International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Comparative Study of Bertalmio and Exemplar Based Image Inpainting

Prof. Tapan Kumar Hazra[1], Rajashree Mitra[2], Jayashree Bhattacharya[2]

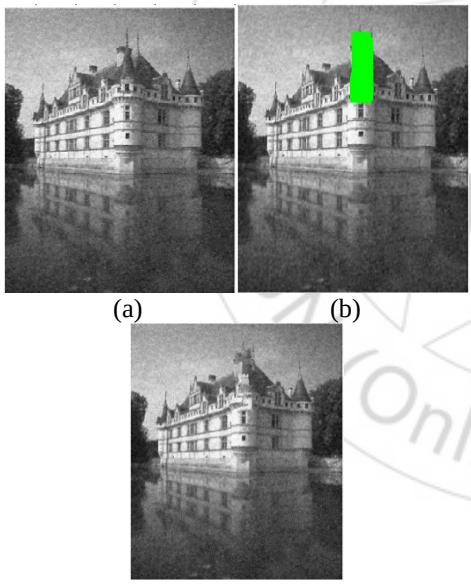
[1]Institute of Engineering and Management, Salt Lake Electronic Complex, Kolkata-700091, West Bengal, India

Abstract: ***Digital image inpainting is a well-known technique for repairing damaged portion of image, reconstructing missing part of the image or removing a selected object in a visually plausible way. Digital image inpainting has a numerous application like image restoration, red eye correction, super resolution, object removal etc. In this paper we provide a detail comparison between Bertalmio's image inpainting and Exemplar based image inpainting algorithm.***

Keywords: Exemplar, patch, image inpainting, isophotes, image texture**.**

1. Introduction

Image inpainting is not a new concept. The aim is to reconstruct the missing portion of an image, removing an unwanted object from the image to make it more legible. The word "inpaint" means to fill in the gaps. In ancient days professional artist use to inpaint the lost part of a historical master piece on the basis of neighborhood and background information. This process was time consuming and there was a possibility of destroying the world-unique art due to direct manual retouch.



(a) (b)

(c)

Figure 1: Restoring the missing part of the image. (a) The original image, (b) The mask of the image, (c) The image after inpanting.

But today there are various digitalized inpainting techniques that can easily and efficiently inpaint the image with least user interference. The user just needs to mention the inpainting region manually. Rest of the task is done automatically. Digital image inpainting found numerous applications like removal of the superimposed text, i.e., date, subtitle and also complete object removal like microphone, wires in special effect. Paper outline: Section 2 contains the overview of inpainting algorithm. Section 3 contains detail explanation of two inpainting algorithms (a) Bertalmio and (b) Exemplar based inpainting. Section 4 contains comparison of the results obtained. In Section 5 conclusion is written.

2. Overview

In Mathematical term, we can describe inpainting as follows: for a sequence, say S, its subset, say X, can estimate the total S as S', such that I(S') = I(X) where I stands for information. For clarity we would like to explain it with an example. Suppose there exists a sequence S = {2,4,X,8,10,12}, where X is unknown. If X is derived as 6 then the sequence S = {2,4,6,8,10,12}is a set of natural even numbers. But if X is derived as 15, then it seems that the sequence is not correct and needs some attention. In inpainting the plausible region gives a natural sequence, whereas for the region required to be inpainted, the sequence shows abnormality.

*The principal of image inpainting***:** Let Ω ϵ D, where D is the image domain and Ω is the portion in D to be inpainted shown in figure 2, δΩ is the boundary of Ω. The objective is to interpolate data from the boundary region δΩ and fill the inpainting region Ω with the appropriate pixel value.

Image inpainting methodology:

- Select the image that needs attention.
- Manually select the portion to be inpainted i.e., the target region (Ω).

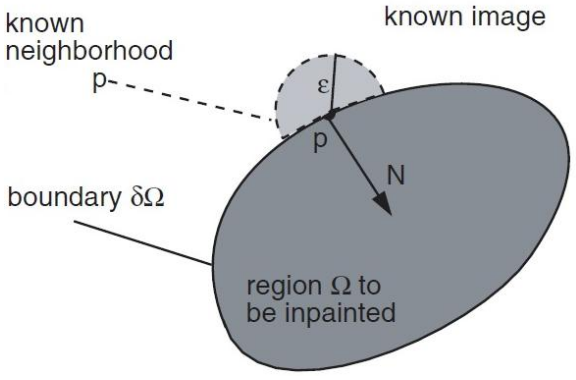


Figure 2: Principle of Inpainting

- Start filling the inpainted region by averaging the pixel value from the surrounding of the target region.
- To reconstruct a plausible image, the inpainting algorithm needs to continue isophote (a curve on a chart joining points of equal value) very smoothly inside Ω.
- Continue the above process till the all the pixel in the target region is inpainted.

Based on the above steps many inpainting algorithms were proposed..

3. Comparative Study

Bertalmio Image Inpainting Algorithm:Technique proposed by Bertalmio will prolong the isophote lines arriving at the boundary of the inpainted region Ω. It says I_0(i,j): [0,M] X [0,N] → $\mathbb{R}$, with [0,M] X [0,N] ⊂ $\mathbb{N}$ x $\mathbb{N}$, is discrete 2D gray level image. Passing the image through an iterative algorithm will give a series of images I (i,j,n): [0,M] X [0,N] X $\mathbb{N}$→$\mathbb{R}$ such that I(i,j,0)=I_0(i,j) and $\lim_{n\to\infty} I\ (i,j,0)$ = I_R (i,j), where I_R (i,j) is the output [6].
In general,

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

where n denotes inpainting time, i and j are the pixel coordinates, Δt is the rate of change and I_t^n(i,j) denotes update of the image I^n(i,j). Now we need to know the updated I_t^n(i,j). The information is smoothly propagated inside the region Ω from its surrounding. If L^n(i,j) is the information to be propagated and $\vec{N}^n$(i,j) is the direction of propagation then,

$$I_t^n(\text{i,j}) = \delta\overrightarrow{L^n}(i,j).\vec{N}^n(\text{i,j}) \qquad (2)$$

where$\delta\overrightarrow{L^n}\ (i,j)$ denotes measure of change in information L^n(i,j). From equation (2) we can we can find the informationL^n(i,j) for a given image and its change in the $\vec{N}$ direction. For a steady state, i.e., where no change takes place, I^{n+1}(i,j) = I^n(i,j). So from (1) and (2) we can say for a steady state, $\delta\overrightarrow{L^n}(i,j).\ \vec{N}^n$(i,j) = 0. Here we can say that same information has been propagated inside the Ω region. To do inpainting in a visually plausible way, L^n(i,j) needs to act as an image smoothness estimator. The author here uses a simple discrete implementation of Laplacian:

$$L^n(\text{i,j}) := I_{xx}^n(i,j) + I_{yy}^n(i,j) \qquad (3)$$

where, I_{xx} and I_{yy} are derivatives of *x* and *y* respectively.
The direction of propagation $\vec{N}$ is normal to the signed distance to δΩ, i.e., at each point (i,j) in Ω the vector $\vec{N}$ (i,j) will be normal to the shrinking δΩ to which (i, j) belongs. Figure 3, shows the direction of propagation, i.e.,$\vec{N}$

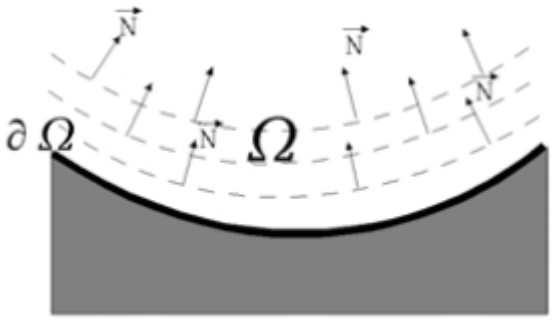


Figure 3: Propagation direction as normal to the signed distance to the boundary of the region to be inpainted Ω [6].

The best choice for $\overrightarrow{N}$ is the isophotes direction. For any point (i,j) the discretized gradient vector ∇I^n(i,j) gives the direction of largest spatial change, that means the 90 degree rotation $\nabla\perp I^n$ (i,j) gives the direction of smallest spatial change. Hence it gives the direction of isophotes. So when we take time into account along with the direction$\vec{N}$, the $\vec{N}\ (i,j,n)$ is same as $\nabla\perp I^n$ (i,j). Here we take time into account, i.e., $\vec{N}(i,j,n)$, to find the isophotes direction to get the desired continuity at$\partial\Omega$, instead of taking fixed $\vec{N}$, which implies that the direction of isophote is well known from the beginning. The smoothness estimator is projected to the variation into the isophote direction, to update the information inside Ω.

This algorithm also include anisotropic diffusion, to ensure correct evaluation of direction field $\vec{N}$. Diffusion can be explained as periodic curving of lines to avoid them crossing each other. Anisotropic diffusion gives the desired result without losing the sharpness of the image (for detail refer to [6]). It applies continuous-time / continuous-space anisotropic diffusion equation:

$$\frac{\partial I}{\partial t}(x,y,t)=g_{\epsilon}(x,y)\kappa(x,y,t)|\Delta I(x,y,t)|,\ \forall\ x,y\epsilon\Omega^{\epsilon} \qquad (4)$$

where, $\Omega^{\epsilon}$is a dilation (to enlarge or expand) of Ω having a circle of radius ∈, κ is Euclidean curvature of isophotes of *I* and $g\epsilon(x,y)$ is a smoothness function. For detail study go through paper [6] and [7]

The input to this algorithm is the image to be restored and the mask (matrix representing inpainted region with 0, and rest with 1) which identifies the portion to be inpainted. At first the original image undergoes anisotropic diffusion and then it enters into the inpainting algorithm, where the inpainted region gets updated. The basic idea is to continue geometric information that arrives at the border δΩ.

Based on the above algorithm Chan and Shen (2001) proposed a new technique known as total variational inpainting model. This model uses Euler-Lagrange equation and anisotropic diffusion. This method works well with small inpainting region, but fails to connect broken edges and inpaint image with great texture pattern. This TV model is then extended to CCD (Curvature-Driven-Diffusion

Model) by Chan and Shen. It include curvature information of isophotes for inpainting curved region.

Disadvantages: Although the algorithm proposed by Bertalmio is one of the basic inpainting algorithm and it works fine for small inpainting region and cartoon image, but it fails to inpaint large inpainting region and textured image. Also it may produce blurred resulting image.

Applications: It has numerous applications such as, image restoration, image segmentation, etc.

Exemplar Based Image Inpainting: Exemplar Based image inpainting is a very well-known and efficient class of image inpainting algorithm. The inpainting algorithm helps us to improve both the structure and texture of the inpainting region. The remarkable feature of this algorithm is that it can inpaint large missing area as well as reconstruct the small defects in the image. It precise way the working of this algorithm is to assign priority to the pixel of the unknown region i.e. target region and selection of the best matching patch for the known region i.e. source region and finally replace the unknown region with the best and appropriate patch from the known region.

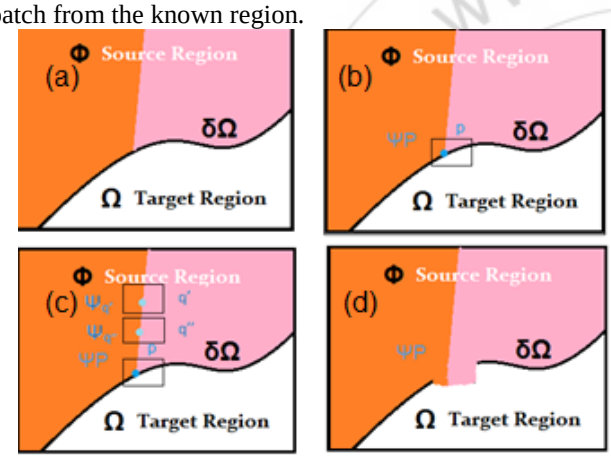


Figure 4: (a) Image to be inpainted, showing Source region (Φ), target region (Ω) & its contour(δΩ) (b) Patch ψ_p centered at point p ϵ δΩ. (c) Most likely candidate patches $\psi_{q'}$ and $\psi_{q''}$ that lies along the boundary between the te to texture in the source region for the patch $\psi_p$in the target region. (d) The best candidate patch is selected and copied to the target region.

The stepwise description of the exemplar based image inpainting algorithm with suitable diagrams is given below:

1) Notations: As shown in figure 4 [3], the region to be inpainted is referred as target region denoted by(Ω), the region or area of the image from where information is taken to inpaint the target region is referred as source region denoted by (Φ).δΩ is used to represent the boundary of the target region also called countour or fillfront.
2) Identify the target region: Target region is manually highlighted with some special color, it is a common fashion to take green color to represent the target region (R=0 ,G=255 ,B=0).
3) Initialize the confidence value: Each pixel has got a certain 'color value' and 'confidence' value which determine our confidence in the pixel value and which is fixed once a pixel is inpainted. Confidence term is also used to determine the amount of reliable information around the pixel p, pixel p have a certain patch of size say 9 X 9 (practical size is lager).During initialization, the function C(p)=0 , for all p ϵ Ω and C(p)=1 for all p ϵ I-Ω.
4) Find the boundary of the target region: It is a practice to first inpaint the region along contour by giving then a temporary higher priority. It is efficient way to do so because it is surrounded with edges that have high confidence value therefore, it makes the algorithm more reliable.
5) Selection and Calculation of the patch priority: By default patch size is taken as 9X9 pixel but in practice patch size should be slightly larger than the largest distinguishable texture element in the image. We denote patch by ψ_p. We use the mean squared error to find the best matching patch from the source region to inpaint the target region

$$MSE=\Sigma\frac{(f_{x,y}-g_{x,y})^2}{N} \quad (5)$$

Where$f_{x,y}$ represent the element in the patch, $g_{x,y}$ represent the element in the patch for which MSE is calculated. N is the total number of element in the patch. The result depends in the order in which patches are selected is very important.

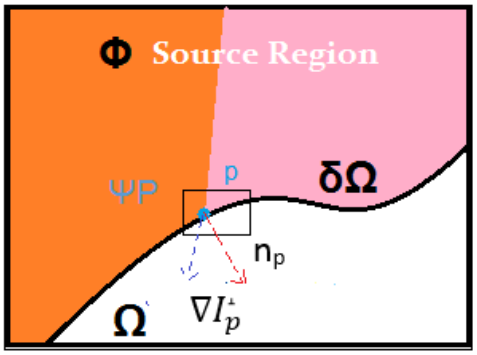


Figure 5: The image is denoted by *I*. Patch Ψ_p,n_p is normal to the contour δΩ of the target region Ω, and $\nabla Ip^{\perp}$ is the isotpote at point p.

According to Figure 4 [3] we have a patch $\Psi_p$centered at point p for some p, we denote its priority by P(p) as a product of two terms:

$$P(p)=C(p)D(p) \quad (6)$$

where C(p) is the confidence term & D(p) is the data term. They are defined as :

$$C(p)=\frac{\Sigma q \epsilon \Psi p \cap (I-\Omega) C(q)}{|\Psi p|} \quad (7)$$

$$D(p)=\frac{|\nabla Ip^{\perp}.n_p|}{\alpha} \quad (8)$$

where, $\Psi_p|$ is the area of Ψ_p, α is the normalisation factor (i.e,α is 255 for a typical grey-level image), n_p is the unit vector orthogonalto the front δΩ in the point p, $^{\perp}$ denotes the orthogonal operator.

The priority P(p) is calculated for every pixel with distinct patch but patch that include the corner of the target region is inpainted first. As confidence term gives the measure of reliability so as data term D(p) is the function that is used to

indicate isophote's strength of hitting the front δΩ at each iteration for the algorithm.

6) Choose the patch with the maximum priority and then replace the patch with the best suitable and matching exemplar (patch) among all the candidate set from the source region and update the confidence value. That is $C(p)=C(\hat{p})$, where, $C(\hat{p})$ is the confidence value of the exemplar (patch) from the source region, C(p) is the confidence value of the patch in the target region that is inpainted.

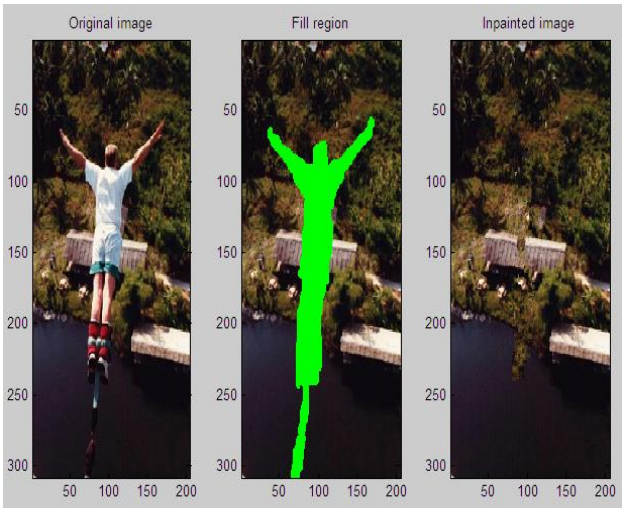
This process continues till all the pixel in the target region is completely inpainted. There are some points that we have noticed during the study and implementation of this algorithm such that in [3] the quality and reliability of the inpainted image is highly dependent on the order in which the filling process proceeds. The traditional concentric layer filling (commonly known as onion peel) process unexpectedly reconstructs curve at the horizontal line between the two different shades color of the image. But the edge driven achieves the artifact-free reconstruction. In [2] it is stated that, often the most appropriate patch lies very close to the patch selected for inpainting. So it is a good approach to restrict the search of the patch up to certain diameter surrounding the region to be inpainted. Doing so the performance of the inpainting algorithm is improved.

4. Results

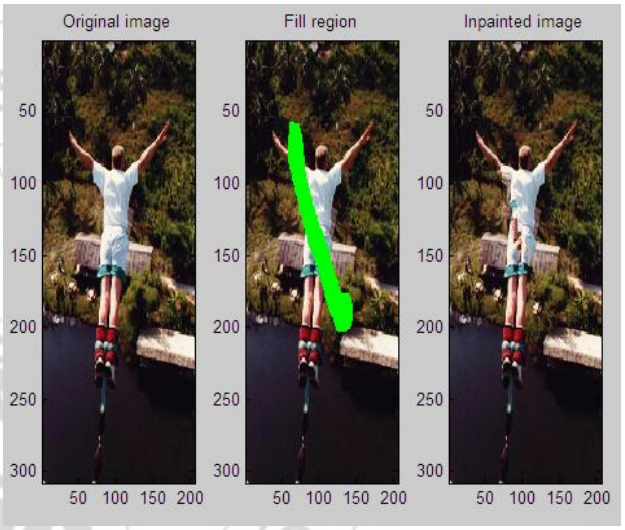
We have analyzed and implemented the Bertalmio and Exemplar image inpainting algorithms on the images given below in this section. The code is executed in Matlab version 10& 13 on Windows 7 operating system. Figure 6, Figure 7 and Figure 8 are the results of Exemplar based image inpainting. All of this sub figure (a), (b), (c) conatins 'Original image','fill region' and 'inpainted image' respectively. Original image is the image on which inpainting is applied. 'fill region' show the mask that should be inpainted with the help of the algorithm. 'Inpainted image' is the result of the of the algorithm.

For Figure 6, in case 1, algorithm removes the person and inpaint the image with the help of surrounding information, as a result the person is removed in the inpainted image. In case 2 some portion of the person's body is selected as the mask and inpainted. In case 3 the rope connected with the man's feet is selected as the mask and in the final result it is showing the man is jumping without rope attached with his feet. Figure 7 is another example of Exemplar based image inpainting algorithm. In this image some portion is selected as the mask which needs to be restored. Finally with the help of the algorithm the mask portion is inpainted and we get the final result. In example 3 we take Lena.jpg as our original image. Case 1, Case 2 and Case 3 shows that the algorithm is capable of inpainting regions with multiple textures.

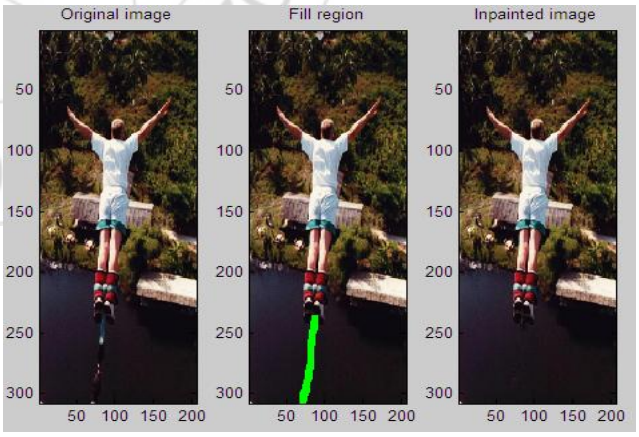
Example1:



(a) Case 1



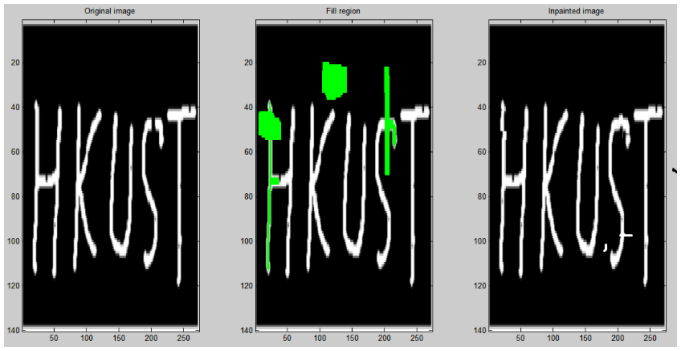
(b) Case 2



(c) Case 3

Figure 6: Bungee Jump Results.

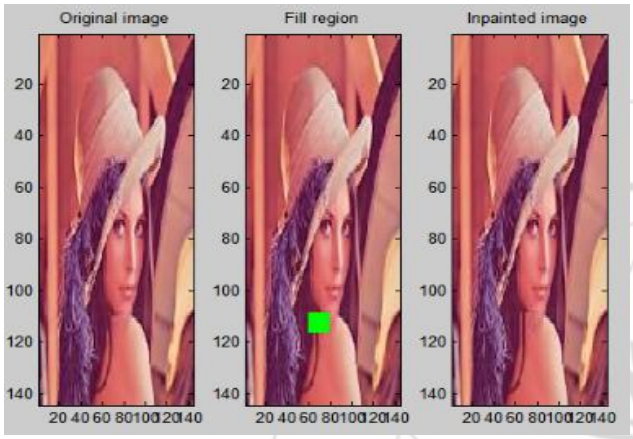
Example2:



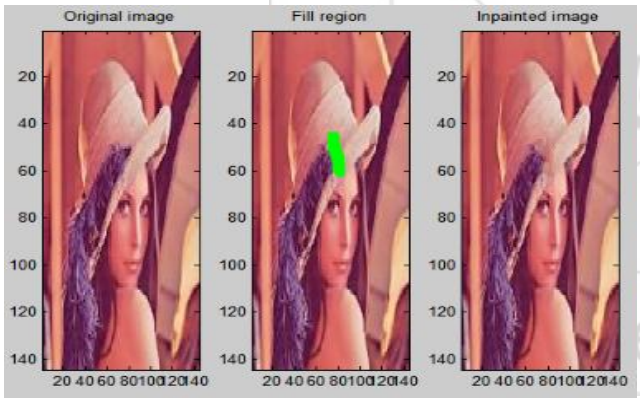
(a) (b) (c)

Figure 7: Text Results

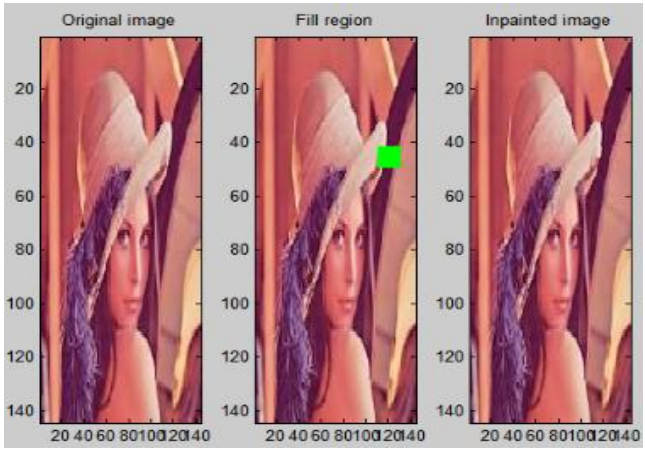
Example 3:



(a) Case 1



(b) Case 2



(c) Case 3

Figure 8: Lena image.

Example 4:

(a) Original Image

(b) Inpainted Image

Figure 9: Text Results

Example 4 is the result of the Bertalmio image inpainting algorithm. Figure 9(a) shows the original image that need to be inpainted. In the original image some portion of the letter are missing but we can see that after applying the algorithm the missing portion of the image as been inpainted as shown in Figure 9(b).

By comparing the results of Exemplar and Bertalmio image inpainting in Figure 7 and Figure 9 respectively we came to a conclusion that former can inpaint large portion of the image but latter algorithm fails.

Table 4.1: Processing time for a common image on each inpainting algorithm

Inpainting Algorithm	Processing Time
Exemplar	20 seconds
Bertalmio	2 seconds

On applying Exemplar based and Bertalmio's inpainting algorithm on Text figure, we get Figure 7 and Figure 9 as results respectively. We observe that the processing time for Exemplar based algorithm is much more than Bertalmio's algorithm, due to the complexity in finding perfect patch.

5. Conclusion

In this paper we see that both Bertalmio and Exemplar based inpainting algorithm is able to restore the image. But Bertalmio's algorithm is more suitable for small region. Also it provides a good result for object removal in a simple background. Exemplar based algorithm also provide good result for restoring missing part of an image. But the processing time is more. To find the best matches patch, the algorithm needs to do too much of computation. So if the source region is large, the time required to inpaint will be

larger, and if the source region is small, we will end up with a poor quality of result.

In future work, we would like to come up with an algorithm which is capable of choosing any one of the two algorithms based on the size and shape of inpainting region or based on the complex background.

References

[1] SanketKhedikar,P.N.Chatur: Digital Inpainting on the Basis of Exemplar Based Method. International journal of Computer Science and Information Technology, Vol.5(3),(2014),3943-3945.
[2] Anupam,PulkitGoyal,SapanDiwakar:Fast and Enhanced Algorithm for Exemplar Based Image Inpainting.IIITAllahabad,India.
[3] A.Criminisi,P.Perez,K.Toyama: Region Filling and ObjectRemoval By Exemplar-Based Image Inpainting.IEEETransaction on image processing,Vol.13, No.9 (2004).
[4] Bhimaraju Swati, Naveen Malviya,Shrikant Lade: Analysis of Exemplar Base Inpainting for Adaptive Patch Propagation using Wavelet TransformISSN 2250-2459,ISO 9001:2008 Certified Journal, Volume 3,Issue 5,(2013).
[5] PranaliDhabekar,GeetaSalunke:The Exemplar-Based Image Inpainting Algorithm through Patch Propagation.IJRTE,ISSN:227-3878,Volume-1,Issue-4,(2012).
[6] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.,: Image inpainting. In: Proceedings of SIGGRAPH 2000, pp.417-424. ACM Press, New York (2000).
[7] Bertalmio, M., Vese, L., Sapiro, G., Osher, S.,: Simultaneous structure and texture image inpainting. IEEE Trans. Image Processing 12, 882-889 (2003).
[8] Rajkumar L Biradar, Vinayadatt V Kohir:The novelimage inpainting Technique Based on Median Diffusion.Sadhana Vol.38, Part 4,(2013), Indian Academy of Sciences.
[9] Vese, L..A., Osher, S.J: Modeling Textures with totalvariation minimization and oscillating patterns in imageprocessing. J. Sci. Comput 19(1-3), 553-572 (2003).
[10] Sangeetha, K., Dr..Sengottuvelan, P.,: Efficient exemplar based image inpainting using multidirectional contourlet transform, Oct 2013.
[11] Drori, I., Cohen-Or, D., Yeshurun, H.,: Fragment-based image completion. In: Proceedings of SIGGRAPH 2003, pp. 303-312. ACM Press, New York(2003).
[12] Rafeal C. Gonzalez, Richard E. Woods : Digital Image Processing.

Author Profile

Tapan Kumar Hazrareceived his M.E degree from Jadavpur University, Kolkata, West Bengal, India. Since from 2003, he is working as Assistant Professor of Department of Information Technology at Institute of Engineering & Management, Salt Lake, Kolkata, West Bengal, India. His research interest includes Design and Analysis of Algorithms, Image Processing, Machine learning, Cryptography.

RajashreeMitra received her B.Tech degree in the year 2013 from Sikkim Manipal Institute of Technology, Majhitar, East Sikkim, India. She has completed her M.Tech degree in the year 2015 from Institute of Engineering and Management, Salt Lake, Kolkata, West Bengal, India.

Jayashree Bhattacharyareceived her B.Tech degree in the year 2011 from College of Engineering and Management Kolaghat, West Bengal, India. Received herM.E degree in the year 2013 from Jadavpur University, Kolkata, West Bengal, India. She has completed her M.Tech degree in the year 2015 from Institute of Engineering and Management, Salt Lake, Kolkata, West Bengal, India. Currently she is working as Assistant Professor of Department of Computer Science and Engineering at University of Engineering & Management, Jaipur, India.