

Image Inpainting by Optimized Exemplar Region Filling Algorithm

Shivali Tyagi, Sachin Singh

Abstract— This paper discusses removing objects from digital images and fills the hole that is left behind. Here, we present a novel and effective algorithm that combines the advantages of these two approaches. We first note that exemplar-based texture synthesis contains the essential process required to replicate both texture and structure; the success of structure propagation, however, is highly dependent on the order in which the filling proceeds. The existing algorithms are combined to improve the efficiency for finding the line association in selected region (like solid objects). Main focus is on data term and confidence term to find line association in selected region which is to be inpainted. The region filling is done from that line associated to other section in selected region.

Keywords— Confidence term, data term, exemplar-based texture, image inpainting, region based algorithm synthesis.

I. INTRODUCTION

Most problems of any image are often corrupted by noise, scanned old photo paper, dust or stains resting on the scanning glass of a scanner, scratched images or others have logos or stamps. Reconstruction of missing parts or scratches of digital images is an important field used extensively in artwork restoration. The algorithms of image denoising and image deblurring do not apply to image inpainting, because the regions to be inpainting are usually large and the information is distorted, whereas in these algorithms the pixels contain both information about the actual data and noise.

The restoration can be done by using two approaches, image inpainting and texture synthesis, whereas the meaning of first approach is restoring of missing and damage parts of images in a way that the observer who doesn't know the original image cannot detect the difference between the original and the restored image. The second approach is filling unknown area on the image by using surrounding texture information or from input texture sample. This technique could be employed to restore digitized photographs especially if a damaged area needs to be filled with some pattern or structure. However texture synthesis usually fails if the area to be reconstructed contains an additional colour or intensity gradient.

Most algorithms combine both texture synthesis and inpainting approaches to restores the image. Both are collectively used to fill holes as they remove unwanted features or holes in the image.

The basic requirement for all image inpainting algorithms is that the region to be inpainted should be selected manually by user, because no mathematical equation is capable of detecting or knowing the region to be inpainting without taking desired area.

In the inpainting based on partial differential equation, the goal was to maintain the angle of arrival [1]. The basic idea was the smooth propagation of information from the surrounding areas in the isophotes direction. The drawback of this method was that the CPU time required for inpainting depends on the size of selected region. Therefore, was a time consuming process as it took nearly 8 to 10 minutes for inpainting.

In inpainting by total variance and curvature-driven diffusion methods proposed by Chan and Shen [2], they used Euler-Lagrange equation and inside the inpainting domain the model simply employs anisotropic diffusion based on the contrast of the isophotes. This was used for inpainting small regions and was also good in removing noise but it did not connect broken lines or edges. The CDD model extended the TV algorithm by taking also geometric information of isophotes while defining strength of the diffusion process, allowing large area inpainting. The major drawback of the TV inpainting model was that it was unable to restore well a single object when its disconnected remaining parts were separated far apart by the inpainting domain.

A. Telea proposed a fast marching algorithm that can be looked as PDE based approach without the computational overheads [4]. It is fast and simple to implement than other PDE based methods, this method produced very similar result comparable to other PDE methods. The main limitation of this method is the blurring produced when inpainting regions thicker than 10—15 pixels, especially visible when sharp isophotes intersect the region's boundary almost tangentially.

Oliveira proposed a fast image inpainting method based on convolution operator [3]. This algorithm produces the result in few seconds with both the diffusion kernels after more than 100 iterations and faster than any image inpainting algorithms but produce blurring.

Exemplar-Based algorithm combines the strengths of both approaches into a single, efficient algorithm [5]. The work decomposes the original image into two components; one of which is processed by inpainting and the other by texture synthesis. The output image is the sum of the two processed components.

Limitations in this are: the synthesis of regions for which similar patches do not exist does not produce reasonable results, the algorithm is not designed to handle curved structures, finally, this algorithm does not handle depth ambiguities (*i.e.*, what is in front of what in the occluded area?).

Manuscript received on January 2013.

Shivali Tyagi, Computer Science and Engineering, SRM University, Ghaziabad, India.

Sachin Singh, Computer Science and Engineering, SRM University, Ghaziabad, India.

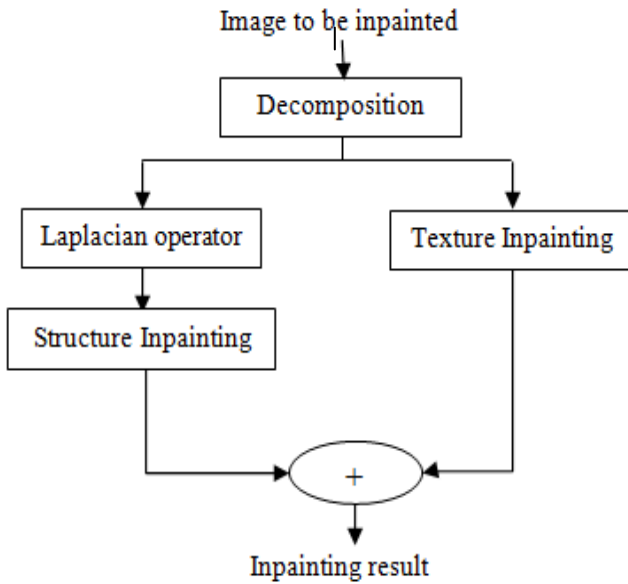


Fig. 1: Exemplar based inpainting flowchart

II. PROPOSED METHODOLOGY

In our inpainting technique, ‘I’ represents the original image. ‘Ω’ represents the target region, i.e. the region to be inpainted. ‘Φ’ represents the source region, i.e. the region from which information is available to reconstruct the image.

Generally, $\Phi = I - \Omega$. Also, we use ‘δΩ’ to represent the boundary of the target region, i.e. the fill front. It is from here that we find some patch that is to be filled.

An exemplar based inpainting algorithm involves the following steps:

- i. **Initialize the target region.** This is generally performed separately from the inpainting process and requires the use of an additional image processing tool. This is performed by marking the target region in some special color. Without any loss of generality, let us consider that the color that the target region will be marked in is green (i.e. R = 0, G = 255, B = 0).
- ii. **Find the boundary of the target region.**
- iii. **Select a patch from the region to be inpainted.** The patch size should be a bit larger than the largest distinguishable texture element in the image. We have used a default patch size of 9 x 9 which can be changed with the knowledge of the largest texture element in the image. We denote the patch by ψ_p .
- iv. **Find a patch from the image which best matches the selected patch, ψ_p .** This matching can be done using a suitable error metric. We use the Mean Squared Error (please refer eq. 1) to find the best matching patch.

$$MSE = \sum (f_{x,y} - g_{x,y})^2 / N \quad (2)$$

Where, $f_{x,y}$ represents the element of the patch ψ_p and $g_{x,y}$ represents the elements of the patch for which MSE is to be calculated. N is the total number of elements in the patch.

- v. **Update the image information** according to the patch found in the previous step.

A. Extract pixel information from selected region

First, we considered the image which is to be inpainted. Then (manually) select the region to be inpainted. Using equation 2 we extracted pixel values with minimum and maximum mean square error.

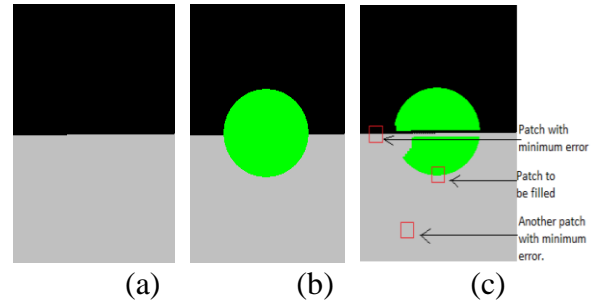


Fig. 2: Information extraction from selected region

In the above fig. (a) is the original image to be inpainted. (b) is region that is to inpainted in the image is selected manually and (c) tells the patches with same mean square error. Selecting the incorrect patch may not produce the most visually plausible result.

B. Compute data and confidence term

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. The priority computation is biased toward those patches which: (i) are on the continuation of strong edges and (ii) are surrounded by high-confidence pixels. Given a patch Ψ_p centered at the point p for some $p \in \partial\Omega$ (see fig. 3), we define its priority $P(p)$ as the product of two terms:

$$P(p) = C(p) * D(p) \quad (3)$$

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

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}, \quad D(p) = \frac{|p \nabla \perp n_p|}{\alpha} \quad (4)$$

Where, $|\Psi_p|$ is the area of Ψ_p , α is a normalization factor (e.g., $\alpha = 255$ for a typical grey-level image), n_p is a unit vector orthogonal to the front $\partial\Omega$ 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 $C(p) = 1 \forall p \in I - \Omega$.

At a coarse level, the term $C(p)$ approximately enforces the desirable concentric fill order. As filling proceeds, pixels in the outer layers of the target region will tend to be characterized by greater confidence values, and therefore be filled earlier; pixels in the centre of the target region will have lesser confidence values.

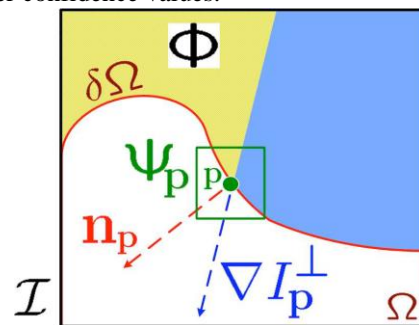


Fig. 3: Representations in an image

In representing the image, the patch is Ψ_p , n_p is the normal to the contour $\delta\Omega$ of the target region Ω and $I(1/p)$ is the isophotes (direction and intensity) at point p.

The entire image is denoted with I .

$D(p)$ is the data term that signifies the strength of the gradient function at that point. The data term is based on the product of isophotes and a unit vector orthogonal to the front of an inpainted area. The data term gives a high priority to inpainted area which has a potential to copy structural information from the source area.

Data term is the function of strength of isophotes hitting the front $\partial\Omega$ at iteration. It boots the priority of a patch in which an isophotes flows into. It also tells which pixel has to be filled first into the target region (i.e., of highest priority). This factor is of fundamental importance in our algorithm because it encourages linear structures to be synthesized first, and, therefore propagated securely into the target region.

In Fig. 4, we can see the pixels as red dots. These pixels are of higher priority and will be filled first with respect to other pixels in the selected region. Therefore, data term is used to find the line association in the selected region. The inpainted is done from this line association towards other section in the selected region. So, we can say that the pixels lying on the line association have higher priority than other pixels in selected region.

The confidence term $C(p)$ may be thought of as a measure of the amount of reliable information surrounding the pixel p . The intention is to fill first those patches which have more of their pixels already filled, with additional preference given to pixels that were filled early on (or that were never part of the target region). Patches that include corners and thin tendrils of the target region will tend to be filled first, as they are surrounded by more pixels from the original image. These patches provide more reliable information against which to match. Conversely, patches at the tip of "peninsulas" of filled pixels jutting into the target region will tend to be set aside until more of the surrounding pixels are filled in.

The confidence term favors out-pointing regions (e.g., sharp bands) in the inpainted area. Thus, smooth or near circular inpainting area results. Pixels in the centre of the target region will have lesser confidence values.

The confidence term summarizes the percentage of useful pixels in a patch. Structural information is not included. For instance, a vertical line and a horizontal line in two different patches (with the same number of useful pixels) could have the same confidence term. We believe the use of confidence term only indicates how much information in a target patch can be used as a reference while we look for similar patches in the source area.

If only the data term is used in the priority, however, edges end up propagating where they shouldn't. It is the harmony between the two factors that creates good results. In this vein, both quantities are normalized (to lie between 0 and 1) by appropriate factors. The data term and the confidence term together results in a balance to select the best patch to be inpainted.

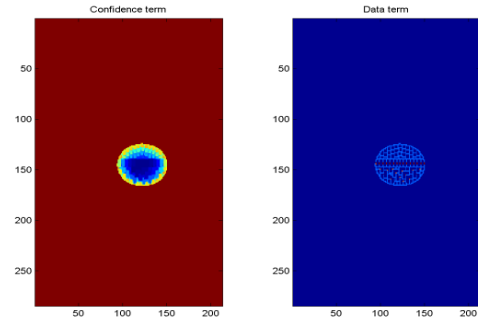


Fig. 4: Confidence Term and Data Term

The above figures tell the pixels with confidence term and data term in the selected region that is to be inpainted.

C. Matrix development

We had used improved Oliveira technique with modified convolution method [7].

The steps for performing the modified algorithm are same as Oliveira method but the difference lies in the position of zero in the weighted average filter. In this modified convolution method there is difference in filling, it takes the image after the region to be inpainted selected and convolves the region with averaging filter having a zero weight at the bottom right corner instead of the center. This change helps in reducing the number of repetition of algorithm to be applied on the image to one. This reduction in repetition avoids blurring of that portion of the image that is convolved.

Only one convolution iteration that convolve the masked region with the kernel

a	b	a	c	c	c
b	a	b	c	c	c
a	b	0	c	c	0

Fig. 5: Modified Oliveira inpainting kernels.

The Fig. 5 shows modified Oliveira method Pseudo code and two diffusion kernels used with the algorithm. $a = 0.073235$, $b = 0.176765$, $c = 0.125$

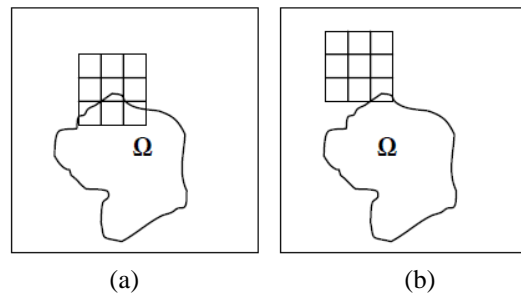


Fig. 6: The Inpainting process

In the above figure we show the different process for inpainting. In (a) the inpainting is done using Oliveira method and in (b) the inpainting is done by proposed method.

D. Applying inpainting algorithms

We use region filling algorithm for filling the pixels in the selected region for inpainting.

Steps of Region filling algorithm:

- (i) Extract the manually selected initial front $\delta\Omega^0$.
- (ii) Repeat until done:
 - Identify the fill front $\delta\Omega^t$. If $\Omega^t = \emptyset$, exit.



- Compute priorities $P(p) \forall p \in \delta\Omega^i$.
- Find the patch Ψ_{p^*} with the maximum priority, i.e. $\Psi_{p^*} | p^* = \arg \max_{p \in \delta\Omega^i} P(p)$.
- Find the exemplar $\Psi_{q^*} \in \Phi$ that minimizes $d(\Psi_{p^*}, \Psi_{q^*})$.
- Copy image data from Ψ_{q^*} to Ψ_{p^*} .
- Update $C(p) \forall p | p \in \Psi_{p^*} \cap \Omega$.

There are various approaches used in region filling such as Seed filling approach and Scan filling approach.

Following is the explanation of each method:

a. Seed filling approach

This approach works at pixel level. This is therefore suitable for interactive inpainting applications.

In seed filling approach we have two types of connectedness, such as:

- 4 – connected region

From a given pixel, the region that we can get by a series of 4 way moves (east, west, north and south).

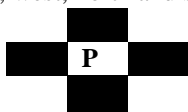


Figure 7: 4-way connectivity

- 8 – connected region

From a given pixel, the region that we can get by a series of 8 way moves (east, west, north, south, north-east, north-west, south-east and south-west).

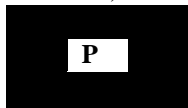


Fig. 8: 8-way connectivity

In seed filling approach we also have two types of algorithm for filling the pixel values in the region, such as:

- Boundary fill Algorithm:

It starts at a point inside a region and paints the interior outward towards the boundary. The boundary is specified in a single color. It can be used for both 4 connected or 8 connected region.

- Flood fill Algorithm

This algorithm is used when an area is defined with multiple color boundaries. It also starts with the point inside the region and replaces a specified interior color (old color) with fill color. It can also be used for both 4 connected and 8 connected approach until all interior points is being replaced.

b. Scan filling approach

This approach works at polygon level therefore gives better performance. Polygon edges are sorted according to their minimum / maximum Y. Scan lines are processed in increasing (upward) / decreasing (downward) Y order. When the current scan line reaches the lower / upper endpoint of an edge it becomes active. When the current scan line moves above the upper / below the lower endpoint, the edge becomes inactive. Active edges are sorted according to increasing X. Filling the scan line starts at the leftmost edge intersection and stops at the second. It restarts at the third intersection and stops at the fourth. We can also fill the polygon by maintaining a data structure of all intersections of polygons with scan lines.

III. RESULT

To demonstrate the effectiveness of the proposed method, we have applied our present method to several images with 200×200 pixels in MATLAB. We used a PC with Core i7 3.47GHz CPU and 12GB memory.

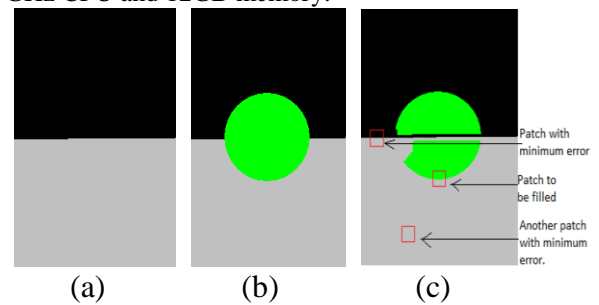


Fig. 9: Information extraction from selected region

In the above figure, (a) is the original image to be inpainted. (b) is region that is to inpainted in the image is selected manually and (c) tells the patches with same mean square error. Selecting the incorrect patch may not produce the most visually plausible result.

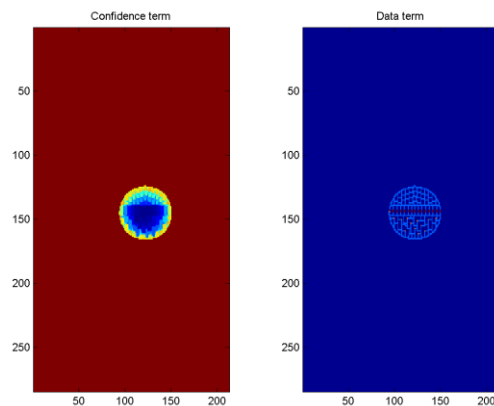


Fig. 10: Confidence Term and Data Term

The fig. 10 shows the pixels with confidence term and data term in the selected region that is to be inpainted.

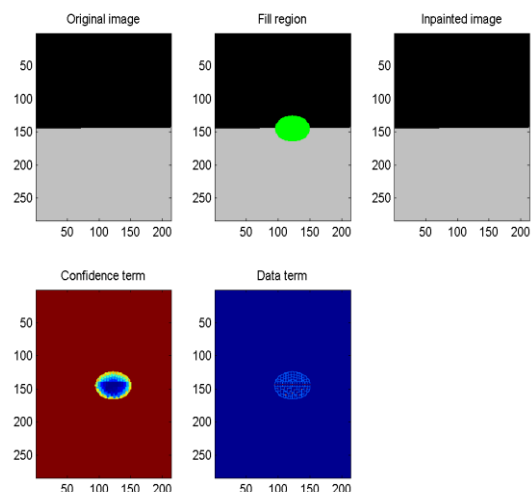


Fig. 11: Inpainted image

The above figures show the original, selected as well as inpainted image with data and confidence term.

IV. CONCLUSION

This paper presents a novel and effective algorithm that combines the advantages of these two approaches. We first note that exemplar-based texture synthesis contains the essential process required to replicate both texture and structure; the success of structure propagation, however, is highly dependent on the order in which the filling proceeds. The existing algorithms are combined to improve the efficiency for finding the line association in selected region (like solid objects). Main focus is on data term and confidence term to find line association in selected region which is to be inpainted. The region filling is done from that line associated to other section in selected region.

In this paper the sets of line associatively is low. The number of lines associated is less by the process used in this project. The future work can be done in increasing the number of lines that can be associated. The association of lines in a mesh region is complicated using the proposed method; therefore future work can be done in improving it.

REFERENCES

1. Marcelo Bertalmio et al "Image inpainting", proceeding of SIGGRAPH 2000 computer graphics processings pp. 417-424.
2. T. Chan and J. Shen. "Non-Texture Inpainting by Curvature- Driven Diffusions (CDD)." Technical Report CAM 00-35, Image Processing Research Group, UCLA, 2000.
3. M. Oliveira , B. Bowen ,R. Mckenna ,Yu-Sung Chang "Fast digital image inpainting" in proceeding VIIP 2001, pp. 261-266, [PUB], 2001.
4. A. Telea "An image inpainting technique based on the fast marching method" journal of graphics tools, v. 9, no. 1, acm press 2004.
5. A. Criminisi, P. Perez and K. Toyama "Region Filling and Object Removal by Exemplar-Based Image Inpainting", in proceeding vol. 13, no. 9, 2004.
6. Arjan Kuijper "Image Processing with Geometrical and Variational PDEs" in processing OEAGM07 pp. 89-96, 2007.
7. Mohiy M. Hadhoud, Kamel. A. Moustafa and Sameh. Z. Shenoda "Digital Images Inpainting using Modified Convolution Based Method" in proceeding SPIE 7340, 73400S, 2009.
8. Muthukumar S, Dr. Krishnan N, Pasupati P, Deepa S "Analysis of Image Inpainting Techniques with Exemplar, Poisson, Successive Elimination and 8 Pixel Neighborhood Methods", in proceeding vol. 9, no. 11, 2010.

AUTHORS PROFILE

Shivali Tyagi, received B.Tech degree in Computer Science & Engineering from Uttar Pradesh Technical University in 2010 and M.Tech degree in Computer Science & Engineering from SRM University, NCR Campus, in 2012. She is currently working as an Assistant Professor in the Department of Computer Science at Dr. K. N. Modi Engineering College, Modinagar.

Sachin Singh, received B.Tech degree in Computer Science & Engineering from Uttar Pradesh Technical University in 2010 and M.Tech degree in Computer Science & Engineering from SRM University, NCR Campus, in 2012.