International Journal of Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-7)
Research Article **July 2015**

Probability based Image Edge Detection using Modified PSOGSA Algorithm: A Review

[1]Kanica, [2]Beant Kaur, [3]Mandeep Kaur
[1]Student , [2, 3] A.P
[1, 2, 3,] Deptt. of ECE, Punjabi University,
Patiala, Punjab, India

Abstract-

This paper presents the review on image edge detection by using particle swarm optimization and gravitational search algorithm. The paper present the various works done by particle swarm optimization and gravitational search algorithm in the field of digital image processing. By observing the previous works the two optimization algorithms particle swarm optimization algorithm and gravitational search algorithm are combined known as PSOGSA. The basic PSOGSA can be modified on the basis of probability for the better results.

Keywords- Edge detection, particle swarm optimization algorithm, gravitational search algorithm, PSOGSA algorithm.

I. INTRODUCTION

Image segmentation is an attempt to partition an image into multiple segments which is also known as sets of pixels or super pixels. The goal of segmentation is to simplify or change the image into something that is more meaningful and easier to analyse. Edge detection and clustering are two basic segmentation methods among the various techniques. Edge detection is basically aimed at identifying the image brightness discontinuities especially along the edges where the intensity tends to change sharply. Canny edge detection is a premium technique used for detecting the edge in an image. Clustering is a process whereby a data set is replaced by clusters, which are collections of data points that "belong together". The specific criterion to be used depends on the application. Pixels may belong together because of the same colour or similarity measure. This work is an enhanced form of edge detection methodology that aids to obtain the best results out of the images.

The "Particle Swarm Optimization" model was brought into light for the first time by Russel Ebenhart and James Kennedy in 1995. As the name suggest the basic operation simulates the behaviour of flocks of birds or the sociological behaviour of a group of people or schooling of fish. Modelled on the mechanism of evolution and natural movement this algorithm provides an alternative tool for locating optimal solutions. Here swarm of several particles are used as the population to that of Genetic Algorithm in which some size to find the best solution which is analogous probable solutions are initialises as the population size. Each particle keeps track of their current positions which correspond to potential solutions of the function to be minimized. Each particle also keeps track of the speed and direction of travel by the particle. Similar to genetic algorithm a fitness value is also associated with each particle which directly depends on the particle's current position. Additionally each particle also remembers its personal best positions which on comparison with the personal best solutions of other particle are used to determine the overall best solution or population. Moreover, this algorithm is also meta-heuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. To be more specific it does not use gradient of the problem being optimized, which means PSO does not require that the optimization problem being differentiable as is required by classic optimization methods. So it can therefore be used on problems that are partially irregular, noisy, change over time, etc [9].Gravitational search algorithm (GSA) is a new population-based search algorithm inspired by universal law of gravity. Algorithm uses the theory of Newtonian gravity and its searcher agents are the collection of masses. In this paper, GSA and law of universal gravity is used to tackle the edge detection problem. Theory of universal gravity is used to detect edge pixels while the movement of agents are computed using the gravitational search algorithm [7].

II. PREVIOUS WORK

J. P.Papa, A. Pagnin, S. A. Schellini, A. Spadotto, R. C. Guido, M. Ponti, G. Chiachia and A. X. Falcao [1] Presented the feature selection of gravitational search algorithm (GSA). The proposed algorithm combines the optimization behaviour of GSA together with the speed of Optimum-Path Forest (OPF) classifier in order to provide a fast and accurate framework for feature selection. The main idea is to use the OPF accuracy over an evaluating set as the fitness function to guide GSA into searching the best solutions. As the number of masses increases, more reliable the solution vector obtained by GSA is. Thus, one needs a fast classifier to train all possible subset of features, denoted by the position of each mass particle. Although the algorithm presented here does not restrict OPF as the only classifier to be applied, it had decided to use it because of its efficiency in training. Another strong point is that OPF does not have parameters to be optimized, such as Support Vector Machines and some neural networks. This paper presents here the experiments to assess the robustness of the proposed GSA-based algorithm for feature selection. Experiments using five different datasets from a wide range of applications comparing the proposed approach with the additional OPF, OPF with Particle Swarm Optimization (OPF-PSO), OPF with Principal Component Analysis (OPF-CA), and OPF with Linear Discriminant Analysis (OPF-LDA) demonstrated that OPFGSA seems to be an interesting strategy for feature selection, since that no consensus was observed about which technique is better than the other to perform the given tasks.

Mahdi Setayesh, Mengjie Zhang and Mark Johnston [2] The goal of this paper is to investigate a new approach to the use of PSO technique for edge detection. Rather than using a simple track of a particular pixel with its neighbors at a time as in many existing edge detection operators, the new approach aims to operate on the entire possible edges or boundaries (entire fitting curves in images) at a time and improve the possible solutions via the automatic evolutionary learning in PSO. The main advantage of an algorithm is detection of edges in one step and there is no need for smoothing, enhancement and localization as pre-processing steps. This algorithm can be used for detecting edges for images with complex objects. The results show that this proposed algorithm outperforms the Sobel and homogeneity edge detectors on the images. In addition, the proposed PSO algorithm can detect complex edges in noisy images without using any pre-processing and post processing algorithms.

Mohammad Alipoor, Sajjad Imandoost and Javad Haddadnia [3] This paper presents a novel edge detection method based on Particle Swarm Optimization. Unlike classical filters that are set by intuitive knowledge, a new filter is proposed on the basis of evolutionary computation**.** A proper synthetic training image and its edge map are used to find an optimum edge filter. The advantage of this method is that an effective edge detection filter can be easily constructed. Exact and highly efficient edge filter is developed and comparative results are provided to certify superiority of this method over classis filters.

Mahdi Setayesh, Mengjie Zhang and Mark Johnston [4] Traditional edge detection approaches often result in broken edges especially in noisy images. This study presents a particle swarm optimisation based algorithm to detect edges continuously in such images. In this paper, a new objective function and a new encoding scheme are proposed to address noise and reduce broken edges. The newly developed algorithm is compared to a modified version of the Canny algorithm as a Gaussian-based edge detector based on Pratt's figure of merit measure. The algorithm to the images in different noise levels is applied and compares the outcome with the modified version of the Canny edge detection algorithm. The results show that our algorithm generally outperforms the Canny edge detector. In addition, our algorithm can detect edges in noisy images without using any pre-processing and post processing algorithms such as a Gaussian filter and an edge linking technique. However, the current version of the algorithm takes a relatively longer time than the Canny edge detector.

Seyedali Mirjalili and Siti Zaiton Mohd Hashim [5] a new hybrid population-based algorithm (PSOGSA) is proposed with the combination of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA). The main idea is to integrate the ability of exploitation in PSO with the ability of exploration in GSA to synthesize both algorithms strength. Some benchmark test functions are used to compare the hybrid algorithm with both the standard PSO and GSA algorithms in evolving best solution. The main idea is to integrate the abilities of PSO in exploitation and GSA in exploration. The results show that PSOGSA outperforms both in most function minimization. The results are also proved that the convergence speed of PSOGSA is faster that PSO and GSA.

Mahdi Setayesh, Mengjie Zhang and Mark Johnston [6] proposed an algorithm based on discrete particle swarm optimisation (PSO) to detect continuous edges in noisy images. A constrained PSO-based algorithm with a new objective function is proposed to address noise and reduce broken edges. The localisation accuracy of the new algorithm is compared with that of a modified version of the Canny algorithm as a Gaussian-based edge detector, the robust rank order (RRO)-based algorithm as a statistical based edge detector, and the previously developed PSO-based algorithm. Pratt's figure of merit is used as a measure of localisation accuracy for these edge detection algorithms. The results show that the new algorithm generally outperformed the Canny and RRO edge detectors and also a previous PSO algorithm for detecting edges in the images corrupted by the Gaussian and impulsive noises.

Om Prakash Verma and Rishabh Sharma[7] This paper present a new approach for edge detection using gravitational search algorithm (GSA) and universal law of gravity. The direct application of any edge detection algorithm on an entire image requires a huge memory which leads to optimization problems. Gravitational search algorithm is an optimization algorithm inspired by Newtonian gravity. Objects in space attract each other by way of gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. In the proposed approach the edges are detected using the law of universal gravity and movement of agents are computed using gravitational search algorithm. The proposed approach is able to detect the edge pixel in an image up to a certain extent. The proposed approach leads to a minimal set of input data to be processed thus making the process much faster and memory-efficient then the edge detection algorithm.

Fatemeh Deregeh and Hossein Nezamabadi-pour[8] In this paper, a new algorithm for image edge detection using Gravitational Search Algorithm (GSA) is proposed. In order to adapt the proposed algorithm to edge detection problem, some modification is applied on the original GSA. Each image pixel is considered as a celestial body and its mass is considered to be corresponding to the pixel's greyscale intensity. To find out the edgy pixels a number of agents are randomly generated and initialized through the image space. Artificial agents move through the space via forces of bodies that are located in their neighbourhood. The results confirmed that the proposed method outperforms the recent Ant edge detection algorithm and also compared to these algorithm, the used population in our algorithm is smaller. Besides, in comparison with Ant algorithm, the proposed method has less parameters and this makes the proposed algorithm more simple. Also in comparison with Sobel method the result of proposed algorithm has more accuracy.

Romesh Laishram[9] This paper attempts to pull out a new and a practical approach for enhancing the underlying delicate architectures of the human brain images captured by a Magnetic Resonance Imaging (MRI) machine in a much better way. Edge detection is a fundamental tool for the basic study of human brain particularly in the areas of feature detection and feature extraction. The edge detection methodology presented in this paper relies on two basic stages: Firstly, the original MRI image is subjected to image segmentation which is done using Particle Swarm optimization incorporating Fuzzy C Means Clustering (PSOFCM) technique. And secondly, canny edge detection algorithm is used for detecting the fine edges. After implementation it was found that this technique yields better edge detected image of the human brain as compared to other edge detection methods**.**

III. PSOGSA ALGORITHM [5]

In recent years, many heuristic evolutionary optimization algorithms have been developed. These include Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Deferential Evolution (DE), Ant Colony (AC), and Gravitational Search algorithm (GSA). The same goal for them is to find the best outcome (global optimum) among all possible inputs. In order to do this, a heuristic algorithm should be equipped with two major characteristics to ensure finding global optimum. These two main characteristics are exploration and exploitation.

1. Exploration is the ability of an algorithm to search whole parts of problem space.
2. Exploitation is the convergence ability to the best solution near a good solution.

The ultimate goal of all heuristic optimization algorithms is to balance the ability of exploitation and exploration efficiently in order to find global optimum. The aim of this paper is the image edge detection by combining the both particle swarm optimisation and gravitational search algorithm named PSOGSA. In PSOGSA, at first, all agents are randomly initialized. Each agent is considered as a candidate solution. After initialization, Gravitational force, gravitational constant, and resultant forces among agents are calculated. After that, the accelerations of particles are defined. In each iteration, the best solution so far should be updated. After calculating the accelerations and with updating the best solution so far, the velocities of all agents can be calculated. Finally, the positions of agents are defined . The process of updating velocities and positions will be stopped by meeting an end criterion. The steps of PSOGSA are represented in fig.1.

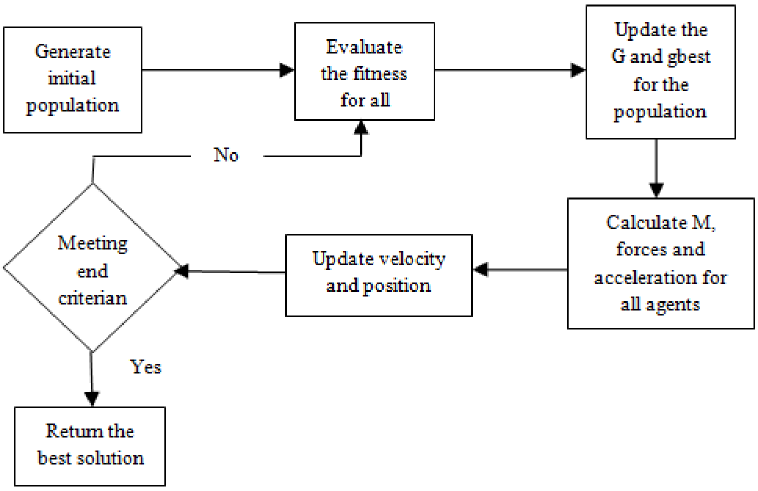

Figure 1 : Steps of PSOGSA.

To see how PSOGSA is efficient some remarks are noted as follow:

In PSOGSA, the quality of solutions (fitness) is considered in the updating procedure. The agents near good solutions try to attract the other agents which are exploring the search space. When all agents are near a good solution, they move very slowly. In this case, the gBest help them to exploit the global best. PSOGSA use a memory (gBest) to save the best solution has found so far, so it is accessible anytime. Each agent can observe the best solution so far and tend toward it. With adjusting weighting factors , the abilities of global search and local search can be balanced.

A. Particle Swarm Optimization Algorithm [5]

PSO is an evolutionary computation technique. The PSO was inspired from social behavior of bird flocking. It uses a number of particles (candidate solutions) which fly around in the search space to find best solution. Meanwhile, they all look at the best particle (best solution) in their paths. In other words, particles consider their own best solutions as well as the best solution has found so far. Each particle in PSO should consider the current position, the current velocity, the distance to *pbest*, and the distance to *gbest* to modify its position. PSO was mathematically modeled as follow:

$$v_i^{t+1} = w v_i^t + c_1 \times \text{rand} \times (pbest_i - x_i^t) + c_2 \times \text{rand} \times (\text{gbest} - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

Where v_i^t is the velocity of particle *i* at iteration *t*, *w* is a weighting function, c_i is a weighting factor, rand is random number between 0 and 1, x_i^t is the current position of particle *i* at iteration *t*, $pbest_i$ is the *pbest* of agent *i* at iteration *t*, and *gbest* is the best solution so far.

The first part of (1), wv_i^t , provides exploration ability for PSO. The second and third parts, c_1 × rand × ($pbest_i$ - x_i^t) and , c_2 × rand × (gbest - x_i^t) represent private thinking and collaboration of particles respectively. The PSO starts with randomly placing the particles in a problem space. In each iteration, the velocities of particles are calculated using (1). After defining the velocities, the position of masses can be calculated as (2). The process of changing particles' position will continue until meeting an end criterion.

Algorithm 1: PSO-based edge detection algorithm[7]:

1. Calculating $Scored$ for each pixel P on an image.
2. For each pixel P on an image do
3. If P is unprocessed and not marked as an edge then
4. Initialize PSO population in feasible search space randomly for pixel P
5. Repeat
6. For each particle (decoded as curve C) do
7. Evaluate $U(C)$ (9), $Score(C)$ (10) and $Curvature(C)$ (12)
8. Evaluate $Fitness(C)$ (13)
9. Update position and velocity of swarm leader if
   $Fitness(C) > Fitness(leader)$
10. For each particle (decoded as curve C) do
11. Find local best particle from neighbourhood
12. Calculate particle velocity (2)
13. Update particle position (1) and apply update rule (3)
14. If $G(C) \neq 0$ or $Prob(C) \leq HP$ then
15. Replace the particle with a new random feasible one
16. Until maximum iterations exceeded or minimum error criteria attained
17. Select best feasible particle in the population and decode it as curve C^*
18. Mark all pixels on curve C^* as an edge
19. If no feasible particle found then
20. Mark all pixels within the red rectangle as processed.

B. Gravitational Search Algorithm [5]

The basic physical theory which GSA is inspired from is the Newton's theory that states: Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. GSA can be considered as a collection of agents (candidate solutions) whose have masses proportional to their value of fitness function. During generations, all masses attract each other by the gravity forces between them. A heavier mass has the bigger attraction force. Therefore, the heavier masses which are probably close to the global optimum attract the other masses proportional to their distances. The GSA was mathematically modeled as follow. Suppose a system with N agents. The algorithm starts with randomly placing all agents in search space. During all epochs, the gravitational forces from agent *j* on agent *i* at a specific time *t* is defined as follow :

$$F_{ij}^t(\mathrm{t}) = \mathrm{G}\ (\mathrm{t})\ \frac{M_{pi}\ (t) \times M_{aj}\ (t)}{R_{ij}\ (t) + \varepsilon} \left(x_j^d\ (t) - x_i^d\ (t)\right), \qquad (3)$$

Where M_{aj} is the active gravitational mass related to agent *j*, M_{pi} is the passive gravitational mass related to agent *i*, G(t) is gravitational constant at time *t*, ε is a small constant, and R_{ij} (t) is the Euclidian distance between two agents *i* and *j*. The *G(t)* is calculated as (4):

$$\mathrm{G(t)} = G_0 \times \exp\ (-\alpha \times iter/maxiter) \qquad (4)$$

Where α and G_0 are descending coefficient and initial value respectively, *iter* is the current iteration, and *maxiter* is maximum number of iterations. In a problem space with the dimension *d*, the total force that acts on agent *i* is calculated as the following equation:

$$F_i^d(\mathrm{t}) = \sum_{j=1, j \neq i}^{N} rand_j\ F_{ij}^d\ (t), \qquad (5)$$

Where $rand_j$ is a random number in the interval [0,1]. According to the law of motion, the acceleration of an agent is proportional to the result force and inverse of its mass, so the acceleration of all agents should be calculated as follow:

$$ac_i^d\ (t) = \frac{F_i^d\ (t)}{M_i\ (t)}, \qquad (6)$$

Where *t* is a specific time and M_i is the mass of object *i*. The velocity and position of agents are calculated as follow:

$$vel_i^d\ (t+1) = rand_i \times vel_i^d(t) + ac_i^d(t), \qquad (7)$$

$$x_i^d\ (t+1) = x_i^d(\mathrm{t}) + vel_i^d\ (t+1), \qquad (8)$$

Where $rand_i$ is a random number in the interval [0,1]. In GSA, at first all masses are initialized with random values. Each mass is a candidate solution. After initialization, velocities for all masses are defined using (7). Meanwhile the gravitational constant, total forces, and accelerations are calculated as (4), (5), and (6) respectively. The positions of masses are calculated using (8). Finally, GSA will be stopped by meeting an end criterion.

Several steps of the GSAED are as follows[8].

(a) Problem space representation: The input image is considered as a two dimensional space, where each image pixel is considered as a celestial body. There are two kinds of bodies: fix bodies which stick on pixels of the image, and bodies that explore the image. The number of explorer bodies is determined by the user and in other case we clearly have one body for each pixel.

(b) Initializing explorer bodies distribution: h agents are located randomly in the image.

(c) Space travelling rule: Explorer agents travel through the image by moving from one place (pixel) to another place (pixel). There are two possibilities: each agent that is standing on a relatively smooth area (background pixels), it would be displaced to a new randomly chosen pixel. In another case, explorer body moves to one of their eight-neighborhood pixels.

(d) Mutation: For moving a body from one place to another, we consider a mutation probability, which makes the body moving vertically to direction of the exerted force.

(e) Field updating rule: After each step, the field level update in places where a body has crossed through them.

(f) Stopping criterion: The end of algorithm achieves by a pre-defined number of bodies changing place. Each cycle contains a fix number of steps. The number of cycles is chosen to be one. This value works well enough for all the experiments. After satisfying stopping criterion, the value of edginess for each pixel could be determined via the following equation:

$$\text{Edge}= E/Nm$$

where *Nm* is a normalization matrix, which its elements illustrate how many bodies passed through a special pixel.

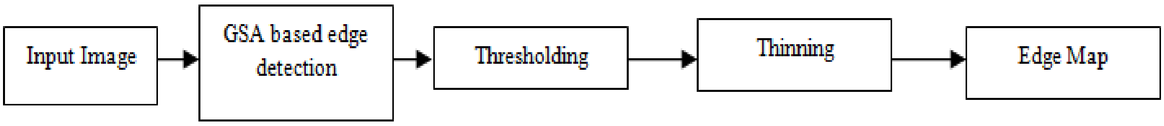

Figure 2: Edge detection using GSAED algorithm.

IV. CONCLUSION

This paper gives the review on image edge detection techniques in which the various researches using particle swarm optimization algorithm and gravitational search algorithm are discussed. The future work can be done using modified PSOGSA for image edge detection for better results. The basic idea of PSOGSA is to combine the ability of social thinking (*gbest*) in PSO with the local search capability of GSA. The basic PSOGSA can be modified on the basis of probability for the better results.

REFRENCES

[1] P.Papa, A. Pagnin, S. A. Schellini, A. Spadotto, R. C. Guido, M. Ponti, G. Chiachia and A. X. Falcao, “Feature Selection Through Gravitational Search Algorithm”, *International Conference*, pp. 2052-2055, 2009.

[2] Mahdi Setayesh, Mengjie Zhang and Mark Johnston, “A New Homogeneity-Based Approach to Edge Detection Using PSO,” *International Conference Image and Vision Computing*, pp.231-237, 2009.

[3] Mohammad Alipoor, Sajjad Imandoost and Javad Haddadnia,“Designing Edge Detection Filters Using Particle Swarm Optimization,” *International Conference*, 2010.

[4] Mahdi Setayesh, Mengjie Zhang and Mark Johnston, “Improving Edge Detection Using Particle Swarm Optimisation”, *International Conference*, 2010.

[5] Seyedali Mirjalili and Siti Zaiton Mohd Hashim, “A New Hybrid PSOGSA Algorithm for Function Optimization”, *International Conference*, pp.374-377, 2010.

[6] Mahdi Setayesh, Mengjie Zhang and Mark Johnston, “Edge Detection Using Constrained Discrete Particle Swarm Optimisation in Noisy Images”, *International Conference*, pp.246-253, 2011.

[7] Om Prakash Verma and Rishabh Sharma, “Newtonian Gravitational Edge Detection using Gravitational Search Algorithm”, *International Conference*, pp.184-188, 2012.

[8] Fatemeh Deregeh and Hossein Nezamabadi-pour, “GSAED a new approach for digital image edge detection*”,6'th International Symposium on Telecommunications*, pp.887-891, 2012.

[9] Romesh Laishram, “A Novel MRI Brain Edge Detection Using PSOFCM Segmentation and Canny Algorithm”, *International Conference*, pp.398-402, 2014.

