Joint Texture-Depth Pixel Inpainting of Disocclusion Holes in Virtual View Synthesis

Smarti Reel*, Gene Cheung†, Patrick Wong* and Laurence S. Dooley*
*Department of Computing and Communications, The Open University, Milton Keynes, UK
E-mail: {smarti.reel, patrick.wong, laurence.dooley}@open.ac.uk
†National Institute of Informatics, Tokyo, Japan
E-mail: cheung@nii.ac.jp Tel: +81-3-4212-2567

***Abstract*—Transmitting texture and depth maps from one or more reference views enables a user to freely choose virtual viewpoints from which to synthesize images for observation via depth-image-based rendering (DIBR). In each DIBR-synthesized image, however, there remain disocclusion holes with missing pixels corresponding to spatial regions occluded from view in the reference images. To complete these holes, unlike previous schemes that rely heavily (and unrealistically) on the availability of a high-quality depth map in the virtual view for inpainting of the corresponding texture map, in this paper a new Joint Texture-Depth Inpainting (JTDI) algorithm is proposed that simultaneously fill in missing texture and depth pixels. Specifically, we first use available partial depth information to compute priority terms to identify the next target pixel patch in a disocclusion hole for inpainting. Then, after identifying the best-matched texture patch in the known pixel region via template matching for texture inpainting, the variance of the corresponding depth patch is copied to the target depth patch for depth inpainting. Experimental results show that JTDI outperforms two previous inpainting schemes that either does not use available depth information during inpainting, or depends on the availability of a good depth map at the virtual view for good inpainting performance.**

I. Introduction

With the advent of sensing technologies, videos of a dynamic 3D scene can now be captured economically by a large array of closely-spaced cameras (*e.g.*, more than 100 cameras are used in [1]). Beside conventional color (*e.g.*, RGB) images, depth images (per-pixel distance between physical objects in the 3D scene and a capturing camera) from the same camera viewpoints can also be acquired using active depth sensors like time-of-flight cameras [2]. Transmitting both texture and depth maps from multiple viewpoints—a format known as *texture-plus-depth* or *video-plus-depth* [3]—enables a user to freely select virtual viewpoints from which to synthesize novel images for observation via *depth-image-based rendering* (DIBR) [4]. It has been shown [5] that allowing a user to interactively select different viewpoints from which to observe a 3D scene can greatly enhance the user's depth perception of the scene, improving his overall visual experience.

In summary, during a horizontal viewpoint change from camera-captured view u to virtual view v, DIBR copies each color pixel $I_u(x, y)$ in reference view u to its new location $I_v(x + \gamma, y)$ in the synthesized image[1] of virtual view v, where the horizontal pixel displacement γ is deduced from the corresponding depth pixel $d_u(x, y)$ in reference view u. In practice, pixels of an object closer to the camera have larger displacements during a viewpoint change than pixels of the background. This means that there may exist one or more spatial region of the background, occluded by a foreground object in the reference view, that become exposed in the virtual view from the large displacement of the foreground object during a viewpoint change. See Fig. 1 for an illustration. The hole with no corresponding pixels in the reference view is commonly called a *disocclusion hole*. Devising a strategy to properly fill in missing pixels in a disocclusion hole—a process called *inpainting* or *image completion* in the literature [6], [7], [8], [9], [10]—is paramount in constructing a visually pleasing virtual viewpoint image.

This work proposes a new Joint Texture-Depth Inpainting (JTDI) algorithm that simultaneously fills in missing texture and depth pixels in the disocclusion holes. Though a similar template matching framework introduced in [11] is used to copy texture pixels from the known region to the unknown region, we derive a new priority term to order filling of pixel patches using available partial depth information. Further, unlike [6], [7], [8] whose inpainting performance depends heavily on the availability of a complete and good-quality depth map in the virtual view for texture inpainting, in JTDI a more realistic DIBR view synthesis scenario is assumed where depth pixels in the disoccluded regions are also missing and challenging to complete. So a joint inpainting algorithm is required to carefully fill in missing pixels in both texture and depth maps. Experimental results show that our proposal outperforms [11] and [7] by up to 1.33dB and 0.83dB in PSNR of the disoccluded texture regions, respectively. Further, we demonstrate that subjective quality of the inpainted areas is also visibly improved.

The outline of the paper is as follows. The overview of related works is presented in Section II. Then an overview of DIBR view synthesis system is given in Section III, followed by discussion of JTDI algorithm in Section IV. Finally, experimental results and conclusion are presented in Section V and

[1] Pixel relocation from a reference view image to a virtual view image is a pure horizontal shift if the camera images are properly rectified *a priori*.

VI, respectively.

II. Related Work

The growing popularity of free viewpoint video means an increased research interest in inpainting of disocclusion holes in DIBR-synthesized images. There are in general two classes of inpainting algorithms: partial differential equations (PDEs)-based schemes like [12] and exemplar-based (template matching) schemes [11], [6], [7], [8], [9], [10]. It is known that PDEs-based schemes do not handle large disocclusion holes well—common if the spacing between neighboring cameras is large, or if the virtual view image is synthesized using one texture/depth map pair of a single camera view. Thus inpainting research for DIBR-synthesized images has been focusing on exemplar-based approach.

The pioneer inpainting work for regular color images with no depth information is [11], which proposed to use *template matching* to fill in missing pixels; *i.e.*, copying a fixed-size pixel patch from a known spatial region to an unknown region. Numerous subsequent works [6], [7], [8], [9], [10] kept the template matching framework in [11] but modified the definition of the priority term (used to determine the order in which missing pixel patches should be filled) and/or the criteria for template matching, using available depth information. The underlying assumption for the majority of these works ([6], [7], [8]), however, is that a complete and good-quality depth map at the target virtual view is available, or can be easily pre-computed *a priori*, for the said computation of the priority term and/or matching criteria.

We argue that this assumption is not realistic for practical DIBR view synthesis systems; disoccluded pixel locations in the target virtual view with missing texture information will also have depth information missing. Further, though depth maps are known to be piecewise smooth, the missing depth pixels can be more complex than a constant background depth value, meaning simple signal extrapolation strategies extending the depth signal of the neighboring background pixels will not always be correct. Thus, in this paper we propose a new algorithm to jointly inpaint texture and depth pixels in disoccluded regions, where *we first leverage on available depth information to fill in texture pixels, then use inpainted texture information to fill in depth pixels*. We found experimentally that this mutual assistance approach between texture and depth information very effective in joint inpainting of both maps.

The recovery of correct depth information at the virtual view is itself important for the case when the reconstructed virtual view is used to synthesize other virtual views. This is indeed the proposal in [13], where a second reference view is first synthesized from the first reference view (with the help of transmitted *auxiliary information* from sender to help complete the target image), so that novel intermediate virtual views between the two reference views can be synthesized via DIBR. Thus, our proposal of jointly inpainting texture and depth maps at the virtual view can also contribute to better intermediate synthesized view quality, if representation of free viewpoint video such as [13] is employed at encoder.

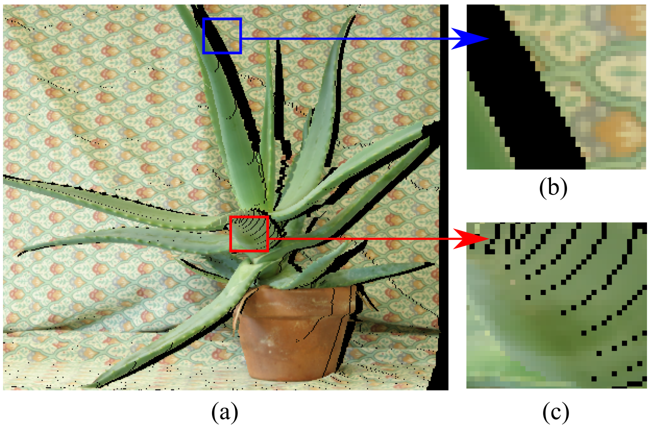


Fig. 1. (a) DIBR synthesized *Aloe* image with (b) disocclusion holes and (c) rounding holes

III. System Overview

We first overview the components of a typical DIBR view synthesis system. To enable synthesis of novel images at arbitrarily chosen viewpoints at decoder, texture-plus-depth format [3] dictates the transmission of texture and depth maps capturing the same 3D scene by one or more closely spaced cameras. The selected viewpoint is synthesized at the receiver via a procedure known as DIBR [4]. DIBR is a pixel-to-pixel mapping such that the reference image pixels are first projected back to the world coordinates using depth map and then reprojected to the virtual image coordinate. This process is also known as 3D image warping [14].

One major drawback when synthesizing novel viewpoint images via DIBR is generation of holes. There are two common types of holes. A *disocclusion hole* is a spatial region that is occluded by a closer object in the reference view, but become visible in the virtual view. Disocclusion holes typically occur at foreground object boundaries. A *rounding hole* is a pixel location in the virtual view that is visible in a reference view, but due to rounding to integer 2D grid positions during 3D warping, it was left unfilled. Figure 1 shows examples of both disocclusion and rounding holes. Rounding holes tend to be small and can be filled easily using conventional filtering techniques [4]. The focus of our paper is on the filling of disocclusion holes.

IV. Algorithm Development

We first overview Criminisi's template matching algorithm for inpainting of regular color images in [11]. We then discuss the two modifications we propose to the base template matching algorithm, so that texture and depth patches can be jointly inpainted.

A. Overview of Criminisi's Template Matching Algorithm

We first define the following terms. The source region (known pixel region) is defined as $\Phi = I - \Omega$, where I and Ω

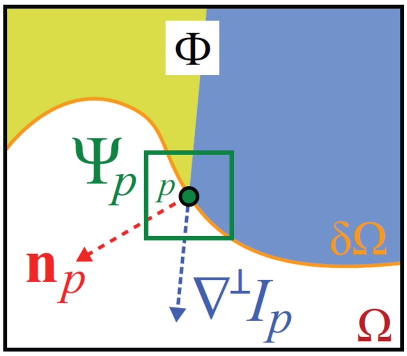


Fig. 2. Abstract illustration of Criminisi's template matching algorithm [11].

are input image and disocclusion hole region, respectively. As shown in Fig. 1, the disocclusion hole region Ω may not be a single contiguous spatial region. The boundary of hole region is defined as $\delta\Omega$. See Fig. 2 for an illustration.

Pixel patches of a pre-selected size $K \times K$ that border the hole region Ω are inpainted in a priority order (to be discussed). Specifically, for a given $K \times K$ target patch $\Psi_{\hat{p}}$ with center pixel location $\hat{p}$, $\hat{p} \in \delta\Omega$, we identify the best matching patch $\Psi_{\hat{q}}$ in the source region Φ that minimizes the matching error:

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} d\left(\Psi_{\hat{p}}, \Psi_q\right) \tag{1}$$

where $d(\Psi_{\hat{p}}, \Psi_q)$ is the Sum of Squared Differences (SSD) between corresponding known pixels of the two patches. In other words, known pixels in $\Psi_{\hat{p}}$ are used as a *template* to find a best matched patch in source region Φ. After $\Psi_{\hat{q}}$ is identified using (1), missing pixels in target patch $\Psi_{\hat{p}}$, $\Psi_{\hat{p}} \cap \Omega$, are filled using corresponding pixels in $\Psi_{\hat{q}}$.

It is stressed in [11] that the order in which missing pixels in Ω are filled is very important; intuitively, pixel patch that can be filled more confidently should be inpainted first. [11] defines a *priority* term $P(p)$ for each boundary pixel $p \in \delta\Omega$ as the product of two terms:

$$P(p) = C(p)\, D(p) \tag{2}$$

where $C(p)$ and $D(p)$ are the *confidence* and *data* terms, respectively. $C(p)$ and $D(p)$ are defined as follows:

$$C\left(p\right) = \frac{\sum_{q \in \Psi_p \cap \Phi} C(q)}{|\Psi_p|}, \;\; D\left(p\right) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \tag{3}$$

where $|\Psi_p|$ is the number of pixels in target patch Ψ_p, α is a normalization factor (*e.g.*, $\alpha = 255$ for a typical gray-level image), n_p is the unit vector orthogonal to $\delta\Omega$ at pixel p, and $\nabla I_p^\perp$ is the isophote (direction and intensity) at pixel p. The confidence term $C(p)$ gives higher priority to the patches which have higher percentage of non-hole pixels. $C(p)$ is initialized to 0 for missing pixels in Ω, to 1 everywhere else. Data term $D(\mathbf{p})$ defines the strength of linear structures hitting the boundary $\delta\Omega$ at each iteration, and is used to encourage propagation of linear structures.

After missing pixels in a patch $\Psi_{\hat{p}}$ are filled, the confidence term $C(p)$ for each newly filled pixel p in the patch is updated as follows:

$$C(p) = C(\hat{p}), \;\; \forall p \in \Psi_{\hat{p}} \cap \Omega \tag{4}$$

Once the confidence values are updated, priorities for the next patch to be filled are computed and this entire process is repeated till all disocclusion holes are filled.

B. Using Depth to Modify Priority Term

Observing that depth information is not used in [11], [7] proposed to modify the computation of the priority term $P(p)$ using depth information as follows. Assuming depth information is available per pixel in the entire virtual view, [7] added an extra term $L(p)$ to $P(p)$ in (2):

$$P(p) = C(p)\, D(p)\, L(p) \tag{5}$$

where $L(p)$ is a *depth variance* term, proportional to the inverse variance of the corresponding $K \times K$ depth patch Z_p:

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum_{q \in Z_p \cap \Phi} \left(Z_p(q) - \bar{Z}_p\right)^2} \tag{6}$$

where $|Z_p|$ is the size of depth patch Z_p, $Z_p(q)$ is the pixel depth value at the pixel location q under Z_p, and $\bar{Z}_p$ is pixel mean value. The intuition is that if a patch has large depth variance, then the patch is likely straddling both foreground and background pixels, which makes the patch difficult to inpaint. The patch should then be assigned a lower priority, influenced by a smaller $L(p)$.

C. New Depth-based Priority Computation

We now discuss proposed modifications to the base template matching algorithm in [11]. Although [7] proposed to give higher priority to patches with smaller depth variance using (5), it does not guarantee the patches to be filled from background to foreground. Since the selection of right priority term is crucial in template matching, as a patch filled from foreground boundary initially will lead to serious error propagation to a large spatial area. Also from *a priori* information we understand that disocclusion areas should always be filled with background pixels. To make sure that background patches are inpainted first, we compute a *depth mean* term which provides higher priority to patches with larger overall depth values. Then the depth mean term is incorporated as a *multiplier* to the original terms $C(p)$, $D(p)$, $L(p)$, which are now combined *additively* instead. The rationale behind adding these terms is to overcome the circumstances where patch priority reduces to zero apart from having high confidence $C(p)$ and low variance $L(p)$ terms. Such a condition occurs when the data term $D(p)$ is zero . The additive combination provides equal weightage to all participating terms. In summary, we revise the priority term $P(p)$ as:

$$P(p) = \left(C(p) + D(p) + L(p)\right) \times \left(Z_{near} - \bar{Z}_p\right) \tag{7}$$

where $Z_{near} = 255$ which is the nearest depth value. Note that unlike (5) in [7], the depth mean term is now clearly the dominant term in the computation of $P(p)$, so that patches further in the background are always selected for inpainting first. Further, unlike [10] where the depth variance term was replaced by a mean term, we keep $L(p)$ in the computation of $P(p)$, so that between two patches that have the same depth mean, the one with the smaller depth variance is favoured.

D. Filling depth disocclusion holes

The key novelty of JTDI algorithm is that we alternate between inpainting of texture pixels using partial depth information, and inpainting of depth pixels using texture information. Specifically, after the best-matched texture patch $\Psi_{\hat{q}}$ is found in the source region Φ, we use the corresponding depth patch $Z_{\hat{q}}$ to fill in missing depth pixels in target depth patch $Z_{\hat{p}}$ as follows:

$$Z_{\hat{p}} = \bar{Z}_{\hat{p}} + \left(Z_{\hat{q}} - \bar{Z}_{\hat{q}}\right) \tag{8}$$

where $\bar{Z}_{\hat{p}}$ and $\bar{Z}_{\hat{q}}$ are the mean depth values of the target depth patch $Z_{\hat{p}}$ (computed using available depth pixels) and the best-matched depth patch $Z_{\hat{q}}$, respectively. In other words, only the depth variance of the matched patch $Z_{\hat{q}}$ is copied to the target, while the depth mean of the original patch $Z_{\hat{p}}$ remains the same.

The rationale for (8) is as follows: Template matching between texture patches just ensures the textural patterns are similar; the patches could be from quite different depths of the 3D scene, *e.g.* same wallpaper pattern recurring on a wall slanted towards infinity away from the camera. Thus, directly copying of depth pixels from best-matched patch (evaluated based solely on texture content) to the target patch, as done in [9], is a tenuous proposition. On the other hand, given the textural content are similar, the depth *gradient* of the best-matched patch is more likely to be similar to the gradient of the target patch, as illustrated in the aforementioned wallpaper example. Thus copying only the variance to the target depth block is arguably more appropriate. Finally, by retaining the original mean depth value in the target patch $Z_{\hat{p}}$, we can achieve piecewise smoothness in the inpainted depth map, unlike simple depth patch copying in [9].

V. Experimentation

In this section, we report the results of applying JTDI approach on various image datasets and comparing against inpainting methods in [11] and [7].

A. Experimental Setup

A simple baseline DIBR view synthesis system has been implemented in Matlab. We evaluate JTDI algorithm with using different Middlebury datasets [15], including `aloe` (427×370), `reindeer` (447×370), `art` (463×370) and `dolls` (463×370). These datasets contain seven camera-captured views of the same static scene, as well as disparity maps for view 1 and 5. For each sequence, we use DIBR to generate reference view 3 using texture and depth maps of view 1. The disocclusion holes in synthesized texture and depth are simultaneously filled using JTDI method. Results from different inpainting methods are judged using both objective measurements and subjective evaluation. To implement [7], which assumes the availability of a complete depth map *a priori*, we first filled the holes in virtual depth map using [11], and then used this inpainted depth map for texture inpainting using [7].

TABLE I
PSNR Comparison for Texture Inpainting *(in dB)*

Image	Criminisi et al. [11]	Daribo et al. [7]	JTDI
`aloe`	27.84	27.53	**28.59**
`reindeer`	27.43	27.96	**28.76**
`art`	23.06	23.38	**23.65**
`dolls`	29.43	29.53	**30.07**

B. Objective Results

To test the objective performance of JTDI algorithm, Peak-Signal-to-Noise Ratio (PSNR) of the disocclusion holes is used. The PSNR values of the inpainted texture maps using JTDI method, [11] and [7] are shown in Table I. The optimal patch-size selected for `aloe` and `dolls` is 5×5 (K = 5) and 7×7 (K = 7) for `reindeer` and `art`. The results demonstrate that JTDI method performs better than [11] and [7]. For `reindeer`, the resulting PSNR increases by up to 1.33dB and 0.80dB compared to [11] and [7], respectively. Similar results have been observed for `aloe`, `art` and `dolls` images.

C. Subjective Results

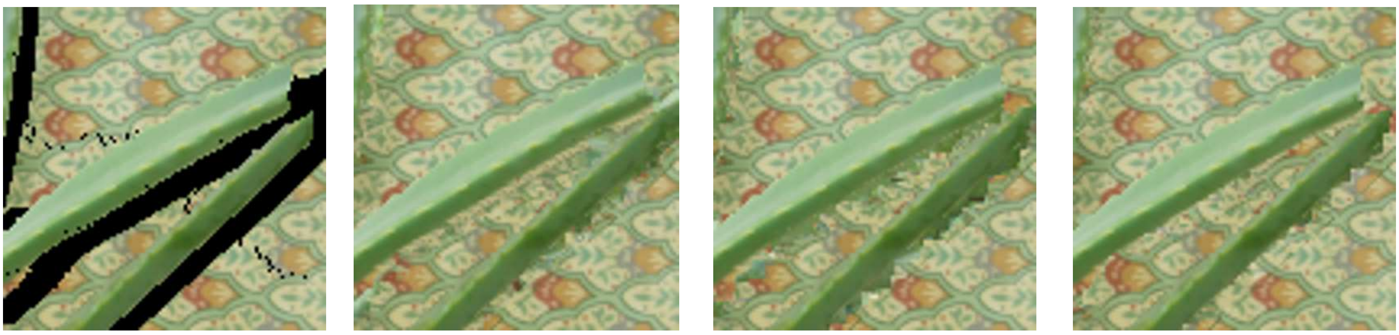
Fig. 3, 4, 5 and 6 shows the subjective comparison of JTDI algorithm with [11] and [7]. The visual results are representative sub-regions for the four Middlebury image datasets: `aloe`, `reindeer`, `art` and `dolls`. It is observed that JTDI algorithm (Fig. 3*(d)*, 4*(d)*, 5*(d)* and 6*(d)*) performs better in preserving the foreground object boundaries compared to [11] (Fig. 3*(b)*, 4*(b)*, 5*(b)*) and 6*(b)* and [7] (Fig. 3*(c)*, 4*(c)*, 5*(c)*) and 6*(c)*. The reduced artefacts are the result of proposed improved priority term, where the filling process begins from background (BG) and move inwards toward foreground (FG).

The comparative results for depth disocclusion filling using JTDI method (Fig. 7*(c)*, 8*(c)*, 9*(c)* and 10*(c)*) and [11] (Fig. 7*(b)*, 8*(b)*, 9*(b)* and 10*(b)*) are shown in Fig. 7, 8, 9 and 10. Clearly, our method provides much better inpainting results then [11]. This shows that inpainting of depth map itself is not trivial and cannot be done simply, as claimed in [6], [7], [8].

Currently, JTDI follows an exhaustive search approach to select the best-matching patch which makes it computationally expensive. Instead of full exhaustive search, probability based random sampling techniques can be deployed to reduce the search complexity with minimal loss in performance.

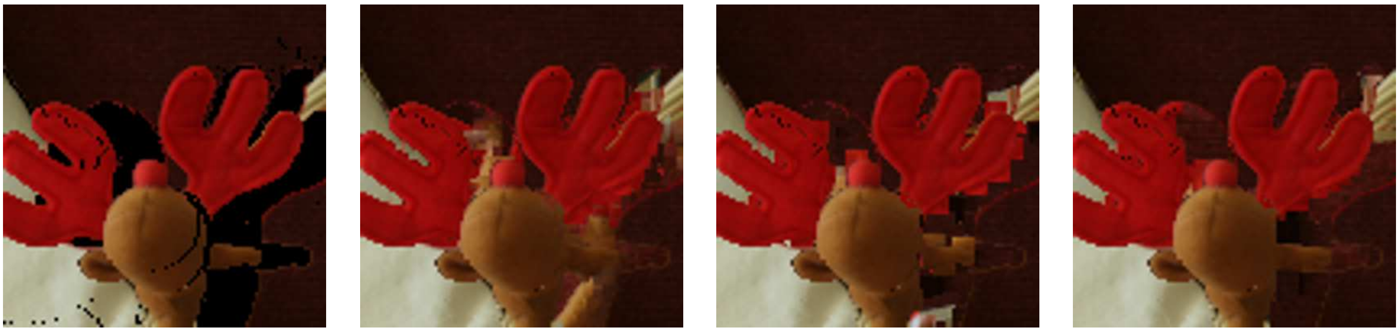
VI. Conclusion

When synthesizing a novel viewpoint image using depth-image-based rendering (DIBR), disocclusion holes appear that



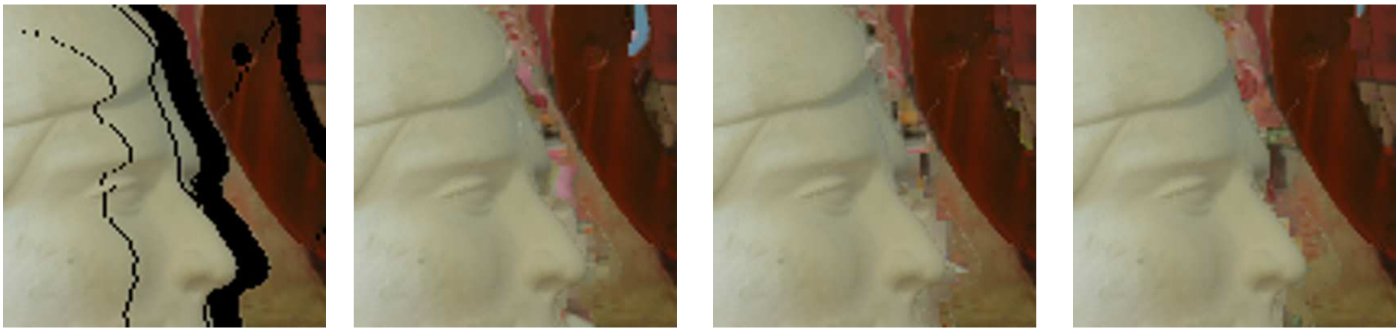
(a) Disocclusion holes (b) Criminisi et al. [11] (c) Daribo et al. [7] (d) JTDI

Fig. 3. Subjective Comparisons for `aloe`.



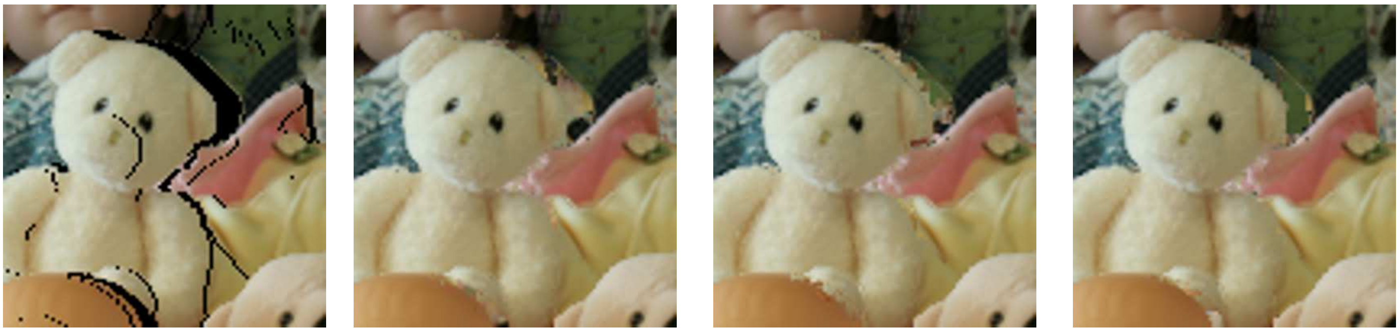
(a) Disocclusion holes (b) Criminisi et al. [11] (c) Daribo et al. [7] (d) JTDI

Fig. 4. Subjective Comparisons for `reindeer`.



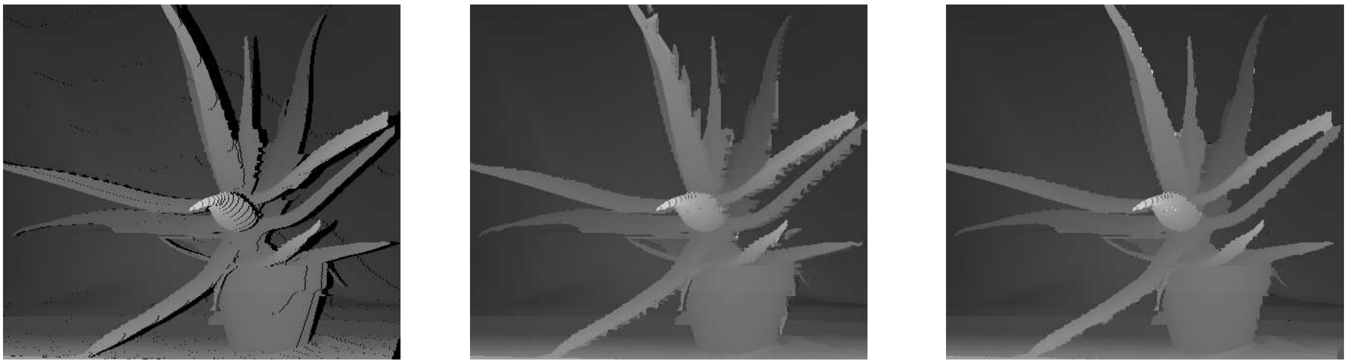
(a) Disocclusion holes (b) Criminisi et al. [11] (c) Daribo et al. [7] (d) JTDI

Fig. 5. Subjective Comparisons for `art`.



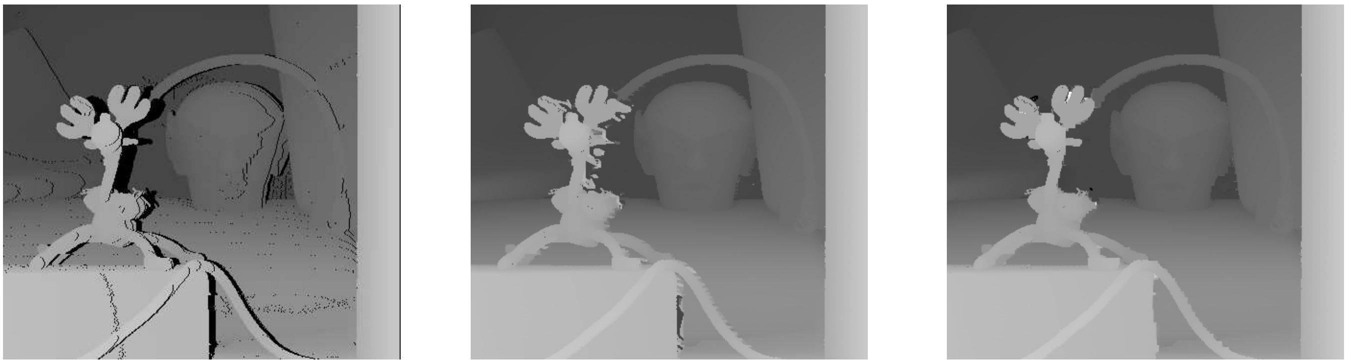
(a) Disocclusion holes (b) Criminisi et al. [11] (c) Daribo et al. [7] (d) JTDI

Fig. 6. Subjective Comparisons for `dolls`.



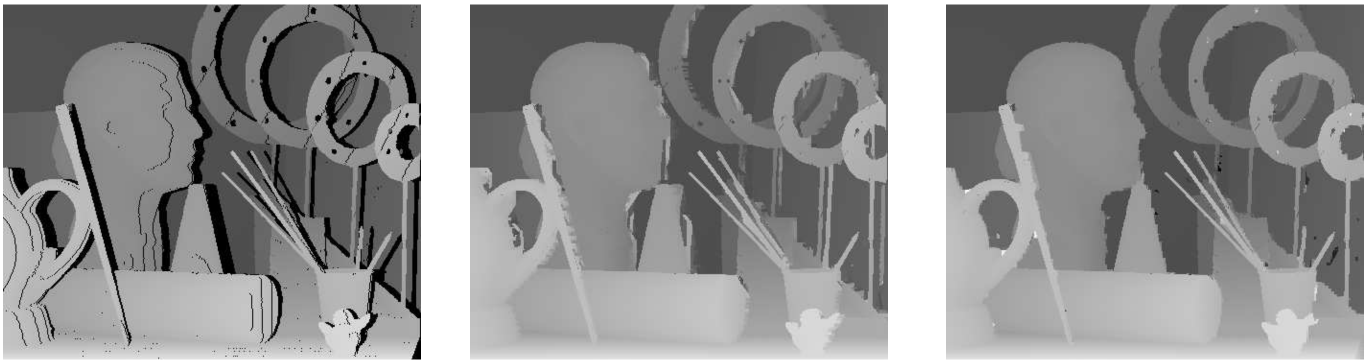
(a) Disocclusion holes (b) Criminisi et al. [11] (c) JTDI

Fig. 7. Depth disocclusion filling for `aloe`



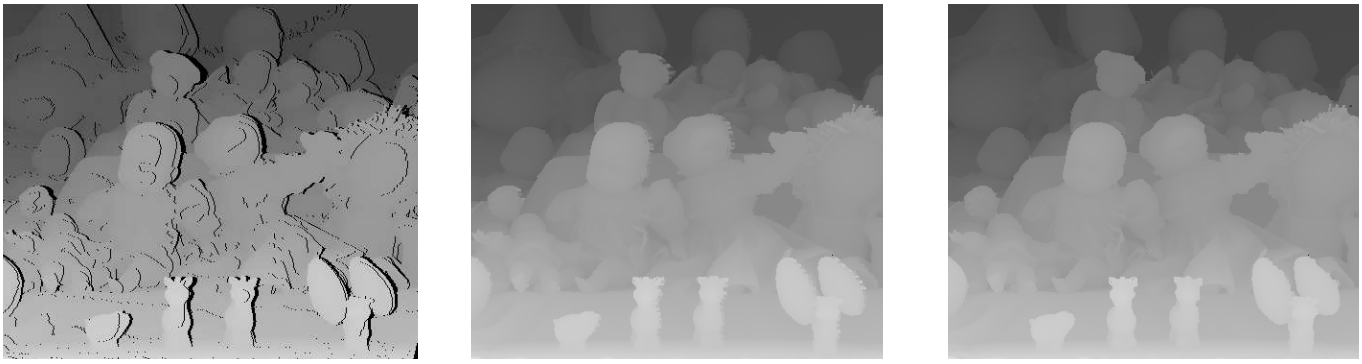
(a) Disocclusion holes (b) Criminisi et al. [11] (c) JTDI

Fig. 8. Depth disocclusion filling for `reindeer`



(a) Disocclusion holes (b) Criminisi et al. [11] (c) JTDI

Fig. 9. Depth disocclusion filling for `art`



(a) Disocclusion holes (b) Criminisi et al. [11] (c) JTDI

Fig. 10. Depth disocclusion filling for `dolls`

correspond to spatial regions of the 3D scene not visible in the reference views. In this paper, we proposed a new inpainting scheme based on template matching in [11], so that missing pixels in both texture and depth maps can be filled simultaneously. In particular, using partial depth information we defined a new priority term to order pixel patches in the disocclusion region to be inpainted. Then for a given best-matched patch in the source region, the depth variance of the best-matched patch is copied to the target patch for depth inpainting. Experimental results show that proposed mutual assistance inpainting approach has noticeable performance gain over [11] and [7] in both objective and subjective comparison.

REFERENCES

[1] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.

[2] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—system description, issues and solutions," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Washington, DC, June 2004.

[3] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing*, San Antonio, TX, October 2007.

[4] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Applications of Digital Image Processing XXXII, Proceedings of the SPIE*, vol. 7443 (2009), 2009, pp. 74 430T–74 430T–11.

[5] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *ACM International Conference on Multimedia*, Singapore, November 2005.

[6] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole-filling method using depth based in-painting for view synthesis in free viewpoint television (FTV) and 3D video," in *Picture Coding Symposium*, Chicago, IL, May 2009.

[7] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *IEEE Multimedia Signal Processing Workshop*, Saint-Malo, France, October 2010.

[8] O. L. Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *IEEE International Conference on Image Processing*, Brussels, Belgium, September 2011.

[9] I. Ahn and C. Kim, "Depth-based disocclusion filling for virtual view synthesis," in *IEEE International Conference on Multimedia and Expo*, Melbourne, Australia, July 2012.

[10] L. Ma, L. Do, and P. de With, "Depth-guided inpainting algorithm for free-viewpoint video," in *IEEE International Conference on Image Processing*, Orlando, FL, September 2012.

[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," in *IEEE Transactions on Image Processing*, vol. 13, no.9, September 2004, pp. 1–13.

[12] D. Tschumperle and R. Deriche, "Vector-valued image regularization with PDEs: a common framework for different applications," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no.4, April 2005, pp. 506–517.

[13] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped sub-block motion prediction in texture map compression using depth information," in *2010 Picture Coding Symposium*, Krakow, Poland, May 2012.

[14] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Symposium on Interactive 3D Graphics*, New York, NY, April 1997.

[15] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.