IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 4 (Nov. - Dec. 2013), PP 43-52
www.iosrjournals.org

Comparative Study and Analysis of Image Inpainting Techniques

Bagadiya Vishal ,Prof.B.A.dixit
(Information Technology, MIT Collage of Engineering / Pune University, India)

Abstract: *Image inpainting is a technique to fill missing region or reconstruct damage area from an image.It removes an undesirable object from an image in visually plausible way.For filling the part of image, it use information from the neighboring area. In this dissertation work, we present a Examplar based method for filling in the missing information in an image, which takes structure synthesis and texture sysnthesis together. In exemplar based approach it used local information from an image to patch propagation.We have also implement Nonlocal Mean approach for exemplar based image inpainting.In Nonlocal mean approach it find multiple samples of best exemplar patches for patch propagation and weight their contribution according to their similarity to the neighborhood under evaluation. We have further extended this algorithm by considering collaborative filtering method to synthesize and propagate with multiple samples of best exemplar patches. We have to preformed experiment on many images and found that our algorithm successfully inpaint the target region.We have tested the accuracy of our algorithm by finding parameter like PSNR and compared PSNR value for all three different approaches.*

Keywords: *Texture Synthesis, Structure Synthesis, Patch Propagation ,imageinpainting ,nonlocal approach, collabrative filtering.*

I. Introduction

In real world, many people need a system to recover the damaged photographs, artwork, designs, drawings etc. Damage may be due to various reasons like scratches, overlaid text or graphics, scaled image etc. Nowadays, powerful photo-editing tools are available for retouching, drawing, and removing object by scissors from images. But, to fill the missing information or reconstruct damage area in an image is still difficult task.

This system could enhance and return a good looking photograph using a technique called image inpainting. Image inpainting modify and fill the missing area in an image in an undetectable way, by an observer not familiar with the original image [1][2]. The technique can be used to reconstruct image damage due to scratches, to remove dates and titles etc. from image.

This method starts with original image and mask image as input. Here mask image specify the object to be removed from the original image. The object to be removed has to be marked by user because it depends on the subjective choice of user. And give the output as reconstructed image. Image inpainting is different from other general image enhancement algorithms in the sense that image enhancement assumes that pixel in the damaged portion of image, contain both the information about real data and the noise, while in image inpainting, the pixel values are all assumed to be missing in the filling domain.

The data exchange through network or use of wireless network increases day to day, this data are in the form text, image, audio, video etc., hence the need for an automatic and fast technique to restore image blocks lost during transmission. Producing stunning special effects in an image also involves lot of image inpainting for removal of artifacts. And some photographs have scratch or distortion. These tasks are conventionally performed manually and requiring lot of time and skills. These problems have motivated us to search for an automatic technique.

Many applications are benefited from image inpainting technology, some of the application can be

Object Removal: This technique can remove small or big any object specified by user from image in visual plausible way.
Stain Image Reconstruction: Stain image can be easily reconstructed by applying inpainting algorithm on the stain part of image.

Correction of Images Corrupted Due to Transmission Error: In wireless transmission there are chances of a loss of image blocks that can be restore by considering the lost part as the inpainting domain.

Scratch Removal: Scratch can be removed from image by applying the inpainting algorithm on the part of the image containing scratch.

Producing Stunning Visual Effects in Image: Special effects such as a bungee jumper diving without a rope can be produced.

II. Theoretical Background

2.1 Digital Image Processing

An image may be define as a two dimensional function f(x, y), where x and y are spatial coordinates, and the value of f at any pair of coordinate (x, y) is called the intensity of the image at that point of the image. The term gray level is used often to refer to the intensity of the monochrome images. Color images are often formed by the combination of separate 2-D images e.g. in RGB color system a color image consist of red, green and blue component images. This is why, many of the technique developed for monochrome images can be extended to color images by processing the three component images separately.

An image may be continues with respect to the x- and y- coordinate, and also in amplitude. To convert such an image in digital from requires that the coordinate as well as the amplitude be digitized. Digitizing the coordinate's value is called sampling while digitizing the amplitude values is called quantization. Thus, when x, y and the amplitude value in an image, are all finite, discrete quantities, it is called a digital image.

2.1.1 Representing Digital Image

Assume that an image f(x, y) is sampled so that the resulting digital image has M rows and N columns as shown in figure 2.1. The values of the coordinates (x, y) now become discrete quantities. These co-ordinates are representing using integer values Thus, the values of the coordinates at the origin are (x, y) = (0, 0). The next coordinate values along the first row of the image are represented as (x, y) = (0, 1). It is important to keep in mind that the notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled.

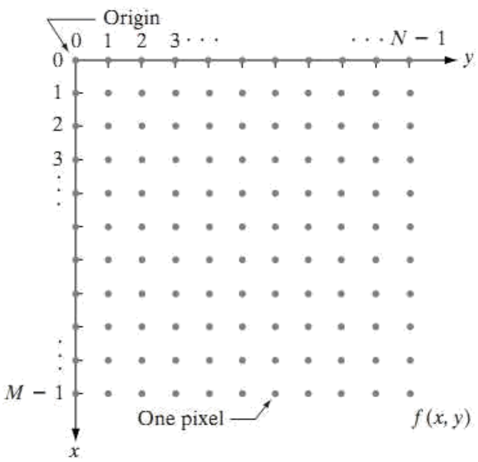


Fig 2.1 Coordinate Convention Used to Represent Digital Images

2.2 Technique for image inpainting

There are two types of techniques for image inpainting:

- Texture Synthesis
- Structure Synthesis

Both these approaches have different direct applications. Structure synthesis is used explicitly for filling holes in an image and texture synthesis is used to create textured pattern, which has extensive application in 3D animation. However, they are methods used to synthesis a pixel given some information about another set pixels. Texture synthesis problem, which requires an input texture can be thought as reducing to the inpainting problem, if we assume that the input texture which it tries to replicate lies in the same image where the region to be synthesized lies. The two approaches can be collectively referred to as hole filling approaches because they try to remove unwanted objects from an image and fills the hole left behind.

Hence, for two dimensional images, one can use both texture synthesis and structure propagation to restore the image but the result produced by an individual technique may not be suited to all kinds of images. A robust method for image inpainting should be able to synthesize structure as well as texture in images.

Technique implemented in this dissertation uses an approach which combines structure propagation with texture synthesis and hence produces very good results. A short introduction of texture synthesis and structure synthesis is given in next two sessions.

2.2.1 Texture Synthesis

Texture synthesis has been an active research topic in computer vision both as a way to verify texture analysis methods, as well as in its own right. Potential applications of a successful texture synthesis algorithm are broad, including lossy image and video compression, occlusion fill-in, foreground removal, etc.

Texture can be classified as either regular (consisting of repeated texels) or stochastic (without explicit texels). However, almost all real world texture lies somewhere between these two extremities and requires to be captured with a single model.

Texture synthesis involves synthesizing an image which matches the appearances of a given texture. The new image may be of arbitrary size and one of the fundamental goals of texture synthesis is that the synthesized image should appear to be generated by the same underlying process as the original image.

2.2.2 Structure Synthesis

Structure synthesis means to fill-in the missing information in such a way that isophote lines arriving at the region's boundaries are completed inside. These methods allow for simultaneous filling-in of multiple regions containing completely different structure and surrounding backgrounds. If structure synthesis is done using PDE based methods than it introduce blur in the image.

III. Implementation Methodology

3.1 Basics of Exemplar based approach

Exemplar based approaches perform well for two dimensional texture as well as with liner image structure. Figure 3.1 shown the missing regions i.e. target region or inpainting domain is denoted by **Ω** and its boundary also specify and the source region is denoted by Φ, remains constant throughout the algorithm and provides sample used in the filling process.

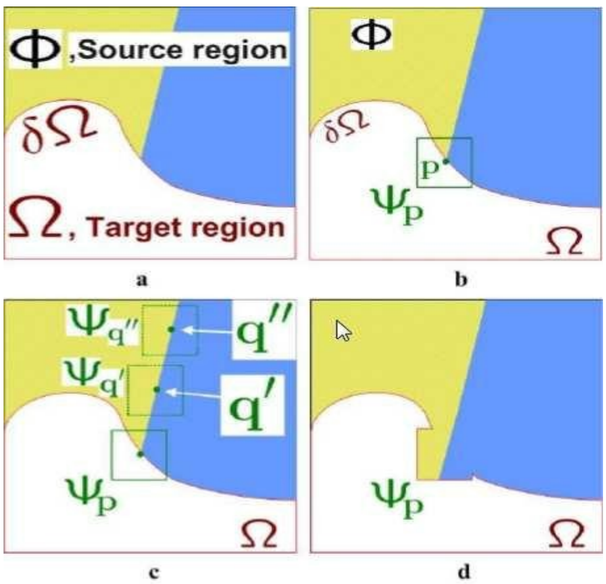


Fig 3.1. An illustration to the exemplar-based inpainting algorithm.

Figure 3.1(a) indicates the original image with the target region **Ω** and the sources region Φ. It also indicates the boundary of the target region δ**Ω**. Figure 3.1(b) shown patch Ψ_p selected on the boundary of the target region which has highest priority. Figure 3.1(c) shown the best match patch for $\Psi_{p.}$ It is find using sum of square of difference method. These missing pixels in Ψ_p are propagated by corresponding pixels in Ψ_q shown in figure 3.1(d). Illustrates that the best matching patch in the candidates set has been copied into the position occupied by Ψ_p, thus achieving partial filling of **Ω**. See that both texture and structure have been propagated inside the target region. Repeat this process until target region fill. The target region **Ω** has, now, shrunk and its front boundary has a new shape now.

We now focus on the single iteration of the algorithm to show how structure and texture are adequately handled. Patch Ψ_p two parts, one belonging to target region **Ω** and other belonging to source region Φ. Only that part which belongs to target region Ψ_p is to be filled because remaining part is already containing the information. From figure 3.1(d), we can say that both structure and texture has been preserved.

3.2 Exemplar Based Approach Using Search Space Window

First, given an input image, the user selects the object to be removed. This step requires user interaction because object to be removed depends on the subjective choice of the user. The part of the image from where the object is to be removed is known as target region or inpainting domain Ω. The sources region Φ is entire image minus the target region. The size of the template window must be specified. This can be 9 x 9 pixels, but in practices required the user to set it to be slightly larger than the largest distinguishable texture element, or "texel", in the source region [6].

In addition to this, user also needs to specify the size of the search window shown in figure 3.2. Use of search window improves execution time because it reduces the searching time for finding a best match patch later in the algorithm. The size of search window depends on the region to be filled and the kind of structure and texture in image.

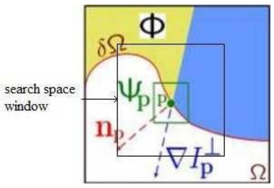


Fig 3.2. Search Space Window

Once these parameters are specifying, the region- filling proceeds automatically. During the algorithm, patches along the fill-front are assigned a temporary priority value, which determines order in which they are filled. Then the algorithm iterates following three steps until all pixels have been filled.

3.2.1Computing Patch Priorities

In the first step, a best edge patch Ψ_p is picked out using priority [6]. This algorithm uses best- first filling strategy that entirely depends on the priority values which are assigned to each patch on the fill-front. The priority computation is biased toward those patches which (1) are on the continuation of strong edges and (2) are surrounded by high-confidence pixels.

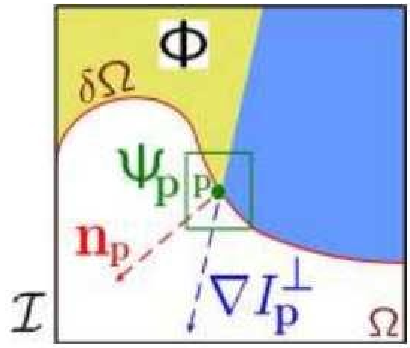


Fig 3.3 Notation diagram

Here target region is indicating by Ω and its boundary is indicating by δΩ, source region is indicating by Φ. Given a patch Ψp centered at the point P for some P€δΩ isshown in figure 3.3. Priority P (p) is defined as the product of two terms.

$$P(p) = C(p)\,D(p)$$

Here, C (p) is the confidence term and D (p) is the data term. They are defined as follows.

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Psi_{\mathbf{p}} \bigcap (\mathcal{I}-\Omega)} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|} \qquad D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha}$$

Where |Ψp| is the area of Ψp, α is a normalization factor (e.g., α=255 for a typical grey-level image), np is a unit vector orthogonal to the fill-front δΩ in the point P, and
denotes the orthogonal operator. The priority 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 ,∀p∈Ω and C (p) = 1, ∀p∈ − Ω.The confidence term C (p) may be thought of as a measure of the amount of reliable information surrounding the pixel P. The idea is to fill first those patches which have more of their pixels already filled. This automatically incorporates preference toward certain shapes of the fill-front. For example, 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.

The data term boost the priority of the patch in which a liner structure flows into.This term is very important because it allows broken lines to correct.

3.2.1Finding Best Match Patch

Once priority is finding for all patches on boundary then take patch Ψp which has highest priority for filling this patch first. We first find the patch Ψq in search window which is most similar to patch Ψp. The most similar patch Ψq is the one which has the minimum difference in the pixel value with patch Ψp. Difference between any two pixels p and q given by using sum of squared difference (SSD) method. It define as

$$SSD(\Psi_p, \Psi_q) = \sum_{i=1}^{M} \mu_i (\Psi_p(i) - \Psi_q(i))^2$$

Where, Ψp (i) and Ψq (i) are the i-th pixel value in respective patches. M is the size of the patch. μi is pixel mask function. An exemplar patch Ψq is a patch with the lowest SSD value. Which is define as

$$\Psi_q = \min_{\Psi_q \in \Phi} SSD(\Psi_p, \Psi_q)$$

Above equation give the patch which is most similar to the patch Ψp in the image which has minimum SSD value Here SSD method takes color value of two pixels for difference.

3.2.3 Copying Best Match Patch And Updating Confidence Values

Once the patch Ψp is filled with new pixel value Ψq, confidence value in the area is updated as follows. C (q) = C (p) for all q belonging to Ψp ∩ Ω.This simple update rule allows us to measure the relative confidence of patch on the fill front. After completion of these three steps then update boundary with updated target region and repeat these three steps until all the pixels in the target region not fill.

Algorithm

Input:

Original Image- It is an image which needs to be inpainted.

Mask Image- This image specifying the object to be removed or the regions to be inpainted. The user marks the object to be removed with white color and other region with black color. Using this image mark the object with red color in original image. This object which is marked in red is removed in visual plausible way.

Patch window size- this parameter gives the size of the patch around the pixel which is compared in the search space, to find a suitable match.*Search space window*- this window limit the search space to a limited area, thereby eliminating the need of searching the suitable patch in whole source region. This improves execution time.

Output:

Inpainted image- Output image include with the removal of object specify in mark image using inpainting algorithm in visual plausible way.

Steps of algorithm are given below:

Step1. Initialize mark variable for all pixel. If pixel belongs to inpainting region set mark variable with 0 else set 1.

Step2. Find boundary of region to be inpaint, if boundary is “empty set” than exit.

Step3. Find priority for all patches on the boundary.

Step4. Select the patch which has highest priority, call that patch, P.

Step5. Find the patch from search window which is best match to patch P, call that patch,Q.

Step6. Copy pixels of patch Q to the patch P, update only those pixels of patch P which has mark value 0 and set mark variable to1, go to step2.

3. 3 Nonlocal-Means Approach

Main issues with the current approach to exemplar-based inpainting is the fact that they use image information from only a single neighborhood. They do not fully exploit content redundancy in an image and, thus, “put all their eggs in one basket”[7].They proposed approach of exemplar-based inpainting, in that approach to use image information from multiple samples within the image and weight their contribution according to their similarity to the neighborhood under evaluation. This concept of weighted aggregation of nonlocal information has proven effective for the purpose of image denoising. Picking only one exemplar patch

Ψq to propagate may lead to mistakes. Thus, Wong and Orchard[7] picked n best non-local exemplar patches Ψqi (i=1,2,...,n).The number of best exemplar patches is fixed in Wong and Orchard's approach. But, it is changeable in Sun and Xu's approach [8], to remove unnecessary exemplar patches.Assuming n best exemplar patches Ψqi (i=1,2,...,n) are picked out, The weight of n best non-local exemplar patches as

$$w(\Psi qi) = e^{-\frac{SSD(\Psi p, \Psi qi)}{h}}$$

A normalized linear combination coefficient is defined as αi

$$\alpha_i = \frac{w(\Psi qi)}{\sum_{j=1}^{n} w(\Psi qj)}$$

At last, Ψq is expressed by synthesizing the n best non-local exemplar patches Ψqi as

$$\Psi_q(x) = \sum_{i=1}^{n} \alpha_i \cdot \Psi_{qi}(x)$$

Where, x is pixel position in the patch. After finding Ψq fill target region of patch Ψp with synthesized patch Ψq. This approach propagates missing pixels in Ψp with counterpart pixels in the synthesized exemplar patch Ψq=ΣαiΨqi. However, propagated missing pixels in Ψp may not be well integrated with known pixels in Ψp.
In this regard X. Wu, W. Zeng and Z. Li[9] proposed the collaborative filtering method to synthesize and propagate with the n best exemplar patches. It focuses on the mean deviation value Δ between known pixels and current filling missing pixel in Ψp.When filling missing pixels in Ψp, although Δ is calculated by information in exemplars Ψqi, the important fundamental value is based on known pixels in Ψp. Therefore, with this propagation method, unreasonable filling result can be alleviated.

Algorithm

Steps of algorithm are given below:
Step1. Initialize mark variable for all pixel. If pixel belongs to inpainting region set mark variable with 0 else set 1.
Step2. Find boundary of region to be inpaint, if boundary is "empty set" than exit.
Step3. Find priority for all patches on the boundary.
Step4. Select the patch which has highest priority, call that patch, ΨP.
Step5. Find the n non-local exemplar patches from search window which is best match to patch ΨP, call that patch, Ψqi, where i=1, 2…, n.
Step6. Calculate the weight of n best non-local exemplar patches w(Ψqi). Using weight of all exemplar patches find normalized linear combination coefficient αi.
Step7. Ψq patch is expressed by synthesizing the n best non-local exemplar patches Ψqi as Ψq=ΣαiΨqi , where i=1, 2,…, n.
Step8. Copy pixels of patch Ψq to the patch ΨP, update only those pixels of patch ΨP which has mark value 0 and set mark variable to1, go to step2.

3. 4 Collaborative Filtering Method

An online rating-based Collaborative Filtering query consists of an array of (item,rating) pairs from a single user. Output of that query is an array of predicted (item,rating) pairs for those items the user has not yet rated. It is the process of filtering for information or patterns by using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. So many collaborative filtering algorithms are used in e-commerce applications such as item rating system. Because the mathematical prototype of synthesizing exemplar patches to propagate is similar to item rating (matrix completion), collaborative filtering algorithms can be introduced into exemplar-based propagation.

Slope one is the simplest form of non-trivial item-based collaborative filtering based on ratings. Since this algorithm is a simple and efficient online collaborative filtering algorithm. In D. Lemire and A. Maclachlan[10][11] approach slope One algorithms work on the intuitive principle of a "popularity differential" between items for users. In a pairwise fashion they determine how much better one item is liked than another. One way to measure this differential is simply to subtract the average rating ofthe two items. In turn, this difference can be used to predict another user's rating of one of those items, given their rating of the other.This method is use in exemplar approached Table 3.1 shows an example to serialize pixels in a two dimensional patch to an array.

Table 3.1. Pixel Notation In A Serialized Patch Ψp

p_0^1	p_0^2	...	p_0^m
p_0^{m+1}	p_0^{m+2}	...	p_0^{2m}
...	...	...	...
$p_0^{m^2-m+1}$	$p_0^{m^2-m+2}$	...	$p_0^{m^2}$

Assuming the k-th pixel in Ψp is unknown marked by * in Table 3.2, pixel arrays of Ψp and Ψqi (i=1,2,...,n) are arranged into a matrix as Table 3.2. P_0^j is the value of j-th pixel in Ψp and P_i^j is the value of j-th pixel in Ψqi,where i ≠ 0.

Table 3.2. Pixel Arrays Of Ψp And Ψqi.

Pixel	1	2	...	k	...	M
Ψp	p_0^1	p_0^2	...	p_0^k *	...	p_0^M
Ψq1	p_1^1	p_1^2	...	p_1^k	...	p_1^M
Ψq2	p_2^1	p_2^2	...	p_2^k	...	p_2^M
...	...	...	...	...	...	...
Ψqn	p_n^1	p_n^2	...	p_n^k	...	p_n^M

The mean deviation value between known pixel j and missing pixel k in Ψp is

$$dev_{j,k} = \sum_{i=1}^{n} \alpha_i \cdot (p_i^k - p_i^j)$$

Where, αi is normalized linear combination coefficient defined in above method.In this method for finding k value we take same row of all exemplar patches and find deviation value for all pixel of that row. If all pixel of that row of Ψp patch is of target region than this method not give correct output so through experiment we implement hybrid approach for that row pixels value find using nonlocal mean approach. The pixel value of missing pixel k is obtained by

$$p_0^k = \frac{\sum_{j=1}^{m} \mu_j \cdot (p_0^j + dev_{j,k})}{\sum_{j=1}^{m} \mu_j}$$

Update µk to 1 and then fill another unknown pixel in Ψp by the same method until there is no unknown pixel in Ψp.

Algorithm

Steps of algorithm are given below:

Step1. Initialize mark variable for all pixel. If pixel belongs to inpainting region set mark variable with 0 else set 1.

Step2. Find boundary of region to be inpaint, if boundary is "empty set" than exit.

Step3. Find priority for all patches on the boundary.

Step4. Select the patch which has highest priority, call that patch, ΨP.

Step5. Find the n non-local exemplar patches from search window which is best match to patch ΨP, call that patch, Ψqi, where i=1, 2…, n.

Step6. Calculate the weight of n best non-local exemplar patches w(Ψqi). Using weight of all exemplar patches find normalized linear combination coefficient αi.

Step7. Suppose k is missing pixel in ΨP, if all pixels of k-th pixel row of Ψp is of target region than used nonlocal mean method otherwise take k-th pixel row of all exemplar patches and find mean deviation value between known pixel j and missing pixel k in Ψp.

Step8. Add this mean deviation value to know pixel value of Ψp patch and new value of pixel k is calculate using average of know pixel value of Ψp patch.

Step9. Update k-th pixel of patch ΨP with new value and set mark variable to1, if all pixels of Ψp patch fill go to step2 otherwise go to step7.

IV. Performance Results

Here we take output image comparison of three approach. First is exemplar based approach, In this approach take only one best exemplar patch to fill the target region.Second approach is Non local mean approach, In this approach take n number of best exemplar patch to fill target region. However, propagated missing pixels in target region may not be well integrated with known pixels. In this regard, the collaborative filtering method is proposed to synthesize and propagate with the n best exemplar patches.

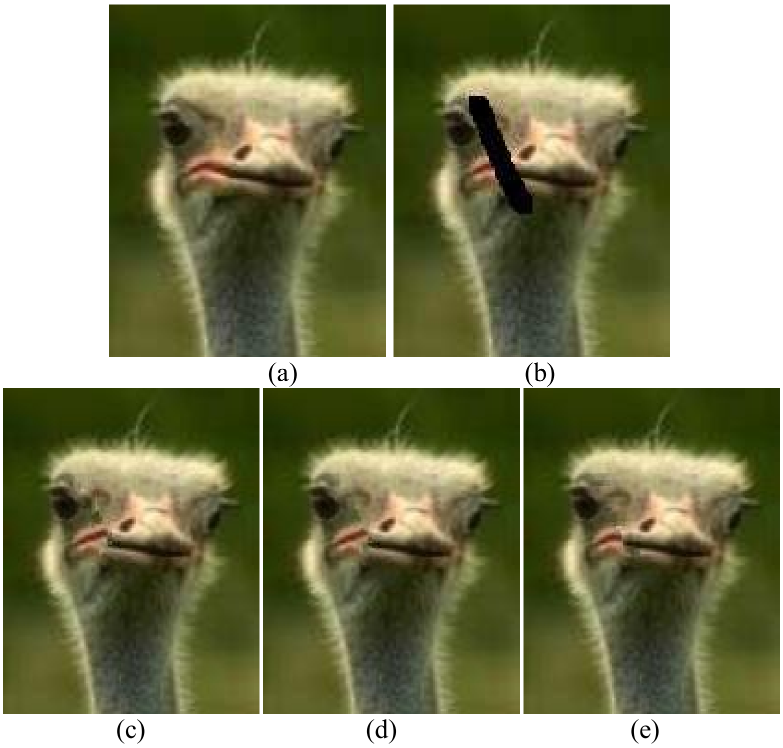


Fig 4.1 Stain Inpainting (a) Original Image, (b) show the Stain in Image (c) Inpainted result with Exemplar based approach (d) Inpainted result of Non local mean approach (e) Inpainted result of Collaborative Filtering Method.

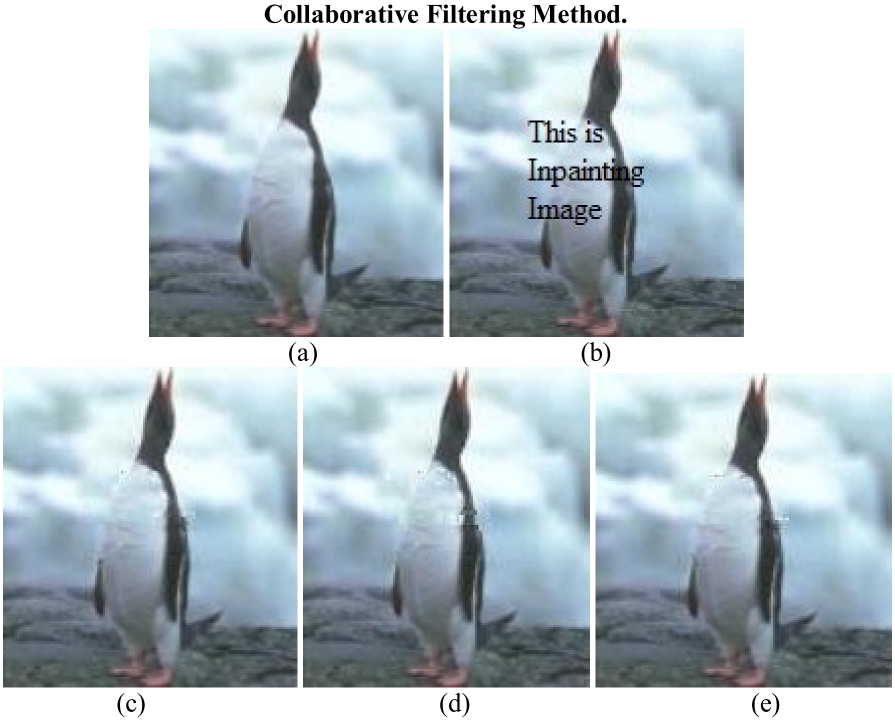
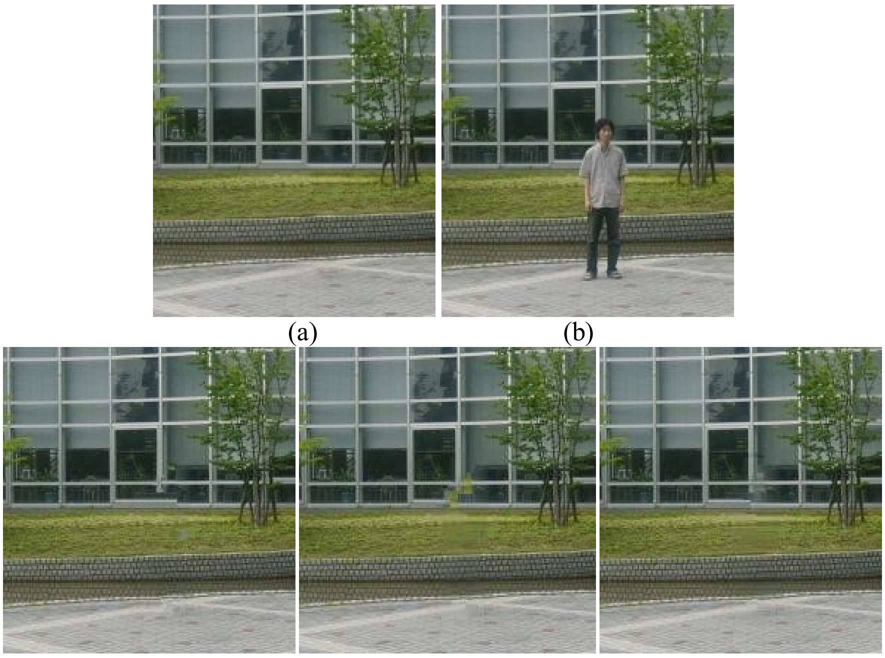


Fig 4.2 Text Inpainting (a) Original Image, (b) show the Text in Image (c) Inpainted result with Exemplar based approach (d) Inpainted result of Non local mean approach (e) Inpainted result of Collaborative Filtering Method.



(a) (b)

(c) (d) (e)

Fig 4.3 Object removal (a) Original Image (b) show the object to removed in Image (c) Inpainted result with one patch propagation (d) Inpainted result of Non local mean approach. (e) Inpainted result of Collaborative Filtering Method.

Table 4.1: PSNR value of above Results

	Exemplar based approach(PNSR)	Non Local Mean Approach(PNSR)	Collaborative Filtering Method(PNSR)
Fig1	32.8745	33.5952	34.5197
Fig2	38.2932	38.1266	38.3910
Fig3	34.5457	34.7436	35.2690

Table 4.1 give the comparisons of different method of Exemplar based approaches second Column show a PSNR value of output image given by Exemplar based approach, third Column show PSNR value of Result of Nonlocal mean approach and fourth column show the PSNR value using collaborative filtering Method. In above result number of best exemplar patches is set to 5.From above comparisons we can conclude that use of collaborative filtering method for inpainting improve the quality of output image.

V. Conclusion & FutureWork

5.1 Conclusion

Image inpainting is a technique to fill missing region or reconstruct damage area from an image. In this dissertation, we have implemented exemplar based approach for image inpainting. This technique considers structure propagation and texture synthesize together which reduced blur in inpainted image. It takes patches window from damage region for inpainting. In exemplar based approach to find best patch it search entire image. In our approach we have searched only in the predefined search window which reduced time complexity without effecting quality of the restored image. Using above experiment we have concluded that output image quality is depend on patch size as well as search window size. In exemplar based approach one sample of best exemplar patch is used to fill the missing information from image. It used local information for patch propagation. Picking only one exemplar patch to propagate may lead to mistakes. Nonlocal Mean approach for exemplar based image inpainting take multiple sample of best exemplar patch with their weight to synthesis target patch. It used nonlocal information is used to fill the target region.

This patch propagation method may give blurred output image in some result or it propagate missing pixels in target patch may not be well integrated with known pixel.Collaborative filtering method used to synthesize and propagate with multiple sample of best exemplar patches. Here we compare results of these three approaches and conclude that uses of collaborative filtering method improve the image quality.

5.2 Future Work

Digital inpainting algorithm aims to automate the process of inpainting, and therefore also need to minimizing the user interaction. This algorithm can be extended which detect inpainted region without user interaction. However, one kind of interaction which is impossible to eliminate is the selection of the inpainting domain because that depends on the subjective choice of the user. At present, algorithm does not work well enough with curved structures, which can be improved. The algorithm can be extended for the removal of moving objects from a video. This will require challenging task of object tracking to be implemented as well. The algorithm can also be extended for automated detection and removal of text in videos. Sometimes videos are inscribed with dates, titles etc which are not required. This kind of text can be automatically detected and removed from images and videos. The user will not have to give any mask image for this desired task.

References

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

[2] T. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions," J.Visual Comm. Image Rep., vol. 4, no. 12, pp. 436-449, 2001.

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. "Navier–Strokes, fluid dynamics, and image and video inpainting." in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, pp. 355–362, 2001.

[4] T. Chan and J. Shen. "Mathematical models for local non-texture inpaintings." SIAM J. on Appl. Math, vol. 62, No. 3, pp. 1019–1043, 2002.

[5] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. "Simultaneous structure and texture image inpainting," in Proc. Conf. Comp. Vision Pat-tern Rec., Madison, WI, pp.1-6, 2003.

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

[7] A. Wong and J. Orchard. "A nonlocal-means approach to exemplar-based inpainting."in Proc. the IEEE Int. Conf. Image Processing, pp. 2600-2603, 2008.

[8] Z. Xu and J. Sun. "Image inpainting by patch propagation using patch sparsity." IEEE Trans.Image Process, vol. 19, no. 5, p.p 1153-1165, 2010.

[9] Xinran Wu, Wei Zeng, Zhenzhou Li," Exemplar-Based Image Inpainting with Collaborative Filtering" in Proc. IEEE Computer Society Sixth International Conference on Image and Graphics, pp. 8–11,2011.37

[10] D. Lemire and A. Maclachlan. "Slope one predictors for online rating-based collaborative filtering." in Proceedings of the SIAM Data Mining Conference (SDM '05), 2005.

[11] D. Billsus and M. Pazzani. "Learning collaborative information filterings." in AAAI Workshop on Recommender Systems, 1998