An Optimal Edge Detection Using Gravitational Search Algorithm

Om Prakash Verma, Rishabh Sharma, Manoj Kumar, and Neetu Agrawal

Abstract—Edge detection is a fundamental tool used in most image processing applications to obtain information from the image as a precursor step to feature extraction and object segmentation. Gravitational search algorithm (GSA) is a new population-based search algorithm inspired by Newtonian gravity. Algorithm uses the theory of Newtonian gravity and its searcher agents are the collection of masses. Masses attract each other by way of gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. In the proposed approach the edges are detected by the local variation in intensity values and the movement of agents is computed using gravitational search algorithm. The proposed approach is able to detect the edge pixel in an image up to a certain extent. The technique can be further extended for finding more edge pixels by modifying the gravitational search algorithm.

Index Terms—Gravitational search algorithm, edge, intensity, Fitness function, optimization.

I. INTRODUCTION

Edge detection is a fundamental of low-level image processing and good edges are necessary for higher level of image processing [1]. Edges are one of the most important visual clues for interpreting images [2]. The process of edge detection reduces an image to its edge details that appear as the outlines of image objects that are often used in subsequent image analysis operations for feature detection and object recognition.

A very important role is played in image analysis by what are termed feature points, pixels that are identified as having a special property. Feature points include edge pixels as determined by the well-known classic edge detectors of Sobel [3], Prewitt [4], Kirsch [5], Canny [6] etc. Russo [7], [8] and Russo and Ramponi [9], designed fuzzy rules for edge detection. Such rules can smooth while sharpening edges, but require a rather large rule set compared to simpler fuzzy methods. Abdallah and Ayman [10] introduced a fuzzy logic reasoning strategy for the edge detection in the digital images without determining a threshold.

Manuscript received December 12, 2012; revised March 27, 2013.

Om Prakash Verma is with the Department of Information Technology, Delhi Technological University (formally DCE), New Delhi, DL 110042, India (e-mail:opverma.dce@gmail.com).

Rishabh Sharma is with the Ericsson India Global Services Pvt. Ltd, Noida, India (e-mail:rishabhs.32@gmail.com).

Manoj Kumar is with the Department of Computer Engineering, Delhi Technological University (formally DCE), New Delhi, DL 110042, India (e-mail:mkg1109@rediffmail.com).

Neetu Agrawal is with the Department of Electronics and Communication, Ambedkar Institute of Technology, New Delhi, DL 110031, India (e-mail:neetu24@gmail.com).

Over the last decades, there has been a growing interest in algorithms inspired by the behaviors of natural phenomena. It is shown by many researchers that these algorithms are well suited to solve complex computational problems. Genyun Suna *et al.* [11] have introduced an edge detection algorithm based on the law of universal gravity in 2007. This algorithm assumes that each image pixel is a celestial body with a mass represented by its grayscale intensity. Verma *et al.* [12] have also developed a novel fuzzy system for edge detection in noisy image using bacterial foraging. Another evolutionary technique known as Particle Swarm Optimization (PSO) [13] employs a swarming in which the movements of the particles are guided by the swarm's local best position as well as global best position in the required search-space. Verma *et al.* [14] have also developed a new approach for edge detection using fuzzy derivative and Ant Colony Optimization. (ACO) algorithm to reduce the discontinuities presented in the image filtered by Sobel operator. Recently Verma *et al.* [15] proposed a new optimal approach for edge detection using universal law of gravity and ant colony optimization. In this approach, the theory of universal gravity is used to calculate the heuristic function which guides the ant towards edge pixels.

Edge detection aims to localize the boundaries of objects in an image and is a basis for many image analysis and machine vision applications. Conventional approaches to edge detection are computationally expensive because each set of operations is conducted for each pixel. In conventional approaches, the computation time quickly increases with the size of the image.

Gravitational search algorithm [16] is an optimization algorithm inspired by Newtonian gravity. Masses cooperate using a direct form of communication, through gravitational force of attraction. Each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertia masses. The lighter masses tend to get attracted towards heaviest mass. The heavier mass presents an optimum solution in the search space.

In this paper, GSA is used to tackle the edge detection problem in optimal manner. Local variation of image intensity value is used to detect edge pixels while the movement of agents are computed using the gravitational search algorithm.

The rest of the paper is organized as the follows. The gravitational search algorithm is briefly reviewed in Section II. The algorithm of the proposed edge detector is presented in Section III. The experimental results are given in Section IV and conclusions are drawn in Section V.

II. GRAVITATIONAL SEARCH ALGORITHM

In GSA, Newtonian laws of gravity are applied to find the optimum solution by a set of agents called masses [16]. In GSA, each mass (agent) has four characteristics namely; 1) position, 2) inertial mass, 3) active gravitational mass, and 4) passive gravitational mass. The position of the mass corresponds to a solution of the problem, and the fitness function is used to determine the gravitational and inertial masses.

The GSA algorithm is mainly comprises of the following steps:

- 1) Identification of search space.
- 2) Initialization.
- 3) Agent evaluation using fitness function.
- 4) Update $G(t)$, $best(t)$, $worst(t)$ and $M_i(t)$ for $i=1,2,..N$.
- 5) Calculation of the total force in all possible directions.
- 6) Acceleration and velocity calculations.
- 7) Updating agents' position.
- 8) Repeat steps 3 to 7 until the stop criterion is reached.
- 9) End

Consider a system with N masses in which position of the i^{th} mass is defined as follows [16]:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \text{ for } i=1, 2, \dots, N \quad (1)$$

where x_i^d is position of the i^{th} mass in the d^{th} dimension and n is dimension of the search space.

Mass of each agent is calculated after computing current population's fitness as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \quad (2)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (3)$$

where $M_i(t)$ and $fit_i(t)$ represent the mass and the fitness value of the agent i at time t , respectively. In case of minimization problem, $worst(t)$ and $best(t)$ are defined as follows:

$$worst(t) = \max_{i \in \{1, \dots, N\}} fit_i(t) \quad (4)$$

$$best(t) = \min_{i \in \{1, \dots, N\}} fit_i(t) \quad (5)$$

The net force, applied by mass j on mass i , $f_{ij}(t)$, is calculated as:

$$f_{ij}(t) = G(t) \frac{M_i(t) M_j(t)}{R_{ij}(t) + \varepsilon} (x_j - x_i) \quad (6)$$

here x_i is the position vector of the i^{th} agent, ε is a small threshold, and $G(t)$ is the gravitational constant, initialized at the beginning of the algorithm. The $G(t)$ is decreased with time to control the search accuracy. $R_{ij}(t)$ is the

Euclidian distance between two masses i and j . Using the Newton's second law of motion, the total gravitational acceleration on the i^{th} agent, is calculated as:

$$a_i(t) = G(t) \sum_{j=1}^n rand_j \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j - x_i) \quad (7)$$

where r and j is a random number within the interval $[0, 1]$, considered to add some stochastic behavior to the acceleration. The velocity and position of the agents are updated as:

$$v_i(t+1) = rand_i v_i(t) + a_i(t) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

k_{best} is the set of first K agents with the best fitness value and biggest mass, which is a function of time, initialized to k_0 at the beginning and decreasing with time. Here k_0 is set to N (total number of agents) and is decreased linearly to 1.

III. THE PROPOSED IMAGE EDGE DETECTION APPROACH

The proposed approach using the gravitational search algorithm leads to a minimal set of input data to be processed. The proposed GSA-based image edge detection approach aims to utilize a set of agents called masses to move on a 2-D image under the influence of gravity.

Algorithm

STEP 1: Initialize the positions of N agents by randomly distributing them on an image I . Initialize the value of the gravitational constant $G(t_0)$

STEP 2: Set the value of k_{best} equal to the number of agents

STEP 3: While ($k_{best} \neq 1$)

STEP 4: For each agent:

- 1) Calculate the total force applied on a single agent by all the other agents in the search space: In this approach total force on an agent is equal to the sum of the force exerted by all the agents individually on a single agent. Force exerted by an agent i at location (x, y) on agent j at location (x_1, y_1) is calculated using the Eq. 6, Where $M_i(t)$ and $M_j(t)$ are the grey value of the pixels located at (x, y) and (x_1, y_1) . $R_{ij}(t)$ is the Euclidian distance between two agents i and j which is calculated using the eq.

$$\|\bar{R}_{ij}\| = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \quad (10)$$

- 2) The total force exerted on an agent is calculated as:

$$f_i(t) = \sum_{j=1, j \neq i}^N rand_j f_{ij}(t) \quad (11)$$

- 3) *Calculate the acceleration and velocity of an agent:* In this approach the acceleration of each individual agent is calculated using the Eq. 7. Using this acceleration, velocity of the individual agent is calculated using the Eq.8.
- 4) *Update the agent's position:* Each individual agent's position is updated using Eq. 9. where velocity is calculated in the previous step.
- 5) *Updating $M_i(t)$, $G(t)$:* As each individual agent's position is updated. The grey value of the pixel on which it lands on becomes its new mass. The heavy masses which correspond to good solutions move more slowly than lighter ones. The $G(t)$ is updated as:

$$G(t) = G(t_0) \left(\frac{t_0}{t} \right)^\beta \quad \beta < 1 \quad (12)$$

- 6) *Fitness evaluation of agents:* In the proposed approach edges are detected using the local variations of image intensity value. For each agent we consider a 3×3 neighborhood (Fig. 1). Information required to find out whether a pixel is an edge pixel or not is determined by the local statistics at that position [17]:

$$f_{fitness} = \frac{V_c(I_{x,y})}{V_{max}} \quad (13)$$

where $I_{x,y}$ is the intensity value of the pixel at (x,y) . $V_c(I_{x,y})$ is a function that operates on the local group of pixels (Fig. 1) around the pixel (x, y) . It depends on the variation of the intensity values on the local group, and is given by [17]

$$V_c(I_{x,y}) = |I_{x-1,y-1} - I_{x+1,y+1}| + |I_{x-1,y} - I_{x+1,y}| + |I_{x-1,y+1} - I_{x+1,y-1}| + |I_{x,y-1} - I_{x,y+1}| \quad (14)$$

V_{max} is the maximum intensity variation in the whole image and serves as a normalization factor.

Fig. 1. A local configuration for computing the intensity variation at (x, y) .

STEP 5: End for loop.

STEP 6: Update k_{best} .

Decreasing the value of k_{best} (no. of agents) by removing agents that did not lead to edge pixels.

The flowchart of the proposed algorithm is shown in Fig. 2.

Fig. 2. Proposed algorithm.

IV. EXPERIMENTAL RESULTS

Conventional approaches to edge detection are computationally expensive because each set of operations is conducted for each pixel. The proposed approach is more focused on the optimization of the edge detection problem. Furthermore, various parameters of the proposed approach are set as follows [16]:

Fig. 3. (a) Original Cameraman image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Edge detection based on the theory of universal gravity and Ant colony optimization (threshold: default) and (g) The proposed approach.

Fig. 4. (a) Original lena image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Edge detection based on the theory of universal gravity and ant colony optimization (threshold: default) and (g) The proposed approach.

$s = \sqrt{M \times M \times 2}$ (the total number of agents), $\Omega = 8$ -connectivity neighborhood, $G(t_0) = 1$, $\beta = 0.2$, $N = 50$, $rand_j$ = rand function (leads to a value between 0 and 1).

The performance of the proposed technique is compared against that of the traditional edge detectors such as Canny, Edison, Rothwell, and SUSAN. The performance is also

compared with the edge detector using universal law of gravity and ant colony optimization [15]. In terms of number of edges, the traditional operators performs slightly better than the proposed approach but the proposed approach gives better results than the edge detector using the universal law of gravity and ant colony optimization. The proposed approach leads to less time consumption and less memory exhaustive

then the traditional operators and thus aptly focusing on the optimization concerns of the edge detection problem.

For images in Fig. 3 and Fig. 4, the captions are as follows: (a) the original image, (b) Canny Edge Detector, (c) Edison Edge Detector, (d) Rothwell Edge Detector, (e) SUSAN Edge Detector, (f) edge detection based on the theory of universal gravity and ant colony optimization (threshold: default) and (g) the result of the proposed approach. The value of threshold for the standard Canny, Edison, Rothwell, and SUSAN operator is selected as default value which gives the ideal edge map. The traditional edge detectors are implemented using the MATLAB toolbox. The proposed method and edge detection using universal law of gravity and ant colony optimization is also coded in the MATLAB.

It is evident from the results that our method finds meaningful edges in most images but is partially successful in presenting the complete edge map.

V. CONCLUSION

In this paper, a new GSA based image edge detection approach has been successfully developed. The proposed approach uses local variations of image intensity in-order to detect edges. The movement of the agents is dependent on the gravitational search algorithm. The proposed approach leads to a minimal set of input data to be processed thus making the process much faster and memory-efficient than the edge detection algorithm [15].

REFERENCES

- [1] M. B. Ahmad and T. S. Choi, "Local Threshold and Boolean Function Based Edge Detection," *IEEE Transactions on Consumer Electronics*, vol. 45, no 3, pp. 674 -679, August 1999.
- [2] E. Gose, R. Johnsonbaug, and S. Jost, *Pattern Recognition and Image Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1996, pp. 298.
- [3] I. Sobel, "Camera Models and Perception," Ph.D. thesis, Stanford University, Stanford, CA, 1970.
- [4] J. M. S. Prewitt, "Object enhancement and extraction," in B. S. Lipkin and A. Rosenfeld (Eds.), *Picture Analysis and Psychopictorics*, New York, NY: Academic Press, pp. 75-149, 1970.
- [5] R. Kirsch, "Computer determination of the constituent structure of biological images," *Computers and Biomedical Research*, vol. 4, pp. 315-328, 1971.
- [6] J. Canny, "Computational Approach to Edge Detection," *IEEE Transaction on Pattern Analysis and Machine intelligence*, no. 6, pp. 679-698, 1986.
- [7] F. Russo, "A new class of fuzzy operators for image processing," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, pp. 815-820, 1993.
- [8] F. Russo, "A user-friendly research tool for image processing with fuzzy rules," in *Proc. the First IEEE International Conference on Fuzzy Systems*, San Diego, 1992, pp. 561-568.
- [9] F. Russo and G. Ramponi, "Fuzzy operator for sharpening of noisy images," *IEE Electron. Lett.*, vol. 28, pp. 1715-1717, 1992.
- [10] A. A. Abdallah and A. A. Ayman, "Edge detection in digital images using fuzzy logic techniques," *World Academy of Sci. Eng. Technol.*, vol. 51, pp. 178-186, 2009.
- [11] G.-Y. Suna, Q.-H. Liua, Q. Liua, C.-Y. Jib, and X.-W. Lia, "A novel approach for edge detection based on the theory of universal gravity," *Pattern Recognition*, vol. 40, no. 10, pp. 2766-2775, 2007.
- [12] O. P. Verma, M. Hanmandlu, and S. A. K. Dhruv, "A Novel Fuzzy Ant System for Edge Detection," *IEEE/ACIS 9th International Conference on Computer and Information Science*, pp. 228 - 233, 2010.
- [13] M. Setayesh, M. Zang, and M. Jonhnton, "A new homogeneity-based approach to edge detection using PSO," in *Proc. 24th International Conference Image and Vision Computing*, New Zealand (IVCNZ), 2009, pp. 231-236.
- [14] O. P. Verma, M. Hanmandlu, P. Kumar, and S. Srivashtav, "A novel approach for edge detection using ant colony optimization and fuzzy derivative technique," *Advance Computing Conference, IEEE international*, pp. 1206-1212, 2009.
- [15] O. P. Verma and R. Sharma, "An optimal edge detection using universal law of gravity and Ant colony algorithm," *World congress on communication and information technologies (WICT)*, 2011.
- [16] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, pp. 2232-2244, 2007.
- [17] H. Nezamabadi-pour, S. Saryazdi, and E. Rashedi, "Edge Detection Using Ant Algorithms," *Soft Computing*, vol. 10, pp. 623-628, 2006.

Om Prakash Verma is the Head of the Department of Information Technology, Delhi Technological University (formerly DCE) having experience of more than 20 years in the field of teaching. Mr. Om Prakash Verma specializes in Image Processing, Signal Processing, Soft Computing and Artificial Intelligence. He has graduated from MNIT Jaipur and did his masters from Indian Institute of Technology Delhi (IITD). He has done his Phd from Delhi University. His work has been published in more than 40 journals and conferences proceedings. Under his guidance, more than 25 students have completed their Masters.