A structure-synthesis image inpainting algorithm based on morphological erosion operation

Hao Guo[1,2], Nobutaka Ono[1], Shigeki Sagayama[1]
1 Sagayama&Ono lab, Department Physics and Computing Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan, 113-8656
2 School of Computer Science and Technology, Dalian Maritime University, Dalian, China, 116026
E-MAIL: guo@hil.t.u-tokyo.ac.jp, guohao0512@hotmail.com

Abstract

Image inpainting provides a means to restore damaged region of an image, such that the image looks complete and natural after the inpainting process. This paper presents a new inpainting algorithm, which implements the filling of damaged region based on morphological erosion and structure feature replication. The method, which is based on a structure/texture feature matching algorithm, can retain isophote continuity by propagating both texture and structure characteristics information of the known patch into the damaged region, and output a complete, natural-looking and non-blurred image. By implementing region-filling through morphological erosion, several pixels instead of one can be restored at every inpainting step, making the method faster than many traditional texture synthesis inpainting algorithms. We demonstrate through several examples the effectiveness of the algorithm in removing large occluding objects as well as thin scratches.

Keywords: *Image restoration, image inpainting, mathematical morphology algorithms, texture synthesis*

1. Introduction

Image inpainting provides a means to restore damaged regions of an image, such that the image looks complete and natural after the inpainting process. Image inpainting has a wide range of applications, such as the reconstruction of scans of deteriorated images by removing scratches or stains, the removal of text and logos from digital images or the creation of artistic effects.

Since Bertalmio et al. developed the BSCB inpainting algorithm [1,2,3], image inpainting has received considerable attention and many inpainting methods have been developed. These algorithms can be grouped in two main classes: (1) “interpolation methods” to fill in small or long and thin gaps, and (2) “texture synthesis methods” to fill in large and thick damaged regions.

Several classical inpainting methods belong to the first group. In the BSCB model, the image smoothness information, estimated by the image Laplacian, is propagated along the isophotes (lines of equal gray value) directions. The Total Variational (TV) model [4] and Curvature-Driven Diffusion (CDD) model [5] use an Euler-Lagrange equation coupled with anisotropic diffusion to preserve the directions of the isophotes. All the above methods essentially solve a Partial Differential Equation (PDE) that describes the color propagation inside the missing region, subject to various heuristics that attempt to preserve the directions of the isophotes. Leave out of account of the complexity of PDE–based algorithms [1,2,3,4,5], their main problem is that both isophote estimation and information propagation are subject to numerical diffusion. Diffusion is desirable as it stabilizes the PDE to be solved, but inevitably leads to blurring of the restoration area. The fast marching method (FMM) model described in [6] propagates an image smoothness estimator along the image gradient, similar to the BSCB model [1,2]. However, as the smoothness estimator is based on a weighted average over a known image neighborhood of the pixel to restore, the method also inevitably leads to blurring. Blurring is unacceptable for the restoration of large and thick damaged regions.

In previous work, several researchers have considered texture synthesis inpainting [7,8,9,10] as a way to fill in large and thick damaged regions. Texture synthesis is suitable to fill in large image regions with "pure" textures – repetitive two-dimensional textural patterns with moderate stochastical variation. Different regions of the texture are always perceived to be similar (high correlation), and each pixel is only related to a small set of neighboring pixels. The damaged domain in many real and synthetic images consists of linear structures and composite textures (multiple textures interacting spatially in many real and synthetic images), and some traditional texture synthesis, such as exemplar-based techniques, which cheaply and effectively generate new texture by sampling and copying color values from the source, have difficulty filling in such domains and say nothing of composite textures. Some methods [11,12] combining interpolation and texture synthesis have been proposed lately, but there speed is affected due to the fact that interpolation is very time-consuming.

Image inpainting mostly encounters difficulties for the restoration of large damaged regions, for which one can stress out two main issues: (1) how to preserve isophote continuity while propagating both texture and structure characteristics information of the known patch into the damaged region, and (2) how to implement the algorithm in a fast and efficient way. In this paper, we propose a new inpainting model based on morphological erosion and structure feature matching. The method can preserve isophote continuity by propagating both texture and structure characteristics information of a known patch into the damaged region, and output a complete, natural-looking and non-blurred image. As several pixels instead of just one can be restored at every inpainting step, the method is faster than many traditional texture synthesis inpainting algorithms. Several examples on real and synthetic images demonstrate the effectiveness of the algorithm in removing large occluding objects as well as thin scratches.

The organization of the paper goes as follows. In section 2, we explain the structure-synthesis image inpainting method based on morphological erosion in details. In section 3, we demonstrate through several examples the effectiveness of our algorithm. We then conclude the paper and list some future work.

2. Image inpainting based on morphological erosion

This section describes the image inpainting method. First, morphological erosion is briefly introduced. Then, the region-filling process based on morphological erosion and the structure/texture features matching algorithm performed after every step of erosion are described. Finally, we give some details on the implementation of the algorithm.

2.1. Morphological erosion

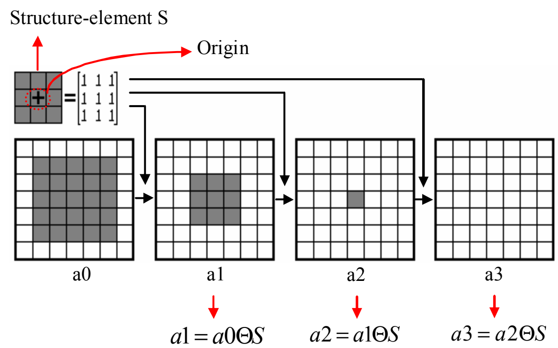


Figure 1. Example of erosion for a binary image

Morphology is a broad set of image processing operations that process images based on shapes. Dilation and erosion are two fundamental morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels from them. Erosion can be used for image inpainting if the damaged region is regarded as a region to be eroded.

Morphological erosion can be defined as follows. If A is an input binary image and B a structuring element, the erosion of A by B, denoted by $A\Theta B$, is given by

$$A\Theta B = \left\{ x \in A \middle| (B)_x \subseteq A \right\}, \qquad [1]$$

where $(B)_x$ designates the structuring element B centered at pixel x. An essential part of the dilation and erosion operations is the structuring element used to process the input image. A structuring element is a matrix consisting of only 0's and 1's that can have any arbitrary structure and size. The structuring elements are typically much smaller than the image being processed. The center pixel of the structuring element, called the origin, identifies the pixel of interest -- the pixel being processed. The pixels in the structuring element containing 1's define the neighborhood of the structuring element. The erosion process is illustrated in Figure 1. In this figure, only one layer of pixels is eroded at every loops of erosion operation.

2.2 Region filling algorithm

The erosion process of a binary image (Figure 1) can be considered as the inpainting process of the binary image. The main advantage of this process is that it explicitly maintains the narrow band that separates the known from the unknown image area, and specifies the next pixel to be inpainted. For gray or color images, if the gray or color information of the unknown pixels can be restored based on information from the neighborhood, inpainting of a damaged area can also be performed using the erosion process. The region-filling process is illustrated in Figure 2.

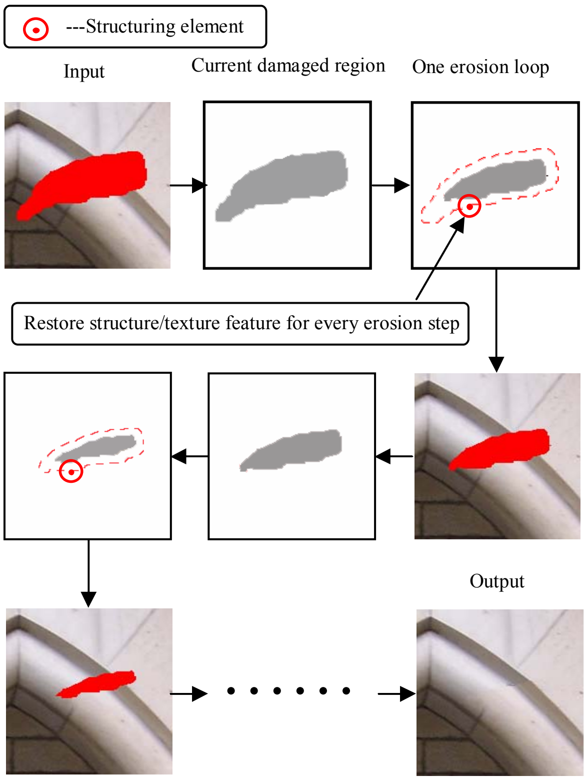


Figure 2. Region-filling process

In brief, to restore the damaged region Ω, a structuring element is iteratively applied to erode and restore all the discrete pixels of the current boundary $\partial\Omega^t$ ($\partial\Omega^t$ denotes the boundary at time t, $t = 0,1,\cdots,T,\cdots$, $\partial\Omega^0$ is the initial boundary), and the boundary is advanced inside Ω until the whole region has been restored.

We now describe one loop of the algorithm. At time t, we extract the boundary $\partial\Omega^t$ between the damaged region and the known image area, and mark the pixels with flags: EDGE, KNOWN, UNKNOWN (see Figure 3), where

• KNOWN designates the pixels outside $\partial\Omega^t$, in the known image area.

• UNKNOWN designates the pixels inside $\partial\Omega^t$, in the region to be restored.

• EDGE designates the pixels between the KNOWN and UNKNOWN which also belong to the damage area Ω.

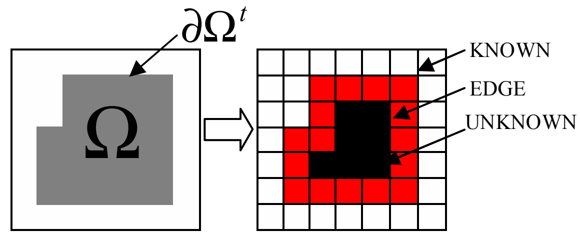


Figure 3. Mark the damaged image with flags

A structuring element is iteratively applied to erode all the discrete pixels of the current boundary. At each step during this erosion, the pixels covered by structuring element are restored with through structure/texture features matching, as described in the following subsection. All restored pixels are flagged as KNOW. The procedure is repeated until there is no more pixels flagged as EDGE. Finally, the flags are updated.

After one erosion loop, one layer of pixels has been restored. The procedure is repeated until there are no more pixels flagged as UNKOWN.

2.3 Restoration of structure/texture features at every erosion step

At each erosion step of each region-filling loop, the next pixel or next patch to be restored is explicitly specified as the scope covered by the structuring element. The inpainting process of gray or color images is not completed until the pixels in the scope are filled and the structure/texture features in the scope are restored.

It is well-known that almost all PDE-based inpainting methods are unable to avoid blurring, and are only effective for long thin damaged regions, for which blurring is not perceived easily, while for large and thick damaged regions, blurring is unacceptable. A complete, natural-looking image can be obtained only if the structure/texture features are restored without blurring. The questions to answer are now (1) what kind of structure/texture features of a gray or color image are to be considered, and (2) where could the structure or texture information be reliably chosen for the damaged regions to be restored.

As for the first question above, a definition can be drawn from Figure 4. Texture features designates gray or color values of different regions which are perceived to be similar or alike, and where each pixel is only related to a small set of neighboring pixels. Structure features are the contour lines of different texture areas, and each pixel has a strong relation to the pixels along the contour line. As for the second question above, some insights may be obtained from manual inpainting techniques. In manual inpainting, the damaged areas closest to known areas are restored first, which means that only the information in a narrow band along the boundary of the known areas is reliable and has a strong correlation with the information to be restored in the damaged areas. After one loop of inpainting along the boundary, a new narrow band along the updated boundary between the damaged and known areas is formed. In this way all the damaged areas are restored.

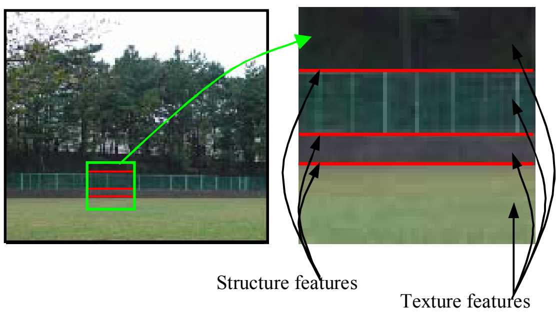


Figure 4. Structure features and texture areas

A structure/texture features matching algorithm is introduced in this section. This algorithm, illustrated in Figure 5, can restore some structure features (some ordinary structures such as linear contours or curves with small curvature) both for large and thick or long and thin damaged regions without blurring. It includes two steps as follows:

First, inside a search scope, the gray or color value of the pixels in the structure features patch of each unknown pixel is compared with that of known pixels left to right and top to bottom, to find the one with maximal similarity. Here, the similarity of two patches is measured using the Sum of Squared Distance (SSD) of all known pixels within them.

Then, when the structure features patch around a known pixel with maximal similarity (minimal SSD) is found in the search scope, the value of the known pixels in the scope of the structuring element are put into the unknown pixels covered by the structuring element, thus completing one step of inpainting. In this way, the structure/texture features of the known areas are propagated into the damaged areas, and at the same time the narrow band can be preserved.

2.4 Some details of the implementation

2.4.1. How to avoid error accumulation

In our method, after one loop of inpainting along the boundary, a new narrow band along the boundary between the damaged area and known area is formed, which information is based on the old narrow band. Thus doing, we can see from the outcomes of some inpainting experiments on real and synthetic images that an error accumulation phenomenon sometimes occurs, as shown in Figure 6. If for some reason the information in the old narrow band is wrong, the error might propagate to the new narrow band step by step.

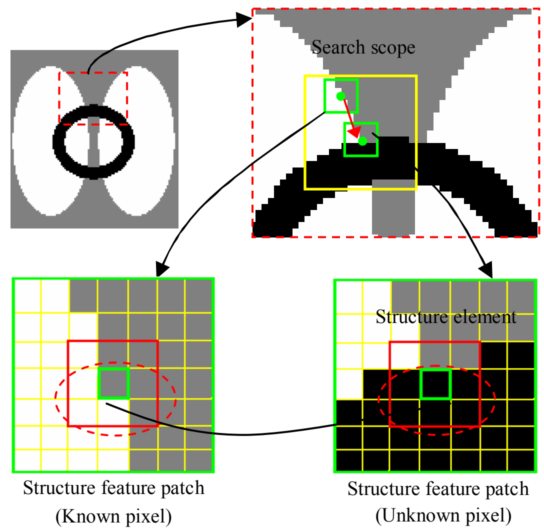


Figure 5. Structure/texture features matching

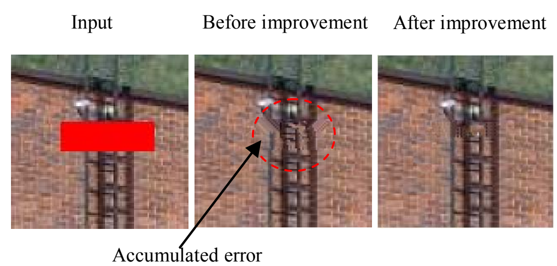


Figure 6: Example of error accumulation

In order to solve this issue, the restoration of structure/texture features for every erosion step is improved: the center of the search scope is adjusted to the original boundary all along, and only the part of the search scope inside the original image is used to reflect the fact that only the information from the original image is reliable. This process is illustrated in Figure 7, and includes the two following steps: first, a center pixel for the search scope is searched on the original boundary closest to the unknown pixel; then, the search scope is defined based on the center pixel, excluding the parts of the original damaged area between the original boundary and the advanced boundary, which have already been restored. Error accumulation can in this way be avoided to some extent (Figure 7).

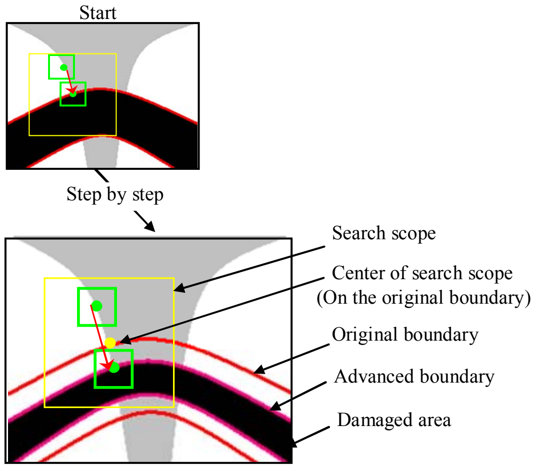


Figure 7: Solving error accumulation

2.4.2. How to speed up the inpainting process

From the erosion process for a binary image illustrated in Figure 1, we can see that only one layer of pixels is eroded at every erosion loop. There are at least two problems here: ignoring the correlation between neighboring pixels and restoring only one pixel at each step of the erosion loop is (1) under-optimal, with information not fully exploited, and (2) leads to a slow restoration speed.

If we follow the process of the restoration of structure/texture features for every erosion step as described in Figure 2, as several pixels instead of just one can be restored at every inpainting step, the method is faster than many traditional texture synthesis inpainting algorithms, and at the same time the narrow band, representing the information reliability, is also preserved.

3 Experimental results

The algorithm is applied to restore a variety of images. All the image examples were taken by the writer himself, and the red areas in the images denote the damaged areas to be restored. Our C++ implementation took about 5 minutes on 1.2 GHz PC for a 800×600 color image with about 18% pixels to be restored. The results in Figure 8 show that the inpainting method is able to restore texture and structure characteristics information for large and thick damaged regions without blurring (here the thickness of the inpainting regions is up to 40 pixels). The example in Figure 9 shows that the method can be used with a wide variety of shapes for the damaged areas. An example of artistic effect is shown in Figure 10. Some comparisons have also have been done such as

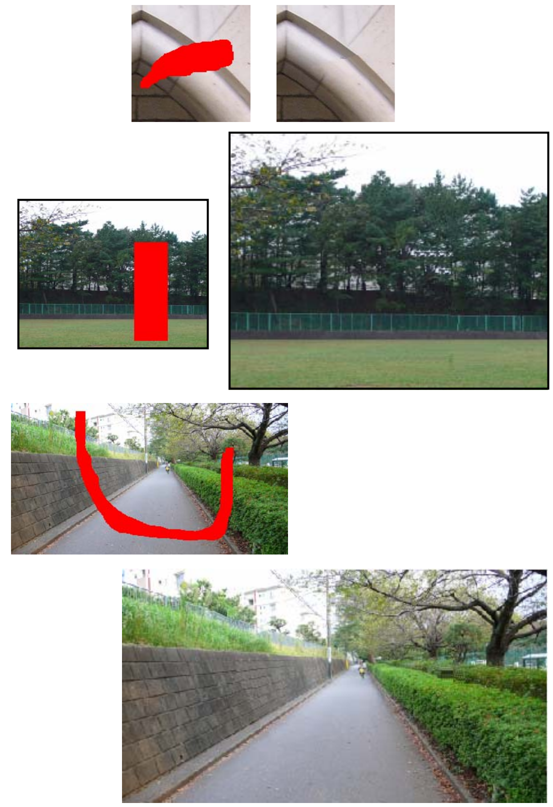


Figure 8. Some inpainting experimental results

Figure 9. Restoration of different damage areas

the one shown in Figure 11, which shows the validity of our method compared to previous work.

From these experimental results, we can see that the method can preserve isophote continuity by propagating both texture and structure characteristics information of the known patch into the damaged region, and output complete, natural-looking and non-blurred images.

4 Conclusion and future work

Inpainting provides a means to restore damaged region of an image. Most inpainting algorithms published in the literature are complex to understand and implement. We presented a new inpainting algorithm, which implements the filling of damaged region based on morphological erosion and the matching of structure features. In brief, the method has the following characteristics:

1) Based on a structure/texture features matching algorithm, this image inpainting method can restore some structure features (some ordinary structures such as linear contours or curves with small curvature) both for large and thick or long and thin damaged regions without blurring
2) Thanks to the implementation of region-filling based on morphological erosion, several pixels instead of just one can be restored at every inpainting step, and the method is thus faster than many traditional texture synthesis inpainting algorithms.

Overall, the presented inpainting method is simple to implement. The method also has some limitations for which more research needs to be done. We plan to work for example on methods to restore complex structure information, such as corners, curves with large curvature, etc.

Where is the clock of the YASUDA hall?

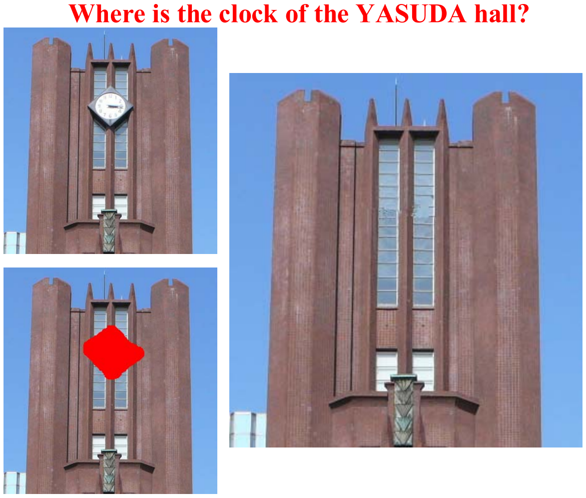
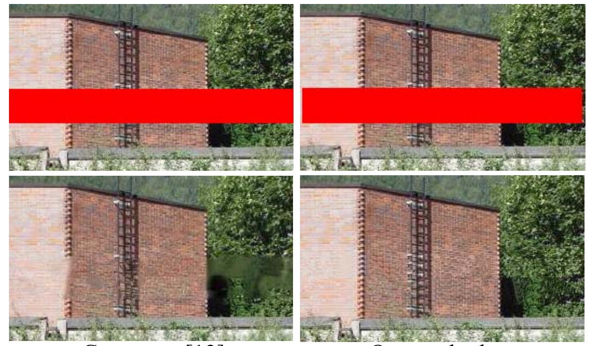


Figure 10: Example of artistic effect

Reference

[1] Bertalmio M, Sapiro G, Caselles V, Ballester C, “Image inpainting”, Proc. of the ACM SIGGRAPH 2000, New Orleans, 2000, pp.417-424.

[2] Bertalmio M, Bertozzi AL, Sapiro G. Navier-Stokes, “fluid dynamics, and image and video inpainting”, Proc. of the IEEE Int’l Conf. on Computer Vision and Pattern Recognition, Volume I. Hawaii, 2001, pp.355-362.



Grossauer [12] Our method

Figure 11: Comparison of different methods

[3] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, “Filling–In by Joint Interpolation of Vector Fields and Gray Levels”, IEEE Transactions on Signal Processing, 10 (8), 2001, pp.1200–1211.

[4] T. Chan, J. Shen, “Mathematical Models for Local Deterministic Inpaintings”, Technical Report CAM 00-11, Image Processing Research Group, UCLA, 2000.

[5] T. Chan, J. Shen, “Non-Texture Inpainting by Curvature Driven Diffusions (CDD)”, Technical Report CAM 00-35, Image Processing Research Group, UCLA, 2000.

[6] Alexandru Telea, “An Image Inpainting Technique Based on the Fast Marching Method”, Journal of Graphics Tools, 9 (1), 2004, pp.25-36.

[7] A. A. Efros, T. K. Leung, “Texture Synthesis by Nonparametric Sampling”, In IEEE International Conf. on Computer Vision, volume 2, 1999, 1033–1038.

[8] L.-Y.Wei, M. Levoy, “Fast Texture Synthesis Using Tree-Structured Vector Quantization”, Computer Graphics (SIGGRAPH 2000 Conf. Proc.), 2000, pp. 479–488.

[9] Peng Tang, “Application of Non-parametric Texture Synthesis to Image Inpainting”, M.S., Computer Science, University of New Mexico, 2004.

[10] Antonio Criminisi, Patrick Perez, Kentaro Toyama, “Object Removal by Exemplar-based Inpainting (PDF) Jun”, WI Proc. IEEE Computer Vision and Pattern Recognition, 2003 (2), pp. 721-728.

[11] Hitoshi Yamauchi, Jörg Haber, Hans-Peter Seidel, “Image Restoration using Multiresolution Texture Synthesis and Image Inpainting”, Proc. Computer Graphics International (CGI) 2003, pp. 120-125.

[12] H. Grossauer, “A Combined PDE and Texture Synthesis Approach to Inpainting”, European Conference on Computer Vision, LNCS 3022, 2004, pp. 214-224.