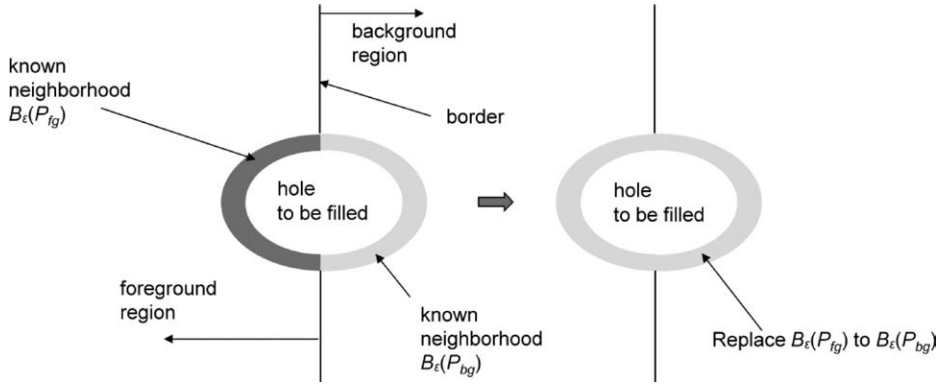

Figure 15. Manipulation of hole to have neighborhood only come from background.

The general in-painting problem is as follow (Telea, 2004): the region to be in-painted Ω and its boundary $\partial\Omega$ are defined and the pixel p belong to Ω would be in-painted by its neighboring region $B_\varepsilon(p)$ as shown in Figure 14.

This concept is quite reasonable for common image in-painting but it should be changed to be applied to hole-filling in view synthesis because $\partial\Omega$ of a certain hole can belong to both the foreground and the background. In this case, we replace the boundary region facing the foreground with the corresponding background region located on the opposite side as depicted in (18). That is, we intentionally manipulate the hole to have neighborhood belonging only to the background as shown in Figure 15.

$$\begin{aligned} p_{\mathrm{fg}} \in \partial\Omega_{\mathrm{fg}} &\rightarrow p_{\mathrm{bg}} \in \partial\Omega_{\mathrm{bg}} \\ B_\varepsilon(p_{\mathrm{fg}}) &\rightarrow B_\varepsilon(p_{\mathrm{bg}}) \end{aligned} \tag{18}$$

where fg and bg represent the foreground and the background, respectively.

To distinguish the foreground and the background, we use the corresponding depth data. In other words, for the two pixels horizontally opposite to each other on the holes boundary, we regard the

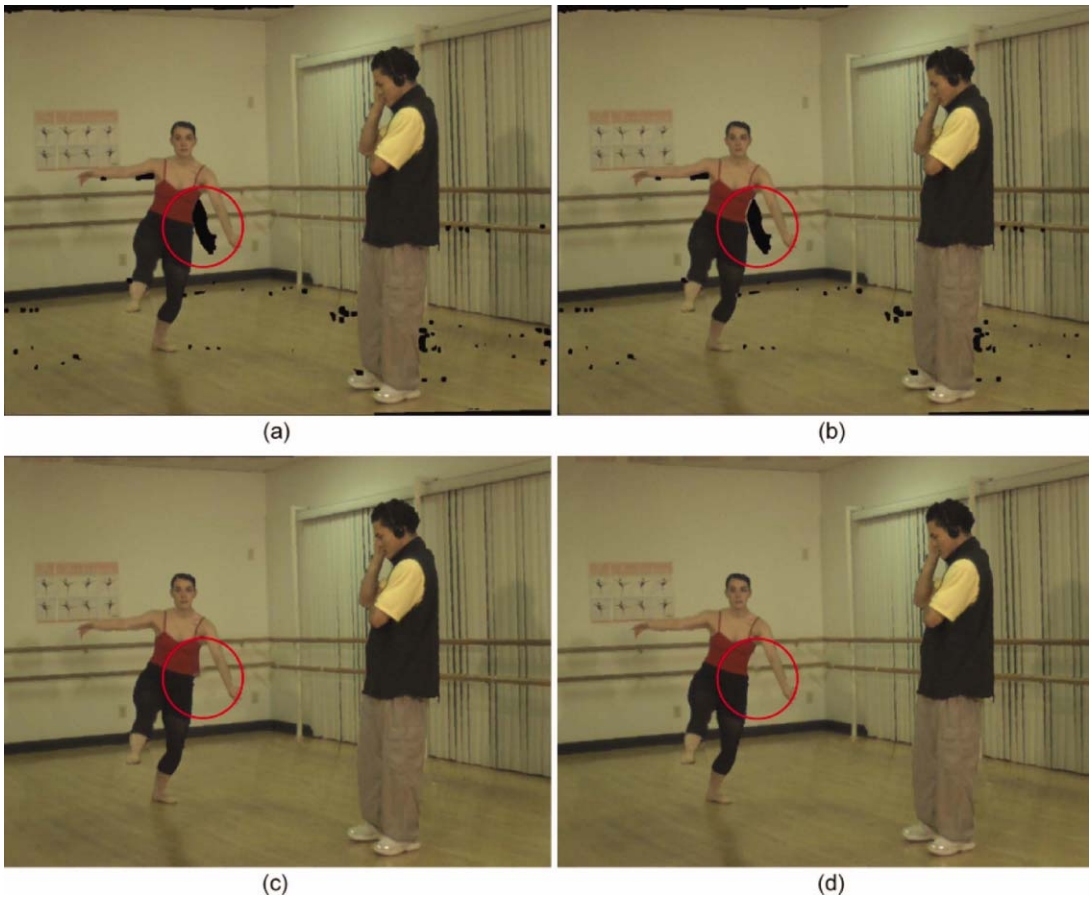

Figure 16. In-painting procedure: (a) image with holes, (b) boundary region copy from background, (c) previous in-painting, and (d) proposed depth-based in-painting. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table I. Experimental results for "depth preprocessing"

Evaluation Measures	Breakdancers		Ballet	
	Without Preprocessing	With Preprocessing	Without Preprocessing	With Preprocessing
PSNR	31.7300	31.6421	31.7773	31.7935
SSIM	0.8381	0.8379	0.8736	0.8739
VQM	3.9973	4.0984	2.6134	2.5351
SPSNR	38.8363	38.9236	38.3793	38.5004
TPSNR	37.8345	38.0415	39.6727	40.7091

pixel having the larger depth value as belonging to the foreground and vice versa. Figure 16 shows the results from the previous in-painting and the proposed depth-based in-painting techniques.

F. Self-Evaluation Metrics. To evaluate the performance of the view synthesis algorithm, generally, we measure the similarity between the synthesized view and the existing original one. The PSNR, SSIM (Wang et al., 2004), and VQM (Xiao, 2000) are widely used but these are only useful when the original view is available for virtual view. In addition, they cannot evaluate temporal consistency that is susceptible to illumination changes and the focus mismatch and to which human eyes are quite sensitive.

To overcome the limitations of the existing evaluation measure, we propose new evaluation metrics named as SPSNR and TPSNR. The SPSNR measure the spatial consistency by checking spatial noise caused by view synthesis. Generally, the view synthesis increases the high-frequency components since the 3D warped images and holes have a lot of high-frequency component. Thus, we can evaluate the spatial consistency by checking the degree of the volume of the increased high-frequency components. From the above concept, the SPSNR is defined as follows:

$$\begin{aligned} \mathrm{SPSNR} &= 10\log\frac{255^2}{\mathrm{SMSE}} \\ \mathrm{SMSE} &= \frac{1}{h\times w}\sum_{i=1}^{w}\sum_{j=1}^{h}[\mathrm{img}(i,j)-\mathrm{img}_{\mathrm{LPF}}(i,j)]^2 \end{aligned} \tag{19}$$

where h and w denote image height and width. We apply the 5×5 median filter as an low pass filter (LPF) to remove spatial noise and its difference image with the original image only contains the high-frequency components. We define the volume of the high-frequency components as SMSE similar to MSE in PSNR and develop the SPSNR similar to existing PSNR.

Table II. Experimental results for "histogram matching"

Evaluation Measures	Breakdancers		Ballet	
	Without Histogram Matching	With Histogram Matching	Without Histogram Matching	With Histogram Matching
PSNR	31.7300	31.8754	31.7773	31.5912
SSIM	0.8381	0.8367	0.8736	0.8714
VQM	3.9973	3.9729	2.6134	2.7049
SPSNR	38.8363	38.7442	38.3793	38.0913
TPSNR	37.8345	37.7891	39.6727	39.5653

The TSPNR to evaluate the temporal consistency is similar with the SPSNR except for its input image is replaced with the difference image between two temporally successive frames in (20). The TPSNR measure the high-frequency components of the temporal changes. The main merit of the proposed measures is it only uses the synthesized view itself.

$$\begin{aligned} \mathrm{TPSNR} &= 10\log\frac{255^2}{\mathrm{TMSE}} \\ \mathrm{TMSE} &= \frac{1}{h\times w}\sum_{i=1}^{w}\sum_{j=1}^{h}[\mathrm{img}D(i,j)-\mathrm{img}D_{\mathrm{LPF}}(i,j)]^2 \\ \mathrm{img}D(t) &= |\mathrm{img}(t)-\mathrm{img}(t-1)| \end{aligned} \tag{20}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested the proposed algorithm on two test sequences: "Breakdancers" and "Ballet." Among the eight views, view 3 and view 5 were selected as reference views and view 4 is set as the virtual view to be synthesized. Each major subalgorithm of the proposed method is evaluated by existing objective evaluation measures, such as PSNR, SSIM (Wang et al., 2004), VQM (Xiao, 2000), and the proposed SPSNR and TPSNR. While a larger value means a better quality for PSNR, SPSNR, and TPSNR, the opposite is true for VQM. In the case of SSIM, the closer the value is to 1, the better is the quality. The proposed view synthesis algorithm was compared with the view synthesis software version 2.3 (ISO/IEC JTC1/

Figure 17. Synthesized images: (a) without depth preprocessing and (b) with preprocessing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Figure 18. Histogram matching: (a) without histogram matching and (b) with histogram matching. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Figure 19. In-painting: (a) previous in-painting and (b) depth-based in-painting. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Figure 20. Synthesized images for "Breakdancers" sequence: (a) reference software and (b) proposed method. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table III. Experimental results for "hole-filling using in-painting"

	Breakdancers		Ballet	
Evaluation Measures	Previous In-Painting	Depth-Based In-Painting	Previous In-Painting	Depth-Based In-Painting
PSNR	31.7300	31.7484	31.7773	32.4967
SSIM	0.8381	0.8384	0.8736	0.8740
VQM	3.9973	3.9852	2.6134	2.5131
SPSNR	38.8363	38.8448	38.3793	38.3821
TPSNR	37.8345	37.8458	39.6727	39.8938

Table IV. Experimental results for "proposed view synthesis method"

	Breakdancers		Ballet	
Evaluation Measures	Reference Software	Proposed Method	Reference Software	Proposed Method
PSNR	31.6292	31.8150	32.1825	32.2854
SSIM	0.8341	0.8365	0.8664	0.8718
VQM	3.9273	4.0628	2.7430	2.5351
SPSNR	38.4073	38.8319	37.8048	38.2107
TPSNR	37.3941	38.0104	39.2467	40.6742

SC29/WG11, 2008b) released by Nagoya University, which is currently used as a reference software in MPEG FTV/3D video standardization activity. The default view blending method in the reference SW was replaced with the proposed base plus assistant method to make a more meaningful comparison.

A. Experimental Results for Depth Preprocessing. The results for depth preprocessing are given in Table I and their corresponding samples of synthesized images are shown in Figure 17. The depth preprocessing does not provide noticeable quality improvements in terms of the existing evaluation measures but it shows some gains for SPSNR and TPSNR. Especially, we can confirm that the temporal consistency of the "Ballet" sequence is enhanced by depth preprocessing. In addition, we can confirm some improvements such as natural smooth boundary for the dancer on the floor and the shapes of the heads of the two dancers standing on the left.

B. Experimental Results for Depth-Based Histogram Matching. As shown in Table II and sample images in Figure 18, the proposed histogram matching improves the subjective quality by reducing the illumination and color changes. However, its objective quality is slightly degraded.

C. Experimental Results for Proposed Depth-Based In-painting. The experimental results for depth-based in-painting are given in Table III and their corresponding synthesized sample images are in Figure 19. The proposed depth-based in-painting fills up the remaining holes using only the pixels located in the background when the holes border with both the foreground and the background. We can confirm the proposed method improves both the subjective and the objective qualities.

D. Experimental Results for the Proposed View Synthesis Method. The proposed view synthesis method consists of various subalgorithms such as depth preprocessing, depth-based 3D warping, depth-based histogram matching, base and assistant view blending, and hole-filling using depth-based in-painting. In this section, the proposed view synthesis method is compared with the reference view synthesis software (ISO/IEC JTC1/SC29/WG11, 2008b). The main tools of the reference software are depth-based 3D warping, hole-filling using in-painting, and weighted sum–based view blending. In this experiment, we replace the view blending method in reference view synthesis software with the base plus assistant method.

The experimental results are given in Table IV and their corresponding synthesized sample images in Figures 20 and 21. We could confirm that the synthesized images by the proposed view synthesis method are both subjectively and objectively better than those of the reference software.

V. CONCLUSIONS

In this article, we have proposed a virtual view synthesis method and self-evaluation metrics for FTV and 3D video. The proposed method consists of four steps: depth preprocessing, depth-based 3D warping, illumination and color difference compensation with a depth-based histogram matching, and hole-filling by a depth-based

Figure 21. Synthesized images for "Ballet" sequence: (a) reference software and (b) proposed method. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

in-painting technique. In addition, a base plus assistant view blending method was introduced for better subjective quality compared with the weighted sum–based view blending. The effectiveness of the proposed method was confirmed by evaluating the quality of the synthesized image using various quality measures including the newly proposed self-evaluation metrics SPSNR and TPSNR. We observed that the proposed method produced both subjectively and objectively better results compared with those by the current reference software being used in the MPEG FTV/3D video standardization activities.

REFERENCES

M. Tanimoto, Overview of free viewpoint television, Proc Signal Process Image Commun 21 (2006), 454–461.

A. Smolic and P. Kauff, Interactive 3D video representation and coding technologies, Proc IEEE, Special Issue on Adv in Video Coding and Delivery, 93 (2005), pp. 99–110.

M. Tanimoto, Free viewpoint television—FTV, Picture Coding Symposium (PCS 2004), Session 5, December 2004.

F. Isgro, E. Trucco, P. Kauff, and O. Schreer, 3D image processing in the future of immersive media, Proc IEEE Trans Circuits Syst Video Technol 14 (2004), 288–303.

X. Chen and A. Luthra, MPEG-2 multiview profile and its application in 3D TV, Proc SPIE-Multimedia Hardware Architectures 3021 (1997), 212–223.

H.A. Karim, S. Worrall, A.H. Sadka, and A.M. Kondoz, 3D video compression using MPEG-4-multiple auxiliary component (MPEG4-MAC), Proc IEE 2nd International Conference on Visual Information Engineering (VIE2005), UK, Glasgow, April 2005.

ISO/IEC JTC1/SC29/WG11, Requirements and application descriptions on multi-view video coding, Doc. N9543, October 2007a.

A. Smolic and D. McCutchen, 3DAV exploration of video-based rendering technology in MPEG, Proc IEEE Trans Circuit Syst Video Technol 14(2004), 348–356.

A. Smolic, H. Kimata, and A. Vetro, Development of MPEG standards for 3D and free viewpoint video, Proc SPIE Opt East, Three-Dimensional TV, Video, and Display IV, Boston, MA, October 2005.

ISO/IEC JTC1/SC29/WG11, Call for comments on 3DAV, Doc. N6051, October 2003.

ISO/IEC JTC1/SC29/WG11, Survey of algorithms used for multi-view video coding (MVC), Doc. N6909, January 2005.

H.Y. Shum, J. Sun, S. Yamazaki, Y. Lin, and C.K. Tang, Pop-up light field: An interactive image-based modeling and rendering system, Proc SIGGRAPH (ACM Trans Graphics) 23 (2004), 143–162.

C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, High-quality video view interpolation using a layered representation, Proc SIGGRAPH (ACM Trans Graphics), August 2004, pp. 600–608.

Z.F. Gan, S.C. Chan, K.T. Ng, and H.Y. Shum, An object-based approach to plenoptic videos, Proc Int Symp Circuits Syst (ISCAS2005), May 2005, pp. 3435–3438.

J. Shade, S. Gortler, L.W. He, and R. Szeliski, Layered depth images, Proc SIGGRAPH (ACM Trans Graphics), July 1998, pp. 231–242.

C. Chang, G. Bishop, and A. Lastra, LDI tree: A hierarchical representation for image-based rendering, Proc SIGGRAPH (ACM Trans Graphics), August 1999, pp. 291–298.

L. McMillan and G. Bishop, Plenoptic modeling: An image-based rendering system, Proc SIGGRAPH (ACM Trans Graphics), August 1995, pp. 39–46.

P.E. Debevec, Y. Yu, and G. Borshukov, Efficient view-dependent image-based rendering with projective texture-mapping, Proc Eurographics Workshop on Rendering, 1998, pp. 150–116.

ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Joint multiview video model (JMVM) 5.0, Doc. JVT-X207, July 2007b.

ISO/IEC JTC1/SC29/WG11, Proposal on standardization of free viewpoint TV (FTV), Doc. M13612, July 2006.

ISO/IEC JTC1/SC29/WG11, Proposal on requirements for FTV, Doc. M14417, April 2007c.

ISO/IEC JTC1/SC29/WG11, Preliminary FTV model and requirements, Doc. N9168, July 2007d.

S.C. Chan, H.Y. Shum, and K.T. Ng, Image-based rendering and synthesis, Proc IEEE Signal Process Mag 24 (2007), 22–33.

M. Levoy and P. Hanrahan, Light field rendering, Proc SIGGRAPH (ACM Trans Graphics), August 1996, pp. 31–42.

S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, The lumigraph, Proc SIGGRAPH (ACM Trans Graphics), August 1996, pp. 43–54.

M. Tanimoto, FTV (Free Viewpoint Television) creating ray-based image engineering, Proc ICIP 2 (2005), 25–28.

M. Droese, T. Fujii, and M. Tanimoto, Ray-Space interpolation based on filtering in disparity domain, Proc 3D Conference 2004, Tokyo, Japan, June 2004, pp. 213–216.

ISO/IEC JTC1/SC29/WG11, Description of exploration experiments in FTV, Doc. N9230, July 2007e.

Z. Wang, A.C. Bovik, H.R. Sheik, and E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, Proc IEEE Trans Image Process 13 (2004), 600–612.

F. Xiao, DCT-based video quality evaluation, Final Project for EE392J Stanford Univ, 2000.

MSU Video Quality Measurement Tool. Available at: http://compression.ru/video/quality_measure/.

A. Fusiello, Uncalibrated euclidean reconstruction: A review, Proc Image Vis Comput 18 (2000), 555–563.

E. Trucco and A. Verri, Introductory techniques for 3D computer vision, Prentice-Hall, Upper Saddle River, NJ, 1998.

R. Hartley and A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, UK, 2000.

J.A. Bangham and S. Marshall, Image and signal processing with mathematical morphology, Proc Electron Commun Eng J 10 (1998), 117–128.

MSR 3D Video Sequences. Available at:http://www.research.microsoft.com/vision/ImageBasedRealities/3DVideoDownload/.

C.J. van den Branden Lambrecht and O. Verscheure, Perceptual quality measure using a spatio-temporal model of the human visual system, Proc SPIE 2668 (1996), 450–461.

ISO/IEC JTC1/SC29/WG11, Improvement of depth map estimation and view synthesis, Doc. M15090, January 2008a.

U. Fecker, M. Barkowsky, and A. Kaup, Improving the prediction efficiency for multi-view video coding using histogram matching, Proc ICIP, 2007.

A. Telea, An image inpainting technique based on the fast marching method, Proc J Graphics Tools 9 (2004), 25–36.

ISO/IEC JTC1/SC29/WG11, Reference softwares for depth estimation and view synthesis, Doc. M15377, April 2008b.