



www.ijarcsse.com

Volume 2, Issue 5, May 2012

ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Image Restoration Using Filling-In Technique for Missing Blocks of Image

Priyanka Rajesh Gulhane

Dept. of Computer Science & Engineering,
Sipna C.O.E.T., Amravati, INDIA.
gulhane.priyanka@gmail.com

V.T.Gaikwad

Dept. of Computer Science & Engineering,
Sipna C.O.E.T., Amravati, INDIA.

Abstract-The filling-in of missing information is a very important technique in image processing. While transmission of image if some blocks of image are lost then instead of using common retransmission query protocols, we aim to reconstruct the lost data using correlation between the lost block and its neighbors. Removing a target object and filling the missing regions of an image is the key technology generally applied to image restoration. The basic idea is fill-in the missing block with the information propagating from the surrounding pixels. Here the aim is to fill-in the gap of missing data in a form that is non-detectable by an ordinary observer. This technique provides a means to restore damaged region of an image, such that the image looks complete and natural after restoration. Applications of this technique include the restoration of old photographs and removal of superimposed text like dates, subtitles, or publicity. The performance of this method is tested for various images and combinations of lost blocks

Keywords — Filling-in, restoration, retransmission.

I. INTRODUCTION

Image processing is a wide area including various applications in it. Since the early days of art and photography, filling-in and inpainting has been done by professional artist. Imitating their performance with semi-automatic digital techniques is currently an active area of research. The filling-in of missing information with applications including image coding and wireless image transmission (e.g., recovering lost blocks), special effects (e.g., removal of objects), and image restoration (e.g., scratch removal) is a very important in image processing.

Before the advent of computers and software such as Photoshop, most photo restoration was done by restoration experts such as museum art restorers. Repairs were applied directly to the damaged photo and consisted mainly of air brushing over the damage. This is still the preferred method for valuable historical photos such as those found in archival collections. This type of work is very expensive and not usually required by the average person wishing repair old damaged family photos.

Fortunately, it is now possible through the use of computers and software, to restore almost any photo at very reasonable prices. Instead of working directly on the damaged photo, a copy is made using a scanner. Once all repairs are made to the copy using computer software, a new print can be produced. The final digital photo file of the repaired photo can be saved as an archival copy and replaces the need for a negative. It is a good idea to restore your entire photo collection in order to stop the unavoidable destruction that will take place no matter how carefully the photos are stored. Once restored, the photos can be copied to CD or DVD for long term storage and safe keeping. Once you have digital copies you need not worry about further damage because digital images do not change at all. Having digital copies will allow you to make

a set of prints, and additional prints at any time in the future.

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and restoration.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

The basic idea is fill-in the missing block with the information propagating from the surrounding pixels. Here the aim is to fill-in the gap of missing data in a form that is non-detectable by an ordinary observer. This technique provides a means to restore damaged region of an image, such that the image looks complete and natural after restoration.

Filling-in missing data in digital images has a number of fundamental applications. They range from removing objects from a scene all the way to retouching damaged paintings and photographs [1]. The basic idea is to fill-in the gap of missing data in a form that is nondetectable by an ordinary observer. In art, this process is called inpainting. Since the early days of art and photography, filling-in and inpainting has been done by professional

artist. Imitating their performance with semi-automatic digital techniques is currently an active area of research. Image inpainting provides a means to restore damaged region of an image, such that the image looks complete and natural after the inpainting process. Image inpainting could also be used to create special effects, for instance specific object removal. Digital image inpainting mainly aims at filling in missing pixels in an unknown region of an image in a visually plausible way.

II. RELATED WORK

Most of the schemes reported in the literature deal with image transmission in error-prone environments. The ensuring reconstruction scheme benefits because, it reduces the need of retransmission. There have been many texture synthesis algorithms proposed over the years. Image Quilting is based directly on texture synthesis [2], so reviewing other algorithms based on that work might provide some insight into possible extensions. Most of these papers, such as [3] use various data structures and search methods to increase the efficiency of the algorithm while maintaining the output quality.

Zhu et. al. model texture as a Markov Random Field and use Gibbs sampling for synthesis. Unfortunately, Gibbs sampling is notoriously slow and in fact it is not possible to assess when it has converged. Heeger and Bergen [4] try to coerce a random noise image into a texture sample by matching the filter response histograms at different spatial scales. While this technique works well on highly stochastic textures, the histograms are not powerful enough to represent more structured texture patterns such as bricks.

De Bonet [5] also uses a multi-resolution filter-based approach in which a texture patch at a finer scale is conditioned on its “parents” at the coarser scales. The algorithm works by taking the input texture sample and randomizing it in such a way as to preserve these inter-scale dependencies. This method can successfully synthesize a wide range of textures although the randomness parameter seems to exhibit perceptually correct behavior only on largely stochastic textures. Another drawback of this method is the way texture images larger than the input are generated. The input texture sample is simply replicated to fill the desired dimensions before the synthesis process, implicitly assuming that all textures are tilable which is clearly not correct.

E. Chang describe a packetization scheme in which the DCT coefficients array generated by JPEG is grouped such that bursty (consecutive) packet loss during transmission is scattered into a pseudo-random loss in the image domain (i.e., consecutive blocks are rarely lost in the image domain). The ensuing reconstruction scheme benefits because, most frequency components can be recovered from adjacent blocks. However, large bursts may cause the errors to cluster in the image, and reconstruction suffers. It should be noted that the packetization scheme proposed in [6], when used with the reconstruction scheme described in our paper, is expected to further improve and provide satisfactory reconstruction results even for very large bursts.

For this the proposed scheme is in the separation of the lost blocks into different classes, followed by the use of state-of-the-art image filling-in algorithms for textured and

structured regions. This is done in a complete automatic fashion and without any side information.

III. PROPOSED WORK

The reconstruction of lost blocks follows three computationally efficient steps-

- a) Mark the region to be filled;
- b) Search for the best matching pixel;
- c) Copy the selected pixel to proper position of the target area in current filling block.

Here the first step is to take the input missing block image and then mark the target region to be filled i.e. missing block region. Once the missing block region is found, then the next step is to search for the best matching pixel from the surrounding area. After finding the best matching pixel, we copy that pixel to the proper position of target area in current filling block. We repeat this process for all the missing blocks. After finishing the filling process we generate the output image. The flow diagram is used to achieve the program structure that is relatively easy for modifying and understanding of program flow. This structure encapsulates data and operation that must be applied to manipulate data or information.

The following fig. shows the flow of data and design of the system.

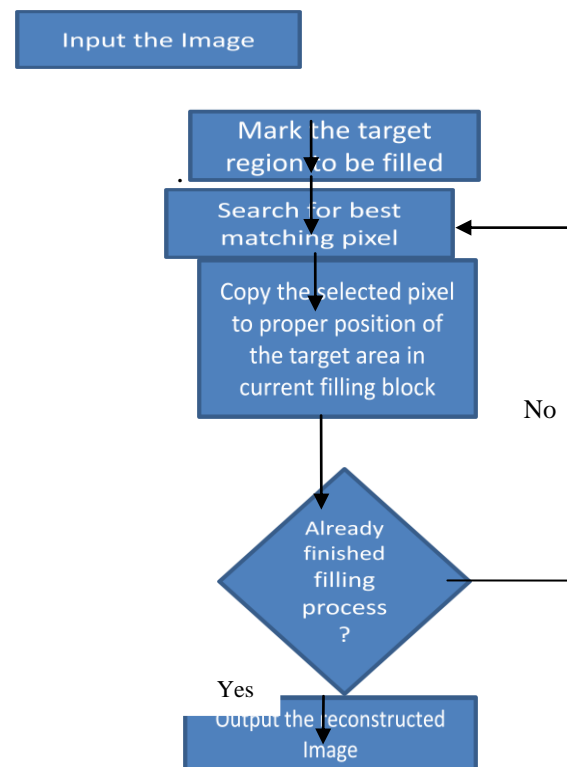


Fig 1 Flow chart of proposed work

In the proposed work we have implemented a system which fills the missing blocks of image in pixel by pixel fashion to restore the image damaged during transmission. The goal of this theme is to provide an effective way of restoring the missing blocks of an image in minimum time reducing the need for retransmission. The aim of image restoration is to restore the image in the form that is non-detectable by an ordinary observer. This technique provides a means to restore the damaged region of an image, such that the image looks complete and natural after restoration.

In the first step the image is taken as an input. Then the algorithm focuses on the salient features of images by finding missing blocks of an image. Once the missing block region is found, then the next step is to search for the best matching pixel from the surrounding area. After finding the best matching pixel, we copy that pixel to the proper position of target area in current filling block. We repeat this process for all the missing blocks. After finishing the filling process we generate the output image.



Fig 2 Original Image



Fig 3 Missing Block Image

Here fig. 2 is the original image and fig. 3 is the missing block image. This missing block image is taken as input to the given implemented system along with the number of iterations. Fig.4 shows the final output of this technique for 5 numbers of iterations. Fig.5 to fig.9 shows the intermediate result with different number of iterations. The time required for getting this restored output image is 16.88 seconds and PSNR value of output image is 45.16 db. Table 1 shows the error removal % for different number of iterations. From table 1 we can see that as the number of iterations increases the error removal % also increases because output of first iteration is given as input to next iteration.



Fig 4 Restored Image using filling-in Technique

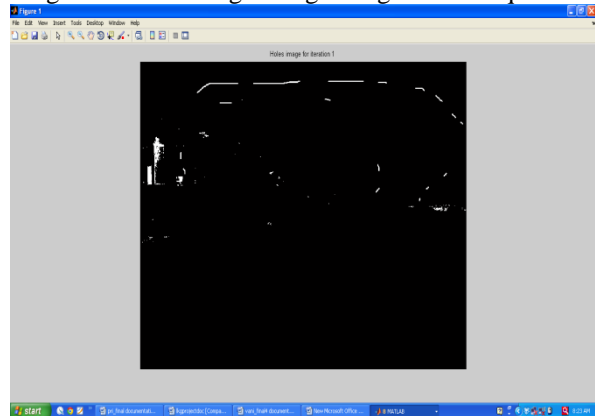


Fig 5 Result of iteration 1

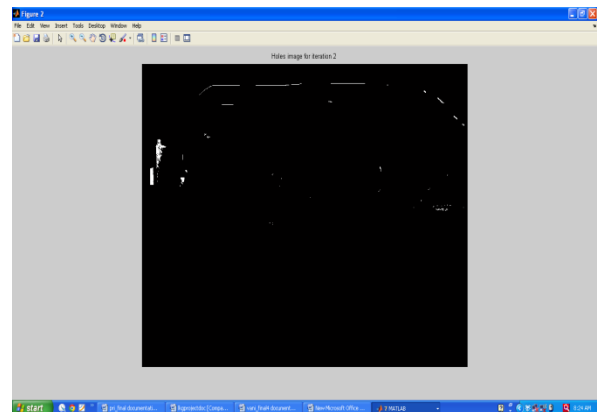


Fig.6 Result of iteration 2



Fig 7 Result of iteration 3

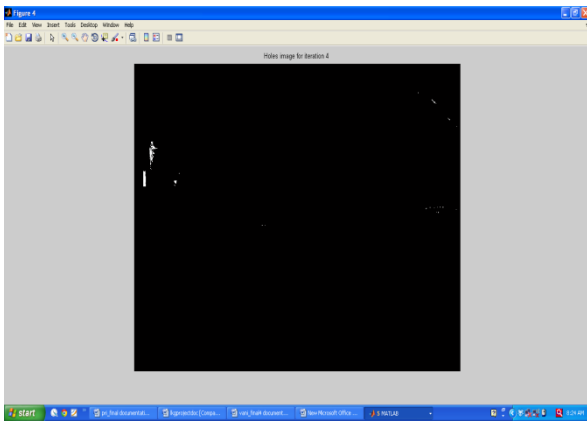


Fig 8 Result of iteration 4

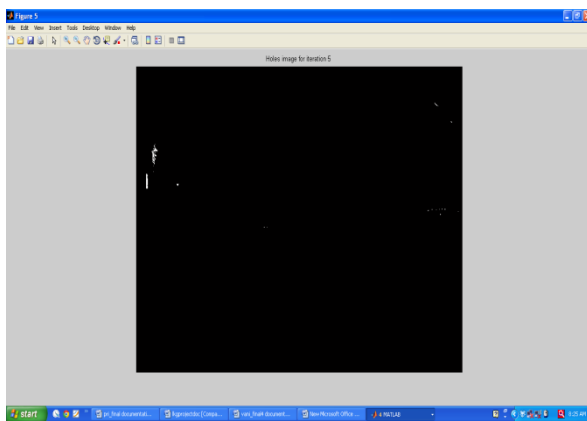


Fig 9 Result of iteration 5

Table 1

Iterations	Error removal %
1	0.40
2	0.48
3	0.51
4	0.53
5	0.55

IV. CONCLUSIONS AND FUTURE DIRECTIONS

The filling-in method is applied to several different types of datasets of missing blocks images. The

results on these data indicate the feasibility of the implemented approach. Here we have shown that as long as the features in the image are not completely lost, they can be satisfactorily reconstructed with the information propagating from surrounding pixels. This eliminates the need for retransmission of lost blocks. When the image resolution is increased, the quality of reconstruction improves and a retransmission request is rarely required, resulting in a better effective data transmission rate.

A number of research directions should be taken following the results reported here. We have tried to use image-dependent information, i.e., information from surrounding pixels, to enhance the performance of image. The reconstruction can be further increased by finding better masks by providing more image information.

In a more general setting, the extension of the approach presented here, to color data needs to be investigated. Since the missing blocks in the different channels need not be in the same image position, information from different channels can be used in the block reconstruction. Adding this to the current neighboring information used is expected to improve even further the quality of the results.

V. REFERENCE

- [1] Coloma Ballester, M. Bertalmio, V. Caselles, "Filling-In by Joint Interpolation of Vector Fields and Gray Levels", *IEEE Transactions on Image processing*, Vol. 10, no.8, August 2011.
- [2] Alexei A. Efros and Thomas K. Leung, "Texture Synthesis by Non-parametric Sampling," in *IEEE Int. Conf. Computer Vision*, Corfu, Greece, Sept. 1999.
- [3] L. Wei and M. Levoy, "Fast texture synthesis using tree structured vector quantization," in *SIGGRAPH* 2000.
- [4] David J. Heeger and James R. Bergen, "Pyramid-Based Texture Analysis/Synthesis," in *Computer Graphics (SIGGRAPH 1995)*, July 1995.
- [5] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proc. ACM SIGGRAPH*, July 1997.
- [6] Chang, "An image coding and reconstruction scheme for mobile computing," in *Proc. 5th IDMS*, Oslo, Norway, Sept. 1998, pp. 137-148.
- [7] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Comput. Graph. (SIGGRAPH 2000)*, July 2000, pp. 417-424.
- [8] Z. Alkachouch and M. Bellanger, "Fast dct-based spatial domain interpolation of blocks in images," *IEEE Trans. Image Processing*, vol. 9, pp.729-732, Apr. 2000.
- [9] K. Karu, A. K. Jain, and R. M. Bolle, "Is there any texture in the image?," *Pattern Recognit.*, vol. 29, no. 9, pp. 1437-1446, 1996.
- [10] Marquina and S. Osher, "Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal," *SIAM J. Sci. Comput.*, vol. 22, pp. 387-405, 2000.