Development of Combined Structure and Texture Inpainting Method

Y. El Glaly*, B. El Desouky**, T. Nazmy*** and E. Atta***
*Research Assistant, Faculty of Education, Suez Canal University, Egypt
**Associate Prof., Faculty of Education, Suez Canal University, Egypt
***Prof., Faculty of Computer & Information Sciences, Ain Shams University, Egypt

Abstract

Image inpainting technique has been widely used for reconstructing damaged old photographs and removing unwanted objects from images. In this paper, we present an improved inpainting method based on the exemplar-based image inpainting technique. The developed method enhances the robustness and effectiveness by including image gradient information during the inpainting process. Presented results show that the developed algorithm can reproduce texture and at the same time keep the structure of the surrounding area of the inpainted region. The method proved to be effective in removing large objects from an image, ensuring accurate propagation of linear structures, and eliminating the drawback of “garbage growing” which is a common problem in other methods.

1. Introduction

Image inpainting is an iterative method for repairing damaged pictures or removing unnecessary elements from pictures. This activity consists of filling in the missing areas or modifying the damaged images in a non-detectable way by an observer not familiar with the original images [1]. Applications of image inpainting range from restoration of photographs, films and paintings, to removal of occlusions, such as large unwanted regions, superimposed text, subtitles, stamps and publicity, from images. In addition, it is of significant importance in restoration of precious work of arts, calligraphies, and paintings in the digital museum with image inpainting technique [13].

A number of algorithms address this image filling issue [2], [3], [7], [8], [12]; these image inpainting techniques fill holes in images by propagating linear structures (called isophotes in the inpainting literature) into the target region via diffusion. They are inspired by the partial differential equations (PDE) of physical heat flow, and work convincingly as restoration algorithms. The user uses mask to specify the region of the input image to be inpainted. The algorithm treats the input image as three separate channels (R, G and B). For each channel, the region is filled by propagating information from the outside of the masked region along level contours (isophotes). Isophote direction is obtained by computing the discretized gradient vector of each pixel along the contour (the gradient indicates the direction of largest color change) and by rotating the resulting vector by 90 degrees. This intends to propagate information while preserving structures. A 2-D Laplacian operator is used to locally estimate the variation in smoothness. Through propagating such variation, the isophote direction is obtained. After every step of the inpainting process, a few diffusion iterations are taken to smooth the inpainted region. Anisotropic diffusion is used in order to preserve edges across the inpainted region [4].

The PDE-based algorithm does not perform well for texture dominated pictures. For such cases the exemplar based algorithm is used instead [5], [10], [11]. With this algorithm, the gap will be filled with non-blur textures, while at the same time preserve and extend the linear structure of the surrounding area. The algorithm uses a best-first algorithm in which the confidence in the synthesized pixel values is propagated in a manner similar to the propagation of information in PDE inpainting algorithm. This algorithm, however, has some problems: firstly, it merely adopts a simple priority computing strategy without considering the accumulative matching errors; secondly, the matching algorithm for texture synthesis only uses the color information; thirdly, the filling scheme just depends on the priority disregarding the similarity. As a result of lacking robustness, their algorithm sometimes runs into difficulties and “grows garbage”. To solve these problems, we propose an enhanced exemplar-based inpainting algorithm.

2. Improved Exemplar-based Algorithm

The exemplar-based inpainting algorithm consists mainly of three iterative steps, until all pixels in the inpainted region are filled. The region to be filled, i.e., the target region is indicated by Ω, and its contour is denoted $\partial\Omega$. The contour evolves inward as the algorithm progresses, and so we also refer to it as the “fill front.” The source region which remains fixed throughout the algorithm, provides samples used in the filling process (figure 1). In order to find the most similar patch in the source region to the target patch, we search the whole source image to find the best fit. The similarity is measured by computing the sum of squared distance in color between each corresponding pixel in the two patches.

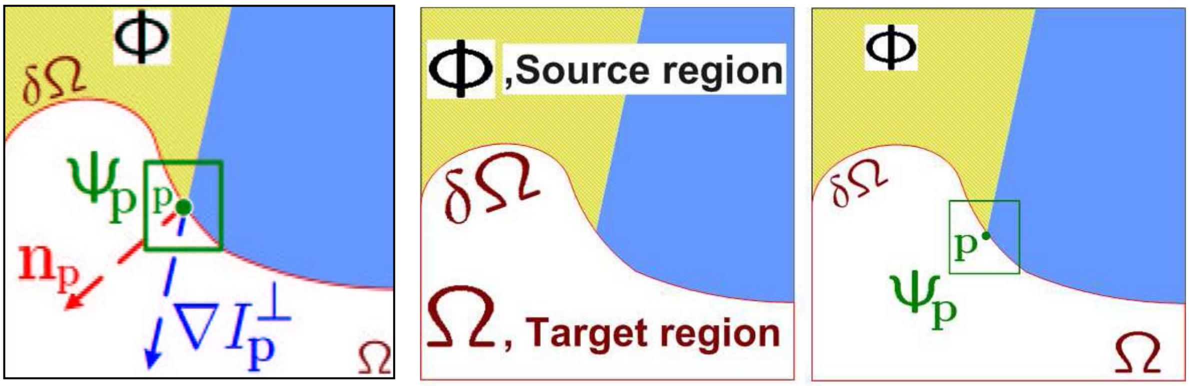


Fig.1 Schematic of exemplar-based inpainting [5]

The similarity measure based only on color is insufficient to propagate accurate linear structures into the target region, and leads to garbage growing. So, we add to this distance function a new term G representing image gradient as an additional similarity metric.

$$G = G(\Psi_p) - G(\Psi_q)$$

Where G is the gradient value for each pixel in the two considering patches. Hence, the similarity function now depends on the difference between the patches according to two criteria, the difference in color and in gradient values.

The details of the algorithm implementation is as follows,

1. Computing patch priorities

Given a patch Ψ_p centered at the point p for some $p \in \partial\Omega$, its priority P(p) is defined as the product of two terms:

P (p) = C (p) D (p)

C (p) is the confidence term and D (p) is the data term, and they are defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \bar{\Omega}} C(q)}{|\Psi_p|} \quad \text{and} \quad , \quad D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{\alpha}$$

Where $|\Psi_p|$ is the area of Ψ_p, α is a normalization factor (e.g., $\alpha = 255$ for a typical grey-level image), and n_p is a unit vector orthogonal to the front $\partial\Omega$ in the point p. The priority is computed for every border patch, with distinct patches for each pixel on the boundary of the target region. The patch with the highest priority is the target to fill.

2. Propagating structure and texture information

Search the source region to find the patch which is most similar to $\Psi_{p^\wedge}$. Formally,

$$\Psi_{q^\wedge} = \arg\min_{\Psi_q \in \Omega} d(\Psi_{p^\wedge}, \Psi_q)$$

The distance $d(\Psi_a, \Psi_b)$ between two generic patches Ψ_a and Ψ_b is simply defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.

$$d = \sum_{i=1}^{i=A} (I_{ai} - I_{bi})^2 + (G_{ai} - G_{bi})^2$$

Where, G presents the image gradient vector, I is the RGB color vector, D is the distance (the larger d is, the less similar they are), and A is the known pixels number in $\Psi_{p^\wedge}$.

Having found the source exemplar $\Psi_{q^\wedge}$, the value of each pixel to be filled, is copied from its corresponding position.

3. Updating confidence values

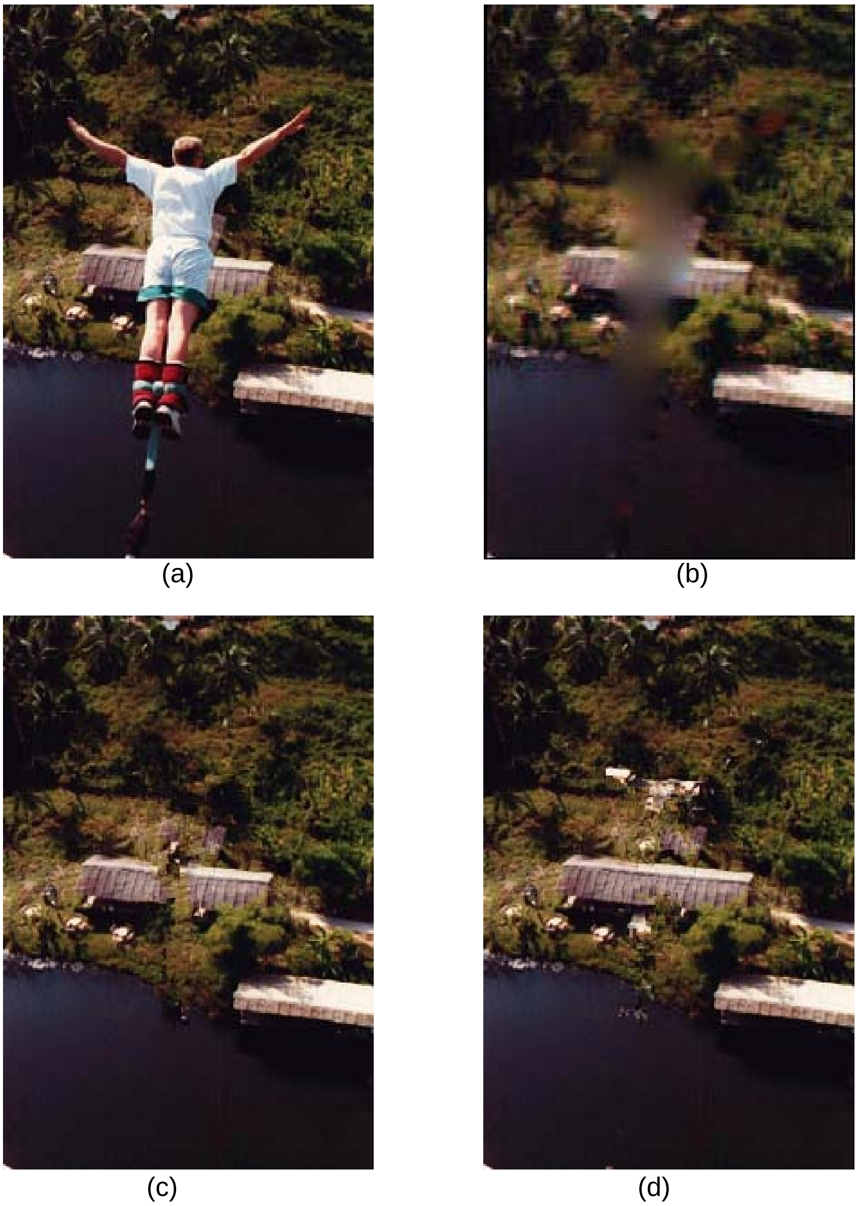
The confidence C (p) is updated in the area delimited by $\Psi_{p^\wedge}$, as follows:

$$c(q) = c(p^\wedge)\ \forall q \in \Psi_{p^\wedge} \cap \Omega$$

As filling proceeds, confidence values decay, indicating that we are less sure of the color values of pixels near the center of the target region.

3. Results

The developed inpainting algorithm is tested on a Pentium IV class PC (256 Mb RAM, 2 GHz). Test images include natural scenes and fullcolor photographs of complex textures. Figure 2 compares the present algorithm with that of reference [1] and [5]. The result of the algorithm used in reference [1] shows a marked image blurring and a loss of details. While that of reference [5] exhibits a break in the linear structure of (the white building) in the image. Better image texture and structure are evident in the result of including the gradient information in the present algorithm.



(a) (b) (c) (d)

Fig.2. (a) Original image ,(b) result of the algorithm of reference [1], (c) result of reference [5], and (d) result of present algorithm

Figure 3.a displays an image for a sail boat in the Nile, the mask used to remove the sail boat from the image, and the inpainted image. It has to be mentioned that creating a suitable mask is essential for obtaining good results, a mask with fairly irregular contours and kinky edges can have an adverse effect on the inpainting process. Another example is displayed in figure 4, that shows the effectiveness of the present algorithm in removing large regions in an image, and still preserves the integrity of the image.

(a) (b) (c)

Fig.3. (a) Original image of a sail boat, (b) the mask used in the inpainting process, (c) the final image after removing the sail boat

(a) (b)

Fig.4. (a) Original image of a natural scene, and (b) the final inpainted image after removing the horses

Figure 5.a shows a view for the sphinx with the pyramid in the background, and algorithm is implemented to remove the sphinx from the image. fig. 5.b depicts the mask used to remove the sphinx from the image. The inpainting process for this image is quite difficult, since the textures present in the image are very close to each other (the pyramid and the sphinx), and even though the gradient information is used, the inpainting algorithm may allow “garbage growing” as evident in figure 5.c. To achieve satisfactory result for this difficult case, the inpainting process is repeated twice to obtain the results shown in figure 5.d. Another image of a different view of the sphinx-pyramid combination is inpainted and displayed in figure 6.a. The same image after removing the sphinx is displayed in figure 6.b and showing an occluded part of the pyramid.

(a)

(b)

(c)

(d)

Fig.5. (a) Original image of the sphinx and pyramid, (b) the mask used in the inpainting process, (c) the image after the first inpainting pass, and (d) the final inpainted image after removing the sphinx.

Fig.6. (a) Original image, and (b) the final inpainted image after removing the sphinx.

The presented results, clearly demonstrate that the algorithm introduced in this paper succeeds effectively in inpainting large regions from images that consists of textures and surrounded by distinctive image structure. The execution time required for the inpainting process depends on the size of the image and the regions to be inpainted, and it ranges from few seconds to several minutes for large images.

4. Conclusions

In this paper, we have presented an image inpainting algorithm based on modifying the exemplar-based image inpainting method. The developed method enhances the robustness and effectiveness by including image gradient information during the inpainting process. Several test images have been used and the results demonstrate that the developed algorithm can reproduce texture and at the same time keep the structure of the surrounding area of the inpainted region. The method proved to be effective in removing large objects from an image, ensuring accurate propagation of linear structures, and eliminating the drawback of “garbage growing” which is a common problem in other methods. The results obtained are preferable to those obtained by other similar methods. The examples presented demonstrate the effectiveness of the modified algorithm

References

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting", Proceedings of SIGGRAPH 2000, New Orleans, USA, pp 417-424, July 2000.

[2] M. Bertalmio, A.L. Bertozzi, and G. Sapiro, "Navier stokes, fluid dynamics, and image and video inpainting", Proceedings of Conf. Comp.Vision Pattern Rec., Hawai, pp.355–362, Dec 2001.

[3] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting". Proceedings of IEEE conference on Computer Vision and Pattern recognition(CVPR’03), 2003.

[4] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion", IEEE-PAMI 12, pp. 629-639,1990.

[5] Criminisi A., Perez P., and Toyama K., "Region filling and object removal by exemplar-based image inpainting", IEEE Trans. Image Processing 9, pp1200-1212, 2004.

[6] Timothy K. Shih and Rong-Chi Chang, "Digital Inpainting – Survey and Multilayer Image Inpainting Algorithms", Proceedings of the Third International Conference on Information Technology and Applications (ICITA’05) IEEE, 2005.

[7] BianRu Li, Yue Qi, XuKun Shen, "An image inpainting method", Ninth International Conference on Computer Aided Design and Computer Graphics, IEEE, 2005.

[8] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors", Proc. IEEE Comp. Vision and Pattern Rec., 2:860-867, 2005.

[9] J. SHEN, "Inpainting and the Fundamental Problem of Image Processing", SIAM News 36(5), June 2003.

[10] A. CRIMINISI, P. PÉRES AND K. TOYAMA, "Object Removal by Exemplar-Based Inpainting", Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’03).

[11] Sung Ha Kang, "Mathematical Approaches to Color Denoising and Image Inpainting Problems", Ph.D. Thesis, University of California, Los Angeles 2002

[12] Manuel M. Oliveira Brian Bowen Richard McKenna Yu-Sung Chang, "Fast Digital Image Inpainting", Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, September 3-5, 2001

[13] Y. El Glaly, E.Atta, T. Nazmy, and B. Desouky, “Digital Inpainting for Cultural Heritage Preservation”, International Conference on Computer Theory and Application, ICCTA, 2006