INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION ENGINEERING & TECHNOLOGY (IJECET)

ISSN 0976 – 6464(Print)
ISSN 0976 – 6472(Online)
Volume 5, Issue 12, December (2014), pp. 258-265
© IAEME: http://www.iaeme.com/IJECET.asp
Journal Impact Factor (2014): 7.2836 (Calculated by GISI)
www.jifactor.com

IJECET

©IAEME

COMPARATIVE STUDY OF DIFFERENT DIGITAL INPAINTING ALGORITHMS

Sudhir S.Kanade,

Professor and Head Electronics and Telecommunication Engineering Department
TPCT's College Of Engineering, Osmanabad (M.S.) India

Sandip S. Gujare

PG Student Electronics and Telecommunication Engineering Department
TPCT's College Of Engineering, Osmanabad, (M.S.), India

ABSTRACT

Image inpainting is the process of filling in missing parts of damaged images based on information gathered from surrounding areas. In addition to problems of image restoration, inpainting can also be used in wireless transmission and image compression applications. This paper gives comparative study of different Image Inpainting Techniques. The proposed work includes the PDE based inpainting algorithm and Texture synthesis based inpainting algorithm.

Keywords: Inpainting, PDE, Texture Synthesis, Image Restoration Isophotes.

I. INTRODUCTION

Image inpainting refers to the process of filling-in missing data in a designated region of the visual input. The object of the process is to reconstruct missing parts or damaged image in such a way that the inpainted region cannot be detected by a causal observer. Image inpainting has been widely investigated in the applications of digital effect (e.g., object removal), image restoration (e.g., scratch or text removal in photograph), image coding and transmission (e.g., recovery of the missing blocks), etc.

The increase in computing power and disk space over the last few decades has created new possibilities for image and movie post processing Typical damages are scratches or stains in photographs, peeled of coatings, or dust particles burned into celluloid. All these flaws create regions where the original image information is lost. Manual restoration of images or single movie frames is possible, but it is desirable to automate this process. Several Inpainting algorithms have been developed to achieve this goal. They may roughly be divided into two categories:

1. Usually PDE based algorithms are designed to connect edges (discontinuities in boundary data) or to extend level. They are targeted on extrapolating geometric image features, lines in some adequate manner into the inpainting domain. especially edges. They create regions inside the Inpainting domain. Most of them produce disturbing artifacts if the inpainting domain is surrounded by textured regions.

2. Texture synthesis algorithms use a sample of the available image data and aim to fill the inpainting domain such that the color relationship statistic between neighbored pixels matches those of the sample. They aim for creating intra region details. If the inpainting domain is surrounded by differently textured regions, these algorithms can produce disturbing artifacts. In this section we will briefly explain the main ideas and the concepts of some of the existing inpainting algorithms.

II. METHODOLOGY

2.1. Partial Differential Equation (PDE) based Inpainting

PDE (partial differential equations) is designed to connect edges or isophotes (line of equal gray values). Bertalmio et al.[1] proposed an image inpainting algorithm based on PDE. Its algorithm will have good results if missed regions are small, but in large damaged regions it will take so long time and won't have good results. Bertalmio et al. [1] use similarities between image processing and fluid dynamic topics, and he proposed an image inpainting algorithm using the Navier Stokes equation. Chan and Shen in [3] use energy functional involving the curvature of the level curves and tries to connect level curves in a smoothing fashion. Masnou and Morel [4] present a new model for image inpainting. They propose a simple but effective algorithm to connect level lines. Telea in [5] propose a fast marching method that can be considered as a PDE method which is faster and simpler to implement than other PDE based algorithms. All of above mentioned algorithms are very time consuming and have some problems with the damaged regions with a large size. PDE technique has been widely used in numerous applications such as image segmentation, restoration and compression. Even for traditional image inpainting.

2.2 Texture Synthesis Based Inpainting

Texture synthesis based algorithms fill in damaged or missed regions using similar neighborhoods in an image, i.e. they try to match statistics of damaged regions to statistics of known regions in the neighborhood of damaged pixels. One of the earliest modes of image inpainting was to use general texture synthesis algorithms to complete the missing regions. The texture synthesis algorithms synthesize new image pixels from an initial seed and strive to preserve the local structures of the image. Earlier inpainting techniques utilized these methods to fill the holes by sampling and copying pixels from neighboring areas. For example, Markov Random Field (MRF) is used to model the local distribution of a pixel and new texture is synthesized by querying existing texture and finding all similar neighborhoods. Their differences lay mainly in how continuity is maintained between the inpainted hole and the existing pixels. These synthesis based techniques perform well only for homogenous texture information would result in a natural completion. Later this effort was extended to a fast synthesizing algorithm by stitching together small patches of existing images referred to as image quilting. Heeger and Bergen in [6] developed a parametric texture synthesis algorithm which can synthesize a matching texture, given a target texture. Igehy et.al in [7] included a composition step to the above method to generate synthetic and real textures. A multi-resolution texture synthesis method which can generate texture under varying brightness conditions was introduced for inpainting by Yamauchi et.al in [8]. Recently, a fast multi-resolution based image completion based on texture analysis and synthesis was introduced by Fang et.al in [9]. In their method, the input image was analyzed by a patch based method using Principal Component Analysis (PCA) and a Vector Quantization (VQ) based technique was used to speedup the matching process of the texture inside the hole region. A. Criminisi [10] uses the simultaneous propagation of texture and structure information is achieved by a single, efficient algorithm. There are innumerable texture synthesis methods other than the aforementioned, but we shall restrict ourselves to illustrate those texture synthesis techniques specifically used for inpainting. While the texture synthesis based inpainting perform well in approximating textures, they have difficulty in handling natural images as they are composed of structures in the form of edges and have complex interactions between structure and texture boundaries. In some cases, they also require the user to specify what texture to replace and the place to be replaced. Hence while appreciating the use of texture synthesis techniques in inpainting, it is prudent to understand that these methods address only a small subset of inpainting issues and are not suitable for a wide variety of applications.

III. IMAGE INPAINTING ALGORITHMS

3.1 PDE based Inpainting Algorithm.

A Partial Differential Equation (PDE) based iterative algorithm proposed by Bertalmio et.al [1] paved the way for modern digital image inpainting. Borrowing heavily from the idea of manual inpainting, this iterative process propagates linear structures (edges) of the surrounding area also called Isophotes, into the hole region denoted by Ω using a diffusion process given by (1)

One of the main drawbacks of this technique is that it underperforms in the replication of large textured regions due to blurring artifact of the diffusion process and the lack of explicit treatment of the pixels on edges. Inspired by this work, Chan and Shen in [4] proposed the Total Variational (TV) inpainting model which uses Euler-Lagrange equation and anisotropic diffusion based on the strength of the isophotes. The TV model was extended to Curvature Driven Diffusion model (CDD) in which included the curvature information of the isophotes to handle the curved structures in a better manner. Tschumperle et.al [11] introduced another PDE based technique referred to as vector valued regularization under anisotropic diffusion framework. These algorithms were focused on maintaining the structure of the inpainting area and hence could not perform as well in texture flling due to blurring artifacts.

Numerical Implementation of the Inpainting Algorithm

- Input:
  - Image to be inpainted,
  - Mask that delimits the position to be inpainted.
- Pre-processing step:
  Whole original image undergoes anisotropic diffusion smoothing. (To minimize the influence of noise on the estimation of the direction of the isophotes arriving at *∂Ω*)
- Inpainting loop: Repeat the following until a steady state is achieved. (Only values inside *Ω* are modified) Every few iterations, a step of anisotropic diffusion is applied.

$$I^{(n+1)}(i,j) = I^{n}(i,j) + \Delta t \,.\, I_t^{n}(i,j), \forall (I,j) \in \Omega \qquad (1)$$

$$I_t^{n}((i,j) = \left[\delta\overrightarrow{L^n}(i,j).\frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|}\right] |\nabla I^{n}(i,j)| \qquad (2)$$

$$\delta\overrightarrow{L^n}(i,j) = (L^n(i+1,j) - L^n(i-1,j), L^n(i,j+1) - L^n(i,j-1)) \qquad (3)$$

$$(i,j) = I_{xx}^{n}(i,j) + I_{yy}^{n}(i,j) \qquad (4)$$

$$\frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|} = \frac{(-I_y^n(i,j), I_x^n(i,j))}{\sqrt{((I_x^n(i,j))^2 + (I_y^n(i,j))^2)}} \tag{5}$$

$$\beta^n(i,j) = \delta\overrightarrow{L^n}(i,j).\frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|} \tag{6}$$

$$|\nabla I^n(i,j)| = \begin{cases} \sqrt{\left(\left(I_{xbm}^n\right)^2 + (I_{xfM}^n)^2 + \left(I_{ybm}^n\right)^2 + (I_{xfM}^n)^2\right)} \text{ where } \beta^n > 0 \\ \sqrt{\left(\left(I_{xbM}^n\right)^2 + (I_{xfm}^n)^2 + \left(I_{ybM}^n\right)^2 + (I_{xfm}^n)^2\right)} \text{ where } \beta^n < 0 \\ 0 \text{ where } \beta^n = 0 \end{cases} \tag{7}$$

β^n (6) is the projection of (3) onto the normalized normal vector (5) (the isophote direction), where (4) is the smoothness estimation (the Laplacian). (7) is a slope-limited version of the norm of the gradient of the image. The subscript indexes b and f denote the backward and forward differences, while m and M denote the minimum and maximum between the derivative and zero.
Ixf= I (i +1, j) – I (i, j) Forward difference in the x-direction
Ixb = I (i, j) – I (i -1, j) Backward difference in the x-direction
Iyf = I (i, j+1) – I (i, j) Forward difference in the y-direction
Iyb= I (i, j) – I (i, j-1) Backward difference in the y-direction
Δ t (1) may be looked upon as a speed factor (i.e., small Δ t gives makes the algorithm converge slower). The algorithm runs in a total of T iterations. The inpainting itself (1) runs in a total of A iterations steps, and B diffusion iterations are performed after the A inpaintings.

3.2 Texture synthesis based Inpainting Algorithm

Texture synthesis algorithms operate essentially on one pixel at a time and determine its value by looking for similar areas in the available image data. In the following we give an overview on texture synthesis algorithms which have been used particularly for inpainting purposes. In the coarsest image several candidates for a best matching patch are searched. In this search process also mirrored and rotated versions of the patches are considered. Once a set of candidate patch positions is found they are transferred to higher levels where the positions are adjusted to the finer resolution by searching in a neighborhood around the best positions found so far. Thus an exhaustive search has only to be performed at the coarsest level, on the finer levels only a small subset of the image has to be considered. Criminisiet.al. and use a confidence and a priority function. The priority of a pixel depends on the confidence and on the gradient magnitude of its surrounding. Pixels lying close to convex corners inside the inpainting domain get high priority since they are surrounded by many high confidence pixels and thus can be reliably inpainted. On the other hand pixels lying close to edges (high gradients) are also assigned high priority such that edges are treated preferably. Continuation of edges tends to build concave spikes into the inpainting domain and the priority of the surrounding pixels decreases. Thus a balanced growing of edges and texture patches is guaranteed. Patches are taken to be fixed size and constant shape (i.e., no rotation or mirroring is considered) and the similarity of patches is simply calculated using sum of squared differences.
The filling process starts with the patch having the highest priority. To fill in the unknown part of the current patch ψ_{px}^{uk} the most similar patch located in a local neighbourhood *w* centered on the current patch is sought. A similarity metric is used for this purpose. The chosen patch ψ_{px}^{*} maximizes the similarity between the known pixel values of the current patch to be filled in ψ_{px}^{k} and co-located pixel values of patches belonging to *W*:

$$\psi^*_{px} = \arg\ \min d\left(\psi^k_{px}\, ,\, \psi^k_{pj}\right)$$

$$\text{s.t.}\quad Coh(\psi^{uk}_{px}) < \lambda_{coh} \tag{11}$$

where d(.) is the weighted Bhattacharya used in [12].Coh(.)is the coherence measure initially proposed by Wexler.

$$Coh\left(\psi^{uk}_{px}\right) = \min\left(d_{SSD}\left(\psi^k_{px}\, ,\, \psi^k_{pj}\right)\right) \tag{12}$$

where d_{SSD} is the sum of square differences. The coherence measure Coh simply indicates the degree of similarity between the synthesized patch ψ^{uk}_{px} and original patches. Compared to previous work [12], there is another substantial difference we only use the best match to fill in the hole whereas a linear combination of the *K* most similar patches is generally performed to compute the patch in [12],.In these cases, the estimated patch is then given by:

$$\psi^*_{px} = \sum W_{p_x, p_j} \times \psi^k_j \tag{13}$$

Where K is the number of candidates which is often adapted locally so that the similarity of chosen neighbours lies within a range $(1+\alpha) \times$ dmin, where dmin is the distance between the current patch and its closest neighbours. Different methods can be used to compute the weights. It could be based on either anon-negative matrix factorization (NMF) or a non local means filter to name a few. Combining several candidates increases the algorithm robustness. However, it tends to introduce blur on fine textures In our method, only the best candidate is chosen. Its unknown parts are pasted into the missing areas. A Poisson fusion is applied to hide the seams between known and unknown parts as shown in Fig.1.

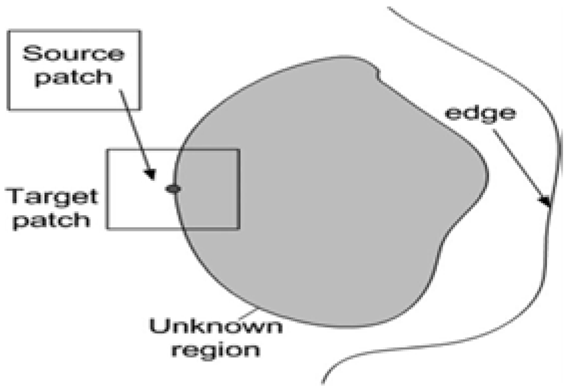
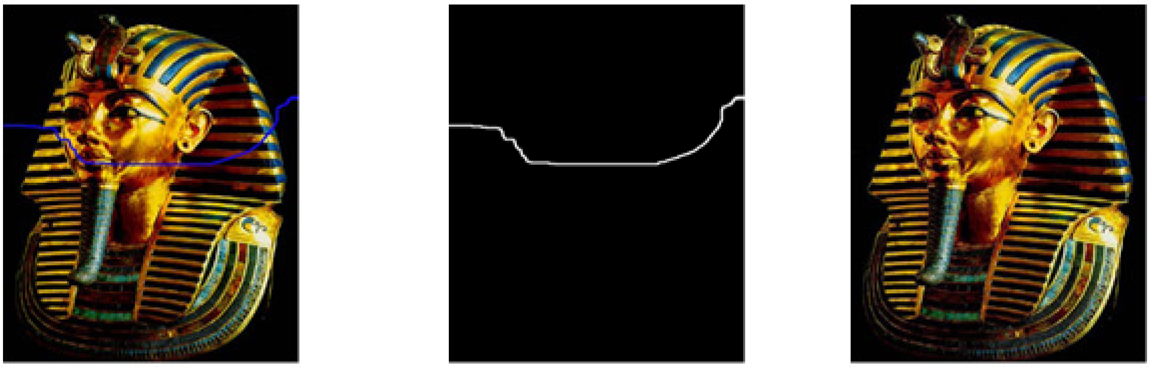


Fig.1: Patch-Wise Texture Synthesis

IV. EXPERIMENTAL RESULTS

In order to verify the effectiveness of image inpainting algorithms which based on pde and texture synthesis in this paper,we use M. Bertalmio [2] and Criminisi [10] algorithms. as shown in figure 3, we add an artificial thin crack using Photoshop to the image The crack is then removed by using the PDE-based digital inpainting algorithm. The image size is 645 by 925, the number of recovered pixels is 4240, and it takes 58 seconds to get this result. Further inpainting iterations would not enhance the image anymore, meaning the PDE algorithm achieves a stable solution. Every ten iterations of inpainting scheme, where the time step equals 0.1. We can realize here that the broken lines are entirely removed, but if the cracks are thicker a blurring effect could be introduced to the image. Since the image presented in figure 2 is colored one, it is treated as a set of three images corresponding to the R, G, B color channels, and the PDE algorithm is applied sequentially to each one. Figure 3 shows the famous bust Statue of Queen Nefertiti before and after inpainting the damaged parts in her crown.

This result is achieved by applying the texture-based inpainting algorithm. The image size is 376 by 525; the number of recovered pixels is 3024, and it takes 114 seconds to achieve this result. The patch size used is 9x9 pixels.



a) **b)** **c)**

Fig 2: (a) The original image, (b) The mask, (c) The result.

4.1 Inpainting Algorithms Parameters

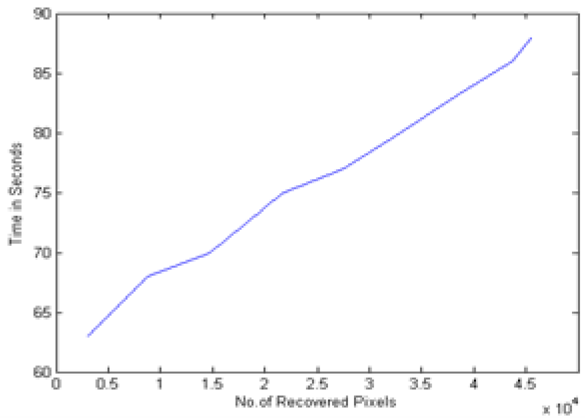
a) **b)** **c)**

Fig 3: (a) The original image, (b) The mask, (c) The result.

4.1.1 Mask Size

The first factor we experiment with is the size of the inpainted region.As expected, we find that the larger the size of the mask, the more time needed by the algorithm to achieve satisfying results. The relationship between the number of recovered pixels and the time taken by the PDE-based inpainting algorithm is shown in figure 4.From this graph we conclude that the relationship between the mask size and the speed of the algorithm is linear. The same procedure is implemented for the texture-based inpainting algorithm; we used the different masks shown in figure 4. and the result also exhibits a
a linear.

The same procedure is implemented for the texture-based inpainting algorithm; we used the different masks shown in figure 4 . and the result also exhibits a linear relationship as shown in figure 4.



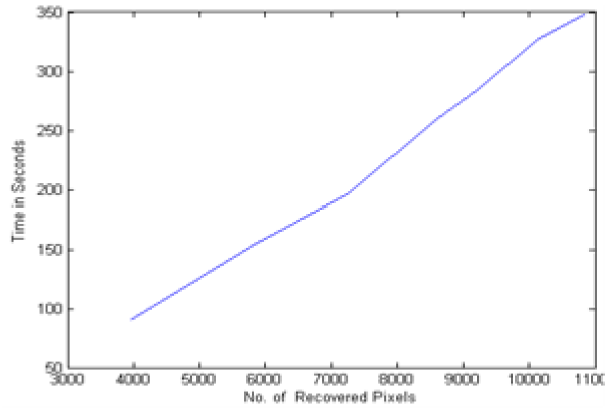


Fig 4: a) PDE algorithm evaluation graph. b) Texture-based algorithm graph

4.1.2. Patch Size

The second important factor that affects the performance of any texture based inpainting algorithm is the patch size. However, it is difficult to set a universal patch size that can be applied to all images. Using large patch size, the filling rate is high, which leads to faster execution time of the inpainting algorithm. However, there are more important implications on choosing the right patch size.

The performance of these methods is compared in Table 1.

Table 1. Performance Analysis of Inpainting algorithms

Parameters	**PDE Based Inpainting**	**Texture Based Inpainting**
Uniform area	good	good
Structure Reconstruction	better for smaller area	depends on nature of image
Texture Reconstruction	poor	good
inpaint area shape	Robust to change	better if matches with patch shape
inpaint area size	Better for smaller area	Robust to Change
Blurring of Edges	yes for larger size	no
Speed	low	low
Other Dependencies	Iterations depend on mask size	Performance depend on Patch Size and shape

V. CONCLUSIONS

In this paper, we have discussed the existing techniques of Image Inpainting. We also compare PDE based Inpainting Algorithm and Texture synthesis based Inpainting Algorithm.. By applying these algorithms to the various images it is found that the choice of using Texture or PDE algorithms depends on the nature of the image to be inpainted. The PDE algorithm is used for structure dominated images to fill-in narrow or crack type regions, while the texture algorithm is more suited for textured images. The time required for the inpainting process depends on the size of the image and the regions to be inpainted, and it ranges form few seconds to several minutes for large images.

REFERENCES

[1] M. Bertalmio, L. Vese, G. Sapiro And S. Ocher, ”Simultaneous Structure and Texture Image Inpainting”Ucla Cam Report 02-47,2003.

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

[3] T. Chan, And J. Shen, "Non-Texture Inpainting By Curvature-Driven Diffusions (Cdd)," J. Vis. Commun. Image Represen, Vol. 12, No. 4, Pp. 436-449, 2001.

[4] S. Masnou, J. M. Morel, “Level lines based disocclusion,” proceedings of the 1998 IEEE In ternational conference of image processing, pp. 259-263.

[5] A. Telea,”An Image Inpainting Technique Based On The Fast Marching Method”, Journal Of Graphics Tools, Vol.9, No. 1, ACM Press 2004.

[6] D.J. Heeger and J.R. Bergen. Pyramid-Based Texture Analysis/Synthesis. In proceedings Of ACM Conf. Comp. Graphics (Siggraph), Volume 29, Pages 229{233, Los Angeles, Ca, 1995.

[7] H. Igehy and L. Pereira. Image Replacement through Texture Synthesis. In Proceedings Of International Conference On Image Processing (ICIP), Volume III, Pages 186{190, 1997.

[8] Hitoshi Yamauchi, Jäorg Haber, and Hans-Peter Seidel. Image Restoration Using Multiresolution Texture Synthesis and Image Inpainting. In Computer Graphics International (CGI 2003), Pages 120{125, Tokyo, Japan, July 2003. IEEE.

[9] C.W. Fang and J.J.J. Lien. Fast Image Replacement Using Multi-Resolution Approach. Jiaya Jia and Chi Keung Tang. Inference of Segmented Color and Texture Description by Tensor Voting. IEEE.

[10] A. Criminisi, P. Peres 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] David Tschumperl and Richard Deriche). Vector-valued image regularization with pde's : A common framework for diÆerent applications.IEEE Transactions on Pattern Analysis and Machine Intelligence , 27(4):506{517, 2005.

[12] O. Le Meur and C. Guillemot, “Super-resolution-based inpainting,” in Proc. 12th Eur. Conf. Comput. Vis., 2012, pp. 554–567.

[13] Mr. Abhijeet A.Chincholkar and Prof. Salim A.Chavan, “An Intelligent Object Inpainting Approach for Video Repairing Using Matlab” International journal of Electronics and Communication Engineering &Technology (IJECET), Volume 5, Issue 2, 2014, pp. 36 - 41, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[14] Mrs.B.A.Ahire and Prof.Neeta A. Deshpande, “A Review on Video Inpainting Techniques” International journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 1, 2013, pp. 203 - 210, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.