International Journal of Information Movement Vol.2 Issue VIII December 2017)
Website: www.ijim.in ISSN: 2456-0553 (online) Pages 165-174

# DYNAMIC PATCH SELECTION IN IMAGE INPAINTING

**Gurhans Pal**
M.Tech Student
Department of Electronics & Communication Engineering,
E-Max Group of Institutions, Kalpi - Naraingarh Road, Gola, Tehsil - Mullana, Dist. Ambala, Haryana, INDIA
E-mail: Guruhanspalsaroha@gmail.com

**Er. Vikas Sharma**
Asstt. Professor
Department of Electronics & Communication Engineering,
E-Max Group of Institutions, Kalpi - Naraingarh Road, Gola, Tehsil - Mullana, Dist. Ambala, Haryana, INDIA
E-mail:Vikas741@gmail.com

**Abstract:**The remarkable demand for universal distribution and consumption of image and video content over various networks has pressured the digital consumer electronics industry to launch an almost infinite variety of electronic devices capable of acquiring, processing, editing and storing these very attractive types of content. The combination of this trend with the increasing demand for higher qualities and larger screen resolutions has posed another challenging problem to the image coding research community: to significantly increase the current image compression efficiency for the various, relevant target qualities. In this context, better exploiting the Human Visual System behaviours and characteristics is largely recognized as an appealing way to target further improvements in terms of image compression efficiency. With this purpose in mind, inpainting-based image coding solutions have recently emerged as a novel coding paradigm to further exploit the imagevisualredundancy, thus allowing increasing the compression efficiency while assuring the required perceived image quality.we have developed a new method on the platform of gravitational search algorithm (GSA) optimisation and Markov Random Field optimised adaptive size blocking in image inpainiting. We replaced the complex MRF with a less complex and more approximate method GSA 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: GSA, MRF

**1.0 Introduction:** As early as the Renaissance, people try to restore damaged paintings in a way that the paintings will properly look like the original for an observer who is unfamiliar with the original. This process is called restoration, conservation, inpainting or retouching. This is done to preserve the paintings and other fine art for future generations. In image processing and computer vision we try to do the same with digital images, this is called image completion, image disocclusion or (digital) image inpainting. 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.

In this paper we have proposed a new method on the platform of gravitational search algorithm (GSA) optimisation and Markov Random Field optimised adaptive size blocking in image inpainiting**.** Our work is based on adaptive sizing of blocks which are to be filled in mask in image inpainting process. As discussed this problem can't be solved my simple mathematics since it is NP hard problem. So we used Gravitational Search Algorithm (GSA) is used which is an iterative algorithm and a script for this purpose is developed in MATLAB.

## 2.0 Gravitational Search Algorithm (GSA)

GSA was introduced by Rashedi et al. in 2009 and is intended to solve optimization problems. The population-based heuristic algorithm is based on the law of gravity and mass interactions. The algorithm is comprised of collection of searcher agents that interact with each other through the gravity force. The agents are considered as objects and their performance is measured by their masses. In the proposed set of rules, agents are taken into consideration as objects and their overall performance is measured by their hundreds. All these objects appeal to each other by means of the gravity force, and this force causes a global motion of all items in the direction of the objects with heavier loads. Hence, hundreds cooperate using an immediate shape of conversation, through gravitational force. The heavy hundreds – which correspond to correct solutions – flow more slowly than lighter ones, this guarantees the exploitation step of the set of rules. In GSA, every mass (agent) has four specs: position, inertial mass, lively gravitational mass, and passive gravitational mass. The function of the mass corresponds to an answer of the trouble, and its gravitational and inertial loads are decided using a fitness characteristic. In different phrases, every mass affords an answer, and the set of rules is navigated by using nicely adjusting the gravitational and inertia loads. By lapse of time, we anticipate that hundreds be attracted with the aid of the heaviest mass. This mass will present a most reliable solution within the search area. The GSA could be taken into consideration as an isolated machine of hundreds. It is sort of a small synthetic world of hundreds obeying the Newtonian laws of gravitation and movement. More precisely, masses obey the following laws [21][22][23]:

**2.1 Law of Gravity:** every particle attracts every other particle and the gravitational force among debris is immediately proportional to the made of their hundreds and inversely proportional to the distance among them, R. We use here R in preference to $R^2$,because consistent with our test consequences, R provides better outcomes than $R^2$ in all experimental cases.

**2.2 Law of motion:** the current velocity of any mass is same to the sum of the fraction of its previous velocity and the version within the velocity. Variation in the speed or acceleration of any mass is equal to the force acted on the device divided by mass of inertia.
Now, recall a device with N agents (masses). We outline the location of the ith agent by way of:

$X_i = X_i^1, \ldots\ldots X_i^d, \ldots\ldots X_i^n$

For i= 1,2,.....N, (1)

In which $X_i^d$ offers the location of ith agent within the dth measurement. At a specific time 't', we define the pressure acting on mass 'i' from mass 'j' as following:

$F_{ij}^d(t)$=G(t)$\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(t)+\varepsilon}X_j^d(t) - X_i^d(t)),$ (2)

wherein $M_{aj}$ is the active gravitational mass associated with agent j, $M_{pi}$ is the passive gravitational mass associated with agent i, G(t) is gravitational constant at time t, ε is a small steady, and $R_{ij}$ is the Euclidian distance between marketers i and j:

$R_{ij}(t) = \| X_i(t).X_J(t) \|$ (3)

To give a stochastic feature to our set of rules, we assume that the entire pressure that acts on agent (i) in a size d be a randomly weighted sum of dth components of the forces exerted from other agents:

$F_i^d(t) = \sum_{J=1,J\neq 1}^{N} rand_j F_{ij}^d(t)$ (4)

Where $rand_j$ is a random range in the c programming language [0,1]. Hence, through the regulation of motion, the acceleration of the agent i at time t, and in path dth, $a_i^d(t)$, is given as follows:

$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$ (5)

Where $M_{ii}$ is the inertial mass of ith agent.
Furthermore, the next velocity of an agent is taken into consideration as a fraction of its contemporary velocity delivered to its acceleration. Therefore, its function and its speed may be calculated as follows:

$V_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$ (6)

$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1)$ (7)

Where $rand_i$ is a uniform random variable in the interval [0,1]. The gravitational constant, G, is initialized at the beginning and can be decreased with time to control the accuracy. In different phrases, G is a function of the preliminary value ($G_0$) and time (t):

G(t)= G($G_0, t$) (8)

Gravitational and inertia masses are certainly calculated by way of the health evaluation. A heavier mass approach a more efficient agent. This means that better agents have higher attractions and stroll greater slowly. Assuming the equality of the gravitational and inertia mass, the values of masses are calculated the usage of the map of fitness. We replace the gravitational and inertial masses by the subsequent equations:

$M_{ai} = M_{pi} = M_{ii} = M_i$ i= 1, 2, ..... N;

$$m_{it} = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (9)$$

$$M_{it} = \frac{m_{it}}{\sum_{j=1}^{N} m_{jt}} \quad (10)$$

Where $fit_i(t)$ represent the fitness value of the agent i at time t.
worst(t) and best(t) are defined as follows (for a minimization problem):

$$best(t) = \min fit_j(t) \quad (11)$$

$$worst(t) = \max fit_j(t) \quad (12)$$

The distinct steps of the proposed algorithm are the followings:
(a) Search space identity.
(b) Randomized initialization.
(c) Fitness assessment of agents.
(d) Update G(t), best(t), worst(t) and $m_{it}$ for i = 1,2,...,N.
(e) Calculation of the full force in distinctive directions.
(f) Calculation of acceleration and velocity.
(g) Updating agents' role.
(h) Repeat steps c to g till the forestall standards is reached.
(i) End.

The principle of GSA is shown in Fig. 1. To see how the proposed algorithm is efficient a few remarks are referred to: – Since each agent may want to look at the performance of the others, the gravitational force is a information moving device. – Due to the force that acts on an agent from its neighborhood sellers, it could see area round itself. – A heavy mass has a big powerful appeal radius and as a result a fantastic intensity of enchantment. Therefore, agents with a higher performance have a greater gravitational mass. As an end result, the dealers generally tend to transport toward the first-class agent.

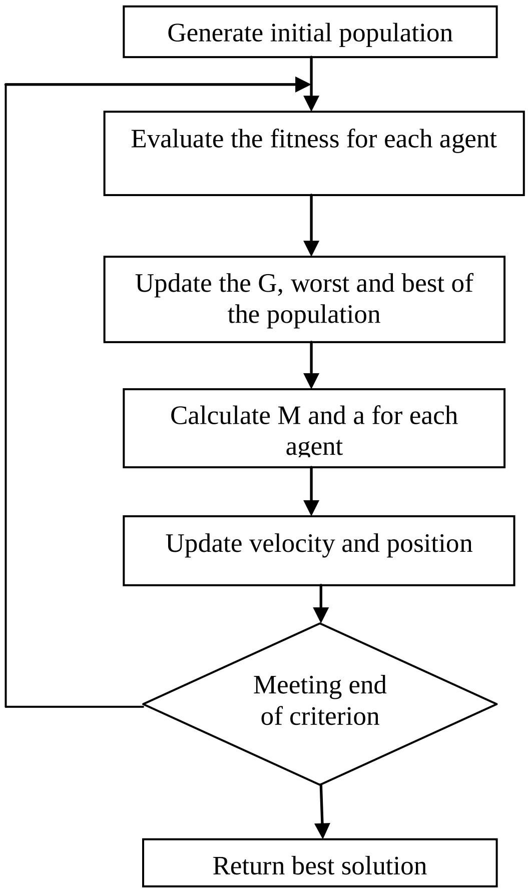



Fig. 1 General principle of GSA

The inertia mass is in opposition to the movement and make the mass motion gradual. Hence, agents with heavy inertia mass flow slowly and for this reason seek the gap greater domestically. So, it can be considered as an adaptive learning fee. – Gravitational regular adjusts the accuracy of the hunt, so it decreases with time (much like the temperature in a Simulated Annealing algorithm). – GSA is a memory-much less algorithm. However, it works correctly just like the algorithms with reminiscence. Our experimental results show the coolest convergence price of the GSA. – Here, we anticipate that the gravitational and the inertia hundreds are the equal. However, for a few packages unique values for them may be used. A larger inertia mass presents a slower motion of agents within the seek area and consequently a extra particular seek. Conversely, a larger gravitational mass reasons a higher appeal of agents. This lets in a quicker convergence. The gravity force causes a global movement where all objects move towards other objects with heavier masses.

## 3.0 Proposed Work

In this section we have defined 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 GSA optimization algorithmSince both are at different poles and no similarity or relation exists between them. out task is to bring them on a same pole and optimise the attributes selection for better accuracy using GSA optimisation. The equilibrium condition between them can be shown in figure 4.2.

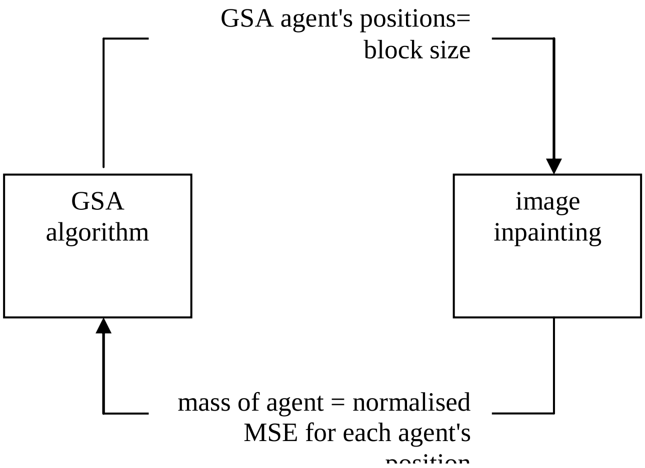



Figure 2: Representation of equilibrium of GSA optimisation and Image inpainting 's block size selection

GSA agents are number of trials in each iteration which we have for different block sizes. Initially these are randomly chosen and MSE is calculated for each block size option which is saved in the form of a matrix of size = (1× number of agents). Agents's position in this case is defined by 2 variables :block size width and height and both variables can be updated independently in each GSA iteration. For the first iteration these are randomly chosen and updated by using equation (1-9). For this new set of available block sizes, patch is filled and MSE is calculated again and saved into the previous matrix but in the next row making the matrix size = (2× number of agents). This way we get a matrix of size = (number of iteartions× number of agents). From this matrix the minimum error row is selected and corresponding block size is best suitable block size for the best matching of source selection for patch filling. The significance of GSA terminology with our application is tabulated in table 1. The flow chart of proposed method is shown in figure 8.

**Table 1: Significance of GSA Terminology In Block Size Tuning**

| **GSA terms** | **In Image Inpainting's Block Size Selection** |
|---|---|
| Agents Position | Block size |
| Dimension for optimisation/ number of variables to be tuned | Total number of dependent variables which is 2 in our case (block length and width) |
| Update in the position of agents | Change is block size to minimize the MSE in patch |

A complete step by step algorithm is explained below.

Step1.Load the patch mask and input image.

Step2.initialise the GSA parameters like number of terations, number of agents, initial G0 and alpha.

Step3.randomly initialise the agents new positions which must be block size.

Step4.call the objective function to fill the patch for selected block size and calculate the mean square error.

Step5.to update the random positions of agents, force and mass has to be calculated by using the equations

$$F_{ij}^{d}(t) = G(t)\,\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(t)+\varepsilon}X_j^{d}(t) - X_i^{d}(t)),$$

$$m_{it}=\frac{fit_i(t)-worst(t)}{best(t)-worst(t)}$$

the respective notations are given in previous chapter

Step6.The new updated position is obtained from the formula

$$X_i^{d}(t+1) = X_i^{d}(t)+V_i^{d}(t+1)$$

the velocity in this case is calculated by using acceleration which is based on force and masscalculated in previous step.

Step7.For this new updated position or values of weights and biases, objective function is again called and accuracy is saved.

Step8. The attributes' positions for which minimum of accuracy is obtained out of previous two set of values, is further considered for updating.

Step9.This process continues till all iterations are not completed.

Step10.The final minimum error is obtained and block size selected for them are used as final set of block which gives higher accuracy.

## 4.0 Results

We have tested the algorithm on many images with different sizes and different size of mask on it. Proposed work has been compared with recently used firefly algorithm for adaptive block size iteration and Markov random field (MRF) algorithm on same test images and our work showed the improvement over both. PSNR and MSE have been used as comparison par. Figure 3(a) shows the input image and figure 3 (b) shows the image with masks over it. We have iterated the image block size algorithm with GSA & firefly algorithm and a total of 40 iterations are done. Regenerated mask by these numbers of iterations is shown in figure 3 (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 3(b) shows the regenerated image with 40 iterations.

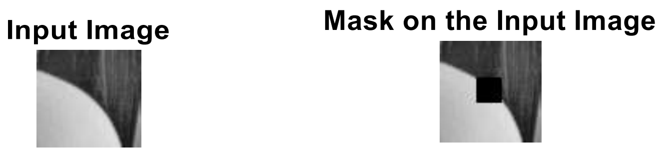



figure 3(a): Input Image (b): Masked Input image

Figure 4 : Masked image with regenerated mask with GSA algorithm in 40 iterations

The comparison is done on the criteria of PSNR, dis-similarity curve and mean square error (MSE) with MRF algorithm using the same adaptive block sizing method. The equation of MSE and PSNR is defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}(I(i,j)-K(i,j))^2 \qquad (13)$$

Where I(i,j) is the image pixel of original image and k(i,j) is the pixel of inpainted image.

International Journal of Information Movement Vol.2 Issue VIII December 2017)
Website: www.ijim.in ISSN: 2456-0553 (online) Pages 159-164

$$PSNR = 10log_{10}(\frac{max^2}{MSE}) \quad (14)$$

Where max is the maximum possible pixal value of the image

These curves are shown in figure 5 (a), (b) and (c) respectively. 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. If PSNR decreases with iterations then algorithm is not correct. Means square error curve in figure 5(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 5(b).

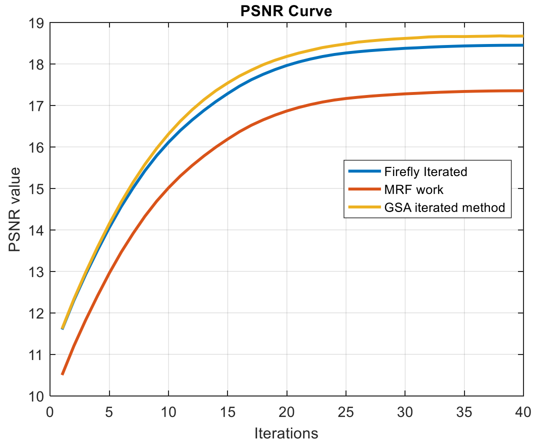



Figure 5(a): comparative PSNR curve

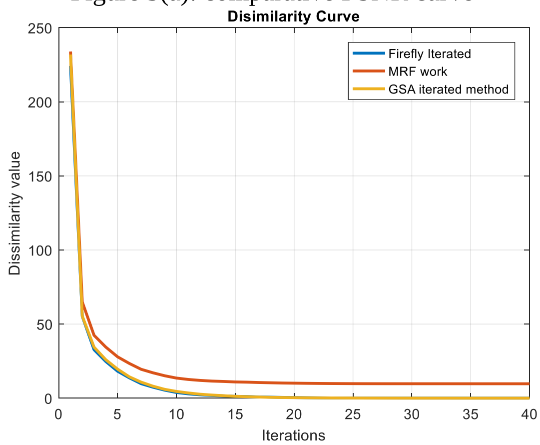



Figure 5 (b): comparative Dis-similarity curve

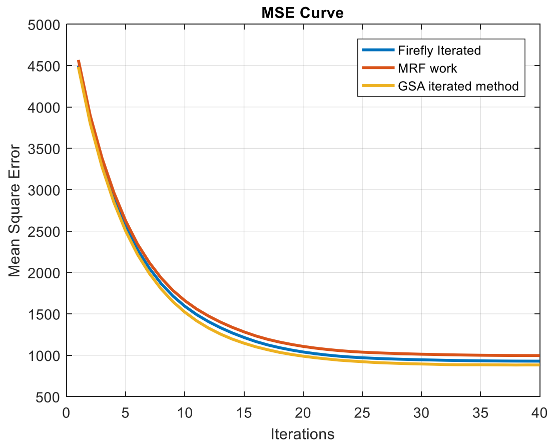



Figure 5(c): comparison in terms of MSE

With change in number of iterations, PSNR values also increase for the same masked image and so the MSE decreases too. Table 2 shows the final PSNR and MSE value for different number of total iterations. And a bar graph for them is plotted in figure 6.

Form the analysis of above table 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. Table 2 shows the images with their results.

## 5.0 Conclusion

It is concluded from results that. we have implemented and analyzed our proposed image inpainting algorithm with its possible contributions and compared with latest developed MRF inpainting algorithm. For the Bertalmio paper three contributions are found, these are using multi-resolution images, inpainting the unknown region inwards and estimating the amount of necessary iterations of the algorithm. These contributions all increase the speed of the algorithm.

- It has been found from the previous paper that local searching like firfly optimisation or MRF increases the speed but should decrease the visual appeal, although in the experiments it increased the visual quality. In our work we have used global searching operations using Gravitational Search Algorithm (GSA)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.
- The user study verified the observation made throughout the project. The technique has it flaws, a low error value means a visual pleasing result but a high error value can result in both a visual pleasing and an unpleasing result. Another conclusion that can be made from the observation is that the MSE is not a correct way of comparing different image inpainting methods.

**Table 2: Comparison of PSNR and MSE for different number of iterations for proposed work and MRF method**

| | Proposed Scheme | Firefly Optimised block selection | MRF scheme |
|---|---|---|---|
| | Iterations=40 | Iterations=40 | Iterations=40 |
| PSNR | 18.67147653 | 18.45069758 | 17.353114 |
| MSE | 882.9430814 | 928.9890669 | 996.6276229 |
| Dis-similarity factor | 0.002438105 | 0.002257917 | 9.718447364 |

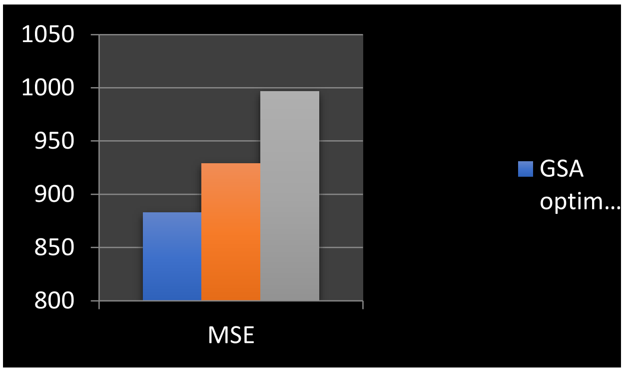
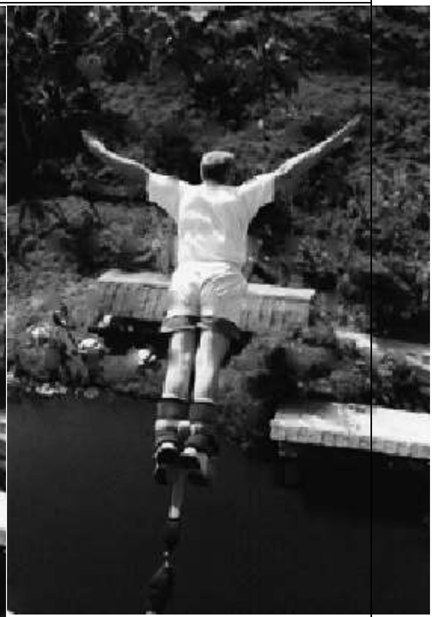
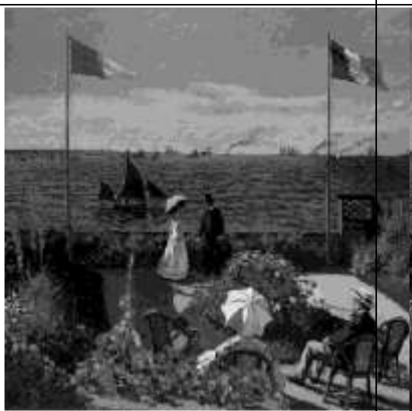
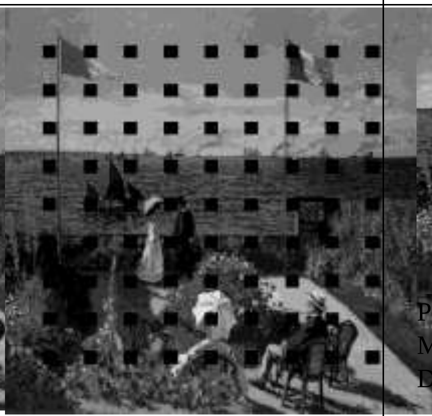
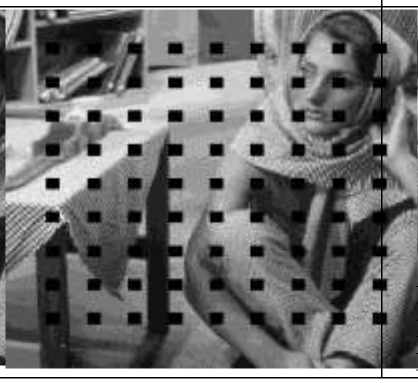



Figure 6: comparative bar chart for SME and PSNR for proposed work, firefly and MRF algorithm

Table 3: Image Inpainting Results For Different Images

| Input Image | Masked Image | Results in 40 iterations |
|---|---|---|
|  |  |  PSNR= 22.365067<br>MSE= 377.201695,<br>Dissimilarity value= 0.027678 |
|  |  |  PSNR= 23.456657<br>MSE= 293.369317,<br>Dissimilarity value= 0.027989 |
|  |  |  PSNR= 25.026725<br>MSE= 204.365641<br>Dissimilarity value= 0.0614 |

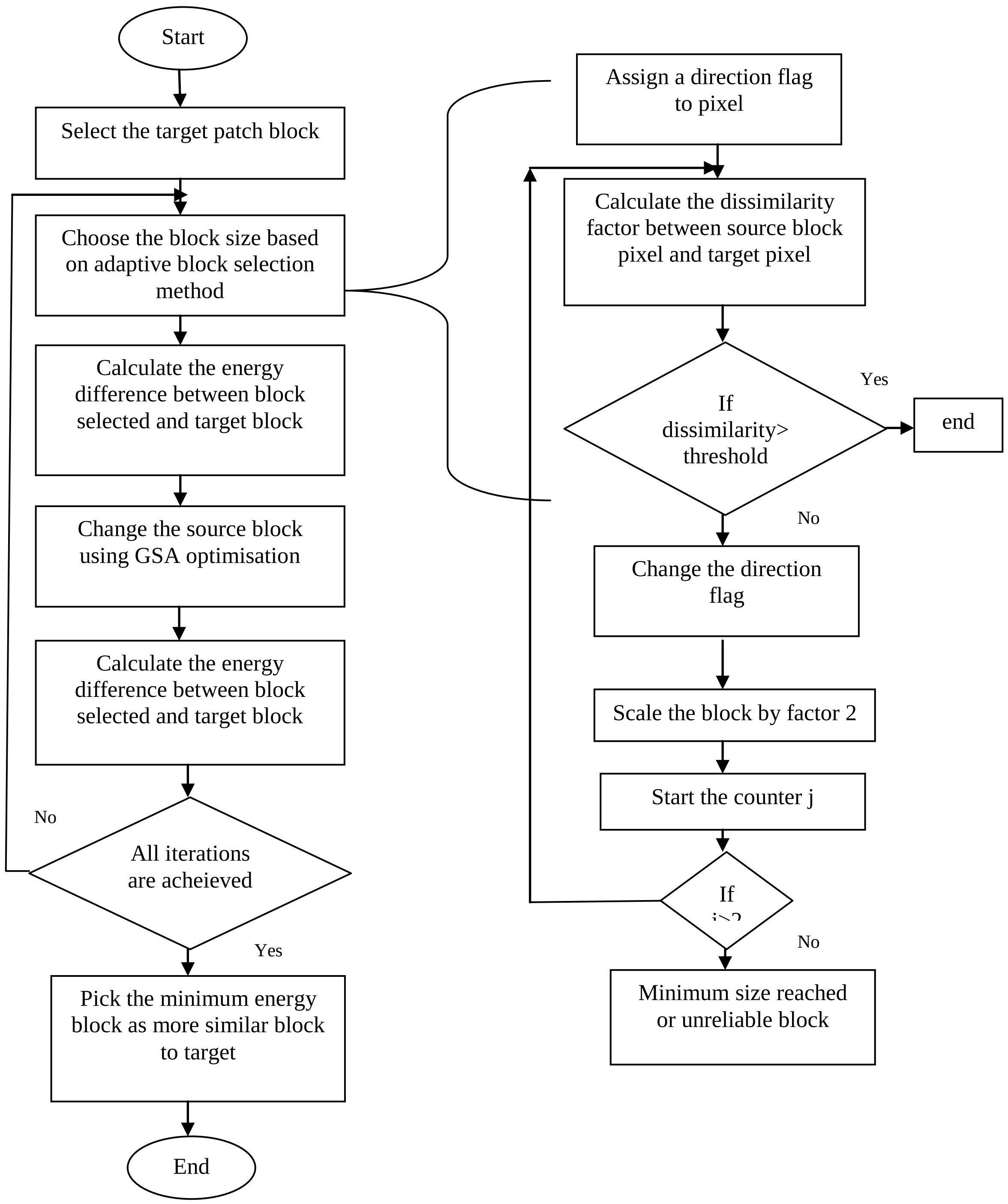



Figure 8 Flowchart of proposed work

## 6.0 References


[1].A. Criminisi, P. P_erez, and K. Toyama, \Region _lling and object removal by exemplar-based image inpainting," IEEE Transactions on Image Processing, vol. 13, pp. 1200-1212, 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, SIGGRAPH '00, (New York, NY, USA), pp. 417-424, ACM Press/Addison-Wesley Publishing Co., 2000.

[3].A. A. Efros and T. K. Leung, \Texture synthesis by non-parametric sampling," in Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99, (Washington, DC, USA), pp. 1033, IEEE Computer Society, 1999.
[4].M. M. Oliveira, B. Bowen, R. Mckenna, and Y. sung Chang, \Fast digital image inpainting," in Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), pp. 261-266, ACTA Press, 2001.
[5].P. P_erez, M. Gangnet, and A. Blake, \Poisson image editing," in ACM SIGGRAPH 2003 Papers, SIGGRAPH '03, (New York, NY, USA), pp. 313{318, ACM, 2003.
[6].A. Telea, \An image inpainting technique based on the fast marching method," Journal of Graphics, GPU, and Game Tools, vol. 9, no. 1, pp. 23-34, 2004.
[7].L.-Y. Wei and M. Levoy, \Fast texture synthesis using tree-structured vector quantization," in Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00, (New 61 York, NY, USA), pp. 479-488, ACM Press/Addison-Wesley Publishing Co., 2000.
[8].J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, \Image completion with structure propagation," ACM Trans. Graph., vol. 24, pp. 861-868, July 2005.
[9].J. Hays and A. A. Efros, \Scene completion using millions of photographs," ACM Transactions on Graphics (SIGGRAPH 2007), vol. 26, no. 3, 2007.
[10].K. A. Patwardhan, G. Sapiro, and M. Bertalmo, \Video inpainting of occluding and occluded objects," in International Conference on Image Processing, pp. 69-72, 2005.
[11].Y. Wexler, E. Shechtman, and M. Irani, \Space-time video completion," Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, pp. 120-127, 2004.
[12].P. Perona and J. Malik, \Scale-space and edge detection using anisotropic di_usion," IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, pp. 629-639, 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, pp. 3106-3115, December 2010.
[14].Q. Chen, Y. Zhang, and Y. Liu, \Image inpainting with improved exemplar-based approach," in Proceedings of the 2007 international conference on Multimedia content analysis and mining, MCAM'07, (Berlin, Heidelberg), pp. 242-251, Springer-Verlag, 2007.
[15].Anupam, P. Goyal, and S. Diwakar, \Fast and enhanced algorithm for exemplar based image inpainting," Image and Video Technology, Pacific-Rim Symposium on Image and Video Technology, vol. 0, pp. 325-330, 2010.
[16].T. Ružic, A. Pižurica, "Context aware image inpainting with application to virtual restoration of old paintings," IEICE International Conference Information and Communication Technology Forum ICTF, 2013, 29-31 May, Sarajevo, Bosnia
[17].Raluca Vreja and Remus Brad,"Image Inpainting Methods Evaluation and Improvement", The Scientific World Journal, Volume 2014.
[18].Jun Zhou, Antonio Robles-Kelly, "Image Inpainting Based on Local Optimisation" 2010 International Conference on Pattern Recognition.
[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. Volume 5, Number 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, Volume 36– No.4, December 2011
[22].Seema Kumari Singh , Prof J.V Shinde," Optimum Patch Selection Using GA in Exemplar Based Image In-painting" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015.
[23].B. Fergani, H.Bensuici, "A discrete firefly algorithm for geometric image inpainting". International Conference on Advanced Technologies For Signal & Image Processing Atsip 2014, At Sousse-Tunisia. Chongwu Tang, Xi Hu, Li Chen, Guangtao Zhai, Xiaokang Yang,"Sample-Based Image Completion Using Structure Synthesis", IEEE 2013
[24].Shivali Tyagi, Sachin Singh,"Image Inpainting By Optimized Exemplar Region Filling Algorithm" International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-6, January 2013
[25].Nuno Miguel Ventura Couto, "Inpainting-based Image Coding: A Patch-driven Approach", Dissertation submitted for Master in Electrical and Computer Engineering in 2010
[26].Sander van de Ven,"Image Inpainting", master Thesis in Utrecht University,2012.