

Region Completion in a Single Image

Yunjun Zhang

Jiangjian Xiao

Mubarak Shah

Computer Vision Lab, School of Computer Science
University of Central Florida, Orlando, Florida 32816, USA

Abstract

Natural images and Photographs sometimes may contain stains or undesired objects covering significant portions of the images. Region completion is a method to fill in such significant portions of an image by using the information from the remaining area of the image. We propose a novel approach to achieve the completion in three steps. First, a spatial-range model is determined to establish the searching order of the target patch. Second, a source patch is selected by measuring the adjusted appearance of the source patch with the target patch and enforcing the searching area in the neighborhood around the previous source patch. Third, a graphcut patch updating algorithm is designed to ensure the non-blurring updating. A number of examples are given to demonstrate the effectiveness of our method.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation I.4.1,4,9 [Image Processing and Computer Vision]: Sampling, Restoration, Applications

1. Introduction

Removing objects or large portions of an image then filling in the missing data is a critical problem in numerous applications, such as image repairing, removing unnecessary objects (Figure 1) or overlaid text in a scene. There are two primary categories of the work that focus on missing image data recovery. One was introduced by Bertalmio [BS00], who used PDE based method to repair damaged images. The idea is to extend inward the structures arriving at the boundaries of the damaged area. For an image in which only small portions are missing, this approach can achieve highly smoothed results. However, the lack of texture in a large reconstructed area is easily visible. Therefore this approach is ineffective for filling in large holes in the natural images. Levin et. al. [ALW03] extended the idea by measuring global image statistics, so that the inpainting results are based on the prior image knowledge besides the local color information.

Recently some researchers have considered texture synthesis based method as a way to achieve image completion [BVS03, ACT03, DCOY03, JT03, IP97]. Criminisi et. al. used the angle between the isophote direction and the normal direction of the local boundary to define the searching order of the patches, so that the structure of the missing



Figure 1: Removing a large object in an image. Left: A campus image with a map board in the front. Right: The board is removed by our method, the perspective deformation of building is restored in the removed region.

region can be filled before filling in the texture [ACT03]. Jia and Tang [JT03] explicitly segmented the unknown area into different homogeneous texture areas using tensor voting method. Drori et. al. [DCOY03] incorporated pyramid image approximation and adaptive neighborhood size together to achieve impressive results. However, the method is slow due to the high computational complexity.

Most previous texture-synthesis based approaches use extensive search to find the source texture patches. This is time

consuming and ignores the local similarity in natural images. For images with perspective deformation, existing approaches may not be applicable if the missing patches have no exactly similar source patches available in the known region due to the deformation. In order to achieve smooth completion results, most existing approaches apply a diffusion method to merge the patches. The drawback of this is that the results get blurred due to the diffusion step.

Our method belongs to the second category. The proposed algorithm has three main steps. First, a spatial-range model is determined to establish the searching order of the target patch. Second, a source patch is selected by measuring the adjusted appearance of the source patch with the target patch and enforcing the searching area in the neighborhood around the previous source patch. Third, a graphcut patch updating algorithm is designed to ensure the non-blurring updating.

2. Region Completion Algorithm

In our approach, the input is a natural or synthetic image containing an area which is damaged or masked out by a mask. This area is also referred to as the unknown area. The output is an image where the unknown area is filled by synthesized texture based on the whole image information. In this paper, we denote the known region by Φ , the unknown area by Ω and the boundary of Ω by $\partial\Omega$. The source and target patches are denoted as Ψ_s and Ψ_t respectively.

2.1. Filling Order

In this paper, the textureiness in a neighborhood area around \mathbf{x}_0 in image $I(\mathbf{x})$ is defined as the following:

$$\mathcal{D} = \frac{1}{|\mathcal{A}|} \int_{\mathcal{A}} g(\xi, \mathbf{x}_0) f(I(\xi), \mu) d\xi$$

where \mathcal{A} denotes the neighborhood area, $f(I(\xi), \mu)$ measures the photometric distance between $I(\xi)$ and the mean color value μ of the neighborhood. Function $g(\xi, \mathbf{x}_0)$ is a Gaussian function in terms of the spatial distance between \mathbf{x}_0 and ξ . Thus, the spatial and range information are well organized together to represent the textureiness of the textures. A large value of \mathcal{D} means that the texture in the neighborhood has more high frequency information while a small value of \mathcal{D} means that the texture in the area is more likely to be a constant color.

In order to keep the performance at a reasonable level, we select five potential patch locations which contain the largest known region, and compute \mathcal{D} for each of them. Next, we select the patch with the smallest \mathcal{D} value as the target patch, Ψ_p . Therefore, the area with less textureiness will be filled in first.

2.2. Patch Matching

After determining the highest priority patch, Ψ_t , a patch matching step is applied to find the source patch, Ψ_s in Φ ,

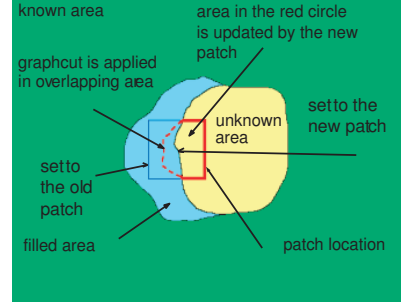


Figure 2: the known area is shown by green and the unknown area is shown by yellow. Two overlapping are shown in the square area. The red dotted line is the minimal cut.

which has the best similarity with Ψ_t . We define the center of the previous target patch, the current target patch, the previous source patch, the current source patch as \mathbf{x}_{t1} , \mathbf{x}_{t2} , \mathbf{x}_{s1} and \mathbf{x}_{s2} respectively. If \mathbf{x}_{t1} and \mathbf{x}_{t2} are close enough, \mathbf{x}_{s2} has a very high possibility to appear around \mathbf{x}_{s1} . Therefore we can reduce the search space Φ_s from Φ to a neighborhood area around \mathbf{x}_{s1} , if the distance between \mathbf{x}_{t1} and \mathbf{x}_{t2} is small.

For the similarity measurement, we make an assumption that the content in the unknown area, Ω , is similar to the content (texture, intensity) of the known region, Φ . The similarity can be expressed as follow:

$$\Psi_s = \arg \min_{\Psi_i \in \Phi_s} \frac{d(\Psi_t, \Psi_i)}{N_t},$$

where the distance $d(\Psi_t, \Psi_i)$ between the two patches is defined as the sum of squared difference (SSD), N_t is the number of pixels in the known area of the target patch, Ψ_t .

This simple approach works well for images without perspective deformation. However, for natural images, this condition may not be true. In order to handle the deformation, we need to estimate the projective transformation parameters between the two patches before applying the similarity measurement.

Given the two patches Ψ_t and Ψ_s , let \mathbf{x}' and \mathbf{x} be the corresponding pixel locations in patch Ψ_t and Ψ_s respectively.

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{\mathbf{A}\mathbf{x} + \mathbf{b}}{\mathbf{c}^T \mathbf{x} + 1}$$

where \mathbf{A} is a 2×2 matrix and \mathbf{b} and \mathbf{c} are 2×1 vectors. We use a pseudo-perspective model to approximate the transformation parameters and design an iterative scheme to achieve a good approximation of the transformation parameters. In order to avoid extreme deformation, we set thresholds on \mathbf{b} and \mathbf{c} to ensure that the transformation parameters are restricted to a reasonable range.

After estimating the pseudo-perspective parameters, the

patch Ψ_s is warped to Ψ_t using the parameters. Then, the similarity measure can be calculated between Ψ_t and Ψ_s .

2.3. Patch Updating

After exploring the spatial similarity between these two patches, we need to update the patch Ψ_t by Ψ_s . We reformulate the problem from a merging problem to a separating problem: given two similar and spatially overlapping patches, where a cut can be made to separate those two patches and to keep the seam least noticeable. This problem can be solved by a maximum flow or minimum cost graphcut algorithm as shown in Figure 2.

In our case, the patches can be cut only in the overlapping region, where both patches have known information. We define each location in the overlapping region as a vertex v_i . Let $C_t(v_i)$ and $C_s(v_i)$ be the color value at the location v_i in Ψ_t and Ψ_s respectively. Therefore, the weight function, $W(v_i, v_j)$, between vertices v_i and v_j can be defined as follows:

$$W(v_i, v_j) = \begin{cases} (\|C_t(v_i) - C_s(v_i)\| + \|C_t(v_j) - C_s(v_j)\|) & \text{if } \{v_i, v_j\} \in \mathcal{N}, \\ \infty & \text{otherwise} \end{cases}$$

where function $\|\cdot\|$ denotes the Euclidean distance between color values, and \mathcal{N} is a 4-connected neighborhood. After defining the weight function as above, the minimal cut can be easily computed by standard graphcut algorithm. A small weight means that if the cut runs between the pair of vertices, the four resulting color pairs $C_t(v_i)$ and $C_s(v_j)$, $C_s(v_i)$ and $C_t(v_j)$, $C_t(v_i)$ and $C_t(v_j)$, $C_s(v_i)$ and $C_s(v_j)$ do not have much difference. Therefore, the cut gives the least noticeable seam. On the contrary, the large weight between two vertices implies that a seam between the two vertices is more noticeable.

In order to start the cut, some vertices on the current boundary $\partial\Omega$ must be constrained to belong to the new patch and a few vertices on the boundary of the old patch and in the area of Φ will be set to stay in the old patch, Figure 2. Since the image information in the known region Φ should not be changed, this graphcut step is not applied to Φ .



Figure 3: Left: Images of a wall with a missing part. Middle: The result using the graphcut updating method. Right: The result using the direct updating method. There is noticeable structure discontinuity inside of the blue circle area.

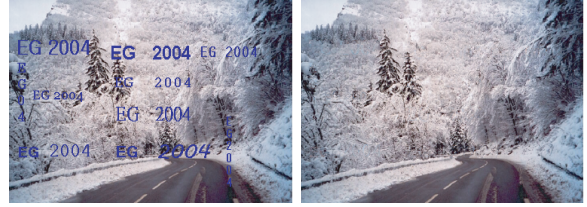


Figure 4: Text removing. Left: A snow field image with super-imposed text. Right: The text is removed by our method.

3. Experimental Results and Comparisons

Our algorithm has been applied to a number of images ranging from stochastic textures to natural images with projective deformation. Since the quality of the results apparently corresponds to the human perception of the appearance in the completed images, we visually demonstrate the results and comparisons without giving any quantified measurements.

Figure 1 demonstrates the advantage of using projective transformation compensation to match the target patch. This image is taken of an university campus containing a building, a car and a map board. The building in the background has a large noticeable projective deformation. Our method can remove the map board and restore the region of the building in the missing area reasonably.

Figure 3 compares the results obtained by graphcut updating method with the result obtained by direct updating method. The graphcut version maintains the wall structure very well, however, the direct updating version fails on a portion of the image. We also remove the super-imposed text from a snow field image in Figure 4. In Figure 5 and Figure 6, we compare our results with other proposed methods. In both cases, our results outperform the other results.

4. Conclusion and Future Work

In this paper, we introduced a novel technique to fill in large missing areas in the natural images. Our approach propagates reasonable texture information into the missing area to make a visually plausible image without noticeable artifacts.

We separate this hard region completion problem into three critical steps. First, a spatial-range model is determined to establish the searching order of the target patch. Second, a source patch is selected by measuring the adjusted appearance of the source patch with the target patch and enforcing the searching area in the neighborhood around the previous source patch. Third, a graphcut patch updating algorithm is designed to ensure the non-blurring updating.

We intend to extend our approach from image region completion to video region completion. The difficulties in video region completion include global motion compensation and maintaining consistency of the unknown area over the whole video sequence.

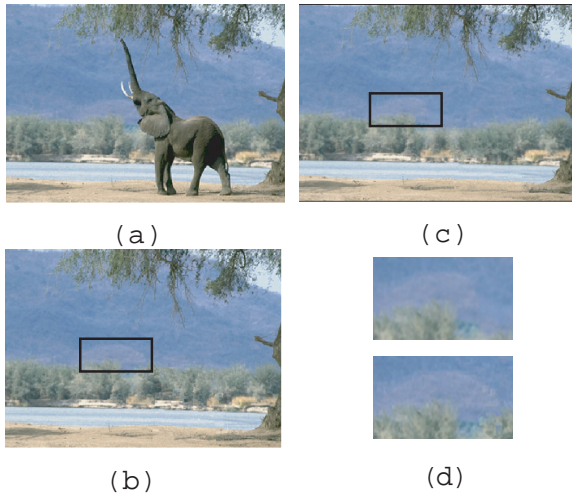


Figure 5: Comparison with the fragment-based completion approach. (a) Original image. (b) Results obtained by our method. (c) Results obtained by the fragment-based approach [DCOY03]. (d) The enlarged areas from the marked region in (c) and (b) respectively. Our results are the bottom one, and contains more continuous background landscape.

References

[ACT03] A. CRIMINISI P. P., TOYAMA K.: Object removal by exemplar-based inpainting. *CVPR* (2003). 1

[ALW03] A. LEVIN A. Z., WEISS Y.: Learning how to inpaint from global image statistics. *ICCV* (2003). 1

[BS00] BERTALMIO M., SAPIRO G.: Image inpainting. *SIGGRAPH* (2000). 1

[BVS03] BERTALMIO M., VESE L., SAPIRO G.: Simultaneous structure and texture image inpainting. *CVPR* (2003). 1, 4

[DCOY03] DRORI I., COHEN-OR D., YESHURUN H.: Fragment-based image completion. *ACM Transactions on Graphics* 22, 3 (2003), 303–312. 1, 4

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 341–346.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2* (1999), p. 1033.

[HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based

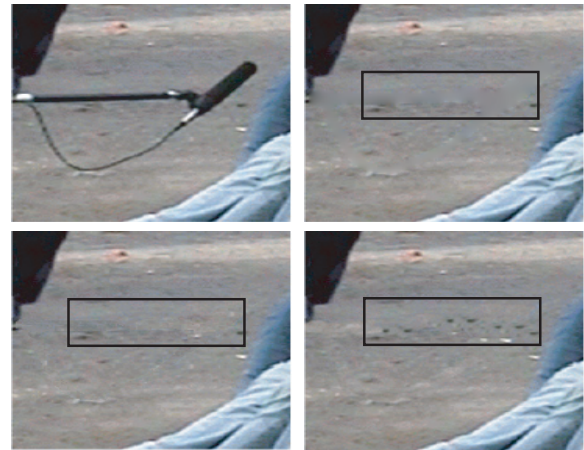


Figure 6: Algorithm comparison. Top left: Original image, where the microphone needs to be removed. Top right: Results obtained by inpainting method [BVS03]. The texture in the removed area can not be constructed. Bottom left: Results obtained by our method. Bottom Right: Result obtained by [JT03]. This image contains a lot of similar texture patterns in the removed area. In the area marked by the black rectangles, the results obtained by our approach provides more detailed and coherent result than the others.

texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM Press, pp. 229–238.

[HT96] HIRANI A. N., TOTSUKA T.: Combining frequency and spatial domain information for fast interactive image noise removal. *Computer Graphics* 30, Annual Conference Series (1996), 269–276.

[IP97] IGEHY H., PEREIRA L.: Image replacement through texture synthesis. In *Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 3* (1997), p. 186. 1

[JT03] JIA J., TANG C.-K.: Image repairing: Robust image synthesis by adaptive nd tensor voting. *CVPR* (2003). 1, 4

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics* 22, 3 (2003), 277–286.

[MP95] MANN S., PICARD R.: Video orbits of the projective group: a new perspective on image mosaicing. *Technical Report 338, MIT Technical Report*. (1995).