

# Region Filling and Object Removal by Exemplar based Image Inpainting

Sayali Baheti<sup>1</sup> Payal Shah<sup>2</sup> Deeksha Jagtap<sup>3</sup> Amruta Pawar<sup>4</sup>

<sup>1,2,3,4</sup>Department of Computer Science & Engineering  
<sup>1,2,3,4</sup>PES MCOE, Pune, Maharashtra, India

**Abstract**— Image Inpainting is the art of restoring lost parts of image by reconstructing them using background information. Recently various approaches have been proposed. In the past, this problem has been addressed by two classes of algorithms: (i) “Texture Synthesis” algorithms for generating large image regions from sample textures, and (ii) “Structure Propagation” techniques for filling in small image gaps. In this paper we are providing an effective algorithm that contains the advantages of above two methods. The exemplar-based texture synthesis contains the essential process required to replicate both texture and structure. The existing algorithms are combined to improve the efficiency for finding the line association in selected region. Data term and confidence term are used to calculate the patch priority. With the help of priority term region filling is done that defines filling order of patches in the image. A number of examples on real and synthetic images demonstrate the effectiveness of our algorithm in removing large occluding objects as well as thin scratches.

**Key words:** Object Removal, Image Inpainting, Simultaneous Texture and Structure Propagation, Exemplar based Inpainting



(a) Original Photograph (b) Inpainted Photograph  
 Fig. 1: Removing Large Objects from Images

Though the techniques of texture synthesis are effective, they have difficulty in filling holes in photographs of real world scenes consisting of composite textures. On the other hand, the structure-oriented scheme obtains the missing region by propagating linear structure into the target region. Their drawback is that the while filing large regions, diffusion process introduces blur. The technique presented in this paper combines the strengths of both approaches into a single efficient algorithm. As with structure oriented technique we pay special attention to linear structures. But, linear structures in the target region only influence the fill order of is an exemplar based texture synthesis algorithm. In this paper we have proposed an efficient exemplar based Image Inpainting algorithm with an improved priority term that defines the filling order of patches in the image. The analysis of both theoretical and experimental results of exemplar based algorithms [6] provides a good framework for us to extend our contribution to this category. This idea stems from the texture synthesis technique proposed in [9] in which the texture is synthesized by sampling the best match patch from the known region.

The rest of the paper is organized as follows: Section 2 presents related work done in image Inpainting techniques, section 3 includes mathematical model, and section 4 includes a description of our proposed method and our region filling algorithm. The experiment results and conclusion are shown in section 5.

## I. INTRODUCTION

Removing objects and repairing damaged regions are very interesting and tedious task. 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. Image Inpainting is a technique for removing such undesired objects in images and reconstructing the missing regions in a visually plausible way. Reconstruction of missing parts or scratches of digital images is an important field used extensively in artwork restoration [3]. Now a days Image Inpainting has become a fundamental area of research in image processing. Image Inpainting has many applications in digital effect i.e. object removal, image editing, image resizing, image restoration e.g. text removal in photograph, transmission and image coding e.g. recovery of missing blocks. 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. These algorithms the pixels contain both information about the actual data and noise [5].

The conventional schemes for image Inpainting are divided into two categories

- 1) Texture oriented
- 2) Structure Oriented [1]

In the previous work, several researchers have considered texture synthesis as a way to fill large image regions with textures-repetitive two dimensional textural patterns [3-8]. Example of texture synthesis is the exemplar based technique.

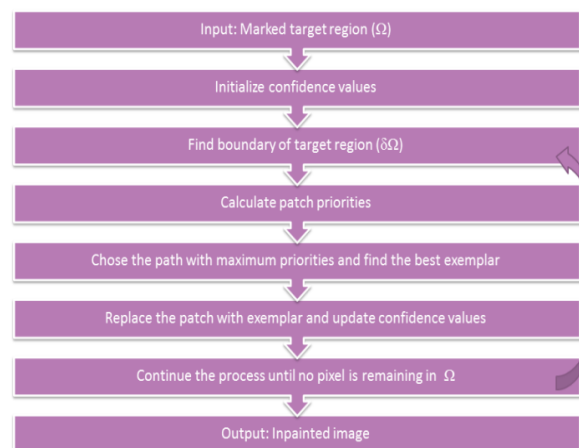


Fig. 2: Exemplar Based Inpainting Flowchart

## II. RELATED WORK

Currently there are very few accepted technologies/tools for carrying out the work of image inpainting. In the past decade there has been considerable work concerned with Image Inpainting. Image Inpainting is still in the beginning stage and many researches are being carried out in this area. There are some softwares and libraries available for inpainting purpose. E.g. "restoreInpaint" [12] is an open source library which provides functionalities to detect and automatically restore cracks etc. from damaged images. Another software is "Photo-Wipe" [11] by Hanov Solutions. It provides tools for selecting the target region and then provides options to carry out the inpainting process with varying time and quality.

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

There are mainly three categories of image inpainting algorithms. First category of algorithms is for restoring films/videos, but this is not very useful for image inpainting as there is limited information of various frames. Another class of algorithms deals with the reconstruction of textures from the image (e.g. [4]). These algorithms utilize samples from the source region to rebuild the image. The third class of algorithms tries to rebuild the structural components of an image such as edges and object contours etc.

The authors of paper [1] presented a pioneering work in this respect. It was able to recover most of the structural features from the image but failed while recovering very large regions. Another algorithm proposed in paper [10] involved the use of mask to achieve inpainting. The mask that they choose for inpainting is decided interactively and requires user intervention. They prepare the mask such that the center element in the mask is zero. This means that no information about a pixel is extracted using its own. But this algorithm also works only for small regions and cannot inpaint large regions in the image.

Authors in [4] proposed an inpainting algorithm to fill in holes in overlapping texture and/or cartoon image synthesis. Their algorithm is a direct extension of morphological component analysis that is designed to separate linearly combined texture and cartoon. There have been a very few algorithms that utilize the advantages of both the image inpainting methods i.e. the structure recreation and texture synthesis algorithms. One such work was proposed in the paper by Criminisi et al. [3]. They proposed a pioneering approach in this field that combined structural reconstruction approach with the texture synthesis approach in one algorithm by combining the advantages of both approaches.

In this paper, we propose an extension to earlier inpainting algorithms with a focus on improving the computational complexity of the approaches along with some other improvements such as speed and accuracy

## III. MATHEMATICAL MODEL

$IList = getImage(iRegion, iImage);$

Here  $iList$  represent the list of the images to be Inpainted. The function  $getImage$  was used to get the proper list of images, the function passes two variables  $iRegion$  which means inpainted region and  $iImage$  which contains Inpainted images.

$I \leftarrow (R,C,P,E);$

Here,

- $I$  represents the inpainting process,
- $R$  represents the inpainting region,
- $C$  represents the Confidence value
- $GP$  represents the Generate Patches.

And  $E$  represents the Exemplar.

Set Theory Analysis:

- 1) Identify the Regions  $R = \{r_1, r_2, r_3, \dots, r_n\}$   
Where 'R' is main set of parts inside the region like  $r_1, r_2, r_3, \dots$
- 2) Identify the Confidence terms  $C = \{c_1, c_2, c_3, \dots, c_n\}$   
Where 'C' is main set of Confidence terms like  $c_1, c_2, c_3, \dots$
- 3) Identify the Patches,  $P = \{p_1, p_2, p_3, \dots, p_n\}$   
Where 'P' is main set of Patches like  $p_1, p_2, p_3, \dots$
- 4) Identify the Exemplar,  $E = \{e_1, e_2, e_3, \dots, e_n\}$   
Where 'E' is main set of Exemplar like  $e_1, e_2, e_3, \dots$
- 5) Identify failure cases as FL. Failure occurs when -  
 $FL = \{F_1, F_2, F_3, \dots, F_N\}$ 
  - $F_1 = \{f_i^j f_i^k\}$  if Improper input Image Format }
  - $F_2 = \{f_i^j f_i^k\}$  if Select Improper region for inpainting }
- 6) Identify success case SS:-  
Success is defined as-
  - $SS = \{S_1, S_2, S_3, S_4, S_5\}$
  - $S_1 = \{s_i^j s_i^k\}$  if Gives proper input image format }
  - $S_2 = \{s_i^j s_i^k\}$  if Select proper region for inpainting }
  - $S_3 = \{s_i^j s_i^k\}$  if Select Proper Confidence Value }
  - $S_4 = \{s_i^j s_i^k\}$  if Select proper exemplar to replace patch }
  - $S_5 = \{f_i^j f_i^k\}$  if Save proper output image format }

7) Initial conditions as  $I_0$

Users have to give Proper input image format. User have to Select proper image region for inpainting.

8) Inpainting algorithm is NP Complete problem

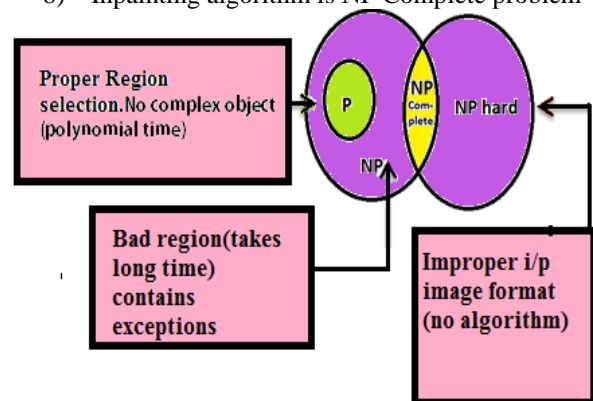


Fig. 3: Computational Complexity of Inpainting Algorithm

#### IV. METHODOLOGY

Adopting notations similar to that used in Criminisi's inpainting algorithm [3], the region to be filled i.e. the target region or masked region is indicated by  $\Omega$  and its contour is denoted by  $\delta\Omega$ . As shown in fig. 3 the contour evolves inward as the algorithm progresses and so it is also called as "fill front". The source region  $\Phi$  which remains fixed throughout the algorithm, provides samples used in the filling process. Formally we could express the inpainting problem in this way: Given an image  $I$  with a target region  $\Omega$ , fill in each pixel inside  $\Omega$  with a pixel value taken from  $\Phi$ . Suppose that the square template  $\Psi_p \subset \Phi$  centered at point  $p$  (fig 4) is to be filled. The best match sample comes from source region patch  $\Psi_q \subset \Phi$ . In the e.g. in fig 5, we see that if  $\Psi_p$  lies on the continuation of an image edge, the most likely best matches will lie along the same edge.

An exemplar based inpainting algorithm involves the following steps:

- 1) 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.  $R = 0, G = 255, B = 0$ ).
- 2) Find the boundary of the target region.
- 3) 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  $9 \times 9$  which can be changed with the knowledge of the largest texture element in the image. We denote the patch by  $\psi_p$ .
- 4) Find a patch from the image which best matches the selected patch,  $\psi_p$ . This matching can be done using a suitable error metric. We use the Mean Squared Error to find the best matching patch.

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

Where,  $f_{x,y}$  represents the element of the patch  $\psi_p$  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.

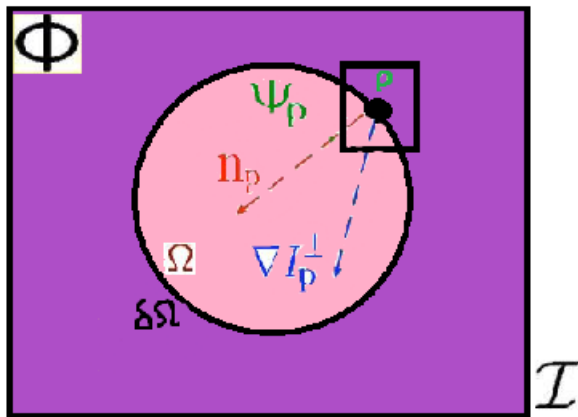


Fig. 4: Image Inpainting Geometry

- 5) Update the image information according to the patch found in the previous step.

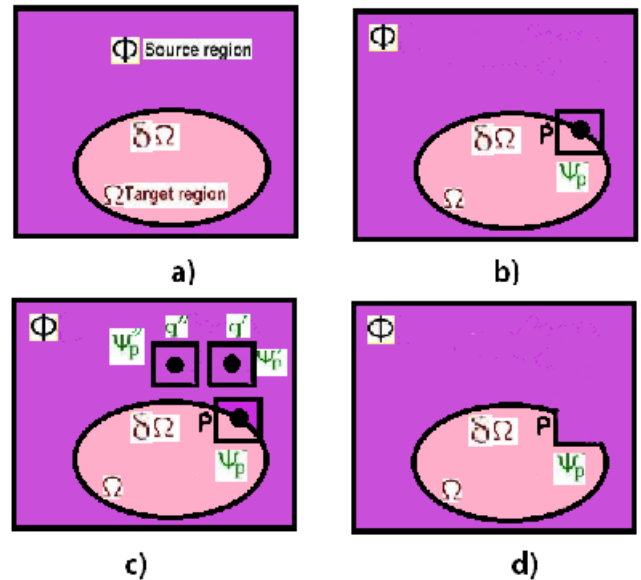


Fig. 5: Illustrations Of The fill-In Propagation By Region-Based Synthesis. (A) Original Image Is Divided Into Source Region (Composed Of Green And Yellow Areas) And (White Area) Target Region. (B) Patch  $\Psi_p[X,Y]$  Centered In  $P[X,Y]$  On The fill Front  $\delta\Omega$ . (C) Most Similar Patch  $\Psi_q[X,Y] \in \Phi$ . (D) Inpainting Result For  $\Psi_p[X,Y]$ .

As mentioned earlier, the result does depend considerably on the third step wherein a patch is selected to be inpainted. The result that we obtain would almost always depend on the selection order and thus there have been approaches that try to define this selection order so that the result is improved.

##### A. Computing Patch Priorities and Patch Selection:

In the procedure of patch selection, patch priority should be defined to encourage the filling-in of patches on the structure with highest priority. Confidence term can serve as a measure of the amount of reliable information surrounding the pixel point  $p$  in an image. When this point is surrounded by many pixels from the original image, the more reliable information from the source region is contained within the patch that is matched. In Criminisi et al's algorithm, the best patch can be selected by using the priority function  $P(p)$  which is defined as the sum of two factors.

$$P(p) = C(p) * D(p).$$

Instead of using this multiplicative definition, we have proposed the addition of weights to the priority term as  $P(p) = a * RC(p) + b * D(p)$  for  $0 \leq a, b \leq 1$  and  $a + b = 1$

Where  $a$  is the component weight for  $C(p)$  and  $b$  is the component weight for  $D(p)$ . The regularized confidence term  $RC(p)$  is denoted as

$$RC(p) = (1 - \omega) * C(p) + \omega \text{ for } 0 \leq \omega \leq 1$$

Where  $\omega$  is the regularizing factor for controlling the curve smoothness. By changing the priority for the patches on the  $\delta\Omega$  now we are able to find a patch  $\psi_p$  with the highest priority. Here, we use a default mask size of  $9 \times 9$  pixel as an exemplar region. We will compute the priority of each pixel on the boundary from the target region, and then according to the level of the priority order to sequentially fill in the patch information. Finally, the priority of the boundary in the target region will be updated.

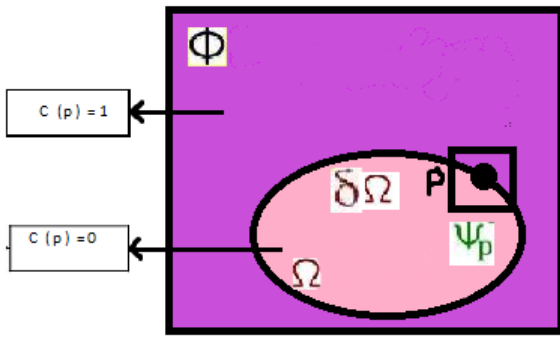


Fig. 6: Boundary Confidence Value Set.

### B. Search for the Best Matched Patch:

Most often, the patch that most resembles the selected patch lies very close to the patch selected to be inpainted. Based on this assumption, we provide an approach on how to reduce the computational complexity of the algorithm.

For the patch  $\psi_p$  with highest priority, we would find the best matched patch of it in  $\Phi$ . To increase the robustness of the similarity between exemplar patch and matching patch based on the square of Euclidean distance we design a new similarity matching function. The diameter of the surrounding region to search is calculated at run time by taking into account the region to be inpainted.

We search for the best exemplar from a rectangle defined by (P1, P2) and (Q1, Q2). These coordinates can be found as follows:

- $P1 = \max(0, p - pc/2 - cpr - MDx/2)$
- $P2 = \max(0, p - pr/2 - cpc - MDy/2)$
- $Q1 = \min(\text{width}, p + cpr/2 + cpr + MDx/2)$
- $Q2 = \min(\text{height}, p + cpc/2 + cpc + MDy/2)$

The diameter of the search region is calculated at run time.  $pr$  and  $pc$  are the number of rows and columns in the patch.  $MDx$  and  $MDy$  are constants that represents minimum diameter for the X and Y directions respectively.  $cpr$  and  $cpc$  are the number of continuous green (which has been used as target mask) pixels. In the process of searching for the best matching patch, we found out that our method could improve the inpainting speed and ensure the matching accuracy at the same time.

### C. Propagation of Exemplars:

When all priorities of the target region have been computed, we find the patch  $\Psi_p$  with highest priority. Then, we fill it with the source region  $\phi$  which is the most similar to  $\Psi_p$ . The exemplar based propagation is conducted to recover the missing pixels in the selected patch  $\Psi_p$ . Several exemplar patches  $\Psi_{qi}$  are selected and missing pixels in  $\Psi_p$  are propagated by synthesizing corresponding pixels in  $\Psi_{qi}$ . The exemplar patches are selected from the sum of squared difference (SSD) measure between  $\Psi_p$  and a patch  $\Psi_q$  in  $\Omega$  as

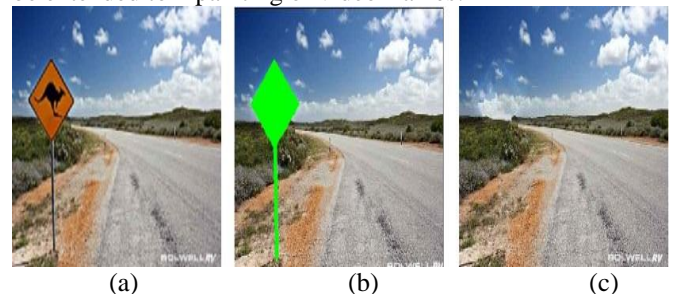
$$SSD(\psi_p, \psi_q) = \sum_{i=1}^M (\psi_p(i) - \psi_q(i))^2$$

When the patch  $\psi_p$  has been filled with pixel values of the source block, the relative confidence of patches on the fill front is change. We must update the confidence  $C(p)$ . We restore the lost area by copying the known image in the corresponding position. However, the inpainting process would change some pixels' state, such as the original point

on the boundary may become known pixel, the original point in  $\Omega$  may become a new boundary point and so on. In order to do the inpainting in a correct order, it is necessary to update the boundary data, confidence information in time. Finally, the priority of the boundary in the target region will be updated.

## V. EXPERIMENTAL RESULTS AND CONCLUSION

In this section, we evaluate the proposed image inpainting algorithm on a variety of natural images. The algorithm is applied to both images as shown in following figures. Figure 8f shows how our filling algorithm achieves the best structural continuation in a simple, synthetic image. But for the special cases if there are many small details in the target region selected by the user, then user can change the patch size of the filling propagation. By increasing patch size, the time to inpaint the input image will decrease but the textural properties won't be accurate. Figure 7 shows how our filling algorithm achieves the best structural continuation in a simple, synthetic image. We applied the proposed algorithm to inpaint the missing region after object removal and we display some results as shown in Fig 7 also results for text n scratch removal are shown in Fig 8. Removal of date stamp is another application of image inpainting which is shown in Fig 9. We present a novel algorithm that can remove objects from the image in a way that it seems reasonable to the human eye. Our approach extends an exemplar based inpainting method along with a priority term that defines the filling order in the image. This approach is capable of propagating both linear structures and two dimensional textures into the target region. The result of object removal is an image in which the selected object has been replaced by a visually plausible background that mimics the appearance of the source region. This is robust algorithm for example based image inpainting, which can be adapted to any image contents of different characteristics. One important parameter of the algorithm is the size of the patch. With bigger patch size, the filling rate is high, thus the program runs faster. However, there's more important implications on choosing the right patch size. While this paper addresses inpainting of still photographs, it can also be extended to inpainting of video frames.





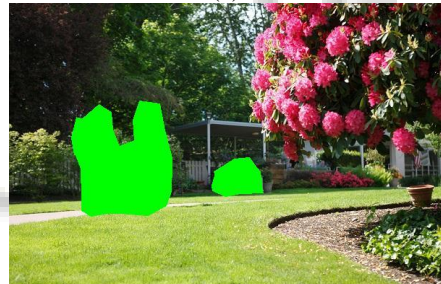
(b)



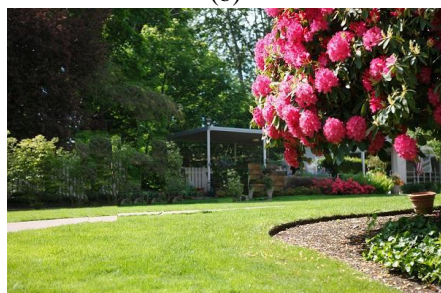
(C)



(a)



(b)

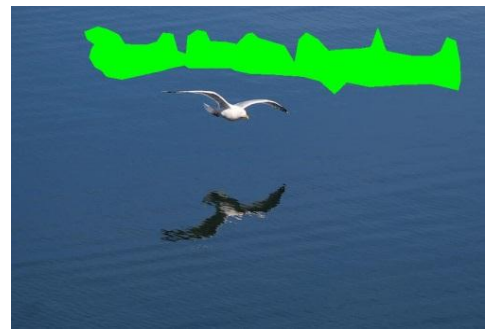


(c)

Fig. 7: Object Removal: A) Original Image To Be Inpainted B) Target Area Selection For Inpainting C) Inpainted Image



(a)



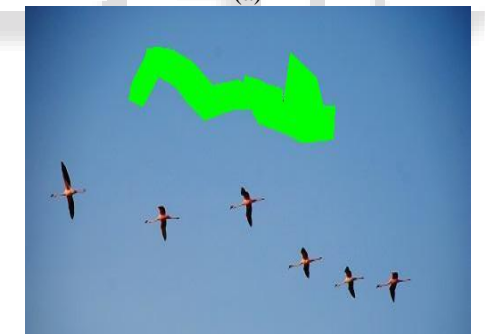
(b)



(c)



(a)



(b)



(c)

Fig. 8: Text And Scratch Removal: A) Original Image To Be Inpainted B) Target Area Selection For Inpainting C) Inpainted Image

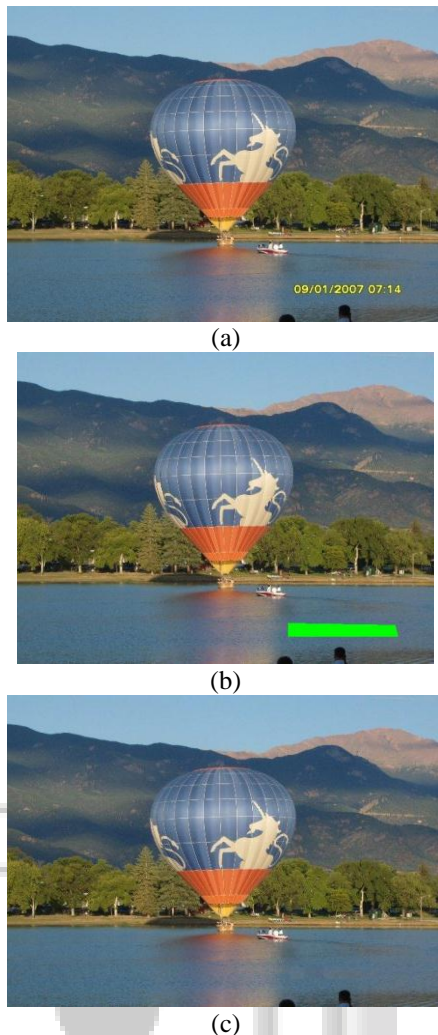


Fig. 9: Date Stamp Removal: A) Original Image To Be Inpainted B) Target Area Selection For Inpainting C) Inpainted Image

## VI. ACKNOWLEDGEMENTS

We express our deepest gratitude to our project guide, Prof. Bhagwan Phulpagar, for providing all the material possible, providing valuable suggestions and encouraging throughout the project work.

## REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," Proceedings of the 27th annual conference on Computer graphics and interactive technique, 417-424, 2000.
- [2] M. Burger, H. Lin, and C.B. Schonlieb, "Cahn-Hilliard Inpainting and a Generalization for Grayvalue Images," UCLA CAM report, 08-41, 2008.
- [3] A. Criminisi, P. Perez, and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," IEEE Transactions on Image Processing, 13(9), 1200-1212, 2004.
- [4] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho, "Simultaneous Cartoon and texture image inpainting using morphological component analysis (MCA)," Journal on Applied and Computational Harmonic Analysis, 340-358, 2005.

- [5] P. Elango, and K. Murugesan, K, "Digital Image Inpainting Using Cellular Neural Network," Int. J. Open Problems Compt. Math., 2(3), 439-450, 2009.
- [6] M.J. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using Sparse Representations," The Computer Journal, 64-79, 2009.
- [7] G. Forbin, B. Besserer, J. Boldys, and D. Tschumperle, "Temporal Extension to Exemplar-Based Inpainting applied to scratch correction in damaged image sequences," Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2005), Benidorm, Espagne, , 1-5, 2005.
- [8] R.C. Gonzalez, and R.E. Woods, Digital Image Processing, 2nd ed. Pearson Education, 2002.
- [9] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Interpolation of missing data in image sequences," IEEE Transactions on Image Processing 11(4), 1509-1519, 1995.
- [10] M.M. Oliveira, B. Bowen, R. McKenna, and Y.S. Chang, "Fast Digital Image Inpainting," Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP2001), Marbella, Spain, 261-266, 2001.
- [11] Photowipe, <http://www.hanovsolutions.com/?prod=PhotoWipe>
- [12] Restore Inpaint, <http://restoreinpaint.sourceforge.net/>
- [13] C.B. Schonlieb, A. Bertozzi, M. Burger, and H. Lin, "Image Inpainting Using a Fourth-Order Total Variation Flow," SAMPTA'09, Marseille, France, 2009.
- [14] T. Shih, et al., "Video inpainting and implant via diversified temporal continuations," Proceedings of the 14th annual ACM international conference on Multimedia, 133-136, 2006.
- [15] Studio Lighting, <http://www.studiolighting.net/>
- [16] Wen-Huang Cheng, Chun-Wei Hsieh, Sheng-Kai Lin, Chia-Wei Wang, and Ja-Ling Wu, "Robust Algorithm for Exemplar-Based Image Inpainting," The International Conference on Computer Graphics, Imaging and Vision (CGIV 2005), Beijing, China, 64-69, 2005.
- [17] A. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling," Proceedings of IEEE International Conference on Computer Vision, Greece, 1033-1038, 1999.
- [18] Q. Chen, Y. Zhang and Y. Liu, "Image Inpainting With Improved Exemplar-Based Approach," Multimedia Content Analysis and Mining, LNCS, vol. 4577/2007, pp.242-251, 2007