IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
ISSN: 2319 – 4200, ISBN No. : 2319 – 4197 Volume 1, Issue 3 (Nov. - Dec. 2012), PP 38-45
www.iosrjournals.org

Image Enhancement using Ant Colony Optimization

Kanika Gupta[1], Akshu Gupta[2]

[1] *(Electronics & Communication Engineering, Punjab Technical University, India)*
[2]*(Electrical Engineering, Thapar University, India)*

Abstract: *Applications of the Ant Colony Optimization (ACO) to solve image processing problem with a reference to a new automatic enhancement technique based on real-coded particle ant colony is proposed in this paper. The enhancement process is a non-linear optimization problem with several constraints. The objective of the proposed ACO is to maximize an objective fitness criterion in order to enhance the contrast and detail in an image by adapting the parameters of a novel extension to a local enhancement technique. The feasibility of the proposed method is demonstrated and compared with Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) based image enhancement technique. The obtained results indicate that the proposed ACO yields better results in terms of both the maximization of the number of pixels in the edges and the adopted objective evaluation. Computational time is also relatively small in the ACO case compared to the GA and PSO case.*

Keywords: *Ant Colony Optimization, Particle Swarm Optimization, Genetic Algorithms, Image enhancement*

I. Introduction

Genetic Algorithms (GA) have been used to evolve computer programs for specific tasks, and to design other computational structures. The recent resurgence of interest in AP with GA has been spurred by the work on Genetic Programming (GP). GP paradigm provides a way to do program induction by searching the space of possible computer programs for an individual computer program that is highly fit in solving or approximately solving the problem at hand. The genetic programming paradigm permits the evolution of computer programs which can perform alternative computations conditioned on the outcome of intermediate calculations, which can perform computations on variables of many different types, which can perform iterations and recursions to achieve the desired result, which can define and subsequently use computed values and subprograms, and whose size, shape, and complexity is not specified in advance. GP use relatively low-level primitives, which are defined separately rather than combined a priori into high-level primitives, since such mechanism generate hierarchical structures that such mechanism generate hierarchical structures that would facilitate the creation of new high-level primitives from built-in low-level primitives.

Particle Swarm Optimization (PSO) is one of the modern heuristic algorithms that can be applied to non linear and non continuous optimization problems. It is a population-based stochastic optimization technique for continuous nonlinear functions. PSO was developed in 1995 by Dr. James Kennedy, a social psychologist, and Dr. Russell Eberhart, an electrical engineer. PSO term refers to a relatively new family of algorithms that may be used to find optimal (or near optimal) solutions to numerical and qualitative problems. It is easily implemented in most programming languages and has proven both very effective and quick when applied to a diverse set of optimization problems. PSO was discovered through simulation of a simplified bird flocking model. Dr. Kennedy and Dr. Eberhart stated in Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both Genetic Algorithms (GAs) and Evolutionary Programming (EP). Unlike GAs and EP, PSO is a simple concept and is very easy to implement. The developers of PSO stated in Particle swarm optimization as developed by [Kennedy and Eberhart] comprises a very simple concept, and paradigms can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed.

Ant Colony Optimization (ACO) is the result of research on computational intelligence approaches to combinatorial optimization originally conducted by Dr. Marco Dorigo, in collaboration with Alberto Colorni and Vittorio Maniezzo. The fundamental approach underlying ACO is an iterative process in which a population of simple agents repeatedly construct candidate solutions; this construction process is probabilistically guided by heuristic information on the given problem instance as well as by a shared memory containing experience gathered by the ants in previous iteration. ACO has been applied to a broad range of hard combinatorial problems. Problems are defined in terms of components and states, which are sequences of components. Ant Colony Optimization incrementally generates solutions paths in the space of such components, adding new components to a state. Memory is kept of all the observed transitions between pairs of solution components and a degree of desirability is associated to each transition depending on the quality of the solutions in which it occurred so far. While a new solution is generated, a component y is included in a state, with a probability that is proportional to the desirability of the transition between the last component included in the state, and y itself. The main idea is to use the self-organizing principles to coordinate populations of artificial agents that collaborate to solve computational problems. Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. The rules specifying the interactions among the system's constituent units are executed on the basis of purely local information, without reference to the global pattern, which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence. For example, the emerging structures in the case of foraging in ants include spatiotemporally organized networks of pheromone trails. In this paper, a real-coded ACO is applied to adapt the gray-level intensity transformation in the image.

II. Genetic Programming

The best optimal hybrid filter is chosen with the help of GA. GA's, discovered by Charles Darwin are based on natural selection. They employ natural selection of fit best individuals as optimization problem solver. Optimization is performed through natural exchange of genetic material between parents. Off springs are formed from parent's genes. Fitness of off springs is evaluated. The fittest individuals only are allowed to breed. In computer world, selection is done by fitness function. Matching of parents is represented by cross-over and mutation operation. The GA has good global search ability but lacks local search ability. The process of GA is shown in Figure 1.

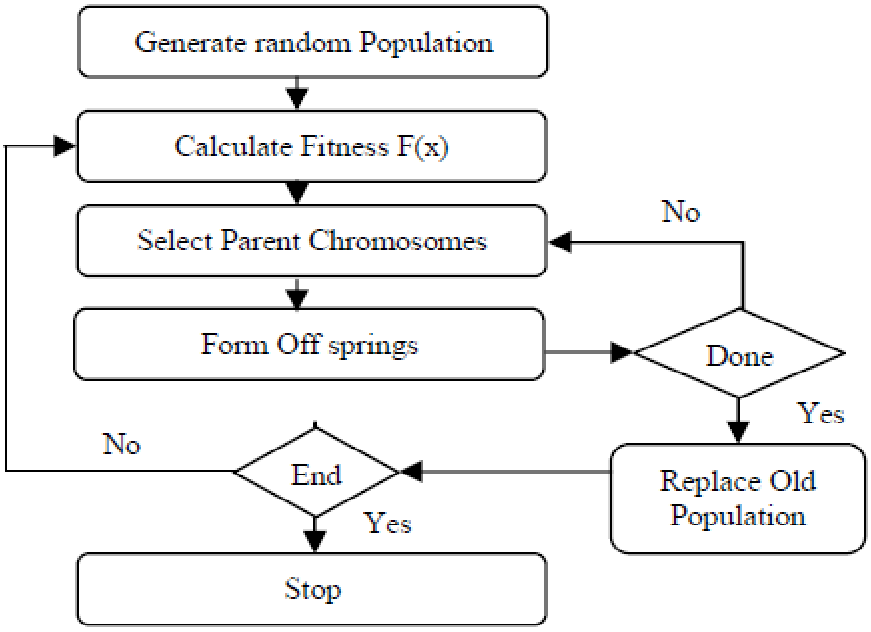


Figure 1: Process of Genetic Algorithm

III. Partcle Swarm Optimization Algorithm

PSO is initialized with a group of random particles (solutions). The algorithm then searches for optima through a series of iterations. The particle's fitness value is evaluated on each iteration. If it is the best value the particle has achieved, the particle stores the location of that value as *pbest* (particle best). The location of the best fitness value achieved by any particle during any iteration is stored as *gbest* (global best). Using *pbest* and *gbest*, each particle moves with a certain velocity, calculated by Equations 1, 2, and 3.

$$Vi = wV_{i-1} + c1 * rand() * (pbest - pL) + c2 * rand() * (gbest - pL) \tag{1}$$

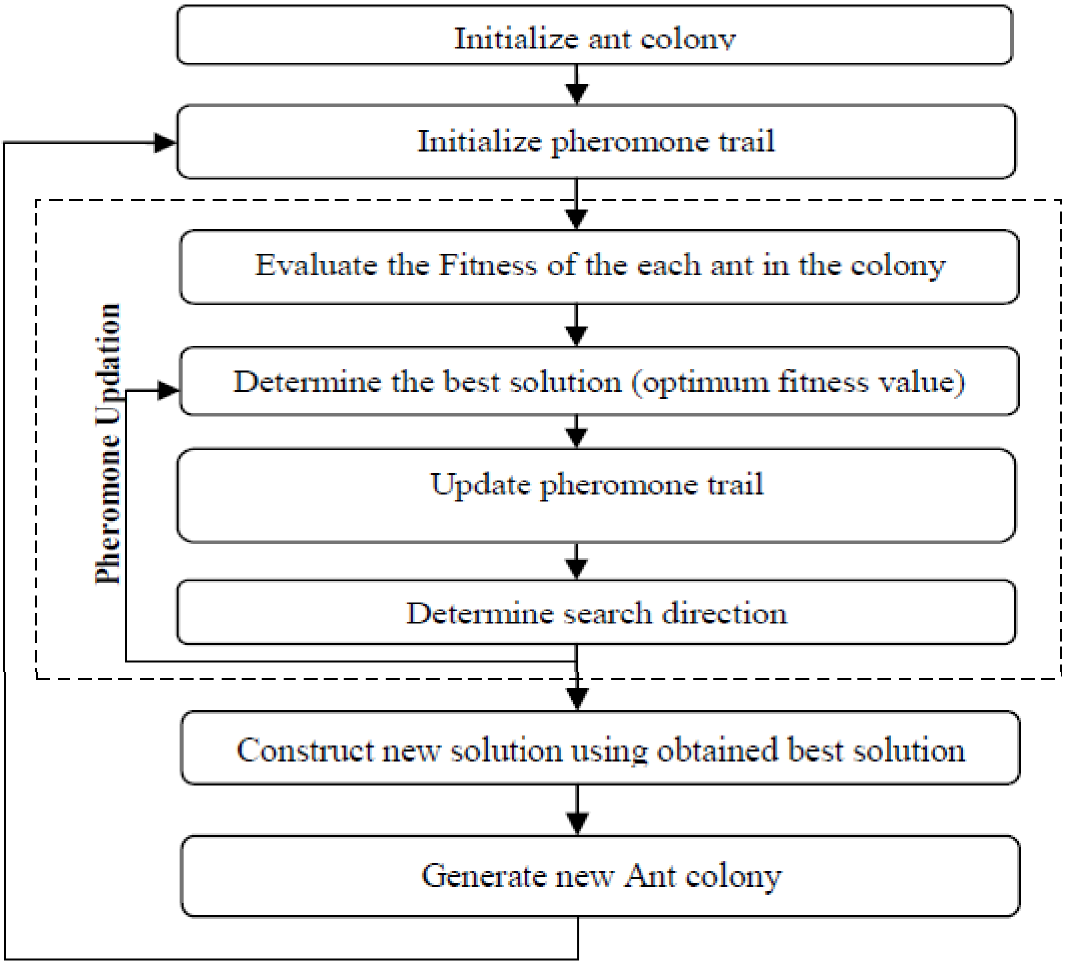
$$pL = pvL + Vi \tag{2}$$

$$w = 1 / iterNum \tag{3}$$

V_i is the current velocity, V_{i-1} is the previous velocity, *pL* is the present location of the particle, *pvL* is the previous location of the particle, *rnd* is a random number between (0, 1), *c*1 and *c*2 are learning factors or stochastic factors, and *iterNum* is the current iteration number.

IV. Ant Colony Optimization Algorithm

ACO has been recently developed as a population based Meta heuristic that has been successfully applied to several NP-hard combinatorial problems. The ACO is the one of the most recent techniques for approximate optimization methods. The main idea is that it is indirect local communication among the individuals of a population of artificial ants. The core of ant's behaviour is the communication between the ants by means of chemical pheromone trails, which enables them to find shortest paths between their nest and food sources. The role of pheromone is to guide the other ants towards the target points. This behaviour of real ant colonies exploited. The ACO is consisted of three main phases; initialization, pheromone update and solution phase. All of these phases build a complete search to the global optimum. At the beginning of the first iteration, all ants search randomly to the best solution of a given problem within the feasible solution space, and old ant colony is created at initialization phase. After that, quantity of Pheromone is updated. In the solution phase, new ant colony is created based on the best solution from the old ant colony. Then, the best solutions of two colonies are compared. At the end of the first iteration, feasible solution space is reduced by a vector that guides the bounds of search space during the ACO application. Optimum solution is then searched in the reduced search space during the algorithm progress. The ACO reaches to the global optimum as ants find their routes in the limited space.


Initialize ant colony
Initialize pheromone trail
Pheromone Updation
Evaluate the Fitness of the each ant in the colony
Determine the best solution (optimum fitness value)
Update pheromone trail
Determine search direction
Construct new solution using obtained best solution
Generate new Ant colony

A simple ACO Algorithm (Figure 2) consists of the steps:

The algorithm starts with ant colony being placed randomly on the search space. The main steps include,

- Initialize ant colony.
- Initialize the pheromone trail.
- Evaluate fitness of the each ant in the colony.
- Determine the best optimum solution among all ants in the colony.
- Update Pheromone trail.
- Determine the search direction for further search.
- Construct new solution using obtained best optimum solution.

- Generate new colony.

1. Initialisation of Ant Colony

Image is represented as a matrix of pixel values and the random population with the set of optimal magnitudes obtained for the hybrid filter combinations. First optimal magnitudes of the image are obtained by using Adaptive thresholding with Foreground and Background Clustering algorithm as described below.

1.1. Adaptive thresholding with foreground and background clustering algorithm:

In the foreground and background clustering (FBC) approach to document image binarization, each pixel is assigned to a foreground cluster or a background cluster. Pixel clustering is based on a variant of the K-means algorithm due to McQueen, where the cluster means are updated, and each time a data point is ascribed to a cluster. Since only one background and one foreground are assumed, K=2, i.e. only two clusters are considered, which makes the overall implementation easy.

1.1.1. Region Selection

Divide the document into all inclusive mutually exclusive sub regions. Select the document sub region for which the threshold be computed, and a region containing the sub region that will be used to determine the threshold sub region. For example, the region may consist of *N* contiguous scan lines, where the sub region is the centre *M* scan lines, with *M<N.*

1.1.2. Initialization

Initialize the background cluster mean and the foreground cluster mean to be the same as computed for the previous sub region. If there is not one, set the two initial cluster means with a large separation between them. For each pixel inside the region, iterate between steps 3 and 4.

1.1.3. Pixel Assignment

Assign each pixel to the nearest cluster.

1.1.4. Cluster mean update

After each new pixel assignment update the relevant cluster mean.

1.1.5 Threshold Calculation

After all pixels in the region have been assigned, set the threshold for the sub region equal to the average of the foreground and different combinations which are closer to the obtained optimal magnitude.

2. Initialize Pheromone Trail

The pheromone trail is the basis for proceeding into next iterations. In this context pheromone trail is also a group of hybrid filters combinations that equal the size of the ant colony (no of hybrid filter combinations that was randomly initialized). The hybrid filter combinations in the pheromone trail are randomly selected. The pheromone trail helps the ant colony to proceed for further search in solution space by providing search direction for the ants in the initialized colony. The pheromone trail is being updated in every iteration.

3. Evaluation of fitness

An individual's fitness of every ant in the colony is measured by the sum of intensities of edges in an enhanced Image, because a gray image with a visual good contrast includes many intensive edges. The sum of intensities of edges *E(k)* included in the enhanced image is calculated by the following expression:

$$E(K) = \sum_x \sum_y \surd \square h_k (x ,y)^2 + \square v_k(x ,y)^2$$

$$\square h_k(x, y) = g_k(x+1, y-1) + g_k(x +1, y) + g_k(x+1, y+1) – g_k(x-1, y-1) – g_k(x-1, y) – g_k(x-1, y+1)$$

$$\square v_k(x, y) = g_k(x-1, y+1) + g_k(x, y+1) + g_k(x+1, y+1) – g_k(x-1, y-1) – g_k(x, y-1) – g_k(x+1, y-1)$$

The fitness of the individual 'n' is obtained by the following expression:

$$F(x_n) = rand(x_n/E_k)$$

where rand (xn) contains all hybrid filter combinations that gives optimal magnitudes.

4. Determine the best ant in colony

In this step the best ant among other ants in the colony is determined. An ant is a hybrid combination of filters and ants are compared on the basis of the fitness value. An ant having the best fitness value is being selected and stored for further updating of solution spaces. The search direction depends on the best ant.

5. Updating Pheromone trail

The pheromone trail is updated using a neighbourhood search done for each of the ant which is in the 10% range of the original ant's fitness value, represents one of the hybrid filter combination which was randomly selected. The pheromone trail is updated by selecting the best of these neighbours (randomly 5 for each ant) comparing with the original ant and best of these two is taken as the
Combination for the next iteration.

6. Determine Search Direction

The way for exploring the search space of solutions is being directed by the best solution and the pheromone trail.

7. Generate New Colony

The search direction is being determined from the above steps and search space is being explored. By doing so, the search space increases and gives a maximum probability of getting the global optimum solution.

8. Completion of Evolution

The population is evolved by the ACO algorithm using evolutionary rules described above. Random Ant's information is given to each individual in the first generation. The individual with the best fitness value is selected in the population in every cycle.

V. Aco, Pso And Ga Control Parameters

In the objective enhancement criterion we need to find the solution of the fitness function. The following combinations of the control parameters are used for running ACO. The number of ants (that get the information related to food location) is 2000. Food threshold value which is the edge value between two pixels is 10. Distance from hive (home of ants at the centre position) is 200. The maximum no. of iterations the ACO executes is set to 100.

In the objective enhancement criterion we need to find the solution of the fitness function $F(z)$ where a, b, c, and k are set to be the swarms.

The following combinations of the control parameters are used for running PSO. The number of particles is 30. Dimension of particles is four since the parameters need to be tuned are 4. Range of particles is the positive real numbers. The maximum change one particle can take during one iteration is 20. Learning factors or acceleration constants are equal to 1.3. The searching is a repeat process and the stop condition or the maximum number of iterations the PSO executes is set to 200. Inertia weight is set at 0.6 and 0.9. Using the above control parameters, the PSO is executed and the results are obtained.

The following combinations of the control parameters are used for running GAs. The chromosome structure had four parameters to be estimated. The selection mechanism of using GAs is binary tournament and K-elitism with K = 5. GAs was used with population size, crossover probability and mutation probability of 1000, 0.9, 0.03, respectively.

VI. Results And Discussion

The optimization problem considered in this paper is to solve the enhancement problem using ACO. Our objective is to maximize the number of pixels in the edges, increase the overall intensity of the edges, and increase the measure of the entropy. After that, the histogram of the enhanced image approaches the required uniform distribution. In order to evaluate the ACO-based enhancement method, we compared the proposed method with PSO and GA-based enhancement method using three selected images. They are the Cameraman, Tire and Pout.

For each ACO or PSO or GA run we report three values:

- The performance of the algorithms by computing the objective evaluation functions in terms of the fitness value.
- The computational time per run of each algorithm.
- The efficiency in terms of the number of edgels which gives an indication of the performance of the proposed algorithm.

The objective evaluation criterion in terms of fitness score is employed to rank the proposed method; the results are given in Table 1 for typical runs. It can be shown that the results obtained using ACO when compared with the results obtained using PSO and GA reveals the following fact:

- The fitness value using ACO is more when compared with the fitness value using PSO and GAs for the same number of generations.
- The computational time is less in case of ACO when compared with that of PSO and GAs.
- The image that contains the highest number of edgel pixels can be rated as having high detail contents as shown in Table 2 and 3.

It is clear from Table 2 and 3 that the ACO-based method achieves the best detail content in the enhanced images when compared with the number of edgels in the enhanced image using PSO and GAs and both all three are greater than the number of edgels in the original image. This ensures that the ACO method yields better quality of solution compared to PSO and GAs. Thus, the above facts reveal the superior properties of ACO when compared with PSO and GAs. So, the proposed ACO method yields high quality solutions with better computation efficiency. It can be shown from Figure 3, that the brightness and contrast of the enhanced images using ACO, PSO and GAs appear visibly and is more than the brightness and contrast of the original images. Also, it can be shown clearly, that the brightness of the enhanced images using ACO is better than the brightness of the enhanced images using PSO and GAs.

ACO has been successfully applied for image enhancement application and demonstrated that ACO gets better results in a faster, cheaper way compared with PSO and GA evolutionary method. Also ACO is more attractive than PSO and GA in the way that there are few parameters to adjust compared with the large number of parameters adjusted when PSO and GA is run. All in all, these reported values and the results shown in Figure 3 give a good explanation of the superior of using ACO for image enhancement compared to PSO and GAs.

Table 1: The fitness value of ACO, PSO and GAs

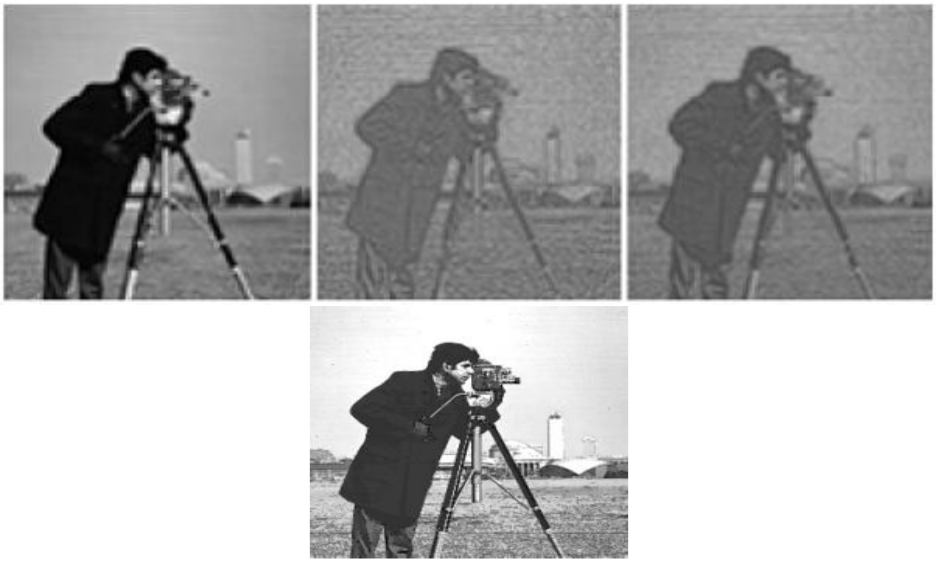
Image/Fitness	ACO-based	PSO-based	GAs-based
Cameraman	133	128.821	102.988
Tire	148	136.398	130.030
Pout	87	10.450	2.972

Table 2: The number of edgels as detected with Sobel automatic edge detector

Image	Original	GA	Diff with GA	PSO	Diff with PSO
Cameraman	2485	2575	90	2674	189
Tire	1823	1917	94	2020	197
Pout	1492	2040	548	2048	556

Table 3: The number of edgels as detected with Sobel automatic edge detector

Image	Original	ACO	Diff with ACO
Cameraman	2512	5411	2899
Tire	1605	3962	2357
Pout	2024	5841	3817



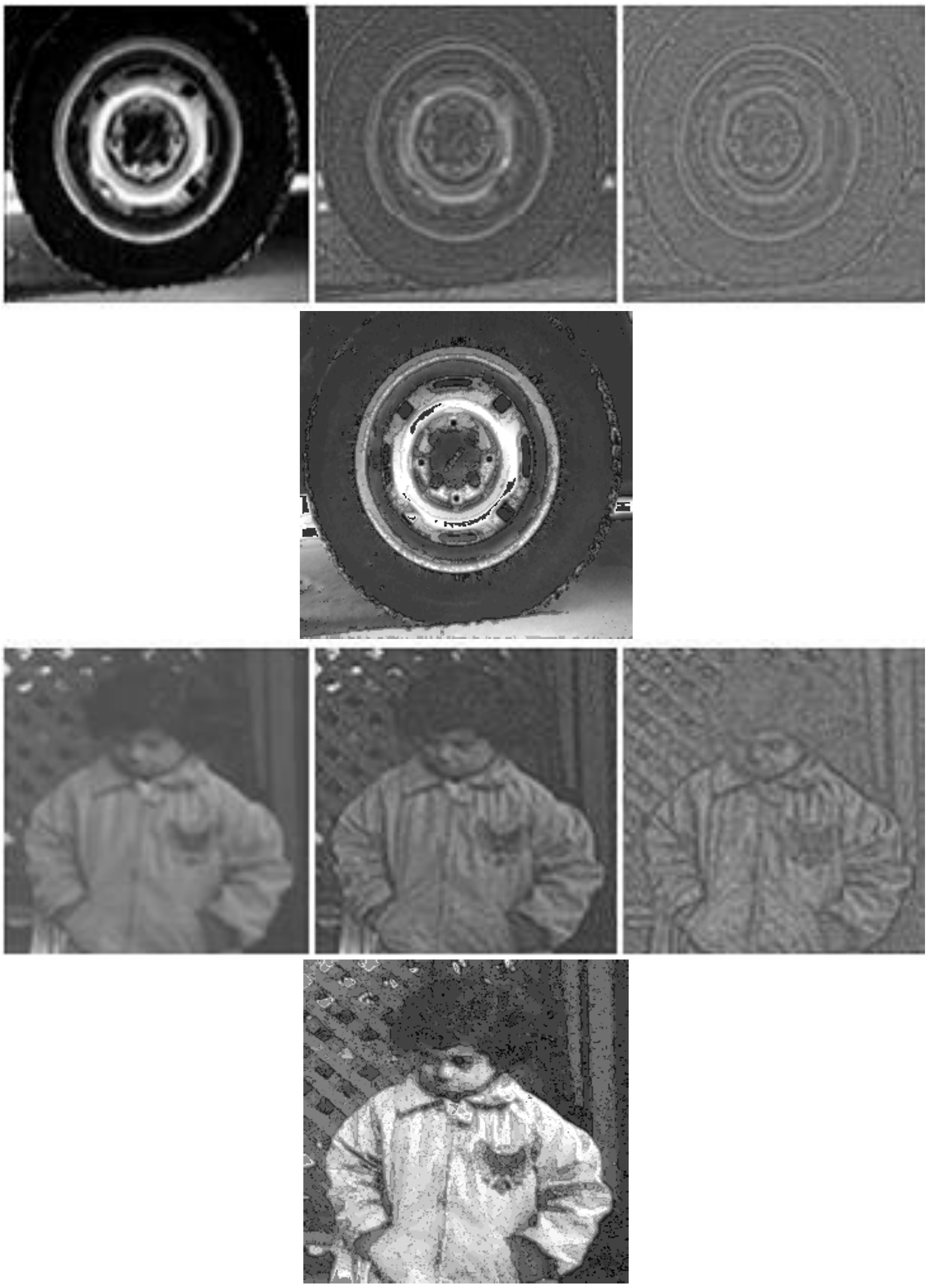


Figure 4: Enhancement results: leftmost-original image; then-GA based method; then-PSO based method; then-ACO based method. For the images a) Cameraman b) Tire c) Pout

VII. Conclusion & Future Work

In this paper, a new approach to automatic image enhancement using real-coded ACO is implemented by specifying a suitable fitness function proportional to the number and intensity of the edgel pixels and to the entropic measure of the image. The objective of the algorithm was to maximize the total number of pixels in the edges thus being able to visualize more details in the images. The algorithm is tested on three selected images. The results obtained are tabulated and compared with the results obtained using PSO and GAs. It is clear from the obtained results that the proposed ACO based image enhancement is better than the PSO and GAs based image enhancement in terms of quality solution and computational efficiency. The proposed ACO based image enhancement method may be extended in several ways, such as: fine tuning of the ACO parameters in order to reduce the number of ants and reducing the maximum number of iterations. Another extension is to code local parameters of the method that applies to each neighbourhood.

References

[1] B. Bhanu, Sungkee Lee, J. Ming. “Adaptive image segmentation using Genetic algorithm”.IEEE transactions on systems, man and cybernetics, 1995, Vol.25, No.12, 1543

[2] Gonzalez and Woods (2001), Digital Image Processing Edition,567-633, Prentice Hall

[3] B. TirimulaRao, K. VenkatRao, G. KiranSwathi, G. PruthviShanthi, and J. SreeDurga. "A Novel Approach to Image Edge Enhancement Using Smoothing Filters." The Icfai University Journal of Computer Sciences (April 2009), Vol.3, No.2, 37

[4] OzgurBaskan, SonerHaldenbilen, HuseyinCeylan, HalimCeylan. “A new solution algorithm for improving performance of ant colony optimization” of Applied Mathematics and Computation 211(2009) 75-84, ELSEVIER.

[5] Marco Dorigo, Thomas Stutzle. A Brad Book, The MIT Press, Cambridge, Massachusetts, London, England

[6] TirimulaRaoBenela, SreeDurgaJampala, Sathya Harish Villa, BhargaviKonathala. “A Novel Approach to Image Edge Enhancement using Artificial Bee Colony Algorithm for Hybrid Filters” BICA-09, IEEE Conference, Buvaneshwar, ISBN 9784244- 5612-3/09

[7] TirimulaRaoBenela, Suresh Chandra Satapathy, SreeDurgaJampala, Sathya Harish Villa, S Chandra Sekhar. “A Novel Approach to Image Edge Enhancement using Particle Swarm Optimization Algorithm for Hybridized Smoothening Filters” SRC

[8] AleksanderLazinica. “Particle Swarm Optimization” published in Intech, p.cm., ISBN 978-953-7619-48-0

[9] Malik Braik, AlaaSheta, Aladdin Ayesh. “Image Enhancement Particle Swarm Optimization” Proceedings of World Congress on Engineering 2007, Vol 1, WCE 2007, July 2 ISBN 978-988-98671-5-7

[10] Mantas Paulinas and Andruisusinskas “ Algorithms Applications For Image Enhancement and Segmentation”, ISSN 1392 –124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol.36, No.3, 278

[11] F.Saitoh. “Image contrast enhancement using Genetic ALGORITHM”. IEEE international conference on systems, man, and cybernetics, IEEE SMC’99, 1999, Vol.4, 899[12] Yaow-Ming Chen. "Passive Filter Design Using Genetic Algorithms."IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS (Feb 2003), Vol.50, Issue 1,202

[13] Yusuke Ojima, Satoshi Kirigaya, and Toru Wakahara."Determining Optimal Filters for Binarization of Degraded Grayscale Characters Using Genetic Algorithms."IEEE Document Analysis and Recognition, 2005.Proceedings. Eighth International Conference (Sept. 2005), Vol.2, ISBN: 0-7695-2420

[14] Abutaleb, and A.S.Kamel "A Genetic algorithm for the estimation of ridges in fingerprints." IEEE Transactions on Image Processing (Aug 1999), Vol.8, Issue 8, 1134-139

[15] Ki-Seung Lee, Eun Suk Kim, Won Doh, and DaeHeeYoun. "Image enhancement based on signal subspace approach." I on Image Processing, (Aug 1999), Vol.8, Issue 8,1129

[16] SurapongAuwatanamongkol. "Inexact graph matching using a Genetic algorithm for image recognition." ELSEVIER Publications on pattern recognition letters, Vol.28 1428-437.

[17] M. K. Lee, S, W. Leung, T. L. Pun, and H. L. Cheung. "EDGE DETECTION BY GENETIC ALGORITHM."IEEE Transactions on Image Processing, 2000.Proceedings.2000 International Conference (2000), Vol.1, 478-80.

[18] Andreas E. Savakis. “ADAPTIVE THRESHOLDING USING BACKGROUND CLUSTERING”.IEEE Proceedings of International Conference On Image Processing ICIP’98, 1998, 8821-1, Vol 3,785-789