Comparative Study of Convolution Based Inpainting Algorithms

Dhara Desai[1] Mehul Thakkar[2]
[2]Assistant Professor
[1,2]A. D. Patel Institute of Technology, New Vallabhvidhya Nagar, Gujarat, India

Abstract— Image Inpainting refers to restoring a damaged image in such a manner that manipulation and modification in an image is not recognizable by the observer. In this paper we have simulated and compared results of two different existing algorithms that are based on convolution as given by Oliveira and Hadhoud. The algorithm given by Hadhoud gives faster results but produces strokes in the resultant image while algorithm given by Oliveira is slow but fills the region from all the sides.
Key words: Image inpainting, Image restoration, Fast digital inpainting, Convolution based inpainting.

I. INTRODUCTION

Digital inpainting refers to a specific image restoration task, where missing or damaged portions or consciously masked portion of an image are reconstructed using information from the surrounding area in such a manner that an observer is not able to detect modifications in an image. Digital inpainting has found widespread use in any applications such as restoration of damaged old paintings and photographs, removal of undesired objects and writings on photographs, transmission error recovery in images and videos, computer-assisted multimedia editing and replacing large regions in an image or video for privacy protection. For video inpainting application the algorithm need to be fast enough. The goal of the inpainting technique is to modify the damaged region in an image or video in such a way that the inpainted region is undetectable to an observer [1].

There are three main challenges that are associated with existing and emerging digital inpainting algorithms. First, to improve the structure completion capability of image inpainting algorithms, second, to design a fast and efficient inpainting algorithm that can perform robustly under varying operating conditions, and third, to analyze the perceptual quality of large area image inpainting algorithms. In this paper we have compared two algorithms, one given by Oliveira [10] and the other given by Hadhoud [11] that are fast and convolution based techniques.

II. LITERATURE REVIEW

Image inpainting algorithms can be separated into different categories like texture synthesis based image inpainting, Exemplar and search based image inpainting, PDE (Partial Differential Equation) based inpainting, Semi-automatic inpainting, Fast or convolution based inpainting and hybrid inpainting.

A. Texture Based Inpainting

In texture based inpainting, holes are filled by sampling and copying neighbouring pixels or finding the best patch in the entire image, based on which a pixel value can be decided [2]. The main difference between different texture based algorithms is how they maintain continuity between hole's pixel and original image pixels [12]. Texture synthesis based inpainting methods not perform well for natural images. These methods do not handle edges and boundaries well.

B. PDE Based Inpainting

The first PDE base approach was given by Bertalmio et.al.[3]. that used the concept of isophotes and diffusion process. Main problem with this method is that due to blurring effect of diffusion process replication of large texture is not perform well. Inspired by the work of Bertalmio et al, Chan and Shen proposed two image-inpainting algorithms [4, 5]. One is TV (Total Variational) model [4] that uses anisotropic diffusion and Euler-Lagrange equation. This model was designed for inpainting small regions and while it does a good job in removing noise, it does not connect broken edges. The Curvature-Driven Diffusion (CDD) model [5] extended the TV algorithm to also take into account geometric information of isophotes. CDD can connect some broken edges, but the resulting interpolated segments usually look blurry.

C. Exemplar Based Inpainting

Exemplar based method is very effective and use priority based mechanism to determine order of region filling. This method works very well for a large number of images. It uses a good texture and structure replication. The Problem with this method is, curved structures are not handled properly. Criminisi et.al proposed an algorithm that uses Isophote (linear edges of surrounding area) driven Inpainting and texture synthesis [6].

D. Combined Structure and Texture Based Inpainting

In combined structural and texture based inpainting (also known as hybrid inpainting) PDE and texture synthesis based inpainting methods are combined for filling holes. Main goal here is to decompose an image into texture and structure region. Then corresponding regions are filled by texture synthesis and edge propagating algorithms respectively. This was developed by Bertalmio et.al [7]. The texture was filled by the algorithm defined in [2] and structure was filled by [8]. Here different textural and structural algorithms can be used as a combination.

E. Semi-Automatic

Semi-automatic image inpainting was given by Jian et.al [9] termed as inpainting with Structure propagation follows a two-step process. In the first step a user manually specifies important missing information in the hole by sketching object boundaries from the known to the unknown region and then a patch based texture synthesis is used to generate the texture.

F. Fast Digital Inpainting

Fast digital inpainting was proposed by M. Oliveira [10]. This technique is based on an isotropic diffusion model which performs inpainting by repeatedly convolving the inpainting region with a diffusion kernel. This method was further altered by M. M. Hadhoud in [11] which was fast enough as compared to that given by Oliveira.

III. FAST INPAINTING ALGORITHMS

Very first simple and fast inpainting algorithm was given by M. Oliveira [10] which worked well for images in which relatively small area was to be inpainted. The algorithm given by Oliveira goes as

- Initialize the region to be inpainted by clearing its colour information
- Repeatedly convolving the region to be inpainted with a diffusion kernel.
- The number of iterations is independently controlled for each inpainting domain by checking if none of the pixels belonging to the domain had their values changed by more than a certain threshold during the previous iteration. Alternatively, the user can specify the number of iterations.

Two different diffusion kernels given are as shown in Fig. 1.

The fact used is that the human visual system can tolerate some amount of blurring in areas not associated with high contrast edges. This algorithm reduces several minutes taken by [3, 4] to just few seconds. The results were generated with 100 diffusion iteration or more. Here the algorithm gives similar results as generated by Bertalmio but two to three orders of magnitude faster.

The above diffusion method was further modified by H. Hadhoud in the paper presented in [11]. This algorithm reduces 100 iterations to just one and making the algorithm much faster. This was done by just modifying the diffusion kernel or the filter to zero weight at the bottom right corner instead of the center and making the convolution from the bottom right corner. The kernel is given as shown in Fig. 2.

This modification forbidden the need to iterate the convolution operation because the inpainted pixel produced from the above left neighbourhoods pixels which are all known pixels achieves the goal of iteration from the first averaging iteration.

This modification results in a much faster algorithm, and the results are produced without blurring. This is also effective in removing large objects from symmetric background which the algorithm given by Oliveira was not able to do.

a	b	a
b	0	b
a	b	a

c	c	c
c	0	c
c	c	c

Fig. 1 Oliveira method two diffusion kernels used with the algorithm, a= 0.073235, b = 0.176765, c = 0.125

a	b	a
b	a	b
a	b	0

c	c	c
c	c	c
c	c	0

Fig. 2 Hadhoud method two diffusion kernels used with the algorithm, a= 0.073235, b = 0.176765, c = 0.125.

IV. RESULT AND DISCUSSION

To compare the quality of different inpainting algorithms, not much research has been done. Initially algorithms were compared based on their capability to handle big fill areas, how good algorithm in curved structures, texture replication capability, time taken and algorithm work for how many images etc. In this we have used PSNR (Peak Signal to Noise Ratio) for comparing inpainting algorithms [11].

In this paper we have compared the results of both the algorithms on different types of images and size of region to be inpainted. In the below images, the region to be inpainted is masked by white pixels as mentioned in step one. Fig. 3(a) shows text region to be inpainted on asymmetric mountain background, 3(b) shows the inpainted image by the algorithm given by Oliveira [10], while 3(c) shows the inpainted image by the algorithm given by Hadhoud [11].Both of them gives good results. Fig. 4 shows large area (moon) to be removed from the symmetric background, and the results of both the algorithms. Here algorithm [10] fails to inpaint but algorithm [11] gives successful results. Fig. 5 shows the cracks to be filled in heritage images where the texture is of highly varying pixels. Here algorithm [10] gives comparatively good results as that of algorithm [11]. Fig. 6 shows a large area (man and fishing rod) to be removed from asymmetric background, and failure of both the algorithms. Similarly, if scratches are to be removed as in Fig. 7 both the algorithms work very well. Table 1 shows a comparative analysis of PSNR of both the algorithms on different types of images and the number of pixels inpainted.

From the results it can be seen that the algorithm given by Hadhoud fills the region from the top right corner thus producing strokes in the result. This is not seen in the algorithm given by Oliveira as it tends to fill the region from all sides but on a compromise with time.

(a)

(b)

(c)

Fig. 3: Mountain: (a) Mountain picture superimposed with text (b) Inpainted image by algorithm [10] (c) Inpainted image by algorithm [11].

Image	No. of pixels Inpainted	PSNR of algorithm given by[10]	PSNR of algorithm given by[11]
Mountain	22103	39.2333	38.9722
Moon	2677	41.4801	42.9298
Heritage	6851	22.6083	22.5764
Fishing	4221	22.6121	23.3942
Boats	1390	43.0315	45.7831

Table I: PSNR Comparison

(a)

(b)

(c)

Fig.4 Moon: (a) Moon to be removed from symmetric background (b) Result of algorithm [10] (c) Result of algorithm [11].

V. CONCLUSION

The algorithm introduced by M. Oliveira gives better results when the area to be inpainted is thin and small, although it takes more time as compared to the algorithm introduced by M. M. Hadhoud due to iterations of convolving mask. Though the results of Hadhoud algorithm fills the region faster, it tends to give strokes as it tries to fill the region from the upper left corner and does not takes into consideration the bottom right portion of an image while algorithm by Oliveira fills the region from all the sides.

Algorithm by Hadhoud works well for large object removal from symmetric background as seen in image moon whereas algorithm by Oliveira fails to do so. While both the algorithms fails to work on large object removal from asymmetric background.

Depending upon the application image, we can use algorithm given by Oliveira if the region to be inpainted is small and contains high frequency components. Algorithm by Hadhoud can be used if the region to be inpainted is small or large object is to be removed from the symmetric background.

REFERENCES

[1] G. Emile-Male. The Restorer's Handbook of Easel Painting. Van Nostrand Rein-hold, 1976.

[2] A. Efros and T. K. Leung, “Texture synthesis by nonparametric sampling,” in IEEE Int. Conf. Computer Vision, Corfu, Greece, Sept. 1999, pp. 1033–1038.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in Proc. SIGGRAPH, 2000, pp. 417–424.

[4] Chan, T., Shen, J. “Mathematical Models for Local Deterministic Inpaintings”. UCLA CAM TR 00-11, March 2000.

[5] Chan, T., Shen, J. “Non-Texture Inpainting by Curvature-Driven Diffusions (CCD)”. UCLA CAM TR 00-35, Sept. 2000.

[6] Criminisi, Patrick Perez, and Kentaro Toyama. “Region filling and object removal by exemplar-based inpainting”, IEEE Transactions on Image Processing, September 2004.

[7] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher, “Simultaneous Structure and Texture Image Inpainting”, in IEEE Transactions on Image Processing, Vol. 12, No. 8, August 2003.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in Comput. Graph. (SIGGRAPH 2000), July 2000, pp. 417–424.

[9] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum, “Image completion with structure propagation”, In Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH), Volume 24, July 2005.

[10] M. Oliveira, B. bowen, R. mckenna, Yu-Sung chang "Fast digital image inpainting" in proceedings of VIIP 2001, pp. 261-266, September 2001.

[11] Mohiy M. Hadhoud, Kamel. A. Moustafa and Sameh. Z. Shenoda, “Digital Images Inpainting using Modified

Convolution Based Method", in International Journal of Signal Processing, Image Processing and Pattern Recognition, April 2009.
[12] Pritika Patel, Ankit Prajapati, Sailendra Mishra, "Review of Different Inpainting Algorithms" in International Journal of Computer Applications, Volume 59, No. 18, December 2012.
[13] Naman Turakhia, Ruchin Shah, Manjunath Joshi, "Automatic Crack Detection in Heritage Site Images for Image Inpainting", ICVGIP, December 2012.

