Stereoscopic Inpainting: Joint Color and Depth Completion from Stereo Images

Liang Wang[†] lwangd@cs.uky.edu
Hailin Jin[‡] hljin@adobe.com
Ruigang Yang[†] ryang@cs.uky.edu
Minglun Gong[§] gong@cs.mun.ca

[†]Center for Visualization and Virtual Environments, University of Kentucky, USA
[‡]Advanced Technology Labs, Adobe Systems Incorporated, USA
[§]Computer Science Department, Memorial University of Newfoundland, Canada

Abstract

We present a novel algorithm for simultaneous color and depth inpainting. The algorithm takes stereo images and estimated disparity maps as input and fills in missing color and depth information introduced by occlusions or object removal. We first complete the disparities for the occlusion regions using a segmentation-based approach. The completed disparities can be used to facilitate the user in labeling objects to be removed. Since part of the removed regions in one image is visible in the other, we mutually complete the two images through 3D warping. Finally, we complete the remaining unknown regions using a depth-assisted texture synthesis technique, which simultaneously fills in both color and depth. We demonstrate the effectiveness of the proposed algorithm on several challenging data sets.

1. Introduction

Digital photos have become a ubiquitous part of our everyday life. As a result, image inpainting, a digital image processing technique to seamlessly fill in holes in an image, has received considerable attention in the research community. While most existing inpainting works mainly focus on texture completion on a single image, we in this paper address a novel problem of completing both texture and depth of a stereo image pair after object removal.

Our proposed *stereoscopic inpainting* algorithm is designed to jointly complete missing texture and depth by leveraging the following advantages introduced by the stereo images. First, the region to be filled after object removal may be partially visible in the other camera view, reducing the need to entirely "hallucinate" the color in the holes. Secondly, the depth information from stereo matching can be used to differentiate structural elements and guide the texture synthesis process. Lastly, the consistency of inpainting results on both images and depth maps provides a quality measure, based on which an iterative algorithm can be developed to automatically detect artifacts and refine the completion.

Being a counterpart of conventional color completion techniques, by utilizing stereo images and depth information our approach is able to complete complex salient structures exist in the missing region and provide more plausible texture synthesis results. Experimental results demonstrate that our novel completion framework produces images with higher fidelity and fewer artifacts compared to traditional inpainting works. What is more, besides pure two-dimensional texture synthesis, *stereoscopic inpainting* can also be used to facilitate many interesting applications in 3D (*e.g.* View Synthesis and Image-Based Modeling) since our algorithm makes it more practical to obtain consistent stereo images and depth maps with undesired objects being removed.

1.1. Related work

This work is related to a sizable body of literature on image inpainting, started by the work of [1]. Of particular interest are the example-based approaches [5, 11, 15] which fill missing regions with patches sampled from known areas. To better cope with salient structures in the images, Sun *et al*. [19] proposed a system that allowed the user to specify curves or line segments on which the most salient missing structures reside, and Drori *et al*. [7] proposed to use "point of interest" to further improve the completion quality. Cues from multiple images have also been explored in the past. Kang *et al*. [13] used landmarks to match images and then copied warped patches from different images. Wilczkowiak *et al*. [22] suggested to increase the sampling spaces by considering patches from images taken from different perspectives. This work differs from both [13] and [22] in that we perform depth estimation from the input images and use the resulting depth to guide the sampling process. [2] also uses depth information from photos to perform completion. However their input is a video sequence and the completion process requires a large number of nearby video frames and

photographs (typically 75 as reported in the paper).

In addition to image inpainting, this work also relates to the literature on occlusion handling in stereo vision. Occlusions are one of the major challenges in stereo. A substantial amount of work has been devoted to detecting occlusion areas. A not-so recent survey can be found in [8]. More recent work suggested to solve stereo matching and occlusion detection jointly within energy minimization frameworks [6, 18]. In particular, [12, 6, 18] proposed to complete occlusion areas with disparities from the background. However, they all suffer from significant artifacts where the scene is not frontal-parallel. The occlusion handling algorithm proposed in this work is related to a line of stereo work that based on segmentation or plane fitting [3, 14, 20, 23, 24]. These approaches typically worked by first solving depth in reliable regions, dividing the reliable regions into color segments, fitting planes to the segments, and assigning the rest unreliable pixels to these planes. Optimal assignment can be achieved by minimizing some energy functions [3, 14]. Although this work uses a similar plane extension idea for filling occlusion regions, it differs from the aforementioned approaches: this work is specifically designed for refining depth maps and filling occlusion regions and works with depth results from any stereo algorithm; it takes into account the visibility constraint for deciding plane assignment; we propose a novel algorithm that can obtain a globally smooth result and is also efficient. Depth filling is not a new problem. [21] proposed a texture-synthesis type of approach. It sampled both the images and the depth and propagated inwards from the boundaries of the missing regions with considerations for edges. The major limitation of [21] is that it copies depth values directly and therefore does not work for scenes that are not frontal-parallel. This work can be considered as a first-order extension of [21] that is able to handle scenes containing planes of arbitrary orientations.

2. Stereoscopic inpainting

This paper addresses the following problem: Given a stereo image pair and estimated disparity maps for both views, allow users to remove foreground objects in the scene and then complete the missing color and depth information in uncovered areas. In particular, we assume the input stereo images are rectified and use $\{I_L, I_R\}$ to refer the left and right images, respectively. The disparity maps are precalculated using a stereo algorithm and are denoted as $\{D_L, D_R\}$. We do not require the stereo algorithm to provide accurate disparity estimations in occluded areas, but assume that the occluded areas in both disparity maps are detected. We use $\{O_L, O_R\}$ to denote the sets of occluded pixels in the left and right views, respectively.

We remark that the proposed algorithm can work in conjunction with any existing stereo approach, but produces better results when taking high quality disparity maps as inputs. In this paper the disparity maps are precalculated using the symmetric stereo algorithm [18], which solves stereo matching and occlusion detection simultaneously in a global optimization framework and has been proved to be one of the top performers [17]. Note that, if an alternative stereo algorithm is used, occlusion regions can still be detected using the method of [8].

2.1. Overview of the algorithm

Taking $\{I_L, I_R\}$, $\{D_L, D_R\}$ and $\{O_L, O_R\}$ as input, the algorithm starts with filling the missing disparity information caused by occlusions using a segmentation-based depth filling approach. The results of this step are complete disparity maps for both views, denoted as $\{\bar{D}_L, \bar{D}_R\}$. The user is then asked to label the foreground object to be removed. After object removal, the uncovered pixels have unknown color and depth information, which need to be completed. We use $\{\Omega_L, \Omega_R\}$ to refer the uncovered pixels in the left and right views, respectively. Since some of the pixels in Ω_L are visible in the right view and some in Ω_R are visible in the left view, we warp both $\langle I_L, \bar{D}_L \rangle$ to the right view and $\langle I_R, \bar{D}_R \rangle$ to the left view to fill these pixels. To inpaint the remaining pixels in Ω_L and Ω_R, an iterative texture synthesis procedure is proposed. This procedure fills in both color and depth information simultaneously, as well as automatically detects unreliable solutions using a disparity-color consistency constraint. The final results of the algorithm are color and depth images for both views with foreground object being removed, referred as $\{I'_L, I'_R\}$ and $\{D'_L, D'_R\}$.

2.2. Segmentation-based occlusion filling

In order to infer unknown disparity values for occluded pixels in $\{O_L, O_R\}$, we employ the widely used segment constraint, *i.e.*, the disparity values vary smoothly within each segment and the corresponding 3D surface can be modeled by a plane. Accordingly, we only need to assign a disparity plane to each segment, instead of assigning a disparity value to each individual pixel.

The occlusion filling process for the two views are performed independently and we only discuss one of the views in detail. The process starts by performing the mean shift segmentation [4] on the stereo images to group pixels into a set of color segments $\tilde{S} = \{S_1, S_2, \ldots\}$, where each segment S_i is a set of adjacent pixels with similar colors. Note that a slight over-segmentation is preferable over under-segmentation since it better satisfies the segment constraint.

To fill disparity values for pixels in $O \subset O_L$, we first find out all the segments that overlap with O. This is to compute a set Υ:

$$\Upsilon = \{S | S \in \tilde{S} \wedge \|S \cap O\| > 0\} \quad (1)$$

where $S \cap O$ is the set of pixels that belong to both sets S

Algorithm 1 Pseudocode for disparity plane assignment

```
while Υ ≠ ∅ do
    ⟨t, s⟩ = arg min_{t∈Υ, s∈S̃−Υ} E(t, s)
    Assign the disparity plane for s to t
    Υ = Υ − {t}
end while
```

and O. $\|\cdot\|$ denotes the number of pixels in a set.

Our goal is to assign a disparity plane to each segment in Υ so that the disparity values of all pixels in O can be calculated using the plane parameters of the segment to which it belongs. Depending on the number of occluded pixels in a given segment, two different filling approaches are used.

Plane fitting for (partly) visible segments. A segment S is considered as visible or partially visible if it contains enough pixels with known disparities. The following criteria is used to determine whether the number of known pixels is enough:

$$\|S - O\| > \max(6, \lambda \cdot \|S\|) \tag{2}$$

where $S - O$ is the set difference between S and O and the parameter λ is set to 0.4.

The plane parameters for S can be computed based on pixels in set $S - O$ since their disparity values are already estimated by the stereo algorithm. In this paper, a RANSAC based plane fitting approach is applied [9]. Once processed, segment S is removed from set Υ if $S \in \Upsilon$.

Plane assignment for the remaining segments. Assigning a proper disparity plane to the remaining segments in Υ is more challenging due to the lack of pixels with known disparity values within the segment. To find optimal disparity planes for these segments, we propose a greedy algorithm that works in a best-first filling order.

As shown in Algorithm 1, the algorithm iteratively searches for the segment pair $\langle t, s\rangle$ that minimizes a matching cost $E(t, s)$, where segment s already has disparity plane assigned and t does not. Once such a pair is found, the segment t will be assigned to the same disparity plane as the one for segment s. The matching cost between segments s and t, $E(t, s)$, is defined as a weighted average of three different terms:

$$E(t, s) = E_{clr}(t, s) + \lambda_{adj} E_{adj}(t, s) + \lambda_{vis} E_{vis}(t, s) \tag{3}$$

where the first term, $E_{clr}(t, s)$, measures the color similarity between the two segments. It is defined as:

$$E_{clr}(t, s) = 1 - \frac{\vec{C}_t}{\|\vec{C}_t\|} \cdot \frac{\vec{C}_s}{\|\vec{C}_s\|} \tag{4}$$

where $\vec{C}_t$ and $\vec{C}_s$ are the average color vectors of segments t and s, respectively. The second term, $E_{adj}(t, s)$, is simply a binary function that returns 0 if segments t and s are adjacent and 1 otherwise. This term encourages neighboring segments to be assigned to the same disparity plane. The last term utilizes the weak consistency constraint in [10] to penalize disparity assignments that introduce inconsistent visibility relationships. This constraint dictates that, if a pixel is occluded, it must be occluded by something closer to the viewpoint, *i.e.*, its corresponding pixel in the other view must have a higher disparity value:

$$\begin{aligned} \bar{D}_L(x, y) &\le D_R(x - \bar{D}_L(x, y), y) \\ \bar{D}_R(x, y) &\le D_L(x + \bar{D}_R(x, y), y) \end{aligned} \tag{5}$$

The above constraint is not strictly enforced since the input disparity maps, $\{D_L, D_R\}$, may contain errors. Instead, violation of the constraint is allowed but is penalized using the cost term $E_{vis}(t, s)$. We let function $E_{vis}(t, s)$ return the ratio of pixels in t that violate the weak consistency, when their disparities are computed using the disparity plane of segment s. The two constant scalars λ_{adj} and λ_{vis} are determined empirically. In our implementation $\lambda_{adj} = 0.03$ and $\lambda_{vis} = 0.05$ are used throughout.

2.3. Foreground object removal

After the occluded areas are filled in both views, the user is asked to label the foreground objects to be removed, *i.e.*, specifying $\{\Omega_L, \Omega_R\}$. This can be done using any image editing tool on either the input color image I or the completed disparity map $\bar{D}$. Labeling on disparity map is easier because sharp depth discontinuities and the absence of high frequency texture can facilitate the process in both speed and quality. After the user specifies the objects to be removed in one of the stereo views, the label for the second view can be generated automatically using the pixel correspondence information provided by disparity map $\bar{D}$. However, additional user interactions may still needed to correct the artifacts and noise caused by inaccurate disparities.

2.4. Mutual completion through warping

Removing a foreground object leaves holes in both input images and disparity maps for both views, since there is neither color nor depth information available for the part of the background that is uncovered. However, due to the viewpoint difference between the two stereo images, part of the uncovered region in one view may be visible in the other. Hence, we can use the two stereo images to mutually complete each other through 3D warping [16]. That is if $(x - \bar{D}_L(x, y), y) \in \Omega_R$ we can set:

$$\begin{cases} I'_R(x - \bar{D}_L(x, y), y) = I_L(x, y) \\ D'_R(x - \bar{D}_L(x, y), y) = D_L(x, y) \end{cases} \tag{6}$$

Similarly, if $(x + \bar{D}_R(x, y), y) \in \Omega_L$, we can set:

$$\begin{cases} I'_L(x + \bar{D}_R(x, y), y) = I_R(x, y) \\ D'_L(x + \bar{D}_R(x, y), y) = D_R(x, y) \end{cases} \tag{7}$$

Completion through warping needs to be performed with care. For example, when more than one pixel is warped to the same destination, the one with the highest disparity value shall be used since it is the closest to the viewpoint. Furthermore, in order to prevent errors being introduced by incorrect disparity values, we also enforce the ordering constraint during the warping. Although the ordering constraint is not always satisfied in real scene, it allows us to err on the side of caution.

Once a pixel's color and depth information is filled through warping, it will be removed from $\{\Omega_L, \Omega_R\}$. This simplifies the texture synthesis process to be described in the next section.

2.5. Iterative texture synthesis

The remaining pixels in $\{\Omega_L, \Omega_R\}$ are the parts of uncovered background that are invisible in both stereo images. An iterative process is used in this paper for filling these pixels. Within each iteration, a modified version of the exemplar-based texture synthesis approach [5] is applied to fill both the left and right views independently. The results obtained for the two views are then cross-validated to detect unreliable solutions. Only the reliable ones found are added into the final color and disparity inpainting results $\langle I', D' \rangle$. The iterative process terminates after a maximum of N iterations and the pixels still considered unreliable after the iteration terminates will be filled using the best solutions found through texture synthesis.

Depth-assisted texture synthesis. In [5], the authors propose a scheme for determining the optimal order of filling unknown pixels. While their technique is capable of propagating linear structure and complex texture into unknown areas, it may use samples from irrelevant texture areas and the propagation of these implausible textures can lead to poor results. To alleviate this issue, our approach makes use of the depth information to improve the sampling process. Our modification is based on the following intuitions:

- With the additional depth information available, the optimal sample should be decided based on both color and depth similarities.
- The uncovered background is usually farther away from the viewpoint than the removed foreground object. Therefore the missing region should be filled using samples with smaller disparity values than that of the removed object. This requirement is referred as the view distance constraint.

Since the texture synthesis for the two views is handled independently, we now describe the procedure for one view only. We use Φ to denote the source regions that provide samples in the filling process and use Ψ_p to refer the $\ell \times \ell$ square patch centered at pixel p. Using the same technique described [5], we compute the filling priorities for all the pixels in Ω and process these pixels in order of priority. When handling pixel p, we search in Φ for a patch Ψ_q that satisfies:

$$\Psi_q = \arg \min_{\Psi_k \in \Phi} F(\Psi_k, \Psi_p), \tag{8}$$

where $F(\Psi_s, \Psi_t)$ measures the difference between two generic patches Ψ_s and Ψ_t. Instead of computing the difference using color similarity only as in [5], the matching criterion we used is defined using both color and disparity similarities, as well as the view distance constraint, as:

$$F(\Psi_s, \Psi_t) = F_{clr}(\Psi_s, \Psi_t) + F_{dis}(\Psi_s, \Psi_t) + F_{viw}(\Psi_s, \Psi_t). \tag{9}$$

$F_{clr}(\Psi_s, \Psi_t)$ is the sum of absolute color differences of the already filled pixels in the two patches. $F_{dis}(\Psi_s, \Psi_t)$ measures the disparity similarity of the two patches and is defined as:

$$F_{dis}(\Psi_s, \Psi_t) = \alpha \sum_{\mu} \min(1, |\bar{D}(s+\mu) - \bar{D}(t+\mu)|), \tag{10}$$

where $\mu \in \{x | (t+x) \in \Psi_t \bigcap \Phi\}$. The last term F_{viw} penalizes pixels in set $\Psi_t \bigcap \Omega$ that violate the view distance constraint, *i.e.*

$$F_{viw}(\Psi_s, \Psi_t) = \beta \sum_{\nu} f(\bar{D}(s+\nu), \bar{D}(t+\nu)) \tag{11}$$

where $\nu \in \{x | (t+x) \in \Psi_t \bigcap \Omega\}$ and $f(a, b)$ equals to 1 if $a > b$ and 0 otherwise. Parameters α and β are set to 13 and 30 in our experiments, respectively.

After the optimal exemplar Ψ_q is found, the color of each pixel $p' \in \Psi_p \bigcap \Phi$ is copied from its corresponding position q' in Ψ_q. The disparity of p' is computed using the disparity plane parameter of the segment to which q' belongs. In this manner, our approach allows to simultaneously fill-in both color and depth.

Consistency check. For most image completion approaches there is no decent way to detect visual artifacts since the result is considered as optimal by the algorithm. With additional image available and by performing inpainting for both images independently and simultaneously, the potential incorrect solutions can be detected automatically through consistency check.

Assuming the surfaces in the scene are close to Lambertian, unreliable inpainted results can be detected based on color consistency of corresponding pixels. That is, given the color images $\{I'_L, I'_R\}$ and their disparity maps $\{D'_L, D'_R\}$ completed after object removal, the following constraints should be satisfied for all non-occluded pixels:

$$\begin{aligned} |I'_L(x,y) - I'_R(x - D'_L(x,y), y)| &< \epsilon \\ |I'_R(x,y) - I'_L(x + D'_R(x,y), y)| &< \epsilon \end{aligned} \tag{12}$$

where ϵ is the error threshold and is set to 20 in our experiment.

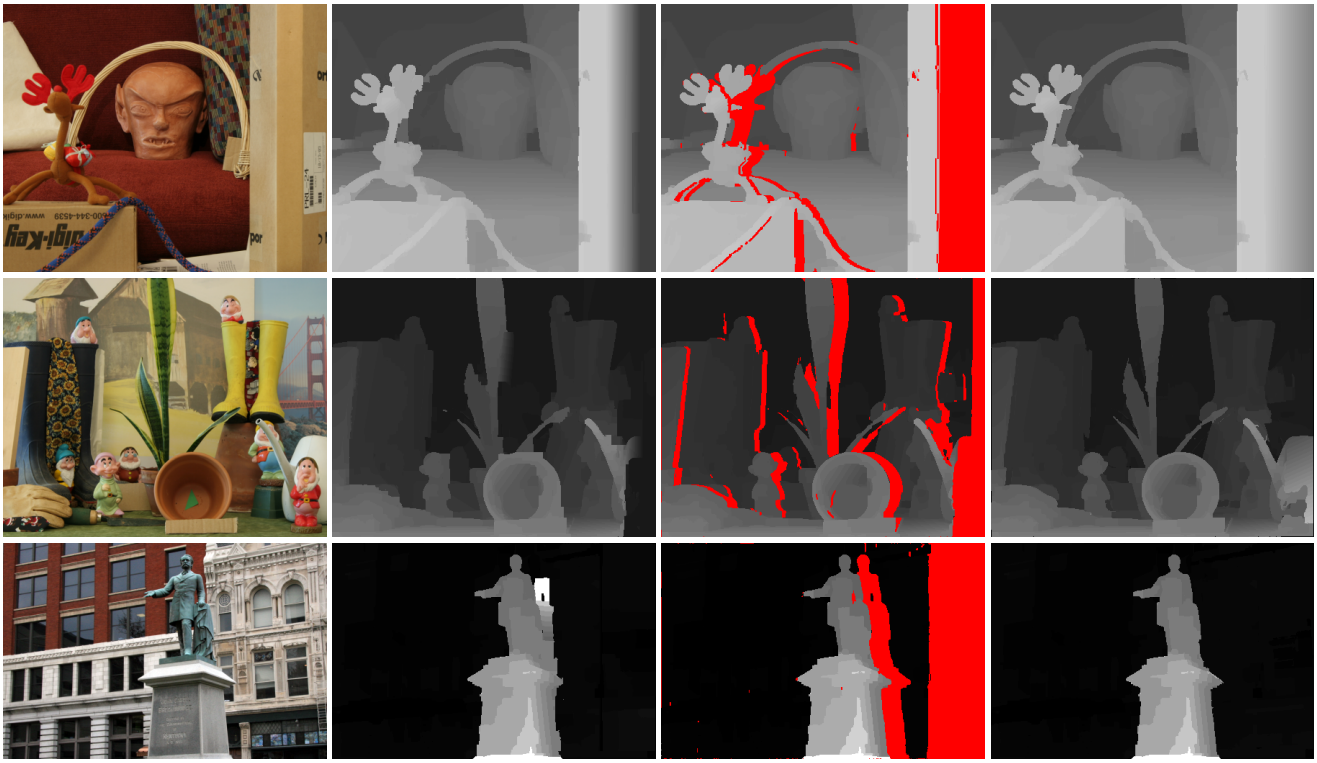

Figure 1. Occlusion filling examples. From left to right: one of the stereo images; disparity maps without occlusion filling; occlusion detection results using [18] (occlusion regions are marked in red); our occlusion filling results.

The above testing is performed on all pixels that are completed using texture synthesis at current iteration, *i.e.*, those in $\{\Omega_L, \Omega_R\}$. If the color and depth inpainting result for a given pixel (x, y) in Ω_L passes the test, it is considered as reliable and we thereby remove pixel (x, y) from Ω_L, as well as pixel $(x - D'_L(x, y), y)$ from Ω_R, if $(x - D'_L(x, y), y) \in \Omega_R$. This symmetric consistency check is applied twice to process the inpainting results generated for both left and right views.

3. Experimental results

In this section we report the results of applying our method on various stereo images and comparing against one of the state-of-the-art automatic image completion methods. In all the experiments the source region is set to $\Phi = I - \Omega$ and the patch diameter ℓ is 11. The sizes of test images are about 500×400.

Figure 1 shows the results of our occlusion filling method. The first column shows three stereo images with slanted surfaces and complicated scene structures. The second column shows the disparity maps without occlusion filling applied. In the third column we show occlusion regions (red) computed from using the algorithm of [18]. Disparity maps after occlusion filling are presented in the last column which shows that our algorithm is able to produce good quality results.

We demonstrate the effectiveness of our iterative inpainting process in Figure 2. The first two images in the first row are the stereo pair with uncovered areas marked in green. The next two images are the results after mutual completion. Note that the sizes of unknown regions are reduced in both images by copying information from the other view. Furthermore, the cane, which is almost totally occluded by the reindeer in the left view becomes partially visible. The second row shows the color and disparity completion results for both views after the first iteration. There are noticeable artifacts, especially in the right image. These artifacts are detected through disparity-color consistency check, shown in green in the first two images of the third row. The next two images in this row show the unreliable pixels found after the fourth iteration. As expected, the number of unreliable pixels reduces as more iterations are used. The last row presents the joint inpainting results after four iterations. More plausible results are achieved for both color and depth inpainting, compared to the ones generated without enforcing disparity-color consistency, *i.e.*, the ones in the second row.

Figure 3 shows more results on color image completion. As can be seen, our algorithm is able to recover salient structures that are largely occluded by the foreground object. This advantage mainly comes from the stereo config-

Figure 5. Inpainting comparisons with the approach of [5]: Our results are shown in Figure 2 and 3

uration which allows the exchange of reasonable amount of visual information between images automatically. By introducing depth information into the texture synthesis procedure our sample regions are more constrained compared to traditional exemplar-based approaches thus reduces the chance of sampling irrelevant texture information. For all these four data sets, usually 3 to 5 iterations can produce visually plausible results. The corresponding depth completion results for these data sets are shown in Figure 4.

Existing image completion algorithms may have difficulties producing satisfactory results for the data sets shown in this paper. Figure 5 shows some unsatisfactory completion results using our implementation of the approach in [5]. We carefully tuned the parameters so the best visual effects are presented. For comparison, corresponding results from our algorithm are given in Figures 2 and 3.

4. Conclusions

We have presented a novel technique that is able to jointly fill in the missing color and depth caused by removing foreground objects from stereo image pairs. Compared with conventional image inpainting approaches, the proposed algorithm makes a good use of the additional view available through: 1) letting the two stereo images to mutually complete each other so that the parts of missing region that are visible in the other view can be reliably filled; 2) using the estimated depth information to facilitate texture synthesis so plausible exemplars can be acquired; and 3) enforcing disparity-color consistency between the inpainting results for the two views so that unreliable solutions can be detected and revised. The experimental results demonstrate that, after the region for removal is identified, *stereoscopic inpainting* can automatically produce very good results.

The use of two images has its disadvantage as well, mainly in data acquisition. However we optimistically think that acquiring stereo images will become easier and easier. The success of our approach depends on the quality of the estimated depth maps. Therefore, it will suffer in cases where stereo methods fail, such as textureless regions and specular reflections.

In summary, *stereoscopic inpainting* is complementary to traditional single-image inpainting. It is particularly effective for more structured inpainting since the depth information provides additional cues as where the proper sample can be copied. Being an automatic method, it is also suited for batch processing of stereo videos.

Acknowledgements This work is supported in part by US National Science Foundation grant IIS-0448185, and a grant from US Department of Homeland Security.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of ACM SIGGRAPH*, 2000. 1

[2] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agarwala, B. Curless, M. Cohen, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Eurographics Symp. on Rendering*, 2007. 1

[3] M. Bleyer and M. Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *SPIE Symp. on Electronic Imaging*, pages 288–299, 2005. 2

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5), May 2002. 2

[5] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2003. 1, 4, 6

[6] Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *Proc. of Intl. Conf. on Computer Vision*, 2005. 2

[7] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *Proc. of ACM SIGGRAPH*, 2003. 1

[8] G. Egnal and R. P.Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1127–1133, August 2002. 2

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. 3

[10] M. Gong and Y. Yang. Fast stereo matching using reliability-based dynamic programming and consistency constraints. In *Proc. of Intl. Conf. on Computer Vision*, pages 610–617, 2003. 3

[11] J. Jia and C.-K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2003. 1

[12] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2001. 2

Figure 2. Iterative inpainting process. First row: stereo pair with uncovered areas labeled in green (two images on the left) and the results after mutual completion (two images on the right). Second row: the joint completion results of both views after one iteration. Third row: unreliable (green) pixels detected after 1 iteration (two images on the left) and 4 iterations (two images on the right), respectively. Fourth row: joint completion results using 4 iterations. Please note the improvements compared to the previous results shown in the second row.

[13] S. H. Kang, T. F. Chan, and S. Soatto. Inpainting from multiple views. In *Proc. Intl. Symp. 3D Data Processing Visualization and Transmission*, pages 622–625, 2002. 1

[14] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 15–18, 2006. 2

[15] N. Komodakis and G. Tziritas. Image completion using global optimization. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 1

[16] W. R. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *Symp. on Interactive 3D Graphics*, 1997. 3

[17] D. Scharstein and R. Szeliski. vision.middlebury.edu/stereo/. 2

[18] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 399–406, 2005. 2, 5

[19] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. In *Proc. of ACM SIGGRAPH*, pages 861–868, 2005. 1

[20] H. Tao, H. S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proc. of Intl. Conf. on Computer Vision*, pages 532–539, 2001. 2

[21] L. A. Torres-Mendez and G. Dudek. Reconstruction of 3d models from intensity images and partial depth. In *Proc. American Assoc. for Artificial Intelligence*, 2004. 2

[22] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla. Hole filling through photomontage. In *British Machine Vi-*

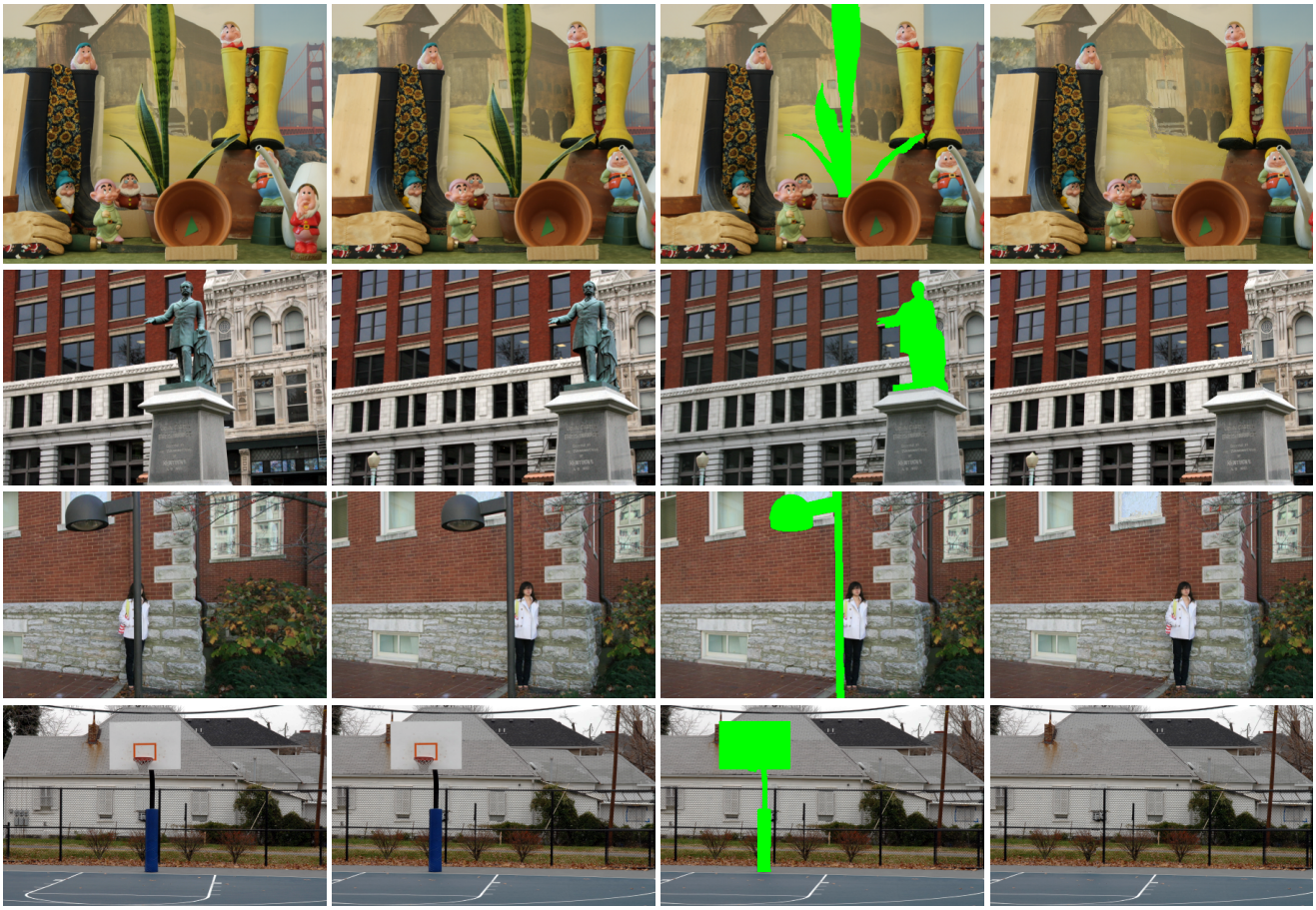

Figure 3. More results from our algorithm. The first two columns show the original stereo images. The third column shows uncovered areas in green. The fourth column demonstrates our image completion results.

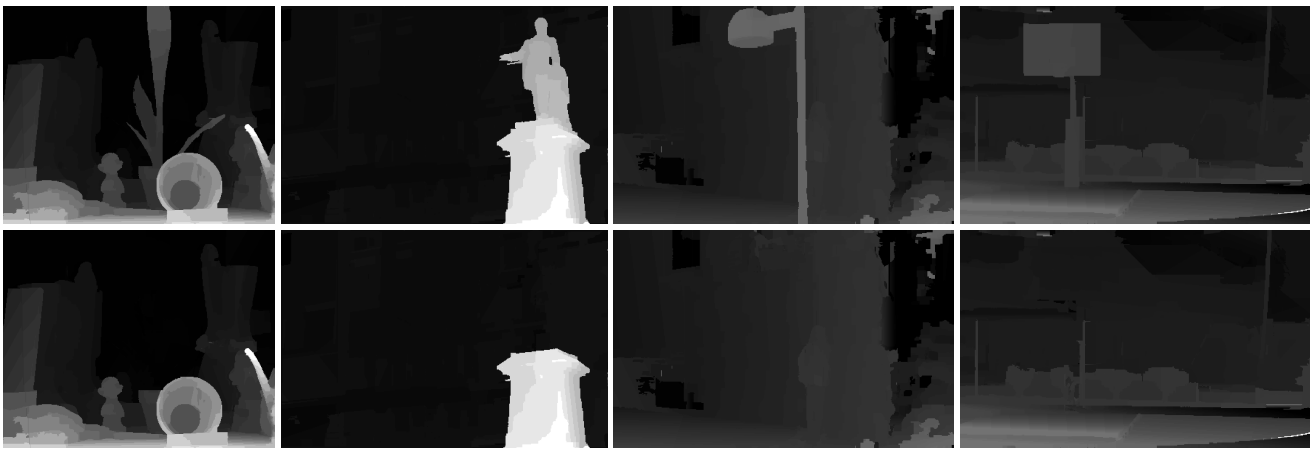

Figure 4. Depth completion results of corresponding images shown in Figure 3. First row: disparity maps after occlusion filling. Second row: disparity maps after object removal and depth completion.

sion Conf., 2005. 1

[23] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2347–2354, 2006. 2

[24] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Proc. of ACM SIGGRAPH*, pages 600–608, 2004. 2