Modified Region Filling and Object Removal by Exemplar - based Image Inpainting

Ankit D. Prajapati
Asst. Prof. in Computer Science & Engineering, FETR, Bardoli, Guajrat

Hetal M. Patel
Asst. Prof. in Computer Engineering, GDEC, Abrama, Navsari, Gujarat

Ankurkumar G. Patel
Asst. Prof. in Computer Engineering, MGITER, Navsari, Gujarat

ABSTRACT
Image Inpainting is technique in which mainly used to filling the region which are damaged and want to recover from unwanted object by collecting the information from the neighbouring pixels. 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 region filling and Object Removal by Exemplar based Image Inpainting algorithm for exemplar based inpainting method by modifying the distance function. 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. Our experimental results show that our method improves the quality of image inpainting compared with the existing exemplar-based image completion algorithms.

Keywords
Object Removal, Image Inpainting, Texture Synthesis, PDE, gradient similarity metric

1. INTRODUCTION
There are lots of advantages multimedia instruments in today’s world peoples are clicking lots of images of theirs and also they were trying to preserve their past pictures. And as the time goes on those pictures got damaged.(cracks, starches, image data loss) Inpainting is the art of restoring lost parts of an image and reconstructing them based on the background information. In the real world, many people need a system to recover damaged photographs, designs, drawings, art works etc. damage may be due to various reasons like scratches, overlaid text or graphics etc. [1]

Inpainting technique has many applications such as, object removal in digital photos, removal of occlusions (date, stamps, logo etc.), such as large unwanted regions, red eye correction, super resolution, restoration of old films and paintings etc. [2] Another use of image inpainting is in creating special effects by removing unwanted objects from the image. Unwanted objects may range from microphones, ropes, some unwanted person and logos, stamped dates and text etc. in the image. During the transmission of images over a network, there may be some parts of an image that are missing. These parts can be reconstructed using image inpainting. Many works on Inpainting have been proposed these recent years [1]. The image is to decompose the original image into a structure and a texture image. Reconstruction of each image is performed separately. The missing information in the structure component is reconstructed using a structure inpainting algorithm, while the texture component is repaired by an improved exemplar based synthesis technique.

2. RELATED WORK ON EXEMPLAR BASED INPAINTING [13]
As it was shown in above that PDE based inpainting algorithms are not sufficient for faithfully reconstructing textured images, nor images with large missing areas. Thus, when inpainting is done with an image restoration purpose in mind, more complex techniques are required, as paintings are composed of both structures and textures. Exemplar-based inpainting methods can overcome this drawback, being able to provide reasonably good quality results, even for large gaps, by combining the isophotes driven inpainting with texture synthesis. The reconstructed visual quality and the reasonability of the filled image are mainly influenced by the filling order. So, we conclude that better performance is obtained using developing a robust priority function. Exemplar based inpainting iteratively synthesizes the target region, by the most similar patch in the source region.[3] According to the filling order, the method fills structures in the missing regions using spatial information of neighboring regions. This method is an efficient approach for reconstructing large target regions. Generally, an exemplar-based inpainting algorithm includes the following three main steps:

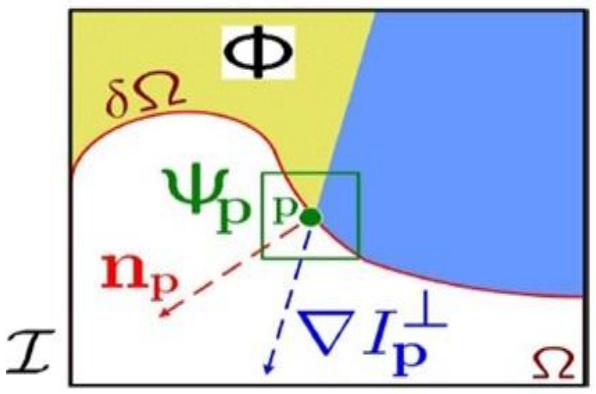


Fig.1. Notation diagram for Exemplar based inpainting [13] In patch Ψ $_p$, then $_p$ is the normal to the contour $\delta\Omega$ of the target region Ω and $\nabla I_p^{\perp}$ is the isophotes at point p. The entire image is denoted with *I*.

First take an input image, the user selects a target region, Ω, to be removed or filled. The source region, Φ , may be defined as the entire image minus the target region (Φ = I - Ω), as a dilated band around the target region, or it may be manually specified by the user. Here we provide a default window size 9 x 9 pixels, but in practice require the user to set it to be slightly larger than the largest distinguishable texture element, in the source region. Once these parameters are determined, the region-filling proceeds automatically.

In algorithm, each pixel maintains its c*olour* value (or “empty”, if the pixel is unfilled) and a *confidence* value, which reflects our confidence in the pixel value, and which is frozen once a pixel has been filled. During the course of the algorithm, patches along the fill front are also given a temporary *priority* value, which determines the order in which they are filled. Then, algorithm iterates the following three steps until all pixels have been filled:

***1) Computing patch priorities*.**

Filling order depends on the priority values that are assigned to each patch on the fill front. Given a patch Ψ_p centered at the point p for some $p \in \delta\Omega$ (see fig.2), we define its priority *P* (p) as the product of two terms:

$$P(p) = C(p) \cdot D(p) \tag{1}$$

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

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I-\Omega)} C(q)}{\Psi_p}, D(p) = \frac{\left|\nabla I_p^{\perp} . n_p\right|}{\alpha} \tag{2}$$

Where $\left|\Psi_p\right|$ is the area of $\left|\Psi_p\right|$, Ω is a normalization factor (*e.g.*, α =255 for a typical grey-level image), n_p is a unit vector orthogonal to the front $\delta\Omega$ in the point p and $\perp$ denotes the orthogonal operator. The priority P(p) is computed for every border patch, with distinct patches for each pixel on the boundary of the target region. During initialization, the function C(p) is set to *C* (p) = 0 . $\forall_p \in \Omega$ and C (p) = 1 . $\forall_p \in I - \Omega$

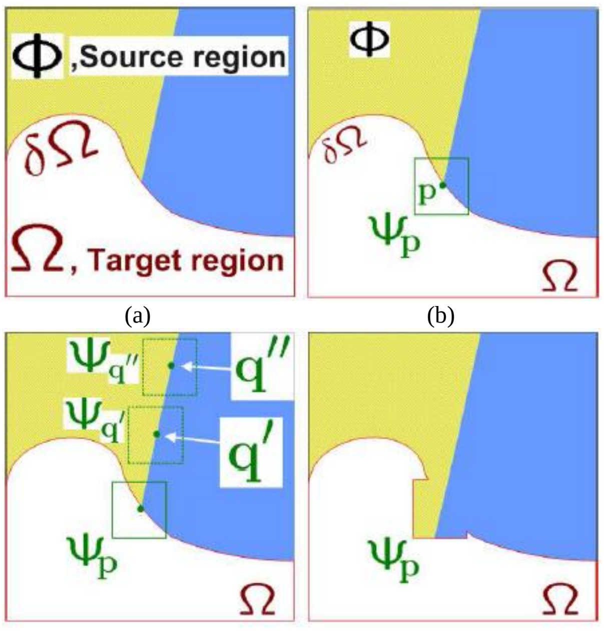


Figure 2: Visualizations of exemplar-based inpainting process.[13] (a) Original image shows sources and target region as well as the boundary contour (b)Patch that was given the highest priority(c) Candidate patches $\Psi_{q'}$ and$\Psi_{q''}$(d) The patch $\Psi_{q'}$ is filled in with the best matching patch.

2) Propagating texture and structure information.

Search the source region to find the patch which is most similar to $\Psi_{\hat{p}}$.

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \phi} d(\Psi_{\hat{p}}, \Psi_{\hat{q}}) \tag{3}$$

Where the distance $\boldsymbol{d(\Psi_a, \Psi_b)}$ between two generic patches is simply defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.

3) Updating Confidences values

In which the boundary $\delta\Omega$ of the target region Ω and the required information for computing filling priorities are updated.

$$C(q) = C(\hat{p}) \ \forall_{q \in \Psi\hat{p}} \cap \Omega \tag{4}$$

In this paper present, the new priority function is able to resist undesired noises.

3. EXISTING REGION FILLING AND OBJECT REMOVAL BY EXEMPLAR-BASED IMAGE INPAINTING [13]

Generally speaking, an exemplar-based image Inpainting algorithm includes four main steps as follows:

Step 1 Identify the target region. The target regions are extracted and the contours are identified manually.

Step 2 Compute filling priorities and select the patch with the highest priority. Each patch Ψp centered at the pixel p (pϵδΩ) has a inpainting priority, which is defined to be the product of Confidence term C(p) and Data term D(p).

We find that the value of C(p) tends to zero with the proceed of the original algorithm, so a regularized confidence term shown in [8] takes the place of the original confidence term.

In equation (1) the confidence term C(p) is the number of known information in the given patch Ψp and is defined as:

The calculation function of the data item D(p) is the most critical point of the algorithm. The accurate estimation of the edge direction determines the value of the data item directly.

If there is a lot of texture information at the edge of the image, it may lead to inaccurate estimation of the structure edge. So we introduce structure tensor to improve the calculation method of D (p).

In equation (2) Where |Ψp| is the area of patch Ψp. During initialization, C(p) is often set to 0 for all pixels in the target region Ω, and to 1 for known pixels in the source region Φ.

The data term D(p), which encourages propagating linear structures, is a function of the strength of isophotes hitting the front δΩ at each iteration and is defined in equation(2).

Also in equation (2) Where α is known as a normalization factor (e.g., α= 255 for a grey-level image), and here $\perp$ is called as isophote at pixel p and also np is a unit vector orthogonal to the contour δΩ at the pixel p. Then find the patch Ψ*p*ˆ with the highest priority.

Step 3 Search the most similar block Ψ*q*ˆ with Ψ*p*ˆ from the source region Φ. Copy the pixels from Ψ*q*ˆ to Ψ*p*ˆ.

Step 4 Update confidence values and the contour of the the target region.
The algorithm iterates the above four steps until all pixels in the target Ω have been filled.

4. IMPROVED ALGORITHM

We find that 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.

$$G = G(\Psi_p - \Psi_q) \qquad (5)$$

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 gradient of an image measure how it is changing. It provides two pieces of information. The magnitude of the gradient tells us how quickly the image is changing while the direction the gradient tells us the direction in which the image is changing most rapidly.
The detail of the algorithm steps are as follows:

1. Computing patch priorities

$$C_N(p) = (1-w)C(p) + w \quad 0<w<1 \qquad (6)$$

where *w* is regularizing factor. So the value of the new confidence term is regularized to *[w, 1]*.

$$D(p) = div(J\nabla I(x, y) \qquad (7)$$

Where *div* is divergence operator. The restoration order of the target patch is no longer only determined by $\nabla I_p^{\perp}$.

$$P(p) = \alpha C_N(p) + (1-\alpha)D(p) \qquad (8)$$

The priority function is defined as the weight sum of regularized confidence term C(p) and new data term D(p). Where α is adjustment coefficient, satisfying 0<α<1.

2. Propagating texture and structure information

Search the source region to find the patch which is most similar to $\Psi \hat{p}$.

$$\Psi_{\hat{p}} = \arg\min_{\Psi_q \in \phi} d(\Psi_{\hat{p}}, \Psi_{\hat{q}}) \qquad (9)$$

Here the distance $d(\Psi_{a,}\Psi_b)$ between two generic patches is simply defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.

$$d = \sum_{i=1}^{i=A} (I_{ai} - I_{bi})^2 + (G_{ai} - G_{bi})^2 \qquad (10)$$

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

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

3. Updating Confidences values

The confidence C (p) is updated in the area delimited by $\Psi\hat{p}$ as follow:

$$C(q) = C(\hat{p}) \ \forall_{q \in \Psi_{\hat{p}}} \cap \Omega \qquad (11)$$

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

5. EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed variance approach and the improvement on several images and compared the so-obtained results with the conventional approaches.

A. Comparison with Criminis's[13] and Lui kui[20] approach

Now we present the comparison of our approach with one presented by Criminisi's in[13].The image in Figure (4) was given us input to the inpainting process that used our approach as well as to our implementation of Criminisi's approach. The result using Criminisi's approach were not that promising whereas our algorithm achieved better result.

The result evaluation is performed by comparing PSNR(the Peak Signal-to-Noise Ratio) between restored image and original image. Generally the higher the PSNR value the larger the similarity of the repaired image to the original. The equation to calculate PSNR as given follows.

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[u(x, y) - u_0(x, y)]^2 \qquad (12)$$

$$PSNR = 10\log\frac{255^2}{MSE} \qquad (13)$$

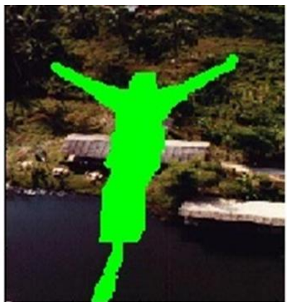
Using our approach, however the best exemplar process well defined and therefore it selected a better patch as shown in Figure. 4, Figure. 5, Figure. 6.

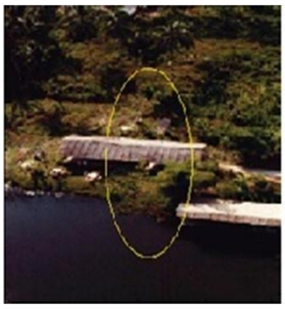
However, PSNR sometimes disaccords with human visual judgments. So SSIM (Structural Similarity) is used as well because the SSIM criterion is closer to human vision system.

B. Real Life Examples

Now we present a few more examples from real life scenes which are captured by us.
Fig 5 shows examples of inpainting using our algorithm. The noise is added randomly to the image and then inpainting was applied. In this figure the structure of the head part is preserved. But the computational time for our algorithm is little longer then [13].
In all experimental, the size of patches is set to 7 x 7, 9 x 9 or 15 x 15 for all images.

(A)


(B)


(C)

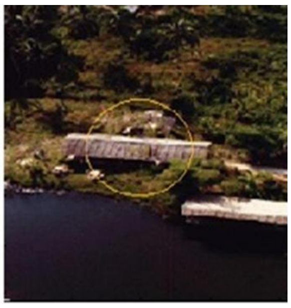
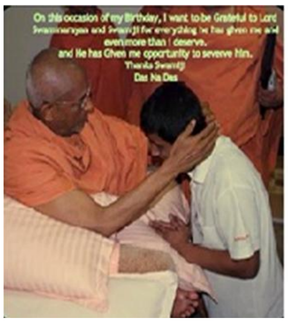
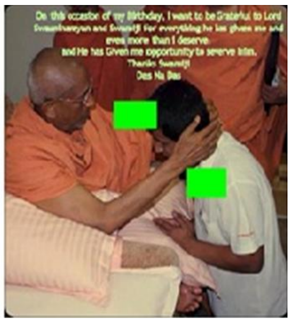
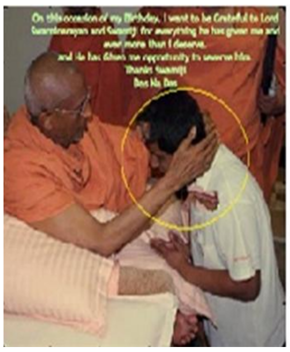

(D)

Figure 3: TEST Image1 the removal of the big object (A)Input image (B)Inpainted image(C)Results of Criminisi's (D)Our results.


(A)


(B)


(C)

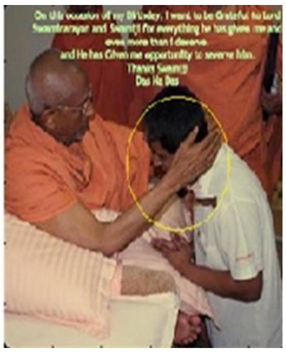
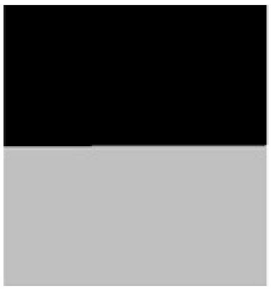
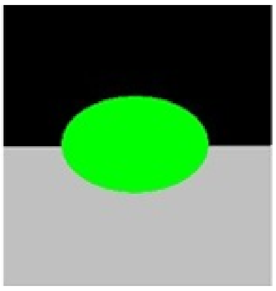
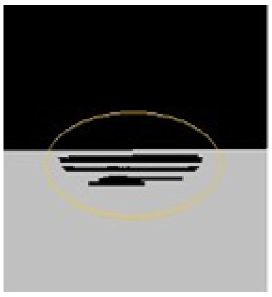

(D)

Figure 4: TEST Image2.(A)Input image (B)Inpainted image(C)Results of Criminisi's (D)Our results.


(A)


(B)


(C)

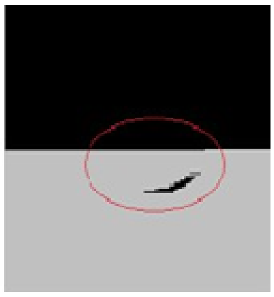

(D)

Figure 5: TEST Image3. (A)Input image (B)Inpainted image(C)Results of Criminisi's (D)Our results.

Table 1. PSNRs OF DIFFERENT IMAGES

No. Of Images	Algorithm of [10]	Proposed Algorithm
Figure 3	53.8855	53.9556
Figure 4	48.2207	50.9038
Figure 5	58.5687	58.6565

6. CONCLUTION

The method proposed by criminisi's [13] includes a similarity measure based only on color is insufficient to propagate accurate linear structures into the target region, and leads to garbage growing. The improved method in which image in paint performs accurately by modified distance function a new term G represents using image gradient as a similarity metric. In this method by using gradient as similarity metric accuracy of the image inpainting result can be increased. This algorithm can be used to remove objects from the image in a way that it seems reasonable to the human eye and provide effective image inpainting result and also enhance the performance.

7. ACKNOWLEDGMENTS

The authors would like to thank to Hetal Patel, Ankur Patel for inspiring discussions; and G. Sapiro, M. Bertalmio, D. Capel, A. Criminisi, M.S.Ishi, Prof. Lokesh Singh, Prof. Manish Agrawal for making some of their images available.

8. REFERENCES

[1] Ankur patel, Shashwat kumar and Ankit parajapati, "Analysis of Exemplar based Image inpainting,"International Journal of Computer Sciences and Information Technology. Vol.5 (1), 2014, pp.800-804.

[2] Bertalmio, Vese L, Sapiro G, Osher S. Simultaneous structure and texture image inpainting," IEEE Transactions on Image Processing, 2003, 12, 882-889.

[3] Rane S, Sapiro G, Bertalmio M. Structure and texture filling of missing image blocks in wireless transmission and compression applications. In IEEE. Trans. Image Processing,2002.

[4] Chan T, Shen J. Local inpainting models and TV inpainting, SIAM Journal on Applied Mathematics, 2001, 62,1019-1043.

[5] Chan T, Shen J. Non texture inpainting by curvature-driven diffusions, Journal of Visual Communication and Image Representation, 2001, 4, 436-449

[6] Heeger DJ, Bergen JR. Pyramid-Based Texture Analysis/Synthesis. In proceedings Of ACM Conf. Comp. Graphics (Siggraph),Los Angeles,Ca, 1995, 29, 229-233

[7] Cheng W, Hsieh C, Lin S, Wang C, Wu J. "Robust algorithm for exemplar based image inpainting"in Proceedings of International Conference on Computer Graphics, Imaging and Visualization, 2005, 64- 69.

[8] Wong, Orchard J. A nonlocal means approach to exemplar-based inpainting, in Proceedings of the 15th IEEE International Conference on Image Processing, 2008,2600-2603

[9] Z. Xu and S. Jian, "Image inpainting by patch propagation using patch sparsity," IEEE Transactions on Image Processing, Vol. 19, 2010, pp. 1153-1165.

[10] Alexei A. Efros and Thomas K. Leung, "Texture Synthesis by Non- Parametric Sampling", IEEE International Conference on Computer Vision, 1033–1038, 1999.

[11] A. Wong and J. Orchard, "A nonlocal-means approach to examplar-based inpainting," presented at the IEEE Int. Conf. Image Processing,2008

[12] W. Cheng, C. Hsieh, S. Lin, C. Wang, and J. Wu, "Robust algorithm for exemplar-based image inpainting," in Processing of International Conference on Computer Graphics, Imaging and Visualization, 2005, pp. 64-69.

[13] A. Criminisi, P. Pérez, and K. Toyama, " Region filling and object removal by exemplar-based image inpainting," IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, Sep.2004.

[14] Zhou, J. & Kelly, A. R. 2010. Image Inpainting based on local optimization. International Conference on Patteren Recognitions(ICPR).

[15] Oliviera, B. Bowen, R. Mckenna, and Y.-S. Chang. Fast Digital Image Inpainting. In Proc. Of Intl. Conf. On Visualization, Imaging And Image Processing (VIIP), Page 261266,2001.