Multimode Automatic Image Inpainting

Shyni Shajahan

PG Scholar,Dept. Of Computer Science & Engg,TKM Institute of Technology,Kollam,Kerala
shyni.shajahan@gmail.com

Abstract- Image Inpainting is the art of filling in missing data in an image. The purpose of inpainting is to rebuild missing regions in a visually conceivable manner so that it seems reasonable to the human eye. There have been several methods proposed for the same. In this paper, an algorithm that improves and extends a previously proposed algorithm and provides faster inpainting. Using this method, one can inpaint large regions (e.g. remove an object etc.) as well as recuperate small portions (e.g. restore a photograph by removing crashes etc.). The inpainting is based on the exemplar based method. The basic idea behind this method is to find areas from the image and replace the lost data with it. This technique can be used in renovating old photographs. It can also eliminat eoverlaid script like dates, subtitles etc.; or entire objects from the image like microphones or wires to produce special effects and obtained good quality results quickly using this method.

Keywords— Inpainting, Exemplar, Priority, Enhanced, Object removal.

IJSER

1. INTRODUCTION

Inpainting is the art of repairing lost parts of an image and rebuilding them based on the background information. This has to be done in an unnoticeable way. The term inpainting is derived from the ancient art of restoring image. Digital Image Inpainting tries to replicate this process and perform the inpainting automatically. Fig 1 shows an example of this method where a building (manually selected as the target region) is replaced by information from the remaining of the image in a visually conceivable way. The algorithm automatically does this in a way that it looks “reasonable” to the human eye. Details that are hidden/ obstructed completely by the object to be removed cannot be recovered by any mathematical method. An objective for image inpainting is not to recuperate the original image, but create image that has a close similarity with the original image.

Such software has several uses. One use is in repairing photographs.With time, photographs get damaged and scratched.

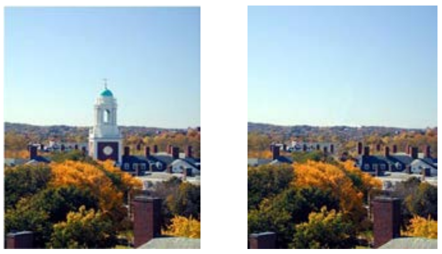


Fig 1: Removing objects using Image Inpainting. (i) The original image [15], (ii) Image with the building removed.

Another use of image inpainting is to removing unwanted objects from the image. Unwanted objects such as microphones, ropes, some unwanted person and logos, stamped dates and text etc. in the image. During the diffusion of images over a network, there may be some parts of an image that are missing. These parts can then be rebuilt using image inpainting. There have also been a few studies on image inpainting for super-resolution and zooming of images [6].

The rest of the paper is prepared as follows. Section II describes the related work in this area. Section III presents key observations and shortcomings of the earlier methods. This is followed by a description of how to improve the inpaintingmethod. Experimental outcomes and Conclusion & Future work then follow in sections IV and V respectively.

2. RELATED WORKS

Currently there are very few accepted tools for carrying out the work of image inpaintingand a lot of researches are being carried out to explore this area. There are a few software products and libraries existing for this intention. E.g.,restoreInpaint [12] is an open source library which provides functionalities to detect and automatically renovatecrashes etc. from damaged photographs. Software currently available for this task is named Photo-Wipe [11]. It provides tools for selecting the region to be inpainted and then provides several options to carry out the inpainting process.

An algorithm may seem to be something similar to noise removal from images. De-noising is focused towards modifying individual pixels whereas inpainting aims at modifying larger regions from the image. De-noising also differs from inpainting in the way that in inpainting there is no information about the image in the region to be inpainted as opposed to noise removal where pixels may contain information about both the real data and noise [1].

Most inpainting methods work as follows: The user selects the region to be inpainted. The image restoration is then carried out automatically. In order to produce a visually Conceivable restoration, an inpainting method must try to renovate the isophotes (i.e. the lines of equal grey values) as smoothly as possible and also proliferate two dimensional textures. Based on these two requirements, the inpainting algorithms are classified as:

There are mainly three classes of algorithms employed for inpainting.First class of algorithms is for restoring films/videos.Another class of algorithms deals with the reconstruction of textures from the image. These algorithms utilize samples from the source region to rebuild the image. Using this method, most of the texture of the image can be rebuilt. The third class of algorithms tries to rebuild the structural features such as edges and object shapes etc.The paper [1] presented ainnovative 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. They prepare the concealment such that the centre element in the mask is zero. This means that no information about a pixel is removed using its own value. It uses the values of its neighboring pixels to determine its value. But this algorithm also works only for small regions .

Another algorithm for recovering small regions and noise in an image is proposed in paper [5]. It caninpaint images with very high noise ratio. Here noises inside the cell with different sizes are inpainted based on background information. They achieved a high accuracy in the field of de-noising using inpainting methods. They provide results that show that an almost blurred image can be recovered with visually good effect. But as with other de-noising algorithms, the approach doesn't work well for large regions.

There have been a very few algorithms that utilize the advantages of both the image inpainting methods i.e. the structure restoration and texture fusion algorithms. One such work was proposed in the paper by Criminisi et al. [3]. The proposed methodis used to combine the advantages of both methods.The traditional concentric-layer filling algorithm [17] for defining the region filling order failed to remake structural features. Another method proposed in [18] tries to improve the time complexity.

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

3. METHODOLOGY

The conventions use throughout this paper are similar to earlier papers that deals with image inpainting [1], [3], [10]. Here, P represents the original image. Q represents the target region, i.e. the region to be inpainted. R represents the source region, i.e. the region from which information is available to rebuild the image. Generally, R =P – Q. Also, we use δQ to represent the boundary of the target region, i.e. the fill front. It is from here that, find some patch that is to be filled.This algorithm is basically an extension to Criminisi's algorithm. Using this algorithm, inpaint large missing regions in an image as well as rebuild small defects.Generally an exemplar based inpainting algorithm involves the following steps:

i. *Initialize the target region.* This is performed by marking the target region in some special colour, 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 anarea from the region to be inpainted.* The area size should be a bit larger than the largest distinguishable texture element in the image. We are used a default area size of 9 x 9 which can be changed with the knowledge of the largest texture element in the image, denote the area by ψ.

iv. *Find anarea from the image which best matches the selected area, ψ.* This matching can be done using a Mean Squared Error tofind the best matching area.

$$\mathbf{MSE} = \sum \frac{(f_{x,y} - g_{x,y})^2}{N} . \quad (1)$$

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

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

In Criminisi's algorithm, the priority function can be calculated by multiplicative form.

$X(P)=Y(P)*Z(P)$

Where Y(p) represents the confidence term for the area and Z(p) the data term for the area.

Also, this paper proposed the addition of masses to different components in the definition of priority term , the modified priority term can now be represented as

$$\mathbf{P(p) = \alpha \times R_c(p) + \beta \times D(p), 0 \leq \alpha, \beta \leq 1 .} \qquad (5)$$

Where α and β are respectively the component masses for the confidence and data terms. Also $\alpha + \beta = 1$ and $R_c(p)$ is the regularized confidence term.

$$\mathbf{R_c(p) = (1 - \omega) \times C(p) + \omega, 0 \leq \omega \leq 1.} \qquad (6)$$

Where ω is regularizing factor for controlling the curve smoothness. Using this confidence term the value of the confidence term is regularized to [ω,1].

The next step in the inpainting process is to find the area with the maximum similarity with the selected area. In the earlier methods, MSE is used for finding the similarity.When two or more areas with the same mean squared error (See Fig 2).for some images, the algorithm produced poor results.

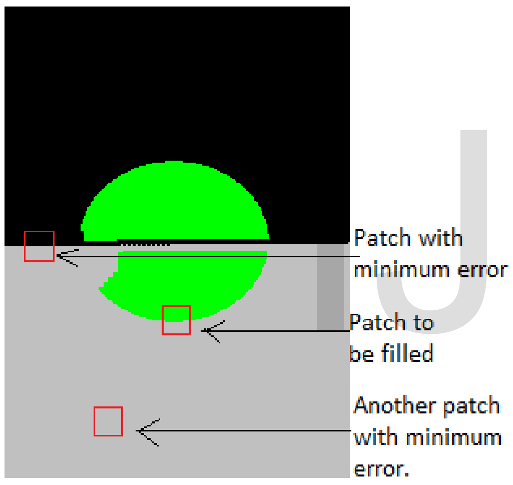


Fig 2: Patches with same mean square error.

The solution to this problem is tocalculate the variance of the areas with same mean squared error. This variance (please refer eq. 8) that use is the variance of the pixel values of the area with respect to the mean (please refer eq. 7) of the pixels from the same area that correspond to the pixels belonging to source region from the area to be inpainted.

$$\mathbf{M} = \frac{\sum \mathbf{f}_{p \in \phi \cap \Psi}}{\#\{p \mid p \in \phi \cap \Psi\}}.$$

$$\mathbf{V} = \frac{\sum (\mathbf{f}_{p \in \phi - \Psi} - M)^2}{\#\{p \mid p \in \phi - \Psi\}}.$$

Where,f denotes the pixel value of the element, #{..} represents the cardinality of the set.

Another improvement that propose to the given method is that Criminisi's approach looked for the best pattern from the complete image. Most often, the area that most resembles the selected area lies very close to the area selected to be inpainted. Based on this assumption, provide amethod on how to reduce the computational complexity of the algorithm. The diameter of the adjacent region to search is calculated at run time by taking into account the region to be inpainted. We search for the best pattern from a rectangle defined by (startX, startY) and (endX, endY). Then find these coordinates by using the maximum number of continuous green pixels in one row as well as a column. Let us assume that these values as cr and cc respectively. Then,calculate the coordinates as follows.

$$\mathbf{startX = max(0, p - \frac{n}{2} - c_r - \frac{D_x}{2}) .} \qquad (9)$$

$$\mathbf{startY = max(0, p - \frac{m}{2} - c_c - \frac{D_y}{2}) .} \qquad (10)$$

$$\mathbf{endX = min(w, p + \frac{n}{2} + c_r + \frac{D_x}{2}) .} \qquad (11)$$

$$\mathbf{endY = max(h, p + \frac{m}{2} + c_c + \frac{D_y}{2}) .} \qquad (12)$$

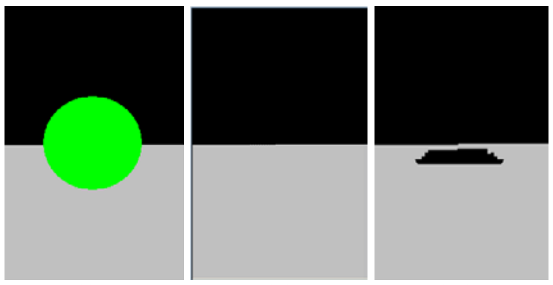
Where h and w are height and width of the image respectively, m and n are number of rows and columns in the area and Dx and Dy are constants that represent the minimum diameter for the X and Y directions respectively.

4. EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed variance approach and the improvement in speed,we are performed tests on several images and compared the so-obtained results with the conventional approaches. Several of the images that present here are taken from the previous literature and cite the appropriate paper.

4.1. Comparison with Criminisi's approach

Now we present the comparison of thismethod withCriminisi's approach. The image in Fig 3 (i) was given as input to the inpainting process that used our method as well as to our implementation of the Criminisi's approach. The results using Criminisi's approach were not that promising whereas our algorithm achieved better results. The difference in the results occurred while searching for the best pattern area.

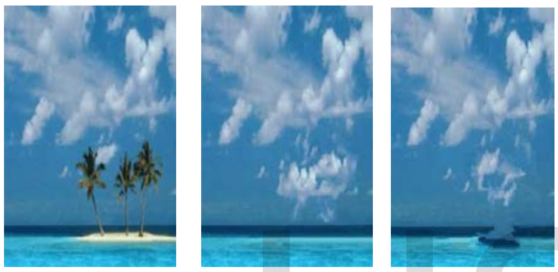


(i) (ii) (iii)

Fig 3.Comparison with Criminisi's approach. (i) Image to be inpainted, (ii) Result using our algorithm, (iii) Result using Criminisi's approach

4.2. Comparison on the basis of time with Criminisi's approach

Using the proposed algorithm, the time taken for inpaintingis considerably reduced.



(i) (ii) (iii)

Fig 4: Comparison with Criminisi's approach on benchmark data. (i) Image to be inpainted [3]. (ii) Result using our algorithm. (iii) Result using Criminisi's approach**.**

Following (Table 1) is a brief comparison of time taken using our method and Criminisi's approach.

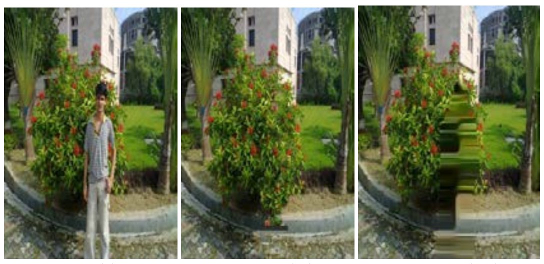
TABLEI. Comparison of our algorithm against Criminisi's algorithm

Serial No.	Image Size (in pixels)	Percentage area to be removed	Time taken (in milliseconds)	
			Criminisi	*Our Algorithm*
1.	124032	0.81	11223	2283
2.	60492	2.80	11546	4042
3.	120000	5.00	53971	27319
4.	60492	14.62	61286	52378
5.	225000	60.79	1357690	1337850

Thus the total time using our algorithm obviously depends on how much area is selected to be inpainted. Inaddition to this, the time would be less if the user selects small but spatially disconnected regions rather than if he selects the same percentage of target region continuously. This is so because we are taken into consideration the number of continuous green (color of the target region) pixels to remove the possibility of finding regions with no available areas.

4.3. Comparison with Photo Wipe © [11]

Now present a comparison of our algorithm with the current existing software for the same task.

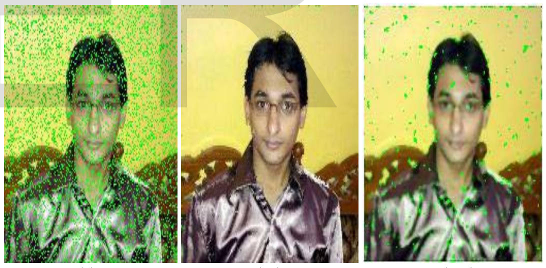


(i) (ii) (iii)

Fig 5: Comparison with Photo Wipe. (i) Original image. (ii) Output from our algorithm. (iii) Output using Photo Wipes.

4.4. Real Life Examples

Now we are present a few more examples from real life scenes.Fig 6 shows an example of noise removal using our algorithm. The noise was added randomly to the image and then the inpainting was applied. The inpainting algorithm was able to achieve a good overall result than that achieved by applying median filter [8] on the image.



(i) (ii) (iii)

Fig 6: Using inpainting to remove noise from the image. (i) Image with noise. (ii) Result after applying our inpainting algorithm. (iii) Result after applying 3x3 median filter .

Now we are present an example of removing an unwanted person from the photograph (See Fig 7).

IJSER

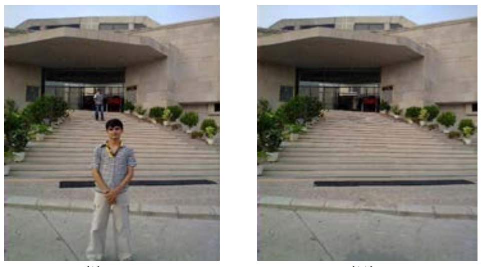


Fig 7: Example of removing unwanted persons. (i) The original image with unwanted persons. (ii) Image with the unwanted persons removed.

5. CONCLUSION AND FUTURE WORK

We are present an algorithm that can remove objects from the image in a way that it seems sensible to the human eye. It can also repair old photographs (e.g. removal of scratches).This method extends an exemplar based inpainting method along with a priority term that defines the filling order in the image. In this algorithm, pixels maintain a confidence value and are chosen based on their priority. The method defines a way of differentiating between areas that have the same minimum mean squared error with the selected area. This method is capable of transmitting both linear structures and two dimensional surfaces into the target region. This method can be used to fill small scuffs in the image or photos as well as to remove larger objects from them. It is also computationally proficient and works well with larger images.

We are looking forward to improving the algorithm so that the computational complexity in terms of quality and accuracy are further improved and also like to improve the inpainting algorithm. Also the inpainting algorithm presented here is not meant to be used for inpainting videos. We are also spendingtime to improve the proposed algorithm, it becomemore strong so that it can be used with videos.

IJSER

ACKNOWLEDGMENT

I wish to thank Prof. P Mohamed Shameem, Head of the Department, Computer Science and Engineering for providing excellent computational facilities and stimulating work environment for carrying out the research work.

REFERENCES

[1]. M. Bertalmio, G. Saprio, 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, Espange, , 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 (VIIP 2001), 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

IJSER