Aditi Beniwal et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 3, Issue 3, September 2016, pp. 11-19

Firefly Based Adaptive Block Image Inpainting

Aditi Beniwal[1], Deepti Ahlawat[2]

[1]M.Tech Scholar, N.C. College of Engineering and Technology, Israna
[2]Assistant Professor, N.C College of Engineering and Technology, Israna

Abstract- In the proposed investigation the developed method on the platform of Markov Random Field optimised adaptive size blocking in image inpainiting. We replaced the complex MRF with a less complex and more approximate method Firefly optimisation algorithm. The conducted performance evaluation shows that the developed inpainting-based image solution outperforms the previous method for the PSNR (peak signal to noise ratio), MSE (mean square error) and dissimilarity metric for all the selected test images, while significantly improves the solution, which is very encouraging.

Keywords: Image Processing, Image inpainiting, Image restore.

I. INTRODUCTION

The goal of image inpainting is filling the target region in the image with visual plausible information. Filling this region with information that could have been in the image. There are several applications for image inpainting, one of these is the restoration of old images and movies by removing cracks from these images and movies. The removal of objects from images, like time stamps or a person. Another application can be image inpainting as the pre-process of other computer vision or image processing applications, an example of this is the use of image inpainting in image-based material editing. It can also be used to fill missing parts in image communication, for example image compression, lost packets retrieval and zooming [1]. An example of image inpainting is shown in Figure 1.1.

Figure 1.1: Example of image inpainting from [1], the original image is shown on the left and the inpainted image on the right.

The main challenges in image inpainting are the continuation of structure, finding the correct information and the speed of inpainting. Many researchers are working to tackle these challenges but still many improvement scopes are still there. In inpainting algorithms we have to find the perfect information to fill in a gap. That information can come from multiple places, adjacent pixels, other parts of the image or other images, it can be challenging to find the best match for each situation. Current research of image inpainting techniques consists of some different approaches. Some approaches are based on pixel by pixel updating by computing partial differential equations. Others are based on copying parts of the input image to the unknown region or even information from a dataset of images [5][6][7]. The introductory block diagram of image inpainting is shown in figure 1.2.

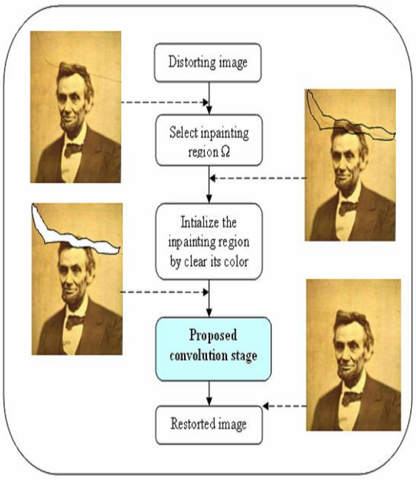


Figure 1.2: block diagram of image inpainting [11]

It has been analysed form the previous work that image inpainting methods in software can be categorized in two categories: geometrical based approach and exempler based approach. Geometrical based approach is simple and fast method but sometimes in case of large contrast difference in images it gives blurring effect whereas such type of issue is not with the exempler based method. But this approach is complex and time consuming, so some times to deal with real time, it fails to show effects [1][13][16].

So in proposed work we will use geometrical based approach and to remove blurring effect we have used modified approach based on blocks of test image. The whole image will be divided into several blocks of adaptive size based on context of chosen block. This reduces the target image blocks which may be similar to patch in image. The block size is also an important factor to match the target block and reduction in execution time. Sometimes a fixed block size matching gives results after many iteration or even doesn't guarantee. Further chosen blocks are fine selected using firefly optimisation algorithm.

In the sections ahead in this paper, literature review of various papers with their important points listed in section II and present algorithm is explained in section III followed by simulated results in section IV and section V inherits conclusion of the survey followed by a list of references. A result table is also shown in appendix at the end of this paper.

II. LITERATURE REVIEW

Image inpainting is an iterative process which ends when any target condition is matched or maximum number of iterations is completed. A general flow chart of image inpainting process is shown in figure 2.1

Bertalmio et al.[2] introduce the term image inpainting to computer science. In the algorithm the region that has to be inpainted will be filled-in by information of the region surrounding the gap. The curves of equal intensity (isophotes) arriving at the boundary are propagated inwards. Because this is the first and one of the most important papers in image inpainting.

Criminisi et al.[1] propose an algorithm inspired by the algorithm by Bertalmio et al.[2] and texture synthesis [3]. In contrast to the paper of Bertalmio et al. the unknown region is inpainted patch by patch. The sequence of patch should be inpainted is based on the isophote information and the amount of known information surrounding the patch. This popular method increased the result of image inpainting a lot. Because of that this method is further explained in chapter 4.

In the algorithm of Oliveira et al.[4], the unknown region of the image is convolved with a Gaussian kernel. To prevent edges to be blurred the user manually specifies barriers for the diffusion. This algorithm is much faster than the other algorithms but it can only be used with very small unknown regions and is less accurate. An example of this algorithm is shown in figure 2.2.

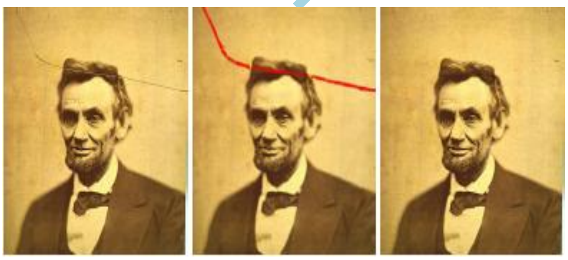


Figure 2.2: Olivera method [4], input, mask and result, note the diffusion barriers near the boundaries of the hair of Abraham Lincoln.

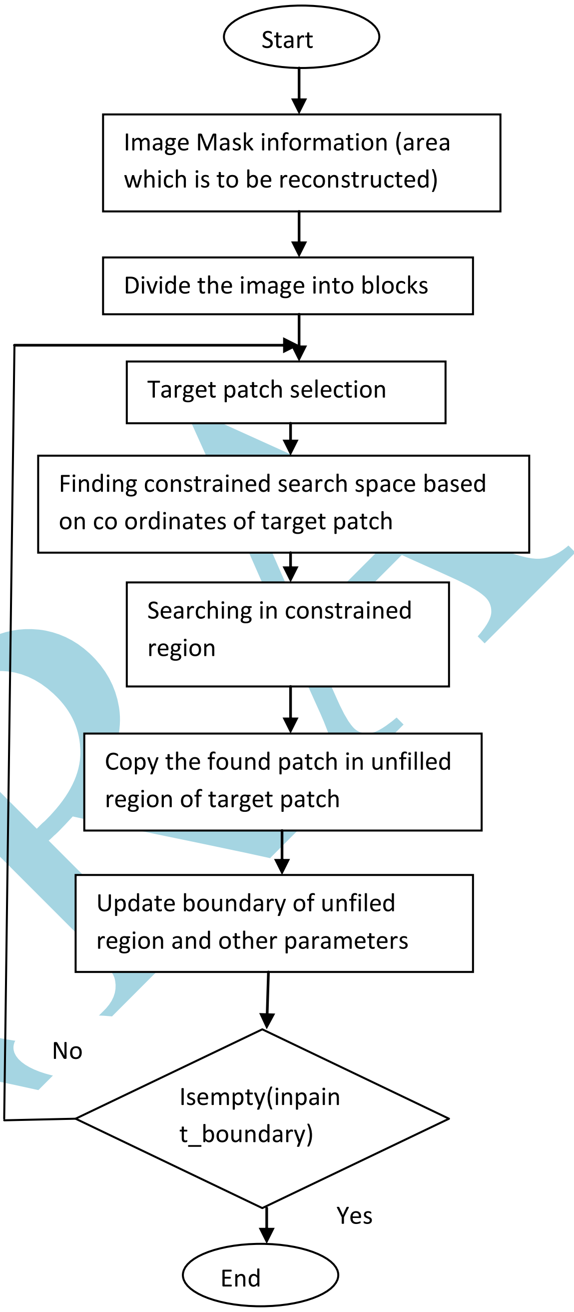


Figure2.1: A general flow chart of image inpainting process

The algorithm of Perez et al.[5] showed how gradient domain reconstruction can be used in image editing applications. The actual pixel values for the unknown pixel values are computed by solving a Poisson equation which locally matches the gradients while obeying the fixed Dirichlet (exact matching) conditions at the seam boundary. Poisson Image Editing can be used best for seamless inserting and local illumination changes but can also be used for image inpainting. The author of [6] proposed a new inpainting algorithm based on propagating the image smoothness estimator along the image gradient. Similar to the algorithm of Bertalmio et al.[2]. The image smoothness is estimated as the weighted average of the known image neighborhood of the pixel to inpaint. The fast marching method (FMM) is used to create a distance function to the initial boundary. The pixels of the unknown region are inpainted in order of the distance to the boundary, proceeding from the smallest to the largest. This method is fast but creates blurry effects with larger unknown regions. An example of this algorithm is shown in Figure 2.3.

Figure 2.3: Telea method [6], input plus mask and result.

Texture synthesis

Texture synthesis is the process of algorithmically constructing the large digital image from a small digital sample image by taking advantage of its structural content [3][7]. Texture synthesis can be also used to all in unknown regions in images, the algorithm of Criminisi et al.[1] is partially based on these algorithms. An example of the algorithm by Efros et al.[3] is shown in Figure 2.4.

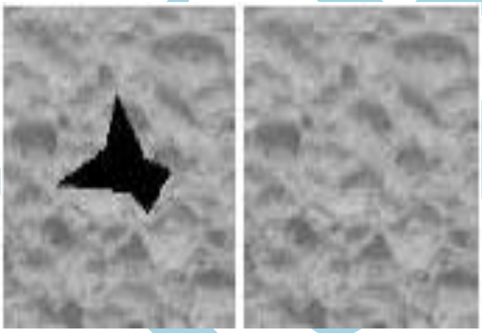


Figure 2.4: Texture Syntheses by Efros et al.[3], input plus mask and result.

Guided method

Sun et al.[8] introduces a new direction in image inpainting. In the authors algorithm the user is also able to specify support lines. These support lines specify where important lines in an image should be continued. This algorithm first alls in the unknown information around the support lines by dynamic programming or belief propagation, depending on the structure of the support lines. After that the rest of the image is filled in by texture propagation. The result of this algorithm is good but we can not compare these with others because this algorithm requires extra input. An example of this algorithm is shown in Figure 2.5.

Figure 2.5: Sun method [8], input, mask, first stage and result.

Multiple images

Hays et al.[9] developed a total different way of image completion, instead of searching in the input image, the algorithm searches throughout a whole database of images to and information to all the missing region. The new area is paste in using Poisson blending [5]. The authors state that their results look better then the algorithm of Criminisi et al.[1] but the algorithm needs a database of two million images for only three scenes. This algorithm gives visual pleasing results but is very slow and is unpractical because it needs the large dataset. An example of this algorithm is shown in Figure 2.6.

Figure 2.6: Hays method [9], input, mask, first stage and result

Video inpainting

Image completing methods can also be used for videos, because a video is a set of multiple images. Naively inpainting each frame will not result in the best result important information about the current frame can be found in the adjacent frames. Patwardhan et al.[10] presented an algorithm for video inpainting of a scene taken from a static camera. This method is an extension of the paper of Criminisi et al.[1]. It extends the idea from a single image to a set of images but also taken in to account the adjacent frames. In this algorithm the frame is separated into a background and a foreground in which the foreground is first inpainted. After this step the background is inpainted, each patch is copied to every frame to get a consistent background. This technique has some nice results but it has some restrictions, it requires a fixed camera position and a stationary background with some moving foreground. Wexler et al.[11] have proposed a method for space-time completion of large damaged areas in the video sequence. They pose the problem of video completion as a global optimization problem with a well-defined objective function. In the algorithm every local patch should be found in the remaining part of the video and globally all these patches must be consistent with each other spatially and temporally. An example of this algorithm is shown in Figure 2.7.

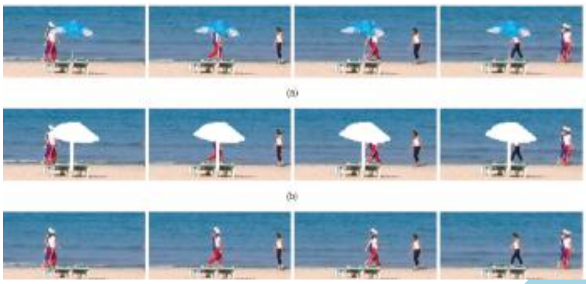


Figure 2.7: Wexler method [11], input, mask and result sequence.

Table of literature survey of image inpainting

Author	Year	Method	Merits	Demerits
Bertalmio M.	2000	Partial differential equation	Fine outcome and preserves all structural information	Results display blurs when applied to large missing regions
Oliviera, B. Bowen and Y.S Chang	2001	Convolution based inpainting	Produces fine results without blurring	For corrupted regions thicker than ten pixels, blurring occurs
Sarab M. Hameed , Nasreen J. Kadhim	2009	Particle swarm optimization	Results do not display blurs	Not applicable for thick regions and curved structure
A. A. Efros and T. K. Leung	2000	Texture Synthesis	Gives impressive results and preserves all structural & textural information	Gives unsatisfactory results of the corrupted regions is spread along most of the image area
J. Sun, L. Yuan and J. Jia	2005	Guided method	Produces fine results without blurring	Requires extra input
J. Hays	2007	Multiple images	Gives visual pleasing results	Very slow and unpractical

III. PRESENT WORK

Proposed work is based on selecting the most appropriate block of image to regenerate the mask of the image. Care has to be taken that size of block should be adaptive rather than fixed number of pixels in the block. Previously Markov random field was used to generate the exact block to replace mask over image but we replaced this algorithm with firefly algorithm. In next sections of this chapter we have explained the firefly algorithm and use of firefly algorithm in selection of appropriate block from rest of image.

Firefly algorithm is an iterative algorithm in which fireflies positions are image patch blocks co ordinates form the test image. The position/ path size is updated by mathematical formulation described in firefly algorithm as

$$x_i = x_i + \beta_0 e^{-\gamma r^2}(x_j - x_i) + \alpha\text{€i}$$

Where 'r' is the distance between any two fireflies i and j at x_i and x_j, respectively is the Cartesian distance which is calculated as

$$r_{i,j} = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2}$$

Where β0 is the attractiveness at r=0. In firefly algorithm the firflies moves towards minimizing the fitness function which is MSE (mean square error) in our case. All firefly move towards to minimize their randomness by updating their positions by above equation. A complete flow chart for the proposed algorithm is shown in appendix. In this work we will define a source region whose selection will be based on adaptive block size iteratively. The energy difference between the patch block and source block is the main deciding factor for the optimum block size selection which is decided by firefly optimization algorithm. Adaptive block size selection reduces the searching space for firefly optimization. The objective function will be used in firefly algorithm will be energy difference and the optimization works towards minimization of this difference.
The significance of firefly algorithm which is a bio inspired algorithm to image inpainting work lies with the terminology used in firefly. Table 1 shows the significance of firefly algorithm terms to proposed work.

Table 1: Significance of firefly algorithm terminology with image inpainting algorithm

Firefly Algorithm	Proposed Image inpainting algorithm
Positions of firefly	Block size for patches
Searching space dimension in which firefly search for brightness	Number of tuning variables to be tuned for better results which is two in this case
Update in the position if firefly	Change in the block size of patch

The position of firefly is decided by the number of variables to be tuned in present application. Like in this case we need to tune only the size of patch blocks which contains the information of number of row pixels and column pixels, so there are two variables to be tuned in image inpainting case. So a firefly' position is defined by two values and each firefly is initialised with different position which gets updated by the light attractiveness factor.

IV. RESULTS

To check the proposed firefly tuned adaptive image inpainting method, we did simulation in MATLAB R2010a. This tool provides a wide range of image processing toolbox and very easy to use syntaxes. Since this is an iterative algorithm so using large images will take time to regenerate the image patch. So we are using gray colored images here though this also works over colored images. Proposed work has been compared with Markov random field (MRF) algorithm for validation and our work showed the improvement over MRF. Results have been compared on the basis of PSNR, MSE. Figure 4.1(a) shows the input image and figure 4.1 (b) shows the image with masks over it. We have iterated the image block size algorithm with firefly algorithm and a total of 20 iterations are done. Regenerated mask by these numbers of iterations is shown in figure 4.2 (a). It can be seen that still there are some blurred masks in the image. These masks are removed when number of iterations is increased. Figure 4.2(b) shows the regenerated image with 40 iterations.

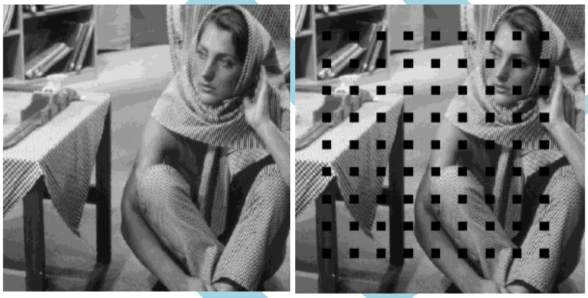


Figure 4.1(a): Input Image (b): Masked Input image

Figure 4.2 (a): Masked image with regenerated mask with firefly algorithm in 20 iterations (b) regenerated masked image in 40 iterations

Comparative results for evaluation parameters are shown in figure 4.3. It is clear from the graph that proposed algorithm is performing better than MRF algorithm. The value of PSNR (peak signal to noise ratio) should be high and with the iterations, it should increase. Means square error curve in figure 4.3(c) is the decreasing curve with iterations and it matches with the practical world condition that error should be decreasing and similar is with figure 4.3(b).

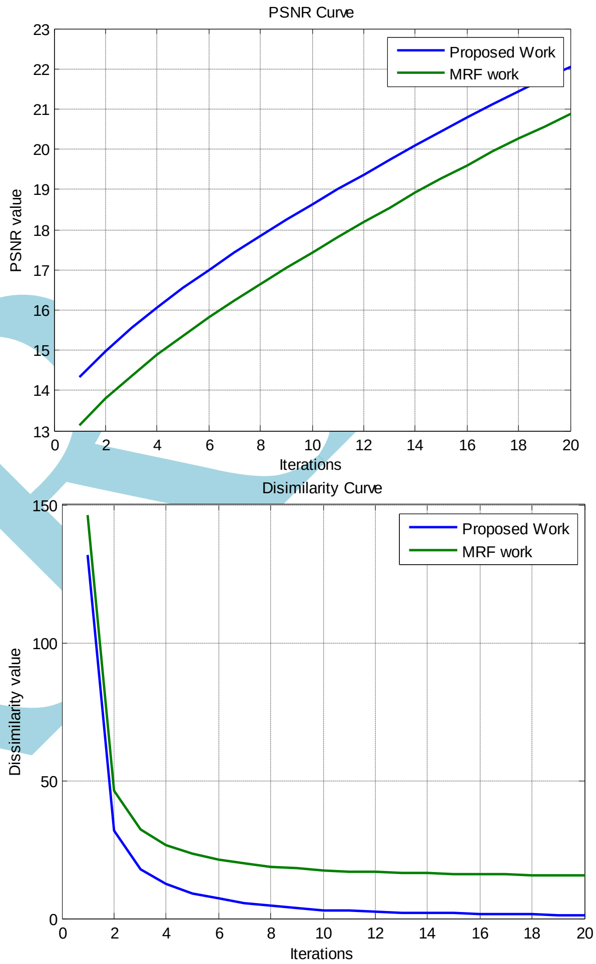


Figure 4.3(a): comparative PSNR curve (b): comparative Dis-similarity curve

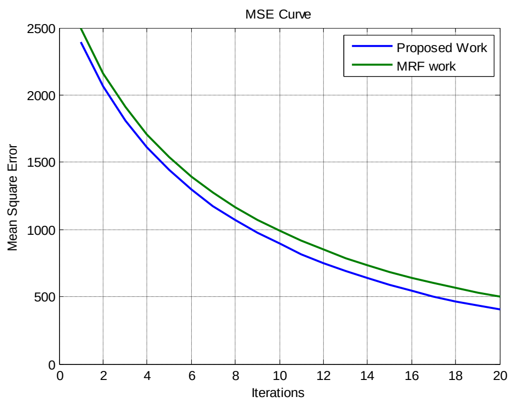


Figure 4.3(c): comparison in terms of MSE

Since, we have tested the algorithm for different number of iterations, so a bar graph for evaluation parameters is shown in figure 4.4. From the analysis of graph it is observed that an improvement of 5.54% in PSNR and 12.62% in MSE is achieved over MRF algorithm with same adaptive block size concept. Results have been tested on various images. Results are shown in appendix in a table.

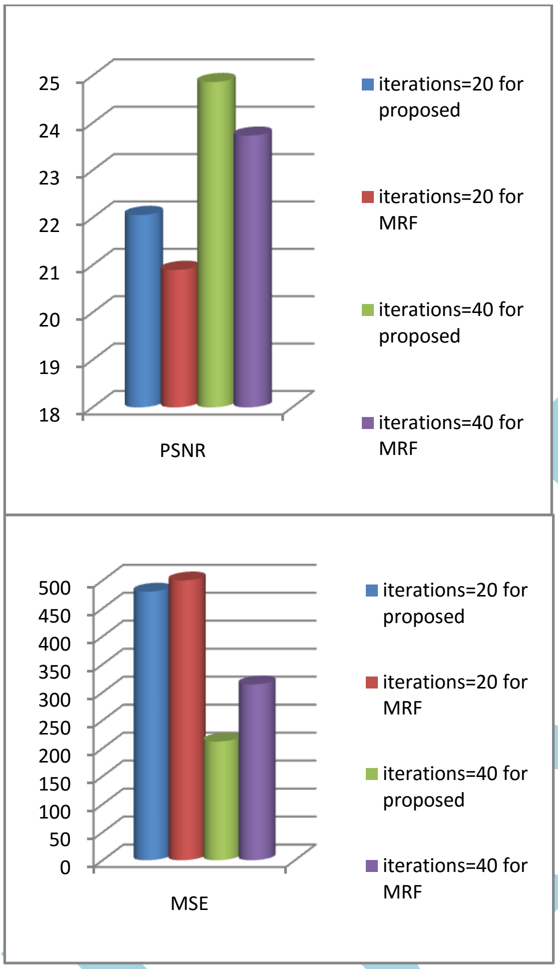


Figure 4.4: comparative bar chart for MSE and PSNR for proposed work and MRF algorithm

V. CONCLUSION

In this paper, we have analyzed the previously used method of image inpainting and suggested our algorithm on that platform. It has been found from the previous paper that local searching like MRF increases the speed but should decrease the visual appeal, although in the experiments it increased the visual quality. In proposed work we have used global searching operations using firefly algorithm which increase the speed and visual quality too by locating the exact match of patch. Calculating MSE of the resulting images with the input image using a virtual mask is the only way of calculating the visual appeal of the resulting image found in literature, but PSNR which is a robust quality measurement is also used by us. It tells how much inpainted image can bear the outside attacks without losing the information.

REFERENCES

[1]. A. Criminisi, P. Pierez, and K. Toyama, “ Region filling and object removal by exemplar-based image inpainting”, *IEEE Transactions on Image Processing*, vol.13, 2004.

[2]. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “ Image inpainting in Proceedings of the 27th annual conference on Computer graphics and interactive techniques”, *ACM Press*, 2000.

[3]. A. A. Efros and T. K. Leung, “ Texture synthesis by non-parametric sampling”, *IEEE Computer Society*, vol.2, 1999.

[4]. M. M. Oliveira, B. Bowen, R. Mckenna, and Y. sung Chang, “ Fast digital image inpainting”, *ACTA Press*, 2001.

[5]. P. Perez, M. Gangnet, and A. Blake, “ Poisson image editing”, *ACM Press*, 2003.

[6]. A. Telea, “ An image inpainting technique based on the fast marching method”, *Journal of Graphics*, vol.9, 2004.

[7]. L.-Y. Wei and M. Levoy, “ Fast texture synthesis using tree-structured vector quantization”, *ACM Press*, 2000.

[8]. J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, “ Image completion with structure propagation”, *ACM Trans. Graph*., vol.24, July 2005.

[9]. J. Hays and A. A. Efros, “ Scene completion using millions of photographs”,*ACM Transactions on Graphics*, vol.26, 2007.

[10]. K. A. Patwardhan, G. Sapiro, and M. Bertalmo, “ Video inpainting of occluding and occluded objects”, *in International Conference on Image Processing*, 2005.

[11]. Y. Wexler, E. Shechtman, and M. Irani, “ Space-time video completion, Computer Vision and Pattern Recognition”, *IEEE Computer Society Conference*, vol.1, 2004.

[12]. P. Perona and J. Malik, “ Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell*”, vol.12, July 1990.

[13]. T.-H. Kwok, H. Sheung, and C. C. L. Wang, “ Fast query for exemplarbased image completion”, *IEEE Transactions on Image Processing*, vol.19, December 2010.

[14]. Q. Chen, Y. Zhang, and Y. Liu, “ Image inpainting with improved exemplar-based approach”, *IEEE Transactions on Image Processing*, vol.18, 2007.

[15]. Anupam, P. Goyal, and S. Diwakar, “ Fast and enhanced algorithm for exemplar based image inpainting”, *IEEE Computer Society Conference*, vol.7, 2010.

[16]. T. Ružic, A. Pižurica, “Context aware image inpainting with application to virtual restoration of old paintings”, *International Conference Information and Communication Technology Forum* , 2013.

[17]. Raluca Vreja and Remus Brad, "Image Inpainting Methods Evaluation and Improvement", *The Scientific World Journal*, 2014.

[18]. Jun Zhou, Antonio Robles-Kelly, "Image Inpainting Based on Local Optimisation", *International Conference on Pattern Recognition*, 2010.

[19]. Sarab M. Hameed , Nasreen J. Kadhim, Mahmood A. Othman, "Image Inpainting Based On Particle Swarm Optimization", *Iraqi Journal of Science*, vol.50, no.2, 2009.

[20]. Zhaoxia Wang, Quan Wang, CS Chang, Ming bai, Zhen Sun, Ting Yang, " Image Inpainting Method based on Evolutionary Algorithm", *International Journal of Digital Content Technology and its Applications*, vol.5, no.4, April 2011.

[21]. K. Sangeetha, Dr. P. Sengottuvelan, " A Novel Exemplar based Image Inpainting Algorithm for Natural Scene Image Completion with Improved Patch Prioritizing", *International Journal of Computer Applications*, vol.36, no.4, December 2011.

[22]. Seema Kumari Singh , Prof J.V Shinde, " Optimum Patch Selection Using GA in Exemplar Based Image In-painting", *International Journal of Computer Science and Information Technologies*, vol. 6, 2015.

[23]. B. Fergani, H.Bensuici, "A discrete firefly algorithm for geometric image inpainting", *International Conference on Advanced Technologies For Signal & Image Processing*, 2014.

[24]. Chongwu Tang, Xi Hu, Li Chen, Guangtao Zhai, Xiaokang Yang, " Sample-Based Image Completion Using Structure Synthesis", *International Journal of Computer Science and Information Technologies*, 2013.

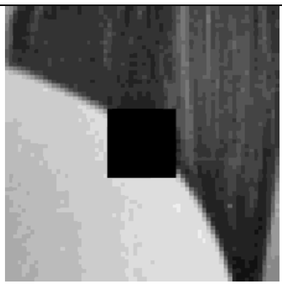
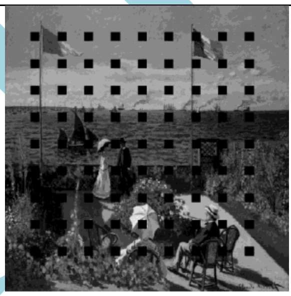
[25]. Shivali Tyagi, Sachin Singh, " Image Inpainting By Optimized Exemplar Region Filling Algorithm", *International Journal of Soft Computing and Engineering*, vol.2, Issue-6, January 2013.

[26]. Nuno Miguel Ventura Couto, " Inpainting-based Image Coding: A Patch-driven Approach", *IEEE Transactions on Image Processing*, 2010.

[27]. Sander van de Ven, " Image Inpainting", *IEEE Computer Society Conference*, 2012.

APPENDIX

Table A.1: Image inpainting results for different images

Input Image	Masked Image	Results in 20 iterations	Results in 40 iterations
 Crop of lena (64x64)		 PSNR= 19.92816 MSE= 661.09550 Dissimilarity value= 0.010630	 PSNR= 20.093279 MSE= 661.09550 Dissimilarity value= 0.002877
 Bungee (512x512)		 PSNR= 21.806062 MSE= 429.01668 Dissimilarity value= 0.413233	 PSNR= 22.375240 MSE= 376.319161 Dissimilarity value= 0.020022
 Base (512x512)		 PSNR= 22.172920 MSE= 394.265105 Dissimilarity value= 0.931369	 PSNR= 23.590052 MSE= 284.49532 Dissimilarity value= 0.025627

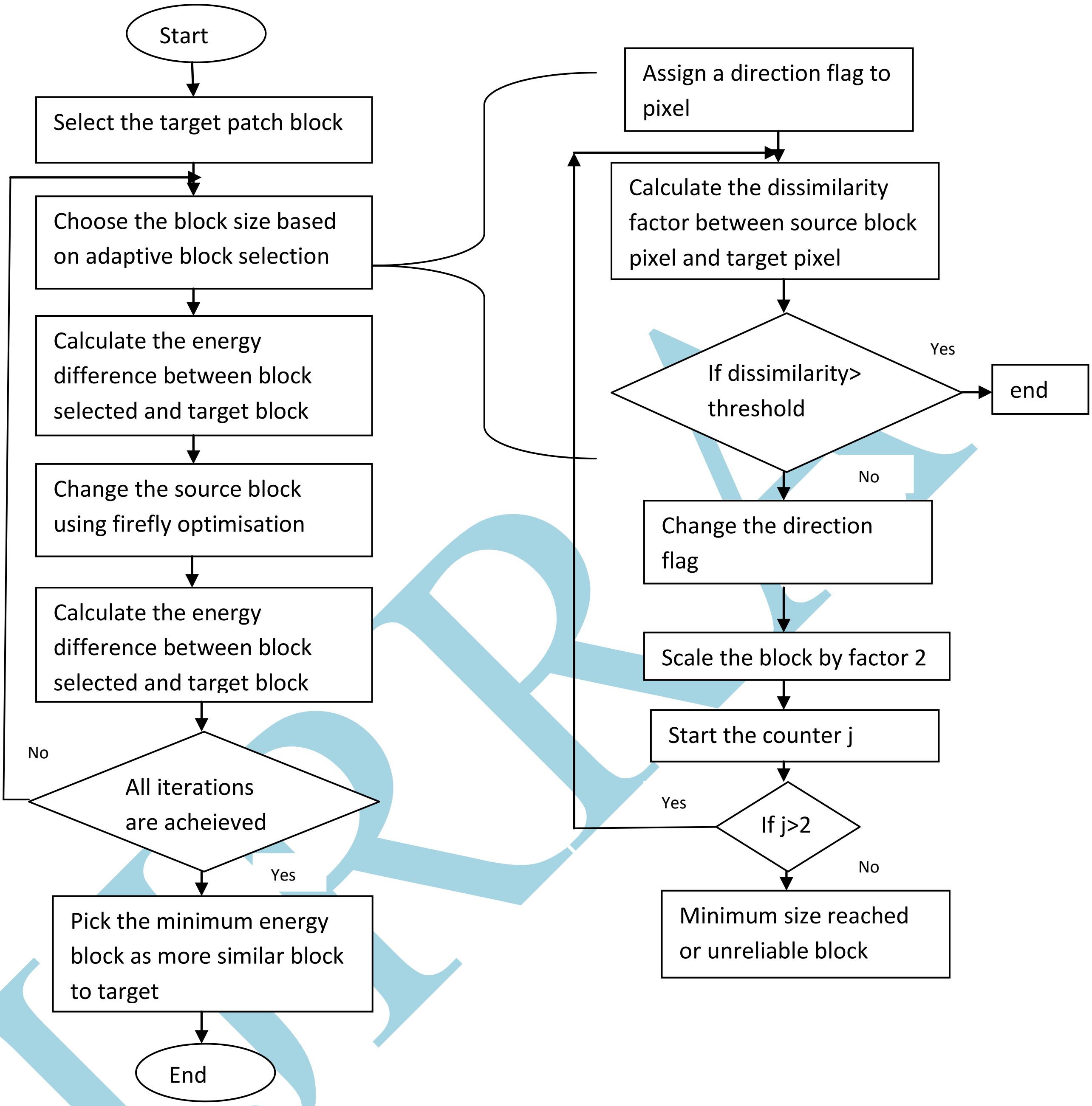


Figure A.1: Proposed algorithm flow chart