Object Removal by Depth-guided Inpainting*

Liu He, Michael Bleyer, and Margrit Gelautz

Institute for Software Technology and Interactive Systems
Interactive and Multimedia Systems Group (IMS)
Vienna University of Technology, Austria
{*liu.he, bleyer, gelautz*}*@ims.tuwien.ac.at*

Abstract
Object removal by image inpainting aims at the visual uniformity of the inpainted blanks among their surroundings. Most inpainting algorithms pursue the structure continuity and texture similarity only in color. In this paper we take the view depth continuity into account and propose a depth-guided inpainting algorithm, in which a single color image and its associated disparity map are inpainted simultaneously. A fast exemplar-based inpainting is applied to fill the blank. Exemplars are randomly selected under depth constraints in initialization and optimized with a nearest neighbor search method in a semi-global way for smooth completion. Experimental results with datasets of different scenes demonstrate the positive impact of depth control in exemplar selection and the efficiency of the proposed algorithm.

1. Introduction and Related Work

As a scientific and technical term, image inpainting is first introduced in [3], which describes the artistic reconstruction process on a damaged digital image. More than the conventional image restoration/interpolation that aims at recovery on a small scale for denoising and super-resolution, the inpainting methods are usually designed for creative color filling in large blanks. Common applications are text [10] and object removal [6] from still images. In both cases, user defined undesired areas are left out and re-filled with color to keep the uniformity of the image. One natural extension of image inpainting is video inpainting [9, 13]. In recent years, inpainting has also started to play a role in disocclusion for image-based rendering [12].

The inpainting algorithms are generally classified into Partial Differential Equation (PDE) based inpainting and texture synthesis based inpainting [13]. In PDE based inpainting, color information is propagated along the isophotes from outside to inside of the blank region, according to a third-order PDE [2, 3, 7]. The filled-in color therefore varies smoothly to keep the continuity of the structure. However, the color changes in texture may be lost. In texture synthesis based inpainting, the target area is filled through texture replication. A texture can be copied from examples or generated procedurally from statistics over the whole image or a serial of images. The most popular texture synthesis approach is exemplar-based inpainting [6], in which the optimal exemplar is selected for each blank pixel by estimating the similarity between the template patch centered at the target pixel and the candidate exemplar. With this method, the repeated texture can be well preserved across the hole boundary. The PDE based and texture synthesis based inpainting methods can also be combined which improves inpainting results [4].

*This work was supported by the Doctoral College on Computation Perception at TU Vienna and the Austrian Science Fund (FWF) under project P19797. Michael Bleyer received funding from the Vienna Science and Technology Fund (WWTF) under project ICT08-019.

While in [6] the blank region is pixelwise inpainted according to the filling priority order, in [8] the coherence among the exemplars for neighboring pixels is enforced by global optimization. Instead of full search for an optimal exemplar, the approximate nearest neighbor search is adopted in [1, 15]. This allows fast exemplar selection, while pursuing smooth results in a semi-global way.

When coming to the problem of object removal by exemplar-based inpainting, our target is not only preserving the color continuity but also the geometric relationship among the remaining objects in the scene, i.e. the blank region should be inpainted with exemplars from the connected background (e.g. in Figure 1(a) , after removing the reindeer the red couch and the white cloth have to be completed). Hence, using depth information as guidance in image inpainting is supposed to improve the results. The depth information can be either measured by active sensing (e.g. Kinect, range camera etc.) or calculated by stereo matching. When using a disparity map generated by stereo matching, in which a pair of stereo images are available, the blank regions can first be filled by mutual completion to reduce blanks left for inpainting. In order to handle the case when the depth is acquired by active sensing, we use only a *single* color image and the corresponding disparity map in our algorithm. The disparity maps are either ground-truth generated by structured light or computed by a stereo matching method from [5]. The color image and the disparity map are inpainted simultaneously.

Our algorithm employs the PatchMatch method [1] for fast exemplar-based inpainting, however, uses a more sophisticated cost function to evaluate the exemplars. This cost function measures the similarity between two patches in terms of color, disparity and Euclidean distance. A related concept of cost function can be fund in [14] which requires a pair of stereo images. The authors calculate disparity maps from the stereo pair and remove the object from both views. The blanks are completed first with mutual information from the other view. The remainder is filled by exemplar-based inpainting with depth constraints on both views. Exemplars are searched through the whole covered region and the optimal ones are those which minimize the patch difference in color and disparity. The further depth constraint is implemented as a term of the cost function which penalize the exemplar with larger disparity than the removed object. The inpainting is performed pixel by pixel, i.e. once an empty pixel is filled up there is no chance to modify the color, even when obvious conflicts among neighboring pixels appear later. Different from [14], our approach only needs on a single image and its disparity map which brings down the work to half size. We use more strict depth constraints which are directly used as precondition in exemplar filtering to reduce the candidate scope. With this measure we do not only enhance the inpainting speed, but also the accuracy of exemplar search. Another advantage of our method is that the inpainting result is optimized dynamically by information propagation between neighboring pixels. As opposed to [14], we do not have the problem of early wrong commitments. For each blank pixel, the exemplar is improved through iterations while kept coherent with its neighbors. Moreover, we use the Euclidean distance between the target pixel and the candidate exemplar in our cost function to encourage sample copies from the nearby region.

2. Depth-guided Image Inpainting

The processing steps of our algorithm are as follows. Taking the original image I and its disparity map D as input, the algorithm starts with filling in the possible holes on the disparity map. The refined disparity map after pre-processing is denoted as D_{ref}. Then we remove the unwanted object from the color image and the disparity map. The uncovered pixels are denoted as Ω, while the remaining pixels are denoted as Ψ. Ψ is considered as source region that offers exemplars to fill Ω. The iterative exemplar-based inpainting method is applied to the color image and the disparity map simultaneously. The final results are a complete color image $I^{'}$ and a corresponding disparity map $D^{'}$.

2.1. Pre-processing of the depth map

All the disparity maps available in our research are not perfect, but contain uncovered holes due to occlusions or limitation of the acquisition tools. Therefore we first fill them up by depth propagation, which is done in eight directions. This means that, to each blank pixel $p_{i,j} \in D$, where (i,j) is the pixel position, the disparity is propagated from its eight connected neighbors so that there are eight candidate disparities for filling, which are denoted as $\{d_{i,j}\}$. Since blind regions usually appear near depth discontinuities where the foreground object partly occludes the background, the missing disparity is assumed to be propagated from the adjacent background. Using this assumption, we select $\min\{d_{i,j}\}$ as the disparity of $p_{i,j}$. Figure 1(b) and Figure 1(c) show the disparity maps before and after pre-processing, respectively. In addition, some disparity maps contain obvious error patches that may impact the exemplar selection[1]. We mark those patches also as holes to be filled in order to refine the disparity maps.

2.2. Foreground object removal

After pre-processing of the disparity map the user is asked to remove the undesired object from the color image to get I_{dmg}. This can be done with several common image editing tools. Using the uncovered blank Ω as a mask to remove the object also from the disparity map we get D_{dmg}. The removal order can be reversed, i.e. the object can be first removed from the disparity map, and then we use Ω to repeat the process on the color image. The user can choose one of the two schemes depending on whether the object is easier to segment in the color or disparity image. It also has to be pointed out that the foreground object is usually fatter in the disparity map generated by stereo matching due to the well-known edge-fattening problem of many stereo algorithms. In this case, removing the object first from the disparity map can avoid inconsistency at the dilated boundary. However this increases the number of pixels that have to be filled in.

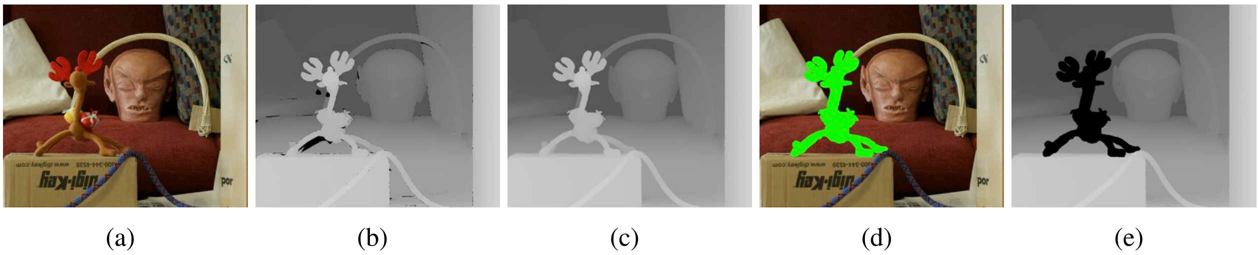

Figure 1. Pre-processing and object removal. From left to right: (a) original color image; (b) disparity map before and (c) after pre-processing (white for foreground and blank for background); (d) color image and (e) disparity map after object removal. Blank regions are marked in green and black, respectively.

2.3. Simultaneous color and depth inpainting

In this most challenging step of our pipeline, we fill the pixels of the blank region Ω to derive an inpainted color image I' and the disparity map D'.

Depth constraint and cost function. In the filling process, we modify the conventional exemplar-based inpainting, which checks only color similarities, based on the following two intuitions: (1) when an object is removed from the scene, the background behind this object becomes visible. Translated to our problem, this means that the inpainted disparity of a blank pixel has to be lower than its original

[1] For example, in the Middlebury ground-truth disparity maps, there are regions where disparity is unknown. Here, the structured light algorithm failed to generate depth information.

disparity (before object removal), because the pixel now shows the background. (2) To prevent visual artifacts, optimal exemplars should be selected based on both color and depth similarities.

We implement the two depth constraints first as hard constraints in exemplar candidate filtering. We take the $l \times l$ (in our experiments $l = 21$) square patch ψ_p centered at $p \in \Omega$ while partly in Ψ as template for exemplar selection. The original disparity at p before object removal is denoted as d_p. Let s of disparity d_s be one of the candidate exemplars, the first depth constraint can be formulated as

$$d_s \leq d_p \tag{1}$$

Moreover, based on the second depth constraint, we assume that the disparity of a blank pixel should be coherent with the disparities of its nonempty surrounding background. Using Equation 1 as a precondition, we define the valid disparity collection of ψ_p as $\{d_{\psi_p}\} = \{d_k \leq d_p \mid k \in \psi_p \cap k \in \Psi\}$. If $\{d_{\psi_p}\} \neq \emptyset$, we formulate our assumption as

$$\min\{d_{\psi_p}\} \leq d_s \leq \max\{d_{\psi_p}\} \tag{2}$$

This restriction further reduces the exemplar search range in large scale, as demonstrated in Figure 2.

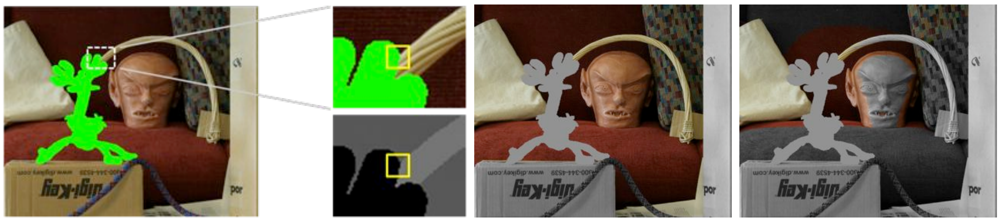

Figure 2. Example of search range restriction by depth. (a) Part of the color image with the inpainting mask is enlarged to give an example. The central pixel in the yellow frame belongs originally to the foreground; (b) the exemplar search range according to Equation 1 is represented with normal color and the excluded area in gray; (c) the search range is further restricted (reduced about 62%) according to Equation 2.

While the exemplars are filtered under hard constraints as described by Equation 1 and 2, the disparity similarity also performs as a soft constraint to guide the further selection. For each blank pixel p, the optimal exemplar q satisfies

$$\psi_q = \arg\min_{\psi_q \in \Psi} E(\psi_p, \psi_q) \tag{3}$$

where $E(\psi_p, \psi_q)$ measures the difference between patch ψ_p and the exemplar patch ψ_q. It is defined as

$$E(\psi_p, \psi_q) = f(E_{clr}(\psi_p, \psi_q)) + \alpha f(E_{dpt}(\psi_p, \psi_q)) + \beta f(E_{dst}(\psi_p, \psi_q)) \tag{4}$$

where $E_{clr}(\psi_p, \psi_q)$ and $E_{dpt}(\psi_p, \psi_q)$ are the sum of absolute differences between ψ_p and ψ_q in color (Euclidean distance between RGB-pixels) and disparity, respectively. $E_{dst}(\psi_p, \psi_q)$ is the Euclidean distance from p to q, which encourages exemplars selected from the nearby region of p. $f(x) = e^{-\gamma x}$ is a norm function that is robust for noise and outliers. In our experiment, α is 1.3, β is 1, γ takes 0.05, 0.01 and 0.1 for E_{clr}, E_{dpt} and E_{dst}, respectively. Note that only pixels in Ψ are valid in the computation of E_{clr} and E_{dpt}. Therefore, for a patch containing n valid pixels, E_{clr} and E_{dpt} are normalized by dividing by n. When the patch size l is smaller than half of the size of Ω, there must be some totally blank patches containing no valid pixels. In this case E_{clr} and E_{dpt} cannot be computed and are set to an infinite value [2].

[2] This measure prevents unreliable offsets from being propagated, however, which may also miss optimal exemplars. Variable patch size or hierarchical inpainting on multi-scaled images can be used to avoid this problem.

Optimization. Denoting the two dimensional vector pointing from the blank pixel in Ω to its exemplar in Ψ as the offset vector $\mathbf{v}$, our goal is translated to find the optimal offset map $\mathbf{V}$. This offset map is then directly used for inpainting. To compute the offset map, we employ the approximate nearest-neighbor algorithm of [1], which starts with a randomly generated initial offset map and refines it iteratively. In each iteration offsets are propagated among pixels in Ω , which is followed by a random search for new exemplars. Different from [1], which uses pure random initialization/search and takes only color similarity into account in offset evaluation, our approach makes use of depth information to reduce search scope and to enhance the correctness in propagation. In the following we explain how the depth constraints are applied in each step.

Initialization. After candidate filtering based on the depth constraints, the exemplars are selected randomly from valid candidates. The colors and disparities of blank pixels are updated. Then we get the initial solution shown in the first column in Figure 4.

Propagation. If we denote the coordinate of p as (i, j) and the offset between p and its exemplar q as $\mathbf{v}(i, j)$, to update the exemplar is actually improving $\mathbf{v}(i, j)$. Let $E(\mathbf{v}(i, j)) = E(\psi_p, \psi_q)$ calculated by Equation 4 denote the patch difference between ψ_p and ψ_q. We strive to reduce $E(\mathbf{v}(i, j))$ with $\mathbf{v}(i-1, j)$ and $\mathbf{v}(i, j-1)$ using the assumption that the neighboring offsets are likely to be the same. In other words, we take $\mathbf{v}'(i, j)$, which is defined as arg $\min\{E(\mathbf{v}(i, j)), E(\mathbf{v}(i-1, j)), E(\mathbf{v}(i, j-1))\}$, to be the new value of $\mathbf{v}(i, j)$, if $E(\mathbf{v}'(i, j)) \leq E(\mathbf{v}(i, j))$. If ψ_p is totally blank, the patch difference is assumed to be $E(\mathbf{v}'(i, j)) = a \cdot \min\{E(\mathbf{v}(i-1, j)), E(\mathbf{v}(i, j-1))\}$, in which a is a propagation factor and is set as 1.02 in our experiments. The described offset propagation is performed in odd iterations to enforce coherent exemplar mapping from up and left to down and right. In even iterations, the propagation is performed along the reverse scan order, which improves $\mathbf{v}(i, j)$ with $\mathbf{v}(i + 1, j)$ and $\mathbf{v}(i, j + 1)$. Note that the magnitude at (i, j) on the offset map V corresponds $E_{dst}(i, j)$. The offset initialization and propagation are illustrated in Figure 3(a) and 3(b).

Random search. At the end of each iteration new candidate exemplars are randomly selected. The selection is under the same constraint as in the initialization. As the depth control reduces the search range to a reasonable size already, in the first three iterations we do not further restrict the search range to avoid getting trapped in suboptimal local minima. In subsequent iterations, we restrict the search to the concentric neighborhoods of the current exemplar (see Figure 3(c)). For each blank pixel (i, j) the exemplar offset $\mathbf{v}(i, j)$ is updated when the cost $E\{\mathbf{v}(i, j)\}$ is reduced by the new exemplar.

Through the above described steps the inpainting problem is converted to building an optimal offset map with combinational use of the color and depth information in exemplar evaluation. Then the color image and disparity map are inpainted simultaneously according to the offset map. To avoid noises and to prevent high frequencies in the inpainted color/disparity image, each empty pixel $p \in \Omega$ is not filled simply by copying the exemplar but taking the median value of the suggested colors/disparities according to offsets within the $l \times l$ square patch centered at p. Figure 4 shows the initial solution as well as the optimization results after the first, third and fifth iterations. Note that the solution converges very rapidly. Therefore we iterate the optimization only five times in our experiments.

3. Experimental Results

We test our algorithm on Middlebury datasets [11], which include ground-truth disparity maps, and outdoor pictures or frames from stereo videos. To obtain the disparity map for the outdoor scenes, we use the stereo matcher of [5]. The maximum and minimum image sizes are 671×555 pixels and 702×278 pixels, respectively. The size of the removed objects is approximately $10 \sim 20\%$ of the image size. A 5-iteration inpainting for a single image and its disparity map with our MATLAB implementation takes up to 2 minutes on a 3.07 GHZ CPU.

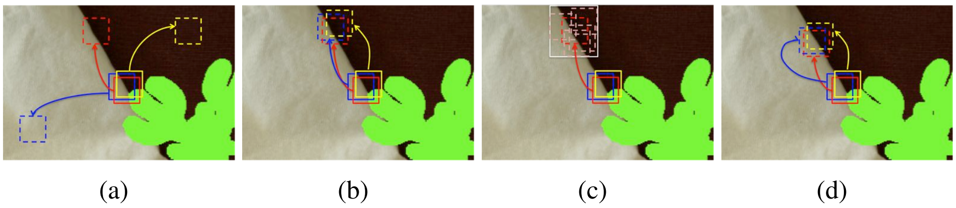

Figure 3. Randomized nearest neighbor search. (a) Initialization: randomly selected exemplar assignments; (b) Offset propagation: the blue patch checks its above neighbor (yellow) and right neighbor (red) and chooses the offset of the right neighbor to improve the mapping; (c) Search: for each empty pixel, a new offset is randomly selected from its neighborhoods in the last few iterations; (d) Optimized result: after new random search and propagation the optimal patches are selected while keeping the coherence between neighboring pixels.

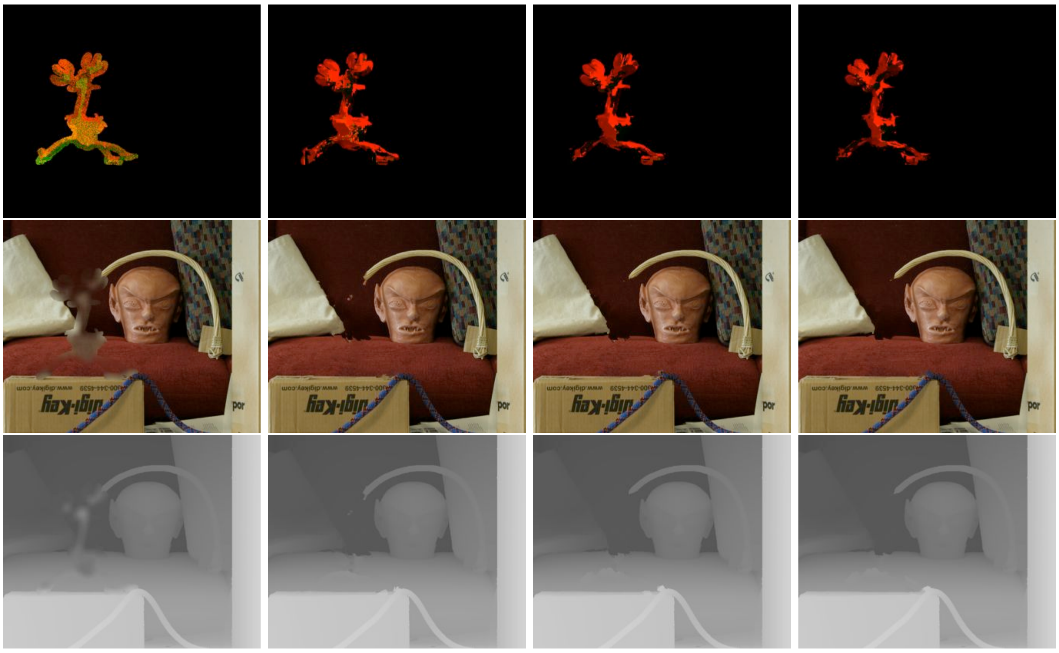

Figure 4. Simultaneous color and disparity inpainting. Rows from up to down are offset maps, inpainted color images and disparity maps. Columns from left to right are the initial solution and the results after the first, third and fifth iteration. In the offset maps, the magnitude is visualized with saturation and the angle with hue.

Figure 5 shows some test results produced by no-depth-guided and depth-guided inpainting with the same filling scheme, i.e. initialization, iterative propagation and random search. On the images inpainted without depth constraints (shown in the third row of Figure 5), some regions of the background (marked with solid frames) are filled with exemplars (marked with dashed frames) from foreground closer than the removed object, foreground closer than the blank surrounding, or background of very different disparities. Such errors are avoided in the results with our algorithm because exemplars can only be chosen from background further than the removed object and the disparities of the exemplars should be similar to the disparities of the outer boundary of Ω. For the same reason, the inpainted disparity maps shown in Figure 6 are reasonable in structure. However, as the disparity is only copied from the source region Ψ, gradual changes along inclined planes are hard to preserve; An example can be seen in the inpainted disparity of the vehicle body in the second column in Figure 6.

4. Conclusion and Future Work

We have presented an algorithm for object removal by exemplar-based inpainting. In contrast to conventional inpainting approaches, our approach makes use of depth information which leads to two

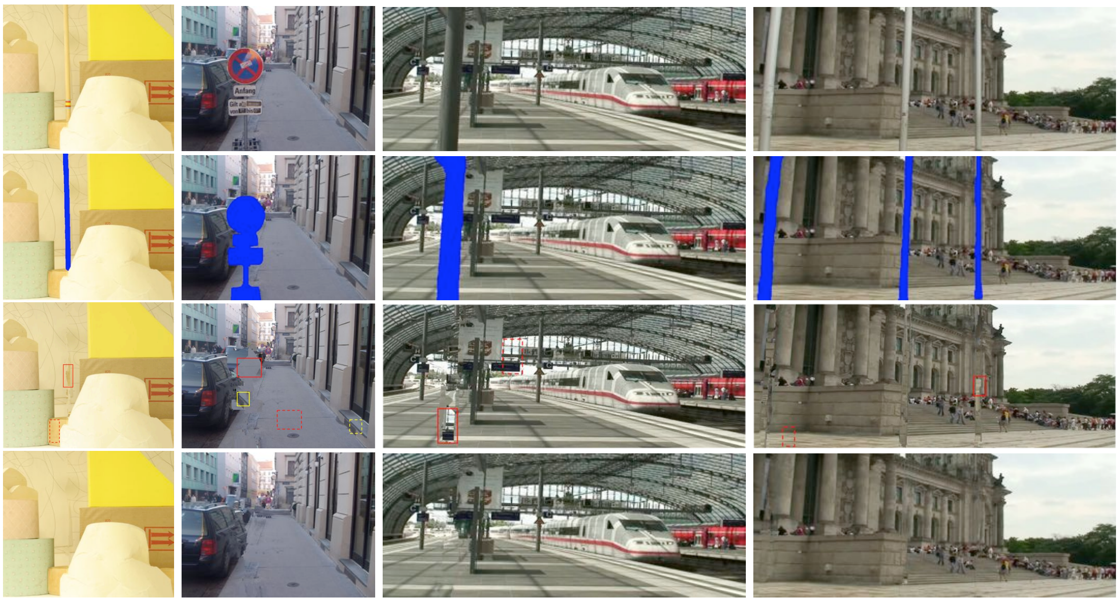

Figure 5. Color inpainting. Columns from left to right are the datasets ”lampshade” from Middlebury [11], ”sign” taken by stereo camera, ”train” and ”parliament” from a stereo video. Rows from up to down are original color images, images after object removal and inpainting results without and with depth guidance. On the inpainted images without depth guidance, some background regions (marked with solid frames) are inpainted with exemplar patches (marked with dashed frames) from foreground closer than the removal object (e.g. ”lampshade”), foreground closer than the blank surrounding (e.g. ”sign” and ”parliament”) or background of very different disparities (e.g. ”train”). Such errors are avoided in the depth-guided results.

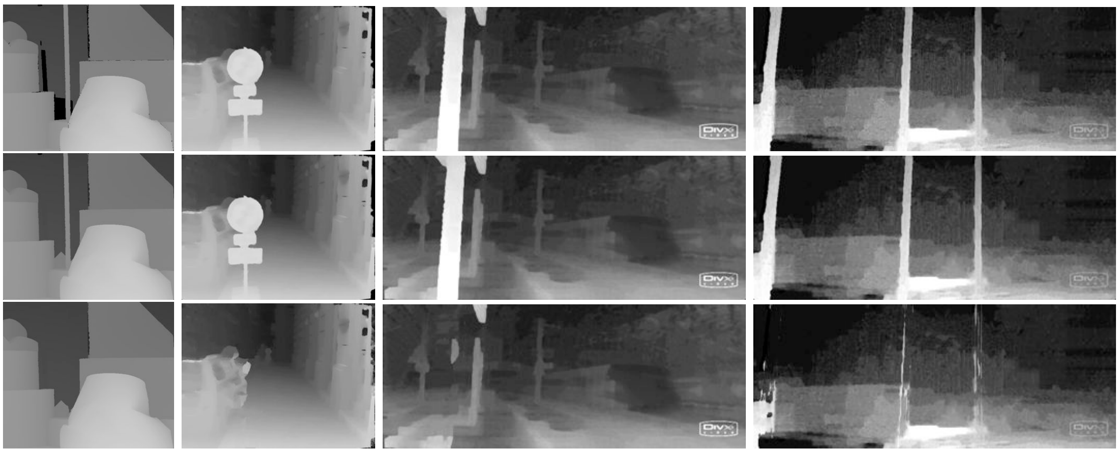

Figure 6. Depth inpainting. Rows from up to down are original disparity maps, disparity maps after preprocessing, inpainting results. Columns from left to right are dataset ”lampshade” from Middlebury [11], ”sign” taken by stereo camera, ”train” and ”parliament” from stereo video. Errors around the inpainted region in ”parliament” is due to the edge-fattening problem in stereo matching.

advantages. (1) We fill depth and color images in a combined fashion. Hence, the disparity inpainting helps the color inpainting and vice versa. (2) We only allow selecting exemplar patches that make sense from a 3D perspective, i.e. a blank pixel must be filled with colors originating from background objects. We have used the recently-proposed PatchMatch algorithm [1] to optimize our inpainting results. This makes our algorithm computationally tractable and circumvents the problem of early wrong commitment to wrong exemplar (such as e.g. in [13]). In our experiments we have shown that our depth-guided inpainting technique produces results superior over standard inpainting that looks at colors only. Future work will concentrate on extending our method to inpainting of video material where the challenge is to preserve visual coherency of inpainting results over time.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAPH '09*, pages 1–11. ACM, 2009.

[2] M. Bertalmio, Bertozzi A.L., and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *CVPR '01: Conference on Computer Vision and Pattern Recognition*, pages I–355–362, 2001.

[3] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles. Image inpainting. In *SIGGRAPH '00*, pages 417–424. ACM, 2000.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. In *CVPR '03: Conference on Computer Vision and Pattern Recognition*, pages II–707–12, 2003.

[5] M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *VISAPP '08: International Conference on Computer Vision Theory and Applications*, pages 415–422, 2008.

[6] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *CVPR '03: Conference on Computer Vision and Pattern Recognition*, pages II–721–728, 2003.

[7] J. Jia and C. K. Tang. Image repairing: robust image synthesis by adaptive nd tensor voting. In *CVPR '03: Conference on Computer Vision and Pattern Recognition*, pages 643–650, 2003.

[8] N. Komodakis and G. Tziritas. Image completion using global optimization. In *CVPR '06: Conference on Computer Vision and Pattern Recognition*, pages I–442–452, 2006.

[9] K.A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007.

[10] E.A. Pnevmatikakis and P. Maragos. An inpainting system for automatic image structure - texture restoration with text removal. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2616–2619, 2008.

[11] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR '07: Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[12] Z. Tauber, Ze-Nian Li, and M.S. Drew. Review and preview: Disocclusion by inpainting for image-based rendering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(4):527–540, 2007.

[13] M.V. Venkatesh, Sen-Ching S. Cheung, and J. Zhao. Efficient object-based video inpainting. *Pattern Recognition Letters*, 30(2):168–179, 2009.

[14] L. Wang, H. Jin, R. Yang, and Gong M. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *CVPR '08: Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[15] Y. Wexler, E. Shechtman, and Irani M.. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.