A Novel Texture Synthesis Based Algorithm for Object Removal in Photographs

Feng Tang, Yiting Ying, Jin Wang, and Qunsheng Peng

State Key Lab of CAD & CG, Zhejiang University,
310027 Hangzhou, P. R. China
{tang, ytying, jwang, peng}@cad.zju.edu.cn
http://www.cad.zju.edu.cn

Abstract. Natural images and photographs sometimes may contain stains or undesired objects covering significant portions of the images. Inpainting is a method to fill in such portions using the information from the remaining area of the image. In this paper, we propose a novel photograph editing framework that utilizes texture synthesis techniques. Major contributions of our algorithm are: 1) a constraint-based candidate patch searching method which limits the searching within neighboring region with similar texture; 2) a metric of Coherence Confidence for selecting the best fit candidate preventing error accumulation and propagation; 3) integration of graphcut optimization to make the seam visually invisible. Experiments show that our system can efficiently handle different cases especially large regions in complex background.

1 Introduction

In practice, we often meet with such problems as old photos spoiled by ink or old paintings full of scratches after long time reservation. Besides, photos may contain undesired large objects to be removed, for example, a passing-by person may drop into the view when a photo is taking. Although these two types of problems originate from different scenarios, they are commonly known as inpainting. How to robustly and efficiently solve both problems remains a challenge.

Several works have addressed these problems. For the first type of problems, which contains scratches or small missing regions, there are diffusion based and filter based methods. Bertalmio [1]-[2] filled in the tiny spoiled regions by propagating information from the outside of the masked region along level lines (isophotes). Reference [3] proposes Total Variational (TV) inpainting algorithm that invokes Euler-Lagrange equation and applies anisotropic diffusion [5] inside the inpainting domain based on the contrast of the isophotes. Oliveira [6] filtered the input image using a Gaussian convolution kernel to remove the undesired flaws. Yet such algorithms account for only small local regions. Artifacts of blurs may occur when they are applied to large region of inpainting.

For the second type of problems, a kind of synthesis based methods has been proposed. Liang et al. [7] composed the texture of a large region by selecting

M. J. Maher (Ed.): ASIAN 2004, LNCS 3321, pp. 248–258, 2004.

rectangle candidate patches recursively from the input texture. Efros and Freeman [8], Vivek Kwatra et al [9] searched for irregularly-shaped patches with optimizations, using dynamic programming and graph-cut respectively. Criminisi [10] employed texture synthesis to remove large objects in photographs and obtained impressive results. Such synthesis based methods consist of two major steps: searching and pasting. Searching is the process of finding best matched patch in the source region based on texture similarity. Pasting is the process of attaching the selected patch to the target region at the desired position. Nevertheless, most current synthesis based algorithms suffer from two major limitations. 1) Searching the source patch globally will easily lead to error match, and such error will accumulate and propagate to other areas quickly, even makes the results completely unacceptable. 2) The global search is computationally expensive, which limits the overall fill-in speed. In this paper we propose a photograph editing framework which can robustly remove various kinds of undesired details from photographs. By adopting local constraints for searching and estimating the coherence confidence for the candidate patch, our synthesis-based object removal algorithm achieves better results and performs more robustly and faster than previous methods.

The rest of the paper is organized as follows: section 2 explains our synthesis method for large object removal. Section 3 shows the results of the proposed method and section 4 draws the conclusion.

2 Texture Synthesis Based Image Editing for Object Removal

2.1 Example Based Object Removal

In [10], Criminisi et al proposed an exemplar based inpainting method, which fills in the target region with patches from the source region possessing similar texture. The candidate patches are selected from the whole image with special priority to those along the isophotes (lines of equal gray value) so as to preserve the linear structure during the filling-in. This process is quite similar to patch matching in texture synthesis and the fill-in priority is inspired by the partial differential equations method of physical heat flow.

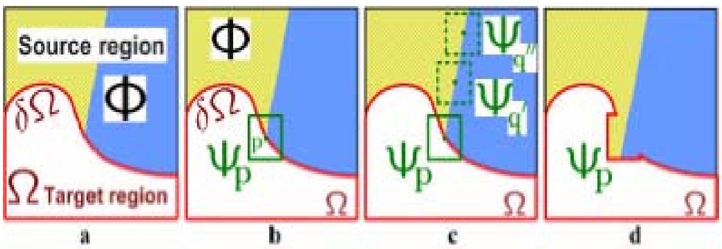

Fig. 1. Example based object removal procedure

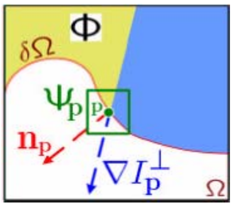

Fig. 2. Priority computation

As shown in Fig.1, suppose the target region is denoted as Ω and its contour as δΩ, Φ is the source region. p is a point on the contour. Let ψp denote the current patch to fill. Although there exist many matched candidates in Φ indicated as ψq', ψq'' (Fig.1 (c)), the selection is made based on the priority of the candidate patches. The priority of each contour point is calculated as follows:

$$P(p) = C(p) * D(p) , \tag{1}$$

where $C(p)$ is the confidence term that indicates the reliability of the current patch, and $D(p)$ is the data term that gives special priority to isophotes as demonstrated in Fig. 2. $\nabla I_p^{\perp}$ is the isophote and n_p is the normal at point p . α is the normalization factor.

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \bar{\Omega}} C(q)}{|\Psi_p|} . \tag{2}$$

$$D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{\alpha} . \tag{3}$$

2.2 Our Algorithm

To achieve robust and fast filling results, we propose a novel texture synthesis method called coherence-based local searching (CBLS) for region filling. As we can see from above, the major disadvantage of Criminisi's method is the global searching, which not only leads to error match but also greatly decreases the system performance. We assume that a photo can be described as a Markov Random Field (MRF) and the neighbor regions provide sufficient information to decide what to fill. Based on this assumption, we propose a local search strategy to find the candidate patch in the neighbor regions.

2.3 Constraint Based Patch Searching

Fig. 3 shows an example of removing some undesired figures from a photo by employing Criminisi's approach. The red region is the target to fill in, the current patch under processing is indicated by a yellow rectangle and the selected candidate is shown in green, both are scaled for clear view in Fig. 3 (b) and 3 (c). Unfortunately, this is an error match. After the error patch is copied to the target region, the consequent patch matching will use this error information to search for the next candidate in the global domain. As shown in Fig. 3 (d), 3 (e) and 3 (f), 3 (g), the initial error will accumulate and propagate to a large region.

To make a correct selection from the candidate pool, we introduce a reference image called segment map which corresponds to a coarse segmentation of the original image. As an initialization step, the whole image is segmented into several seprate regions according to the texture similarity. We adopt Meanshift operator here to achieve this segmentation, for technical details please refer to [11]. Suppose the whole image is denoted by I, it is divided into N regions with each region as T_i, then we have:

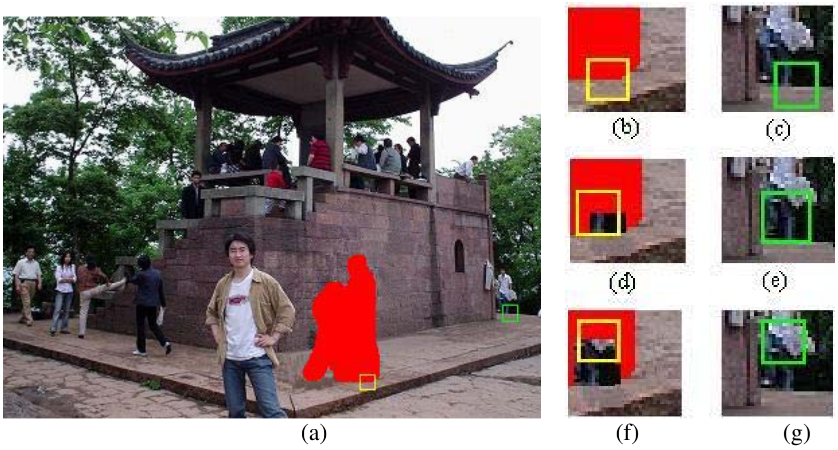

Fig. 3. Error propagation. Red region is the target to fill in. the yellow rectangles and green rectangles in (b), (c), (d), (e), (f), (g) indicate respectively the current patch and the selected candidate patch under processing and patches at successive steps

$$I = \bigcup_{i=1}^{N} T_i \quad \text{and} \quad T_i \cap T_j = \varnothing \quad (1 \le i,j \le N)\,. \tag{4}$$

During the synthesis, the patch matching is restricted within the relevant regions which the current patch overlaps with (an example is shown in Fig. 4):

1) *If the current patch is completely within one segment* T_i, *searching is limited within* T_i, *this eliminates the type of error in Fig. 3.*
2) *If the current patch intersects k segments* $T_i \sim T_{i+k}$, *searching is limited in the combined region of all the k segments* $T_i \cup T_{i+1} \cup \ldots \cup T_{i+k}$.

Although a local search is performed, we may still find many candidate patches. We adopt the concept of coherence confidence to select the best fit from candidate pool. The key idea of our approach is that the texture should be spacially coherent, we then choose the one from the candidates such that it has the greatest coherence confidence (measured in formula (7)) with the neighbor patches already filled.

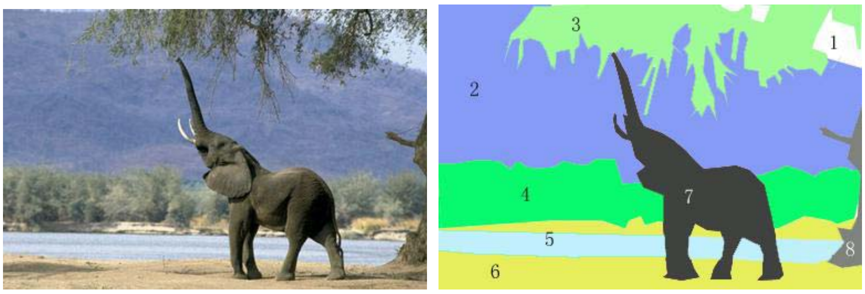

Fig. 4. Original image and corresponding segment map. To remove the elephant, our algorithm only searches in the neighborhood regions 2, 4, 5 and 6

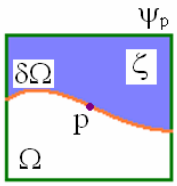

Fig. 5. Structure of a partially synthesized patch. ψ_{p} is the patch under processing centered at point p. ζ is the already synthesized region and Ω is the unsynthesized region

2.4 Coherence Confidence

Our definition of the coherence confidence is based on the nearest pixels whose positions on the source region (where it comes from) are known, using neighbor synthesized pixels and make them vote. This is quite similar to that of k-nearest neighbor classifier. As indicated in Fig. 5, suppose the current patch is centered at p and it overlaps with already filled region. We find k candidate patches $\psi_{p_1}, \psi_{p_2}, \ldots, \psi_{p_k}$. Let point qi be a point in ζ and it comes from point $\mathbf{c}_{q_i}$, pi be a point in ψ_{p}-ζ and it's corresponding point in current patch ψ_{p_i} at $\mathbf{c}_{p_i}$. We define the mean distance MD_ζ of region ζ as follows:

$$MD_\zeta = \sum_{i=0}^{N} \| \mathbf{q}_i, \mathbf{c}_{q_i} \| / N \quad , \tag{5}$$

where N is the number of points in ζ,and $\| * \|$ is the Euclidian distance.

The mean distance of region $\psi_{p}\text{-}\zeta$ can be defined as:

$$MD_{\psi_p\text{-}\zeta} = \sum_{i=0}^{M} \| \mathbf{p}_i, \mathbf{c}_{p_i} \| / M \quad (6)$$

where M is the number of points in $\psi_{p}\text{-}\zeta$.

The *coherence confidence* $CC_{\psi_{p_i}}$ of patch ψ_{pi} is defined as:

$$CC_{\psi_{p_i}} = \frac{1}{| MD_{\zeta} - MD_{\psi_p\text{-}\zeta} |} \quad (7)$$

From the candidate pool, we select the patch $\psi_{\hat{q}}$ with the highest *coherence confidence*.

$$\psi_{\hat{q}} = \max_{\psi_q} \arg CC_{\psi_q} \quad (8)$$

Using the above criteria, we can determine the best match in the candidate pool.

2.5 Graphcut Optimization

Direct paste of selected candidate patch to the current patch may still lead to great visual discontinuity. We implement a "Graphcut" optimization to find the best cut in the overlap region, such that the newly added patch can seamlessly join the already synthesized regions. Details of "Graphcut" can be found at [12]. The algorithm is briefly described as follows:

1. Construct a flow graph as Fig. 6

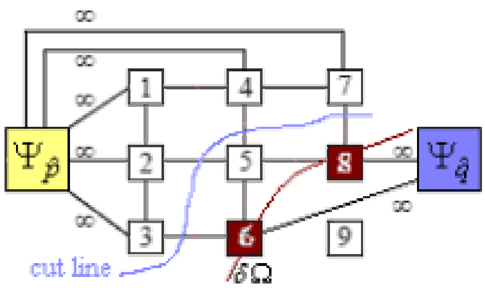

Fig. 6. Flow graph

The patch is deemed as a start knot in the graph and the candidate patch as the end knot. The flows of points $p \in \delta\Psi_{\hat{p}} \cap \Omega$ on the boundary of the current patch are set to be infinite to the start knot; the already synthesized pixels $p \in \partial\Psi_{\hat{p}} \cap \Phi$ are connected to the end knot and the corresponding flows are set to be infinite. The points in the current patch $\Psi_{\hat{p}} \cap \Phi$ are then connected to its corresponding patch, and the flows are calculated as follows:

$$M(s,t,\Psi_{\hat{p}},\Psi_{\hat{q}}) = \left\|\Psi_{\hat{p}}(s) - \Psi_{\hat{q}}(s)\right\| + \left\|\Psi_{\hat{p}}(t) - \Psi_{\hat{q}}(t)\right\| , \tag{9}$$

where $\Psi_{\hat{p}}, \Psi_{\hat{q}}$ are the current patch to fill and the candidate patch respectively, and s, t are corresponding points. $\left\|\Psi_{\hat{p}}(s) - \Psi_{\hat{q}}(s)\right\|$ is the L-2 distance.

$$\left\|\Psi_{\hat{p}}(s) - \Psi_{\hat{q}}(s)\right\| = \sum \sqrt{(R(s_{\hat{p}}) - R(s_{\hat{q}}))^2 + (G(s_{\hat{p}}) - G(s_{\hat{q}}))^2 + (B(s_{\hat{p}}) - B(s_{\hat{q}}))^2} . \tag{10}$$

2. Perform the min-cut on the flow graph

3. The points in the starting knot (1, 2, 3, 4, 7 in Fig. 6) keep their original value in $\Psi_{\hat{p}}$; the points in the end knot (5, 6, 8 in Fig. 6) are selected from the candidate patch $\Psi_{\hat{p}}$. Other points are also selected from $\Psi_{\hat{p}}$

3 Experiment Results

We have implemented the proposed algorithm on a home PC with Duron 1.2GHz CPU, 512M RAM using Visual C++. We provide two methods for the user to select the region for editing. One is an eraser tool with which users can replace the details in the specified region with uniform color. The other is a contour tool with which users can outline the contour of the region to fill.

Fig. 7 – Fig. 13 are inpainting results by applying our synthesis-based method. The areas to be repainted are all very large, occupying more than 10% of the entire image. The yellow contours are specified by the user to indicate the target regions. Experiments show that our method can robustly remove very large objects under complex background. We also show the comparison of our method with Criminisi's method in Fig. 9. It can be seen that our algorithm produces much better results, and attains much faster speed.

4 Conclusion and Future Work

We have proposed a novel synthesis-based inpainting algorithm, by taking into consideration of constraints during synthesis. Our method can effectively and efficiently handle large complex regions. Compared to previous methods, our algorithm has the following three characteristics: 1) A constraint-based candidate patch selection method is suggested which limits the searching within neighboring regions of similar texture; 2) the desired candidate is selected based on coherence confidence preventing the error accumulation and propagation; 3) The selected patch is integrated into the original image with Graphcut optimization to make the seam visually invisible.

Experiments show that our approach provides a powerful photo editing tool for undesired object removal. In the future, we will try to find automatic method that can infer the structure information in the unknown region to guide the texture fill-in. Another possible direction is to find a better texture difference metric that can capture both structure and intensity difference, for example, Garbor based feature representation. In addition, we would extend our framework to video inpainting.

Acknowledgements

This work is partially supported by Natural Science Foundation of China under grant 60033010 and National 973 program (No. 2002CB312101).

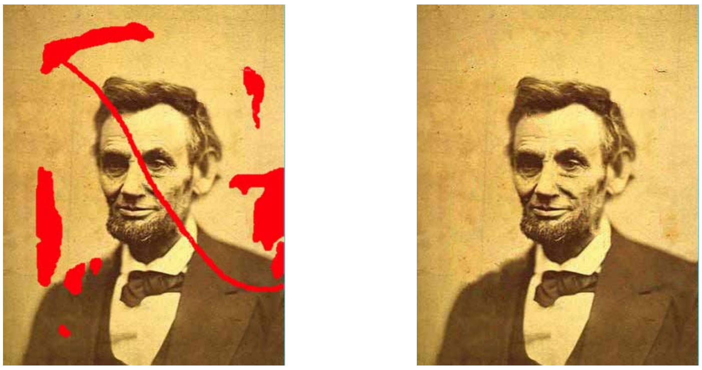

Fig. 7. Removal of scratches on "Lincon's portrait". Left is the spoiled image and right is our inpainting result

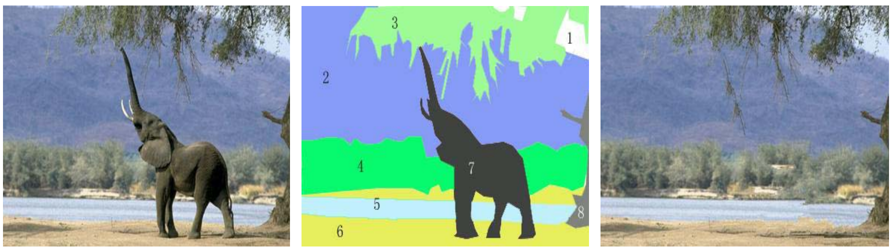

Fig. 8. Removal of the elephant from the photo. Left is the original image, middle is the segment map and right is our result

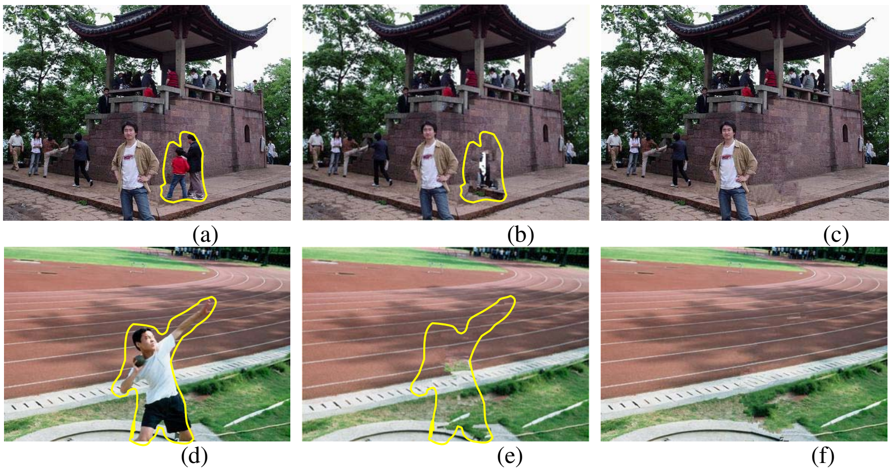

Fig. 9. Removal of large objects in photos. (a) and (d) are the original image; (b) and (e) are Criminisi's results. As can be seen, errors propagation is serious and the results are not satisfactory. (c) and (f) are our results. The contours in yellow are specified by the user

Fig. 10. Removal of a large object in photo "Park"

Fig. 11. Removal of a human in the photo "Girl"

Fig. 12. Removal of the islands from the scene

Fig. 13. Removal of ocean wave

References

1. Bertalmio, M, Sapiro, G., Caselles, V., Ballester, C. Image Inpainting. SIGGRAPH 2000, pages 417-424.
2. Bertalmio M. Bertozzi A. L Sapiro. G. Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting Computer Vision and Pattern Recognition (CVPR'01) - Volume 1 December 08 - 14, 2001 Kauai, Hawaii
3. Chan, T., Shen, J. Mathematical Models for Local Deterministic Inpaintings. UCLA CAM TR 00-11, March 2000.
4. Chan, T., Shen, J. Non-Texture Inpainting by Curvature-Driven Diffusions (CCD). UCLA CAM TR 00-35, Sept. 2000.
5. Perona, P. Malik, J. Scale-space and edge detection using anisotropic diffusion. IEEE-PAMI 12, pp. 629-639, 1990.
6. Oliveira M., Bowen B., McKenna R., and Chang Y-S., Fast Digital Image Inpainting, in Proceedings of the Visualization, Imaging, and Image Processing IASTED Conference, Marbella, Spain, 261-266, Sept. 2001.
7. Liang, L., Liu, C., Xu, Y., Gguo, B., and Shum, H.-Y. 2001. Real-time texture synthesis using patch-based sampling. ACM Trans. Graphics 20, 3, 127–150.
8. Efros, A., Freeman, W. 2001. Image quilting for texture synthesis and transfer. In SIGGRAPH'01, 341–346.
9. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk and Aaron Bobick Graphcut Textures: Image and Video Synthesis Using Graph Proc. ACM Transactions on Graphics, SIGGRAPH 2003

10. Criminisi A., Perez P. and Toyama K. Object Removal by Exemplar-Based Inpainting. CVPR, Madison, Wisconsin, June, 2003.
11. Comaniciu D., Meer P.: Mean Shift: A Robust Approach toward Feature Space Analysis, IEEE Trans. Pattern Analysis Machine Intell., Vol. 24, No. 5, 603-619, 2002
12. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk and Aaron Bobick Graphcut Textures: Image and Video Synthesis Using Graphcut Proc. ACM Transactions on Graphics, SIGGRAPH 2003.