Space-Time Hole Filling With Random Walks in View Extrapolation for 3D Video

Sunghwan Choi, *Student Member, IEEE*, Bumsub Ham, *Student Member, IEEE*, and Kwanghoon Sohn, *Senior Member, IEEE*

***Abstract*— In this paper, a space-time hole filling approach is presented to deal with a disocclusion when a view is synthesized for the 3D video. The problem becomes even more complicated when the view is extrapolated from a single view, since the hole is large and has no stereo depth cues. Although many techniques have been developed to address this problem, most of them focus only on view interpolation. We propose a space-time joint filling method for color and depth videos in view extrapolation. For proper texture and depth to be sampled in the following hole filling process, the background of a scene is automatically segmented by the random walker segmentation in conjunction with the hole formation process. Then, the patch candidate selection process is formulated as a labeling problem, which can be solved with random walks. The patch candidates that best describe the hole region are dynamically selected in the space-time domain, and the hole is filled with the optimal patch for ensuring both spatial and temporal coherence. The experimental results show that the proposed method is superior to state-of-the-art methods and provides both spatially and temporally consistent results with significantly reduced flicker artifacts.**

***Index Terms*— Depth image-based rendering, hole filling, inpainting, random walks, temporal coherence, view synthesis.**

I. Introduction

THREE-DIMENSIONAL video technologies offer more realistic experience to the user through stereoscopic movies and television, video games, and clinical applications. Autostereoscopic displays provide 3D depth perception without the need for wearing glasses by simultaneously showing a number of slightly different views [1]. These views have been captured by using a number of real cameras which are placed at different viewpoints. It needs, however, high transmission bandwidth and data storage requirements as well as considerable equipment cost for multi-camera setups. Consequently, synthesizing *virtual* views from limited views is of increasing interest.

The depth image-based rendering (DIBR) is one of the most important techniques to render virtual views at different viewpoints using a 3D warping process, where a color image and its associated per-pixel depth information are exploited based on the principle of projective geometry [2]. During a 3D warping process, some parts of an occluded background region might be visible in a synthesized virtual view, i.e., a disoccluded region. It is referred to as a "*hole*" in the literature. When a virtual camera is located between two real cameras, most hole regions can be filled by combining image and depth information from multiple views. However, in some applications, the view extrapolation is inevitable, especially when the end-to-end coverage of successive pairs of cameras is insufficient to the requirement of the autostereoscopic system for sensible 3D viewing. Therefore, the hole filling problem becomes more crucial when a virtual view is located beyond the field of view of real cameras. In this case, the hole filling problem is nontrivial, since the virtual view should be extrapolated from the nearest single view only. Furthermore, there exists no texture related to these hole regions. Such challenging condition prevents the use of conventional hole filling methods commonly used in the view interpolation.

In this paper, a hole filling framework for extrapolating a virtual view from a single view is presented. In a 3D video,[1] the proposed method fills the hole regions by using the space-time exemplar-based inpainting method. Based on the geometric characteristics of hole occurrence, i.e., the hole is mainly located between the foreground object and the background scene in that the 3D warping process alters the position of foreground objects, hole boundaries are labeled as "foreground" or "background." With this, the background of the scene is then automatically segmented by making use of the random walker segmentation [3]; thus, the hole areas are filled from the background textures only. In our framework, we introduce a new probabilistic filling strategy, i.e., patch candidates are dynamically estimated in a space-time domain. The patch candidate selection is posed as a labeling problem, and it is solved with the random walks. The quality of virtual views is thus ensured to be visually plausible and time-consistent. This paper extends our early work [4] in a number of ways. First, we review the random walk-based inpainting process in detail and extend the algorithm in [4] to deal with videos as well as images. Second, an intensive experimental study and comparison with other methods are presented.

Manuscript received June 11, 2012; revised February 21, 2013; accepted February 21, 2013. Date of publication March 7, 2013; date of current version April 17, 2013. This work was supported by The Ministry of Knowledge Economy, Korea, under the Information Technology Research Center Support Program Supervised by the National IT Industry Promotion Agency under Grant NIPA-2012-H0301-12-1008. The associate editor coordinating the review of this paper and approving it for publication was Dr. Stefan Winkler.

The authors are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: shch@yonsei.ac.kr; mimo@yonsei.ac.kr; khsohn@yonsei.ac.kr).

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/TIP.2013.2251646

[1]The 3D videos refer to single color and its associated depth videos in this literature.

This paper is organized as follows. Section II presents the related works, and Section III briefly reviews the random walker segmentation algorithm. The proposed inpainting framework is presented in Section IV. Then, the performance of the proposed method is demonstrated in Section V. Finally, Section VI discusses limitations and concludes this paper.

II. Related Works

In the literature, a number of methods have been proposed for filling a hole region during view synthesis. They can be generally classified into two categories based on the domains of operation, either depth-based or image-based methods.

The depth-based methods handle hole regions by applying a low-pass filter to the depth map as a pre-processing step in the DIBR process [1], [2], [5], [6]. A Gaussian filter has been applied to the depth map to minimize or completely remove hole regions. The main drawback of these approaches is that they change the scene geometry, i.e., smoothing lowers the depth gradients, introducing geometric distortions.

The image-based methods fill the hole regions in the image domain by applying either an averaging filter [7] or an advanced inpainting method [8], [9]. The candidates for recovering a hole region are sampled from within its vicinity, and then an adequate pixel or a patch is inserted into the hole region. In [7], the hole region was filled by using the depth-guided interpolation in which the neighboring color pixels having a low depth value were used. The alternative works [8], [9] adopted an inpainting algorithm for the filling process. Inpainting methods infer the missing information by propagating structure and texture information from known regions to regions to be filled. In general, two approaches are described: the structure-based and the exemplar-based methods. The structure-based inpainting methods leverage partial differential equations (PDEs), where structure components such as edges are diffused into the hole region [10], [11]. The recovered region is semantic compared to the interpolation, since the structure components are continued toward the hole region. The diffusion, however, causes the filled region to be blurred when the hole area is large. Consequently, the structure-based inpainting [8], [9] suffers from blurring artifacts and is applicable only to the small baseline configuration. The exemplar-based inpainting methods recover missing information by propagating structure as well as texture information into the hole region [12], [13]. Based on the priority-based filling strategy, these approaches update an unknown region with an exemplar patch that best matches the region to be filled. Compared to the structure-based methods, the exemplar-based inpainting methods typically yield better results without introducing blur artifacts at large hole regions [13].

State-of-the-art algorithms for hole filling in view synthesis have adapted the exemplar-based inpainting method [4], [14], [15], [16]. The geometric model of view synthesis can be modeled into the filling priority. In [14], Azzari *et al.* filled the hole region with non-local means of similar patches. Daribo and Saito [15] introduced the depth similarity cost in a patch matching procedure to restrain the samples coming from a background region. The inverse variance of depth

(a) (b) (c) (d) (e) (f) (g)

Fig. 1. Examples of the problem of foreground texture penetration in existing algorithms. (a) Hole regions. (b) Criminisi's method [13]. (c) Daribo's method [15]. (d) Wexler's method [21]. (e) VSRS [27]. (f) Proposed method. (g) Ground truth. Hole regions are painted in yellow.

information was used in the priority computation, so that foreground and background parts can be implicitly distinguished. From the algorithmic perspective, this work lies in a line of work that started from Efros and Leung [12], through Criminisi *et al.* [13], to Daribo and Saito [15]. The existing image-based hole filling algorithms mainly suffer from the penetration problem of the foreground texture, since the fill front and the sample space have been set without distinguishing between the foreground and the background. Therefore, the region at the foreground object with the highest priority might be extended inward the hole region, which leads to penetration of foreground textures as shown in Fig. 1. It can be remedied by constraining only the background to be the sample space: the background modeling methods such as the dynamic scene reconstruction [17], [18], [19] and the background subtraction methods [20] can be employed, where the foreground and background models of the scene were derived from the silhouette information. However, in case that the modeled object has a similar color appearance to the background, these methods are unstable [17], [20]. Another drawback is that they only work frame-by-frame, thus temporal correlation in a video is ignored, causing annoying flicker artifacts in a rendered video.

For a decade, various video inpainting methods have been proposed to maintain temporal consistency between frames. A space-time patch matching criterion is mainly considered for maintaining temporal consistency in a filled area [21], [22]. Wexler *et al.* [21] posed the hole filling problem as a global optimization to enforce global coherence. Specifically, the mean-shift voting scheme was used to select the optimal one from the estimated space-time patch candidates. Also, it was proved that the object trajectory in filled areas is successfully recovered by using the space-time patch matching criterion. Inspite of active research on video inpainting, there have been only a few attempts to maintain the temporal consistency of a filled region in view synthesis [23], [24]. Furthermore, none of them focus on the hole filling problem for view extrapolation.

Recently, stereoscopic inpainting approaches for jointly filling color and depth information have been proposed [34], [35]. They aim to fill the hole regions located in either mutually consistent region (seen in both images) or half-occluded region (seen in one image and occluded in the other) in a stereo image pair. The mutually consistent region must guarantee stereo consistency for 3D viewing. For this, the works of [34] and [35] use the weak stereo consistency constraint to encourage the appearance of the filled region to look identical.

In contrast, the half-occluded region has no depth cues and thus, the stereo consistency in this region is not valid. This region should be filled with the unoccluded parts of the scene by using monocular information only. In our method, we deal with the hole regions which originate from half-occlusion. The hole regions, therefore, are filled with the unoccluded parts of the scene as in [34] and [35].

III. Review on Random Walker Segmentation

We now briefly introduce the random walker segmentation algorithm [3]. Let us assume that an image is represented by a weighted and undirected graph $G = (V, E)$, where V is a set of vertices (nodes) and E is a set of edges. A node $v \in V$ corresponds to a pixel of the image, and an edge $e \in E \subset V \times V$ represents the connection between the nodes. An edge connecting two nodes v_i and v_j is denoted by e_{ij}. With a user-interaction, V can be classified into a set of initially labeled nodes V_M and a set of unlabeled nodes V_U, such that $V = V_M \cup V_U$. The weight imposed on an edge e_{ij} is denoted by $w(e_{ij})$ or simply w_{ij}, which describes the cost affecting a random walker's choice. Since the graph is undirected, $w_{ij} = w_{ji}$. Typically, the Gaussian kernel is used to represent the gray-scale image structure as follows:

$$w_{ij} = \exp\left(-\beta(g_i - g_j)^2\right) \tag{1}$$

where g_i corresponds to the image intensity value at node v_i, and β is a free parameter that controls the kernel bandwidth.

Based on the connection between the random walks and potential theory, inferring random walker probabilities is equivalent to solving the combinatorial Dirichlet problem with prescribed boundary conditions [3]. A harmonic function u is the solution for the Dirichlet problem. It satisfies the Laplace equation $\Delta u = 0$ and minimizes the Dirichlet integral $\int_\chi |\nabla u|^2 d\chi$ with a domain χ, where Δ and ∇ are the Laplacian and differential operator, respectively.

For discretizing the Dirichlet integral, we define the combinatorial Laplacian matrix as

$$L_{ij} = \begin{cases} d_i, & \text{if } \ i = j \\ -w_{ij}, & \text{if } \ v_i \text{ and } v_j \text{ are adjacent nodes} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where d_i is the degree of node v_i (i.e., $d_i = \sum_j w(e_{ij})$). Typically, a 4-neighborhood system is used to define the relation between neighboring nodes. Note that the random walks formulation can easily be extended to the space-time domain by adapting a high order neighborhood system in the Laplacian matrix.

The use of the Laplacian matrix allows us to formulate the combinatorial Dirichlet integral as

$$D[x] = \frac{1}{2} x^T L x \tag{3}$$

where x is a harmonic function that minimizes (3) and is a vector that consists of all nodes on the graph. Note that it is a weighted discrete version of the Dirichlet integral. Pre-labeled seed points V_M are used to prescribe boundary conditions for the Dirichlet problem, where the locations of the seed points are fixed at unity while others are set to zero. To infer the probabilities that a random walker starting at a seed point first reaches the remaining unseeded points, elements in L and x are reordered, such that (3) can be rewritten as

$$\begin{aligned} D[x_U] &= \frac{1}{2} \begin{bmatrix} x_M^T & x_U^T \end{bmatrix} \begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix} \begin{bmatrix} x_M \\ x_U \end{bmatrix} \\ &= \frac{1}{2} \left(x_M^T L_M x_M + 2 x_U^T B^T x_M + x_U^T L_U x_U \right) \end{aligned} \tag{4}$$

where L_M and L_U represent the intra-dependences of the seeded points and the unseeded points, respectively, and B and B^T describe the inter-dependences between the seeded and unseeded points. Differentiating $D[x_U]$ with respect to x_U and calculating the root yields a system of linear equations described by

$$L_U x_U = -B^T x_M. \tag{5}$$

Let $S = \{s_k | k = 1 \ldots K\}$ denote the set of labels, where each label s_k is given by the user-interaction, and K is the number of labels. We introduce the $|V_M| \times 1$ vector f^{s_k} for each label s_k and define the value at node $v_j \in V_M$ as

$$f_j^{s_k} = \begin{cases} 1, \text{ if } \ Q(v_j) = s_k \\ 0, \text{ if } \ Q(v_j) \neq s_k \end{cases} \quad \forall v_j \in V_M \tag{6}$$

where $Q(v_j)$ represents a label function for the seeded point $v_j \in V_M$ which returns the node's associated label $s_k \in S$. Let $x_i^{s_k}$ denote a random walker's probability at node $v_i \in V_U$ for each label $s_k \in S$. The solution to (5) for each label s_k can be calculated by solving

$$L_U x^{s_k} = -B^T f^{s_k} \tag{7}$$

where x^{s_k} denotes the $|V_U| \times 1$ vector consisting of random walker's probabilities $x_i^{s_k}$ for all nodes $v_i \in V_U$. The segmentation is achieved by assigning the label s_k to each unlabeled node $v_i \in V_U$ based on the probability $x_i^{s_k}$ that a random walker starting from node v_i first reaches the initially-s_k-labeled node $v_j \in V_M$. Finally, each node is classified by the label that has the highest probability.

We emphasize that every unlabeled node is assigned to the probability that can be interpreted as a confidence level for being a specific node (i.e., prelabeled node). In our hole filling framework, 1) we classify a 3D video into foreground and background labels using the random walker segmentation algorithm in order to explicitly define the sample space, and 2) we further utilize the random walker's probabilities to select candidate patches in a space-time domain in order to yield an optimal patch for filling.

IV. Proposed Hole Filling Framework

The algorithm uses color images and their associated depth maps of a *warped* 3D video as input. The system diagram of the proposed hole filling framework is presented in Fig. 2.

The proposed method consists of two main stages: sample space selection and space-time hole inpainting. In the first stage, hole boundaries are labeled as "foreground" or "background" based on the geometric characteristics of hole occurrence [16]. With knowledge of hole boundary classification, the sample space of each frame is then selected using

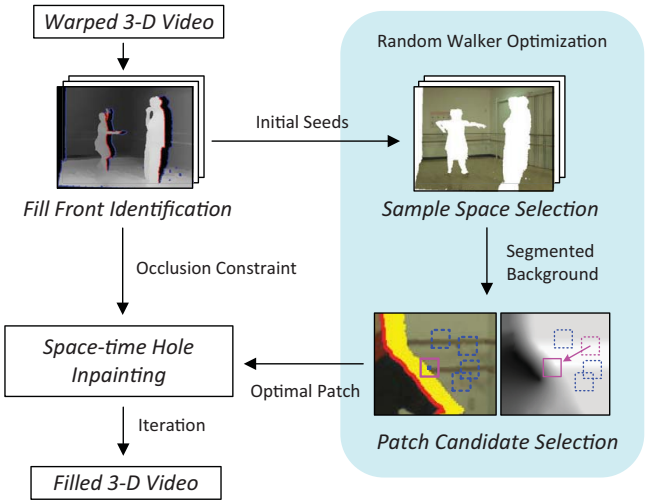


Fig. 2. Proposed hole filling framework for view extrapolation. The proposed method consists of two main stages. In the first stage, hole boundaries are identified as "foreground" or "background." These labeled boundaries then serve as initial seeds for selecting the sample space. In the second stage, the space-time inpainting process is performed using the identified boundaries and the sample space. The candidate patches are dynamically selected, and the optimal one is used to fill the hole region.

the random walker segmentation algorithm [3]. In the second stage, we perform space-time inpainting to fill the hole regions with information from the sample space. The filling order is prioritized based on the occlusion constraint. The optimal patch is selected from the candidate patches in a space-time domain. The patch candidate selection is formulated as a labeling problem which we solve with random walks. The random walker's probabilities are used to identify candidates for the optimal patch. The hole filling process iterates until there is no hole region.

For ease of algorithm exposition, we adopt notation used in the inpainting literature [13]. The important symbols are listed in Table I.

TABLE I
NOTATIONS FOR HOLE FILLING FRAMEWORK

$\mathbf{p}$	Point in an image domain
I^t	Input warped image at frame t (N: maximum frame)
D^t	Input warped depth map at frame t
Ω	Hole region
$\delta\Omega$	Fill front (hole contour)
Φ	Sample space (segmented background region)
$\hat{\Phi}$	Region of patch candidates
$\Psi_{\mathbf{p}}$	2D patch centered at point $\mathbf{p}$
Ψ_i	2D patch centered at node v_i
$\Gamma_{\mathbf{p}}$	Space-time patch centered at point $\mathbf{p}$
$x^s_{\mathbf{p}}$	Random walker's probability of the label s at point $\mathbf{p}$

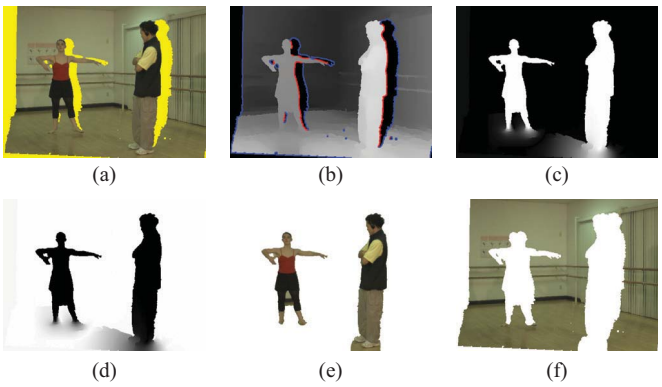


Fig. 3. Sample space selection process ("Ballet" sequence, frame 45). (a) Hole regions. (b) Foreground label (s_1, red) and the background label (s_2, blue) serve as initial seeds on the depth map. (c) and (d) Random walker's probability map of the foreground label (x^{s_1}) and the background label (x^{s_2}), respectively. By assigning each unlabeled pixel to the label with the highest probability, the segmented regions associated with the (e) "foreground" label and the (f) "background" label can be obtained.

A. Sample Space Selection

1) Fill Front Identification [16]: As the boundary of the hole region is generated by splitting the depth discontinuity, the fill front can be efficiently identified according to the sign of the Laplacian of a depth map. By applying the Laplacian operator to the depth map, the pixel along the hole boundary with a negative (positive) Laplacian value belongs to the foreground (background) boundary, since the higher (lower) depth value corresponds to the foreground (background) boundary at the depth discontinuity. It can be formulated as

$$F^t(\mathbf{p}) = \begin{cases} 0, \text{ if } (\Delta D_0^t)_W(\mathbf{p}) < 0 \\ 1, \text{ if } (\Delta D_0^t)_W(\mathbf{p}) > 0 \end{cases} \quad \forall \mathbf{p} \in \delta\Omega^t \tag{8}$$

where D_0^t is a reference depth map at frame t used in a warping process, and $\delta\Omega^t$ is the contour of the hole at frame t. Subscript W represents a warping operator, such that $(D_0^t)_W = D^t$. $(\Delta D_0^t)_W$ represents the warped Laplacian of the reference depth map. Note that the Laplacian operator is applied to D_0^t, and then ΔD_0^t is warped to the desired virtual viewpoint. Finally, as shown in Fig. 3, the hole boundary is classified into foreground ($\delta\Omega_F$) and background ($\delta\Omega_B$) boundaries based on the sign of the warped Laplacian of the reference depth map, such that $\delta\Omega^t = \delta\Omega^t_F \cup \delta\Omega^t_B$ and $\delta\Omega^t_F \cap \delta\Omega^t_B = \emptyset$. That is, a hole boundary that belongs to the foreground (background) is defined as non-occluded (occluded) boundary pixels that circumscribe a hole region. In our framework, we define the identified background boundary $\delta\Omega^t_B$ as the fill front for each frame t. With the hole boundary classification data, in the next section, we will explain how the sample space is defined in the 3D video in order to handle the foreground texture penetration problem.

2) Random Walks for Sample Space: For constraining background regions to be the sample space, it is needed to segment the 3D video into foreground and background regions.

This process can be seen as reconstructing the background model of the scene. Recently, some relevant works have been developed such as the dynamic scene reconstruction [17], [18], [19] and the background subtraction [20]. In the dynamic scene reconstruction, since a complete model of the scene were generated from all the available information from multiple views, the sample space that build upon the complete scene model increased the possibility that the candidate patches used in the hole filling contain a ground truth texture. In case of a stationary camera motion with a given single view, background subtraction methods can also be employed to define the

sample space. In our framework, the above reconstruction methods might be incorporated into our algorithm according to the camera configuration. These methods are, however, usually unstable when the modeled object has a similar color appearance to the background [19], [20]. Also, prior knowledge about the scene is needed to infer initial models [17], [18]. Instead, we solve the problem of background model reconstruction with the random walker segmentation using a *depth* map. Thanks to the initial labels of (8), the random walker algorithm can fully automate the task of foreground and background segmentation in contrast to the original work [3] which requires user-specified seeded points.

We apply the random walker segmentation algorithm [3] to the associated depth map as shown in Fig. 3. For each frame t, we represent a depth map D^t as a weighted and undirected graph $G = (V, E)$. We modify the original weighting function for the random walker bias (discussed in Section III) as

$$w_{ij} = \exp(-\sigma_D |\mathbf{d}_i^t - \mathbf{d}_j^t|) + \varepsilon \tag{9}$$

where $\mathbf{d}_i^t$ represents the depth value at node v_i in frame t. ε is a small constant (e.g., 10^{-6}), and σ_D balances the sensitivity of the depth similarity cost (e.g., $\sigma_D = 80$ in our experiment). Note that depth values are normalized prior to applying (9), such that $0 \leq |\mathbf{d}_i^t - \mathbf{d}_j^t| \leq 1$.

Based on the fill front identification result in (8), we can automatically assign initial seeds to the random walker algorithm. Let us define a label set $S = \{s_1, s_2\}$, where labels s_1 and s_2 correspond to the foreground and background, respectively. As shown in Fig. 3, we assign labels s_1 and s_2 to the nodes that correspond to the foreground boundary points $\delta\Omega_F$ (red) and the background boundary points $\delta\Omega_B$ (blue), respectively. A node $v_i \in V_M$ that is assigned to either s_1 or s_2 serves as an initial seed in the segmentation process. With initial seeds, segmentation is achieved by solving the algebraic function in (7). The probability map for each label can then be obtained as shown in Fig. 3(c) and (d).

By assigning each unlabeled node to the label with the highest probability, the segmented regions for each frame t can be obtained as shown in Fig. 3(e) and (f). The segmented region associated with the foreground s_1 [Fig. 3(e)] and the background s_2 [Fig. 3(f)] are represented by Φ_F^t and Φ_B^t, respectively. Hereafter, we define the sample space at frame t as $\Phi^t \triangleq \Phi_B^t$. As the initial seeds largely affect the segmentation quality of a depth map, we apply an erosion operation to remove isolated seed points prior to using them in the segmentation process. In our framework, the sample space selection procedure is applied to all of the frames before performing the space-time hole inpainting procedure. The classified hole boundary and the segmented background information work as constraints, which will be described in the following section.

B. Space-time Hole Inpainting

We now focus on the joint color and depth filling of the 3D video considering space-time consistency, and expose the strategy for the candidate patch selection under the space-time random walks formulation.

The proposed hole inpainting algorithm inwardly propagates the textural information from the sample space into the interior of the hole region. Traditional exemplar-based inpainting methods search for the optimal exemplar within a rectangular region of pixels using sliding windows. In the proposed method, however, the basic unit is a space-time patch. Hence, both spatial and temporal correlations are taken into account during the inpainting process by enlarging the sampling region to the temporal neighborhood.

1) Priority Computation With Occlusion Constraint: To handle the problem of foreground texture penetration caused by depth ambiguities, we constrain the foreground points on the fill front to have zero priority using the fill front identification result from (8). Furthermore, it is preferred that a higher priority is assigned to a homogeneous depth region, since the patch lying on the intersection between different depth regions may have foreground textures. To this end, the scene structure around the hole area is estimated based on the depth distribution along the fill front. These key constraints are modeled as an occlusion constraint term in the filling priority formulation. Assuming that a frame t is considered, we neglect the superscript t in the following equations for the sake of simplicity.

The proposed priority $P(\mathbf{p})$ is defined as the product of three terms

$$P(\mathbf{p}) = C(\mathbf{p}) \cdot G(\mathbf{p}) \cdot Z(\mathbf{p}) \tag{10}$$

where $C(\mathbf{p})$ is the *confidence* term that indicates the number of valid pixels in the source region, $G(\mathbf{p})$ is the *data* term that represents the strength of the isophotes such as edges. $Z(\mathbf{p})$ is the *occlusion constraint* term that prevents the penetration of foreground textures due to depth ambiguities and it gives special priority to the homogeneous depth plane. $C(\mathbf{p})$ and $G(\mathbf{p})$ are defined as

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Gamma_\mathbf{p} \cap \Phi} C(\mathbf{q})}{|\Gamma_\mathbf{p}|} \quad \text{and} \quad G(\mathbf{p}) = \frac{\left| \nabla I_\mathbf{p}^\perp \cdot \mathbf{n}_\mathbf{p} \right|}{\alpha} \tag{11}$$

where $|\Gamma_\mathbf{p}|$ represents the area of $\Gamma_\mathbf{p}$ and α is a normalization constant (e.g., $\alpha = 255$ for a typical gray-scale image). $\nabla I_\mathbf{p}$ denotes gradient of an image I at the point $\mathbf{p}$. $\mathbf{n}_\mathbf{p}$ represents the vector normal to the hole boundary. $\perp$ is an orthogonal operator. $C(\mathbf{p})$ is initialized to one prior to filling hole regions, and is updated in a recursive manner as the filling process proceeds. Note that $C(\mathbf{p})$ is summed over pixels in a space-time patch. In this manner, the patch with smaller hole region is assigned a higher priority. The patch lying on the continuation of the strong edges is assigned a higher priority by means of $G(\mathbf{p})$, such that significant edges should be continued in the region to be filled. The occlusion constraint term $Z(\mathbf{p})$ is defined as follows:

$$Z(\mathbf{p}) = H(\mathbf{p}) \left\{ 1 - \frac{\max_{\mathbf{q},\mathbf{r} \in \Gamma_\mathbf{p} \cap \Phi} d(\mathbf{q}, \mathbf{r})}{\eta_Z} \right\} \tag{12}$$

where $d(\mathbf{a}, \mathbf{b})$ represents the absolute difference between the depth values of $\mathbf{a}$ and $\mathbf{b}$. η_Z represents a normalizing factor (e.g., $\eta_Z = 255$ as the maximum depth value). $H(\mathbf{p})$ is a binary function that constrains the points of the fill front along the foreground side to be zero priority. $H(\mathbf{p})$ is initialized

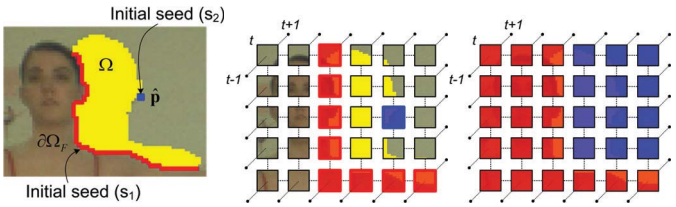


Fig. 4. Illustration of the patch candidate selection procedure with the random walks. (a) Initial seeds of "foreground" (s_1, red) and "background" (s_2, blue) are served at frame t. (b) Each node represents a 2D patch Ψ_i at node v_i. A node that serves as an initial seed is represented as a red or blue box. The red and blue boxes represent the patches centered at each of the red and blue points in (a). (c) After solving a space-time labeling problem with the random walks, each unlabeled node is assigned to its corresponding label, where the red box and blue box correspond to "foreground" and "background," respectively. In addition, the probability of being a certain label is calculated on each unlabeled node. Note that (b) and (c) are a downsampled set of patches from (a) for visualization purposes.

to $H(\mathbf{p}) = F(\mathbf{p})$ for all $\mathbf{p}$, where $F(\mathbf{p})$ is the fill front identification result from (8). $Z(\mathbf{p})$ prefers homogeneous depth areas to handle depth ambiguities in such a way that it assigns a lower priority to patches lying on the intersection between the foreground and background regions in reference to the maximum difference between depth values in $\Gamma_{\mathbf{p}}$. With the occlusion constraint term, artifacts caused by the problem of foreground texture penetration are successfully remedied. After the priority of all the points on the fill front is computed, the patch centered at a pixel with a maximum priority value is selected as the starting point for filling.

2) Patch Candidates as Labeling Problem: Once a point with a maximum priority is selected (i.e., $\hat{\mathbf{p}} = \arg\max_{\mathbf{q}} P(\mathbf{q})$), we estimate the patch candidates which share a significant similarity with the patch $\Psi_{\hat{\mathbf{p}}}$ (i.e., missing patch). In contrast to the conventional method [13], we explore a dynamically generated random walk-based region for candidate patches to select the optimal replacement patch.

For finding the patch candidates, we pose this problem as a labeling problem as illustrated in Fig. 4. Given a set of warped virtual views $Y = \{I^t | \forall t = 1 \ldots N\}$, we introduce a patch-based weighed and undirected 3D graph representation on Y, where a weight imposed on each edge represents the patch dissimilarity cost of neighboring patches over the visible pixels. Note that as opposed to the typical graph formulation described in Section III, in which a node corresponds to its associated pixel location in the image domain, we represent an image as a graph encoding a 2D rectangular patch to let a random walker travel an entire graph for a given patch dissimilarity cost. In addition, we use a 6-neighborhood system in a graph representation to account for temporal information.

We generate a 3D subgraph of size $w_g \times h_g \times t_g$ centered at the maximum priority point $\hat{\mathbf{p}}$. The size should be sufficiently larger than the size of $\Gamma_{\hat{\mathbf{p}}}$ and typically smaller than the size of the space-time volume for computational efficiency. The weight between nodes is defined as

$$w_{i\,j} = M(v_i)M(v_j)\exp(-\sigma_I R(v_i, v_j)) + \varepsilon \qquad (13)$$

where σ_I is a weighting factor, e.g., $\sigma_I = 900$ in our experiment, which controls the sensitivity of the patch difference.

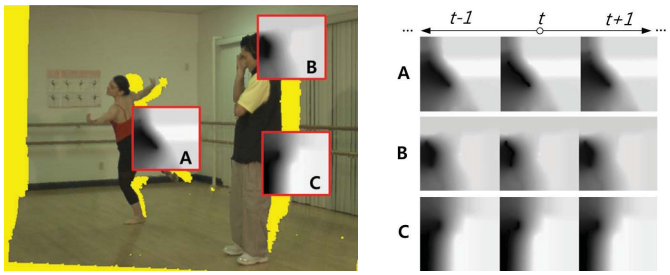


Fig. 5. (a) Examples of the patch candidate selection process when the highest priority patch is selected at the subgraph of A, B, or C. (b) Probability indicating the confidence level of being an optimal patch (i.e., "background" label) is calculated via the random walks. A higher intensity value corresponds to a higher probability.

$\varepsilon = 10^{-6}$ is a small constant. $M(v_i)$ is a verification function which yields 1 (or 0) if the shape of the source region of both Ψ_i and the missing patch $\Psi_{\hat{\mathbf{p}}}$ are the same (or different). Thereby, the random walker bias is only defined on the edge connecting nodes that have no missing information in the source region. $R(v_i, v_j)$ represents the patch dissimilarity cost between two nodes v_i and v_j as follows:

$$R(v_i, v_j) = \frac{1}{\eta_S} \sum_{\mathbf{r} \in \overline{\Psi}_i} \sum_{\mathbf{s} \in \overline{\Psi}_j} \| I(\mathbf{r}) - I(\mathbf{s}) \| \qquad (14)$$

where $\|I(\mathbf{r}) - I(\mathbf{s})\|$ is the Euclidean distance between pixel values at point $\mathbf{r}$ and $\mathbf{s}$. η_S is a normalizing factor. $\overline{\Psi}_i$ represents the set of valid points in the patch at node v_i, where a valid point corresponds to the already filled point of the missing patch $\Psi_{\hat{\mathbf{p}}}$. Pixel colors are represented in the CIE Lab color space. The reason for introducing the patch dissimilarity feature as in (14) is that the candidate patches must have *textural* correlation to the missing patch. Note that the weight function in [3] operates on a pixel-wise basis, which means only the intensity correlation is taken into account. We construct a Laplacian matrix with (2) where the relation between adjacent nodes is extended to the temporal domain (i.e., a 6-neighborhood system).

From the fill front identification result in (8), the nodes associated with the foreground boundary $\delta\Omega_F$ as well as the node with the highest priority $\hat{\mathbf{p}}$ are assigned to the initial seeds, as shown in Fig. 4(a). Given a label set $S = \{s_1, s_2\}$, the foreground boundary nodes (red line) are set to foreground seeds s_1, while the node of the current missing patch (blue dot) is set to an background seed s_2. Finally, candidate patches with the associated random walker's probability can be inferred by finding the roots of $L_U x^s = -B^T f^s$ for all labels. By assigning each unlabeled patch to the label with the highest probability, the region of patch candidates $\hat{\Phi}$ can be dynamically obtained as shown in Fig. 4(c). We define the region of patch candidates $\hat{\Phi}$ as the nodes assigned to the label background (blue). Fig. 5 shows examples of the patch candidate selection process in a space-time domain. Suppose that the filling starts at the subgraph of either A, B, or C. The probability calculated via the random walks indicates the confidence level of being an optimal patch. As shown in Fig. 5(b), the nodes with strong textural correlation to the background seed show a higher probability and are mainly

(a) (b) (c)

Fig. 6. Result of depth filling in "Ballet" at frame 11. (a) Hole regions painted in yellow. (b) Filled depth map. (c) Ground truth.

located in the neighboring background region, while the nodes in the foreground region have relatively a lower probability.

Now, we have the set of candidate patches and the corresponding probability maps. They are fully exploited in the next section to pick an ideal patch for current location to be filled.

3) Optimal Patch with Temporal Coherence: The optimal patch is selected over the region of candidate patches defined on the space-time domain according to the weighted sum of squared difference (SSD) criterion. The weight used in the SSD is defined as

$$W(\mathbf{p}) = 1 - x_{\mathbf{p}}^{s_2} \tag{15}$$

where $x_{\mathbf{p}}^{s_2}$ denotes the random walker's probability of the label background at point $\mathbf{p}$. A patch that has a high probability has a low weight value, such that the candidate patches having a high confidence are preferred in the space-time patch matching process. The optimal patch selection is performed as

$$\Gamma_{\hat{\mathbf{q}}} = \underset{\Gamma_{\mathbf{q}} \in \hat{\Phi} \cap \Phi}{\arg\min}\, W(\mathbf{q}) \mathrm{SSD}(\Gamma_{\hat{\mathbf{p}}}, \Gamma_{\mathbf{q}}) \tag{16}$$

where $SSD(\cdot,\cdot)$ represents the SSD between the previously filled pixels in two space-time patches. Note that foreground textures in a space-time patch can have relatively low matching costs, which should be prevented for lowering mismatching error. We thus remove the region associated with the foreground using the pre-defined sample space Φ. Accordingly, the sampling density of the patches for the probability-based weighted patch matching procedure is defined by the intersection of the region associated with the sample space Φ and the region of patch candidates $\hat{\Phi}$.

Once the best matching patch $\Gamma_{\hat{\mathbf{q}}}$ is found, each pixel-to-be-filled in $\Psi_{\hat{\mathbf{p}}}$ is copied from its corresponding position in $\Gamma_{\hat{\mathbf{q}}}$. This means that searching for the best patch exemplar proceeds in a space-time domain, whereas the filling is performed in a space domain. It is worth noting that copying the space-time patch to the hole region causes annoying blocky artifacts in the temporal domain due to the seams between new and previously-placed space-time patches in a filled area. We also fill depth values of the hole region, as shown in Fig. 6. The depth values in $\Psi_{\hat{\mathbf{p}}} \cap \Omega$ are substituted from the weighted average depth value of the candidate patches with the weight as the random walker's probability x^{s_2}. Furthermore, the background boundary $\delta\Omega_B$ is extended into the newly filled region, i.e., $H(\mathbf{q}) = 1 \;\; \forall \mathbf{q} \in \Psi_{\hat{p}} \cap \Omega$. The pseudo code of the proposed method is presented in Algorithm 1.

The use of a random walker's probability as a weighting factor in the patch matching process makes the filling process more robust to outliers. From the mathematical similarities between a diffusion process and the random walks, the random walker algorithm diffuses matching outliers which potentially lead to mismatching error. Fig. 7 compares the results from a typical SSD-based template matching and those from our probability-based weighted template matching. The typical SSD-based method shows global inconsistency due to the mismatching error, since the matching criterion only enforces local consistency by making local decisions at independent image positions. On the contrary, the proposed method enforces naturalness to the filled region by propagating matching error to the neighborhood. Note that our probability-based filling process relies on the greedy-filling scheme as in [13]. Consequently, the filled region itself might converge to a local minimum in the context of global visual coherence [21]. Nevertheless, the weight $W(\mathbf{p})$ defined by the random walker's probability in the patch matching process allows the filled region to be consistent with its vicinity.

Algorithm 1 Proposed Hole Filling Framework

Input: video I, depth D, reference depth D_0, hole Ω.
a. Identify hole boundaries $F^t(\mathbf{p}), \forall \mathbf{p} \in \delta\Omega^t, t = 1...N$.
b. Select sample space $\Phi^t, \forall t = 1...N$.
c. Initialize $C^t(\mathbf{p}), \forall \mathbf{p} \in I^t, t = 1...N$.
For frame t, from 1 to N
  Do
    a. Compute priorities $P(\mathbf{p}), \forall \mathbf{p} \in \delta\Omega^t$.
    b. Select the space-time patch $\Gamma_{\hat{\mathbf{p}}}$ with maximum priority.
    c. Estimate x^s of space-time patch candidates, $\forall s \in \{s_1, s_2\}$.
    d. Find $\Gamma_{\hat{\mathbf{q}}} \in \Phi$ that minimizes $W(\mathbf{q}) SSD(\Gamma_{\hat{\mathbf{p}}}, \Gamma_{\hat{\mathbf{q}}})$.
    e. Copy image data from $\Gamma_{\hat{\mathbf{q}}}$ to $\Psi_{\hat{\mathbf{p}}}$.
    f. Copy weighted average depth value from $\hat{\Phi}$ to $\Psi_{\hat{\mathbf{p}}}$.
    g. Update $C^t(\mathbf{q}) = C^t(\hat{\mathbf{p}})$ and $H^t(\mathbf{q}) = 1, \forall \mathbf{q} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega^t$.
  Until $\Omega^t = \emptyset$.
End for

(a) (b) (c)

Fig. 7. Performance comparison of (a) typical SSD-based template matching and (b) our probability-based weighted template matching with (c) ground truth.

V. Experimental Results

A. Experiment Setup

To evaluate the performance of the proposed method, we used six public multiview video-plus-depth (MVD) datasets; "Ballet" and "BreakDancer" [25], and "Dancer," "PoznanHall2," PoznanStreet, and Gt_Fly [26]. They consist of real world and computer graphic scenes with stationary and dynamic background. The estimated depth maps are associated for all real world scenes, while the ground truth depth maps are provided only for the computer graphic datasets. We adapted the conventional 3D warping method to extrapolate the desired virtual view from the single view. To validate the superiority of the proposed method, we intentionally placed the virtual camera far away from the reference camera, such

TABLE II
CHARACTERISTICS OF MULTIVIEW DATASETS AND EXPERIMENTAL SCENARIO

Name	Resolution	Frames	Scene Type	Depth Type	Camera Motion	Background Texture	Virtual Camera Pos.	Hole Size (Avg.)
Ballet	1024 × 768	1–100	Real world	Estimated	Stationary	Simple	3 → 4	13.1%
BreakDancer	1024 × 768	1–100	Real world	Estimated	Stationary	Simple	3 → 4	6.5%
Dancer	1920 × 1080	1–100	Computer graphics	Ground truth	Dynamic	Complex	1 → 3	1.3%
PoznanHall2	1920 × 1080	1–100	Real world	Estimated	Dynamic	Simple	6 → 7	2.2%
PoznanStreet	1920 × 1080	150–250	Real world	Estimated	Stationary	Complex	3 → 1	3.4%
Gt_Fly	1920 × 1080	100–200	Computer graphics	Ground truth	Dynamic	Complex	1 → 5	2.2%

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 8. Results of the proposed method and competing algorithms in "Ballet" (frame 23) and "BreakDancer" (frame 75) sequence. (a) Hole regions. (b) Criminisi's method [13]. (c) Criminisi's + Occ. method. (d) Daribo's method [15]. (e) Wexler's method [21]. (f) VSRS [27]. (g) Proposed method. (h) Ground truth. Hole regions are painted in yellow.

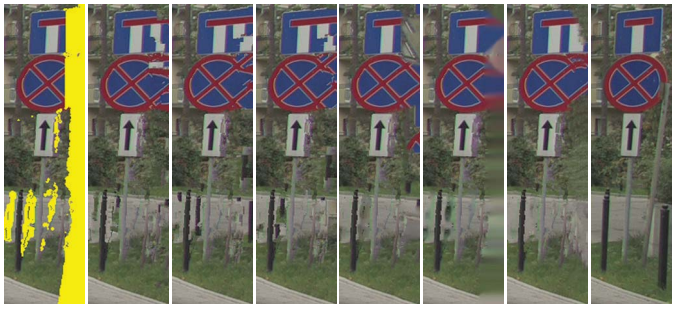
that large hole areas appear during the 3D warping process. The experimental scenario is summarized in Table II.

Five state-of-the-art methods were chosen for comparison. The proposed method was compared with the Criminisi's method [13] and the Daribo's method [15], since they can be directly comparable to our method. To provide a fair comparison for the original work of Criminisi *et al.* [13], we modified the Criminisi's algorithm referred to as "Criminisi's + Occ." by adding the proposed occlusion constraint term to the original priority term. Also, the Wexler's method [21] was evaluated for the performance comparison in terms of temporal coherence. In addition, the view synthesis reference software (VSRS) [27] was evaluated.

All the parameters were fixed during the experiment. The size of a space-time patch Γ was set to $9 \times 9 \times 7$. The size of a 3D subgraph in the patch candidate selection process was set to $100 \times 100 \times 10$. σ_I and σ_D were set to 900 and 80, respectively.

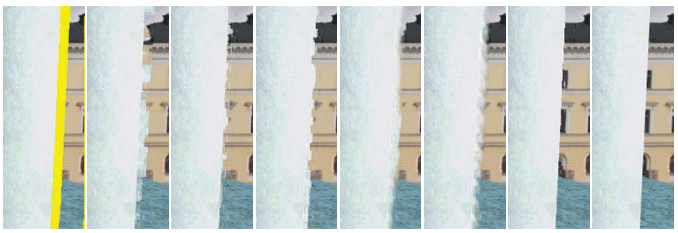
B. Spatial Quality Evaluation

1) Stationary Background Sequence: Fig. 8 compares the results of the proposed method with those of state-of-the-art methods in the case of stationary background motion with simple texture appearance: Ballet and BreakDancer sequences. It shows that the proposed method outperforms others in terms of synthesized appearance. Other methods contain apparent defects, such as the foreground texture penetration and unrealistic recovered regions, due to misleading priorities. Since Criminisi's method is not able to distinguish the foreground and background boundary, most hole regions were filled with foreground textures as shown in Fig. 8(b). Although the occlu-



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 9. Results of the proposed method and competing algorithms in "PoznanStreet" sequence at frame 25. (a) Hole regions. (b) Criminisi's method [13]. (c) Criminisi's + Occ. method. (d) Daribo's method [15]. (e) Wexler's method [21]. (f) VSRS [27]. (g) Proposed method. (h) Ground truth. Hole regions are painted in yellow.

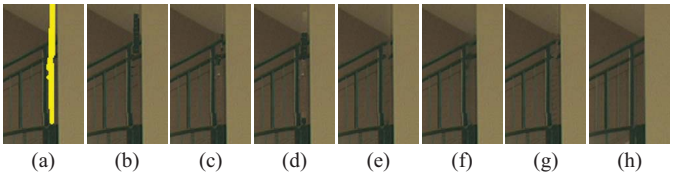


(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 10. Results of the proposed method and competing algorithms in "Dancer" sequence at frame 15. (a) Hole regions. (b) Criminisi's method [13]. (c) Criminisi's + Occ. method. (d) Daribo's method [15]. (e) Wexler's method [21]. (f) VSRS [27]. (g) Proposed method. (h) Ground truth. Hole regions are painted in yellow.

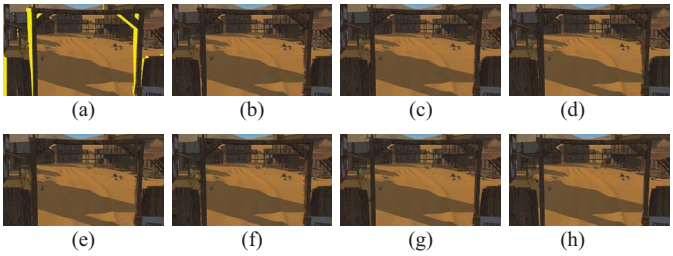
sion constraint term was leveraged in the Criminisi's + Occ. method, it still undergoes the problem that a patch may overlap both foreground and background regions as the filling process proceeds, resulting in expanding foreground textures as shown in Fig. 8(c). This defect can also be seen in Daribo's method [Fig. 8(d)], although it implicitly considers the sample space as the background region by adding a depth similarity cost in a patch matching criterion. As shown in Fig. 8(e), Wexler's method also has the foreground texture penetration problem, since it utilizes all the patches surrounding the hole boundary to infer the missing area. Hence, foreground and background textures are likely to be uniformly distributed in the filled area, resulting in the penetration of the foreground textures. Similar distortions are observed in Fig. 8(f). The VSRS recovers the hole area by using the structure-based inpainting without the boundary distinction, leading to blurring artifacts. The proposed method, however, successfully propagates significant edges into hole areas and shows realistic appearance, as shown in Fig. 8(g). It is achieved by propagating texture and structure from the background region and by selecting the optimal patch in a probabilistic manner. Furthermore, the hole area is filled in a smooth manner based on the guidance of the random walker's probabilities, such that potential outliers caused by the local matching criterion are significantly reduced.

Fig. 9 shows the results of the proposed method with those of state-of-the-art methods in the case of stationary background motion with complex texture appearance:



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 11. Results of the proposed method and competing algorithms in "PoznanHall2" sequence at frame 80. (a) Hole regions. (b) Criminisi's method [13]. (c) Criminisi's + Occ. method. (d) Daribo's method [15]. (e) Wexler's method [21]. (f) VSRS [27]. (g) Proposed method. (h) Ground truth. Hole regions are painted in yellow.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12. Results of the proposed method and competing algorithms in "Gt_Fly" sequence at frame 155. (a) Hole regions. (b) Criminisi's method [13]. (c) Criminisi's + Occ. method. (d) Daribo's method [15]. (e) Wexler's method [21]. (f) VSRS [27]. (g) Proposed method. (h) Ground truth. Hole regions are painted in yellow.

PoznanStreet sequences. Note that a local search is sufficient for simple background appearance such as Ballet and BreakDancer. This is because the performance of the isophote-driven sampling process such as Criminisi's method is not significantly affected from the simple texture area. However, as shown in Fig. 9, the local search fails in the complex textured scene, since a matching error can be propagated as the filling process proceeds. As shown in Fig. 9(b) and (c), the synthesized textures, e.g., grass, through the local search show inconsistent and unrealistic appearance. On the contrary, the proposed method ensures spatial continuity in this region as shown in Fig. 9(g).

2) Dynamic Background Sequence: To evaluate the performance of the proposed method in the dynamic background scene, we tested the dynamic background datasets including Dancer (Fig. 10), PoznanHall2 (Fig. 11), and Gt_Fly (Fig. 12). Except to PoznanHall2 which has simple texture appearance, both Dancer and Gt_Fly have complex textures in the background. Dancer and PoznanHall2 involve very smooth horizontal lateral dolly type of camera motion, while Gt_Fly involves arbitrary camera motion (including zooms) and changes of scale in the moving objects and the background.

As shown in Fig. 10, the proposed method outperforms the competing algorithms when the scene involves horizontal camera motion. Although the camera motion slightly involves zooming, the filled area shows plausible appearance without expending foreground boundaries along the moving objects. As shown in Fig. 11, however, noticeable performance gain is not achieved among competing algorithms. This is because the scene with simple textures does not need the isophote-driven filling criterion. In Fig. 12, the proposed method outperforms

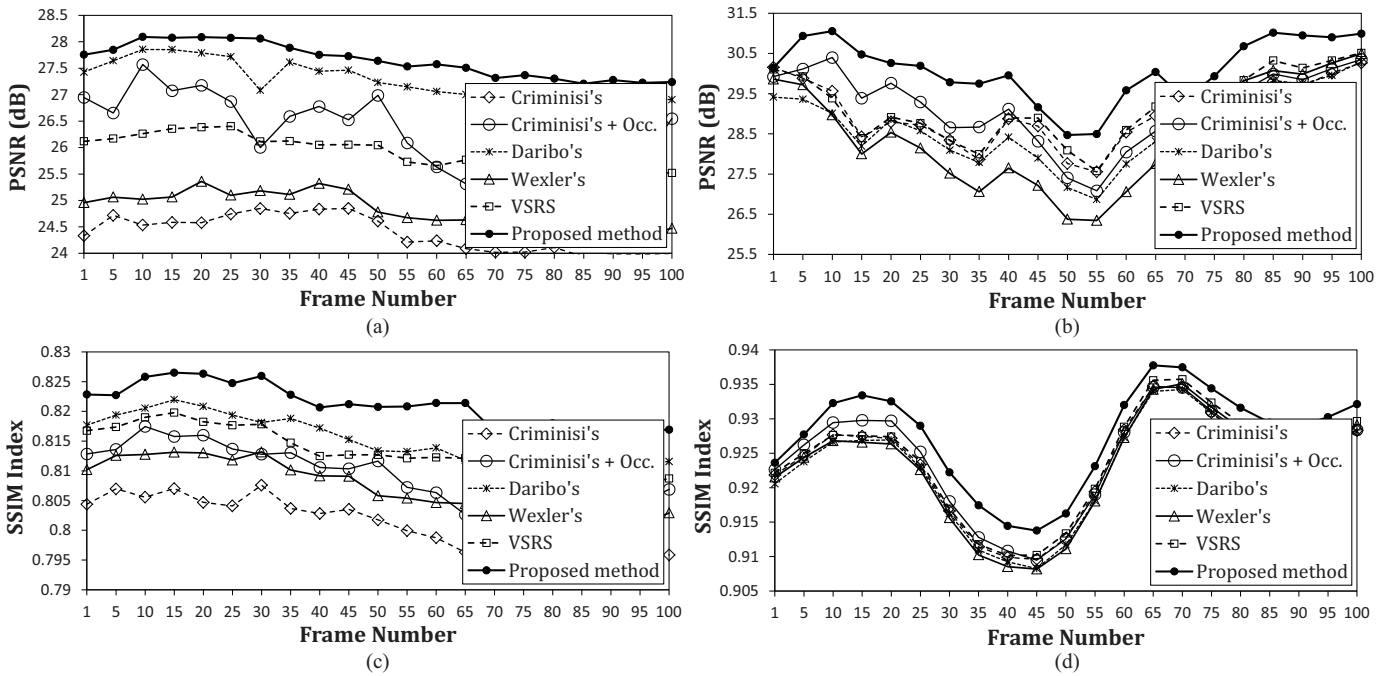


Fig. 13. Quantitative comparisons of the proposed method with competing algorithms over frames. (a) PSNR measures on "Ballet". (b) PSNR measures on "Dancer". (c) SSIM measures on "Ballet". (d) SSIM measures on "Dancer".

TABLE III
QUANTITATIVE QUALITY EVALUATION

Method	Ballet			BreakDancer			Dancer			PoznanHall2			PoznanStreet			Gt_Fly		
	PSNR	SSIM	F-value	PSNR	SSIM	F-value	PSNR	SSIM	F-value	PSNR	SSIM	F-value	PSNR	SSIM	F-value	PSNR	SSIM	F-value
Criminisi's [13]	24.38	0.8000	12.36	29.68	0.8192	9.39	28.90	0.9233	22.13	32.52	0.8429	2.12	25.17	0.7646	22.96	31.59	0.9350	5.91
Criminisi's + Occ.	26.33	0.8088	13.06	30.20	0.8226	7.44	29.06	0.9241	14.13	32.55	0.8431	1.83	25.22	0.7654	20.98	32.04	0.9370	5.52
Daribo's [15]	27.27	0.8151	7.65	30.43	0.8236	6.02	28.59	0.9228	15.39	32.51	0.8430	2.01	25.17	0.7648	21.27	31.50	0.9350	6.82
Wexler's [21]	24.81	0.8068	5.29	30.17	0.8238	4.91	28.39	0.9230	4.97	32.51	0.8434	0.87	25.37	0.7660	9.94	31.77	0.9354	7.26
VSRS [27]	25.90	0.8129	3.10	30.53	0.8289	1.89	29.07	0.9241	7.08	31.28	0.8440	2.75	25.40	0.7678	0.93	30.77	0.9361	6.84
Proposed method	27.65	0.8211	1.95	30.90	0.8298	3.22	30.08	0.9276	2.05	32.47	0.8433	0.73	25.35	0.7681	0.54	32.33	0.9377	5.33

the other methods but some irrelevant patches are seen in the filled areas. The optimal patch for the missing area can not be found to some extent in the scene with the presence of zooming camera motion. Specifically, scene components such as moving objects at each frame changes their scale during zooming, thus the possibility for finding the optimal replacement patch decreases.

3) Quantitative Evaluation: We adapted PSNR and structural similarity (SSIM) [28] index to compare the synthesis results of the proposed method with those of competing algorithms. The measured data at each frame on Ballet and Dancer are shown in Fig. 13. The proposed method shows better results in both PSNR and SSIM measures compared with the other methods. The average PSNR and SSIM values for all datasets are shown in Table III.

C. Temporal Quality Evaluation

Figs. 14 and 15 show the results across sequential frames of the enlarged area in Ballet and Dancer sequence. They demonstrate the space-time consistency property in case of stationary and dynamic background. As shown in Fig. 14(f) and Fig. 15, the proposed method fills the hole regions using temporally consistent background textures, while other methods show less correlation among adjacent frames and produce annoying flicker artifacts. Since the proposed method treats spatial and temporal information uniformly by using the space-time patch matching criterion, the synthesized results are temporally consistent. The temporal appearance of the hole region is reconstructed based only on the neighboring background regions without any interference from moving foreground objects, since the sample space is constrained to the background regions. In addition, the filled regions show strong correlation to the neighborhood in a locally smoothed way in that the random walks propagate similarity through a space and a time domain. Also, no blurring artifacts are seen as the proposed method relies on a copy-and-paste strategy in the filling process.

Since human eyes are very sensitive to temporal inconsistency when watching videos, we measured, similar to [29], the amount of flicker over the frames as follows:

$$T = \frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega^t} \left\| I^t(\mathbf{p}) - I^{t-1}(\mathbf{p}) \right\| \qquad \forall t = 2 \ldots N \qquad (17)$$

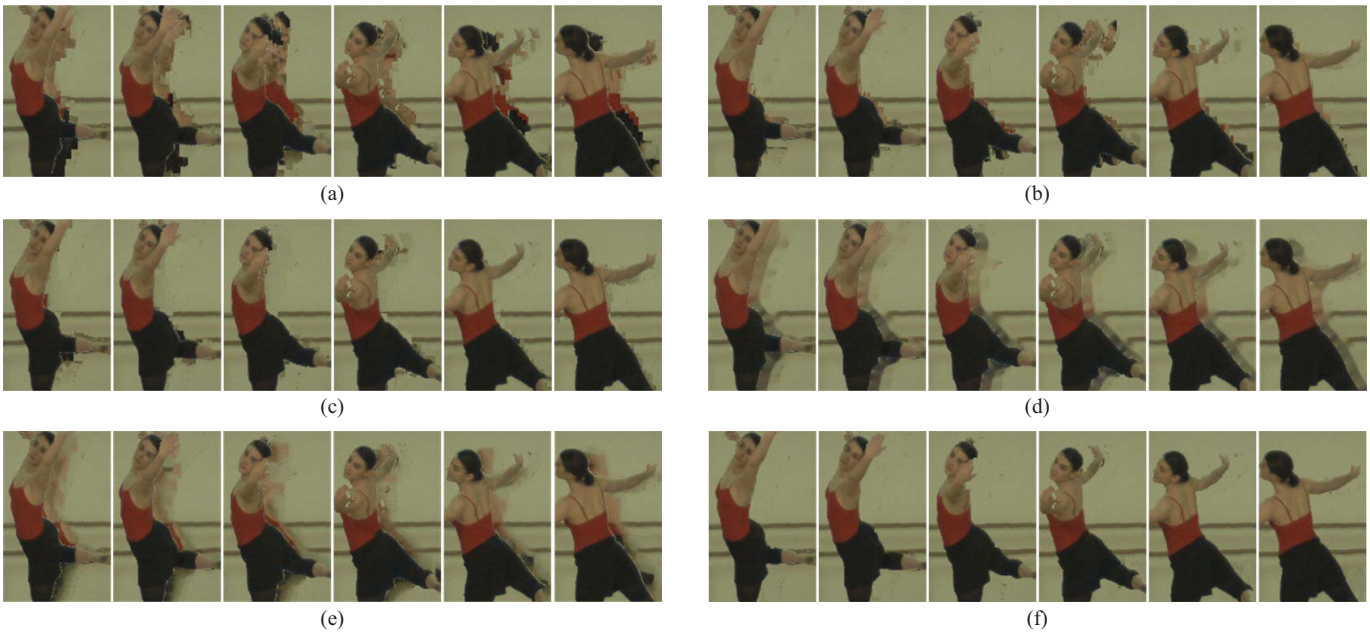


Fig. 14. Results across sequential frames from 17 to 22 in "Ballet" sequence. (a) Criminisi's method [13]. (b) Criminisi's + Occ. method. (c) Daribo's method [15]. (d) Wexler's method [21]. (e) VSRS [27]. (f) Proposed method.

Fig. 15. Results across sequential frames from 95 to 99 in "Dancer" sequence. Hole regions are painted in yellow.

where $|\Omega|$ represents the total number of hole pixels in a video. The T value measures the average number of changes in the inter-frame pixel value in hole regions. The overall flicker artifacts (F-value) can be measured as follows:

$$F = |T_{\text{syn}} - T_{\text{gt}}| \tag{18}$$

where T_{syn} and T_{gt} represent the value T of the synthesized view and the ground truth, respectively. Note that the lower F-value indicates fewer flicker artifacts. The flicker artifacts measured over frames are shown in Fig. 16. The proposed method shows lower flicker artifacts than the competing methods over frames. As shown in Table III, the average F-values of the proposed method are 1.9 and 2.0 on Ballet and Dancer sequences, respectively, while other methods have the average F-values greater than 3.0. All methods show a similar temporal performance on Gt_Fly, although the proposed method achieves the best performance among them. Note that Gt_Fly has low temporal correlation as the camera motion involves zooming. For example, the woody fence in Fig. 12 changes its scale over time. Accordingly, a spatial correlation between

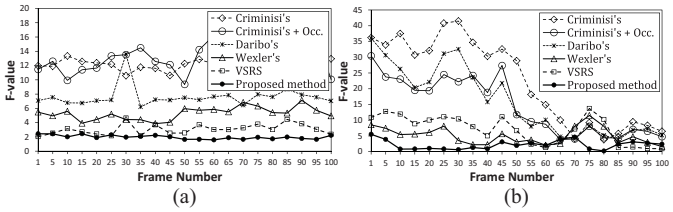


Fig. 16. Time-consistency comparisons of the proposed method with competing algorithms on (a) "Ballet" and (b) "Dancer" sequence over frames.

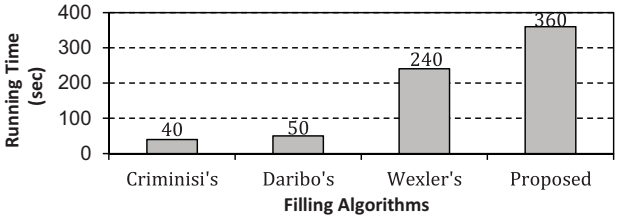


Fig. 17. Complexity comparisons on "Ballet" sequence. The value indicates the average completion time per one frame when the hole size is about 13% of the image size. Note that Wexler's and the proposed method use the space-time information, while the other methods consider spatial information only.

frames is low. The results shown in this paper are also available at the author's website [30].

D. Complexity Analysis

The proposed method is relatively slower than the competing algorithms, since it makes use of a random walks implementation in couple locations. Specifically, the main computational complexity occurs at the candidate patch selection stage, since it solves a linear system for the dynamically generated space-time random walks formulation at each iteration. For example, given Ballet sequence, the sample space selection takes approximately 420 seconds to segment entire

frames of a video. The running time of the space-time hole inpainting is about 360 seconds to complete a frame. In our implementation, we constrained the density of the patches in the patch candidate selection, i.e., $w_g \times h_g \times t_g$, to reduce the sparse matrix size for the random walks formulation. For fast implementation, the linear solver can be accelerated by using GPU [31]. Further speedups can be achieved at the expense of accuracy by adopting K-nearest-neighbor search algorithms, such as the PatchMatch [32], to the SSD-based algorithm described in Section IV-B. The running time of the proposed method and the competing algorithms is presented in Fig. 17.

VI. Conclusion

In this paper, we have proposed a hole filling method to fill a large hole created by the view extrapolation. Based on the geometric characteristics of hole occurrence as well as the behavior of random walkers on a graph, hole regions were reconstructed in a visually plausible and temporally consistent manner.

There are some limitations in our approach. Our method is not able to deal with an arbitrary camera motion, e.g., zoom and rolling, which changes the scale and the orientation of scene components. The filling process might fail unless the ideal patch with the desired orientation exists. In this case, the rotation and scale invariant template matching criterion [33] can be adapted. Also, if the scene has overlapping dynamic objects, it is not sufficient to fill the hole using the background only. The filling might only be possible by making use of non-local patches when the target object is fully visible [21], or by using information from the other views like the complex scene reconstruction [17], [18], [19]. In addition, since our method works with a single extrapolated view, appearances of the multiple extrapolated views would not be consistent with each other, creating artifacts when viewed in 3D. This limitation, however, can be relaxed by producing multiple extrapolated views in a consecutive order. Specifically, the most distant virtual view is first extrapolated. The remaining intermediate views can then be simply interpolated.

Acknowledgment

The authors would like to thank Dr. Dongbo Min and the anonymous reviewers for their valuable comments and suggestions.

References

[1] S. Zinger, D. Ruijters, L. Do, and H. N. Peter, "View interpolation for medical images on autostereoscopic displays," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 128–137, Jan. 2012.

[2] C. Fehn, "Depth-image-based rendering DIBR, compression and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, Jan. 2004.

[3] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[4] S. Choi, B. Ham, and K. Sohn, "Hole filling with random walks using occlusion constraints in view synthesis," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1965–1968.

[5] 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.

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

[7] S. Zinger, L. Do, and P. H. N. de With, "Free-viewpoint depth image based rendering," *J. Vis. Commun. Image Represent.*, vol. 21, nos. 5–6, pp. 533–541, Jan. 2010.

[8] K. Oh, S. Yea, and Y. Ho, "Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3D video," in *Proc. Picture Coding Symp.*, May 2009, pp. 233–236.

[9] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 65–72, Nov. 2008.

[10] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH)*, Jul. 2000, pp. 417–424.

[11] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[12] A. Efros and T. Leung, "Texture synthesis by nonparametric sampling," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1033–1038.

[13] 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.

[14] L. Azzari, F. Battisti, and A. Gotchev, "Comparative analysis of occlusion-filling techniques in depth image-based rendering for 3D videos," in *Proc. 3rd Workshop Mobile Video*, Oct. 2010, pp. 57–62.

[15] I. Daribo and H. Saito, "A novel inpainting-based layered depth video for 3DTV," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 533–541, Jun. 2011.

[16] H. Lim, Y. Kim, S. Lee, O. Choi, J. Kim, and C. Kim, "Bi-layer inpainting for novel view synthesis," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1089–1092.

[17] L. Ballan, G. Brostow, J. Puwein, and M. Pollefeys, "Unstructured video-based rendering: Interactive exploration of casually captured videos," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–64, Jul. 2010.

[18] L. Guan, J. S. Franco, and M. Pollefeys, "Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction," *Int. J. Comput. Vis.*, vol. 90, no. 3, pp. 283–303, Dec. 2010.

[19] A. Taneja, L. Ballan, and M. Pollefeys, "Modeling dynamic scenes recorded with freely moving cameras," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 613–626.

[20] L. Cheng, M. Hong, D. Schuurmans, and T. Caelli, "Real-time discriminative background subtraction," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1401–1414, May 2011.

[21] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[22] S. Kumar, M. Biswas, S. J. Belongie, and T. Q. Nguyen, "Spatio-temporal texture synthesis and image inpainting for video applications," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Sep. 2005, pp. 85–88.

[23] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3D video," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 453–465, Jun. 2011.

[24] C. Cheng, S. Lin, and S. Lai, "Spatio-temporally consistent novel view synthesis algorithm from video-plus-depth sequences for autostereoscopic displays," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 523–532, Jun. 2011.

[25] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004.

[26] H. Schwarz, D. Marpe, and T. Wiegand, "Description of exploration experiments in 3D video coding," ISO, Dresden, Germany, Tech. Rep. MPEG2010/N11274, Apr. 2010.

[27] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," ISO, Archamps, France, Tech. Rep. M15377, Apr. 2008.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[29] M. Schmeing and X. Jiang, "Time-consistency of disocclusion filling algorithms in depth image based rendering," in *Proc. 3DTV Conf. True Vis. Capture, Trans. Display 3D Video*, May 2011, pp. 1–4.

[30] S. Choi, B. Ham, and K. Sohn. (2010, Jul. 27). *Experimental Results* [Online]. Available: http://diml.yonsei.ac.kr/shch/holefilling/

[31] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random walks for interactive alpha-matting," in *Proc. 5th Int. Conf. Visualizat. Imag. Image Process.*, 2005, pp. 1–5.

[32] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 29–43.

[33] A. Mansfield, M. Prasad, C. Rother, T. Sharp, P. Kohli, and L. Van Gool, "Transforming image completion," in *Proc. Brit. Mach. Vis. Conf.*, Aug. 2011, pp. 1–11.

[34] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[35] B. Morse, J. Howard, S. Cohen, and B. Price, "PatchMatch-based content completion of stereo image pairs," in *Proc. 2nd Int. Conf. 3D Imag. Model. Process. Visualizat. Trans.*, Oct. 2012, pp. 555–562.

Sunghwan Choi (S'10) received the B.S. degree in electronic engineering and avionics from Korea Aerospace University, Gyeonggi-do, Korea, in 2009. He is currently pursuing the joint M.S. and Ph.D. degrees in electrical and electronic engineering with Yonsei University, Seoul, Korea.

His current research interests include 3D image and video processing, computer vision, computational aspects of human vision, and image-based modeling and rendering.

Bumsub Ham (S'09) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2008, where he is currently pursuing the joint M.S. and Ph.D. degrees in electrical and electronic engineering.

His current research interests include variational methods and geometric partial differential equations, both in theory and applications in computer vision and image processing, particularly regularization, stereo vision, super-resolution, and HDR imaging.

Mr. Ham was the recipient of the Honor Prize in 17th Samsung Human-Tech Prize in 2011 and the Grand Prize in Qualcomm Innovation Fellowship in 2012.

Kwanghoon Sohn (M'92–SM'12) received the B.E. degree in electronic engineering from Yonsei University, Seoul, Korea, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1983, 1985, and 1992, respectively.

He was a Senior Member of the research staff with the Satellite Communication Division, Electron and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a Post-Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently a Professor with the School of Electrical and Electronic Engineering, Yonsei University. His current research interests include 3D image processing, computer vision and image communication.

Dr. Sohn is a member of SPIE.