We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Provisional chapter

Ant Algorithms for Adaptive Edge Detection

Aleksandar Jevtić and Bo Li

Additional information is available at the end of the chapter

1. Introduction

Edge detection is a pre-processing step in applications of computer and robot vision. It transforms the input image to a binary image that indicates either the presence or the absence of an edge. Therefore, the edge detectors represent a special group of search algorithms with the objective of finding the pixels belonging to true edges. The search is performed following certain criteria, as the edge pixels are found in regions of an image where the distinct intensity changes or discontinuities occur (e.g. in color, gray-intensity level, texture, etc.).

In applications domains such as robotics, vision-based sensors are widely used to provide information about the environment. On mobile robots, images from sensors are processed to detect and track the objects of interest and allow safe navigation. The purpose of edge detection is to segment the image in order to extract the features and objects of interest. No matter what method is applied, the objective remains the same, to change the representation of the original image into something easier to analyze. Digital images may be obtained under different lighting conditions and using different sensors. These may produce noise and deteriorate the segmentation results.

In recent years, algorithms based on swarming behavior of animal colonies in nature have been applied to edge detection. Swarm Intelligence algorithms use the bottom-up approach; the patterns that appear at the system level are the result of local interactions between its lower-level components [2]. The initial purpose of Swarm Intelligence algorithms was to solve optimization problems [7], but recent studies show they can be a useful image-processing tool. The emerging properties inherent to swarm intelligence make these algorithms adaptive to the changing image patterns. This is a useful feature for real-time image processing.

In this work, two edge-detection methods inspired by the foraging behavior of natural ant colonies are presented. Ants use pheromone trails to mark the path to the food source. In

INTECH
open science | open minds

© 2013 Jevtić and Li; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

digital images, pixels define the discrete space in which the artificial ants move and the edge pixels represent the food. The edge detection operation is performed on a set of grayscale images. The first proposed method extracts the edges from the original grayscale image. The second method finds the missing broken-edge segments and can be used as a complementary tool in order to improve the edge-detection results. Finally, the study on the adaptability of the first edge detector is performed using a set of grayscale images as a dynamically changing environment.

The chapter is organized as follows. Section 2 provides an overview of the state-of-the-art edge detectors. Section 3 introduces the basic Ant System algorithm. In Section 4 the proposed Ant System-based edge detector is described. The discussion of the simulation results is also given in this section. Follows the description of the proposed broken-edge linking algorithm and the simulation results in Section 5. The study on the adaptability of the proposed Ant System-based edge detector is given in Section 6. Finally, in Section 7 the conclusions are made.

2. Related work

Edges represent important contour features in the image since they are the boundaries where distinct intensity changes or discontinuities occur. In practice, it is difficult to design an edge detector capable of finding all the true edges in image. Edge detectors give ambiguous information about the location of object boundaries for which they are usually subjectively evaluated by the observers [30].

Several conventional edge detection methods have been widely cited in literature. The Prewitt operator [25] extracts contour features by fitting a Least Squares Error (LSE) quadratic surface over an image window and differentiate the fitted surface. The edge detectors proposed in [31] and [3] use local gradient operators, sometimes with additional smoothing for noise removal. The Laplacian operator [9] applies a second order differential operator to find edge points based on the zero crossing properties of the processed edge points.

Although conventional edge detectors usually perform linear filtering operations, there are various nonlinear methods proposed. In [23], authors proposed an edge detection method based on the Parameterized Logarithmic Image Processing (PLIP) and a four directional Sobel detector, achieving a higher level of independence from scene illumination. In [10], an edge detector based on bilateral filtering was proposed, which achieves better performance than single Gaussian filtering. In [21], authors proposed using Coordinate Logic Filters (CLF) to extract the edges from images. CLF constitute a class of nonlinear digital filters that are based on the execution of Coordinate Logic Operations (CLO). An alternative method for calculating CLF using Coordinate Logic Transforms (CLT) was introduced in [4]; the authors presented a new threshold-based technique for the detection of edges in grayscale images.

In recent years, Swarm Intelligence algorithms have shown its full potential in terms of flexibility and autonomy, especially when it comes to design and control of complex systems that consist of a large number of agents. Metaheuristics such as Ant Colony Optimization (ACO) [6], Particle Swarm Optimization (PSO) [17] and Bees Algorithm (BA) [24] include sets of algorithms that demonstrate emergent behavior as a result of local interactions between the members of the swarm. They tend to be decentralized, self-organized, autonomous and adaptive to the changes in the environment. The adaptability and the ability to learn are very important for systems that are designed to be autonomous.

ACO is a metaheuristic that exploits the self-organizing nature of real ant colonies and their foraging behavior to solve discrete optimization problems. The learning ability, in natural and artificial ant colonies, consists in storing information about the environment by laying pheromone on the path that leads to a food source. The emerging pheromone structures serve as the swarm's external memory that can be used by any of its members. Although a single ant can only detect the local environment, the designer of a swarm-based system can observe the emergent global patterns that are a result of the cooperative behavior.

ACO algorithms have been applied to image processing. Some of the proposed applications include image retrieval [28] and image segmentation [11, 14, 18]. Several ACO-based edge detection methods have also been proposed in literature. Among others, these include modifications to Ant System (AS) [22] or Ant Colony System (ACS) algorithms [1, 8, 32] for a digital image habitat, combined with local gray-intensity comparison for different pixel's neighborhood matrices. Some studies showed that an improved detection can be obtained using a hybrid approach with an artificial neural network classifier [26].

In order to apply artificial ant colonies to edge detection one needs to set the rules for local interactions between the ants and define the "food" that ants will search for. For the edge detection problem, the food are the edge pixels in digital images.

3. Ant System algorithm

Artificial ants, unlike their biological counterparts, move through a discrete environment defined with nodes, and they have memory. When traversing from one node to another, ants leave pheromone trails on the edges connecting the nodes. The pheromone trails attract other ants that lay more pheromone, which consequently leads to pheromone trail accumulation. Negative feedback is applied through pheromone evaporation that, importantly, restrains the ants from taking the same route and allows continuous search for better solutions.

Ant System (AS) is the first ACO algorithm proposed in literature and it was initially applied to the Travelling Salesman Problem (TSP) [5]. A general definition of the TSP is the following. For a given set of cities with known distances between them, the goal is to find the shortest tour that allows each city to be visited once and only once. In more formal terms, the goal is to find the Hamiltonian tour of minimal length on a fully connected graph.

AS consists of a colony of artificial ants that move between the nodes (cities) in search for the minimal route. The probability of displacing the kth ant from node i to node j is given by:

$$p_{ij}^{k} = \begin{cases} \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{h \notin tabu_{k}} (\tau_{ih})^{\alpha}(\eta_{ih})^{\beta}} & \text{if j} \notin \text{tabu}_{k} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where τ_{ij} and η_{ij} are the intensity of the pheromone trail on edge (i, j) and the visibility of the node j from node i, respectively, and α and β are control parameters ($\alpha, \beta > 0$; $\alpha, \beta \in \Re$). The tabu_k list contains nodes that have already been visited by the kth ant. The definition of the node's visibility is application-related, and for the TSP it is set to be inversely proportional to the node's Euclidean distance:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{2}$$

It can be concluded from the equations (1) and (2) that the ants favor the edges that are shorter and contain a higher concentration of pheromone.

AS is performed in iterations. At the end of each iteration, pheromone values are updated by all the ants that have built a solution in the iteration itself. The pheromone update rule is described with the following equation:

$$\tau_{ij(new)} = (1-\rho)\tau_{ij(old)} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{3}$$

where ρ is the pheromone evaporation rate ($0 < \rho < 1$, $\rho \in \Re$), m is the number of ants in the colony, and $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid on the edge (i,j) by the kth ant, and is given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if edge } (i,j) \text{ is traversed by the k}th \text{ ant} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where L_k is the length of the tour found by the kth ant, and Q is a scaling constant ($Q > 0$, $Q \in \Re$).

The algorithm stops when the satisfactory solution is found or when the maximum number of iterations is reached.

4. Ant System-based edge detector

In this section, the AS-based edge detector proposed by [15] is described. The method generates a set of images from the original grayscale image using a nonlinear image enhancement technique called Multiscale Adaptive Gain [19], and then the modified AS algorithm is applied to detect the edges on each of the extracted images. The resulting set of pheromone-trail matrices is summed to produce the output image. Threshold and edge thinning, which are optional steps, are finally applied to obtain a binary edge image. The block diagram of the proposed method is shown in Figure 1.

4.1. Multiscale Adaptive Gain

Image enhancement techniques emphasize important features in the image while reducing the noise. Multiscale Adaptive Gain is applied to obtain contrast enhancement by suppressing pixels with the grey intensity values of very small amplitude and enhancing only those pixels with values larger than a certain threshold within each level of the transform space. The nonlinear operation is described with the following equation:

$$G(I) = A[sigm(k(I-B)) - sigm(-k(I+B))] \tag{5}$$

where

$$A = \frac{1}{sigm(k(1-B)) - sigm(-k(1+B))} \tag{6}$$

where $I = I(i,j)$ is the grey value of the pixel at (i,j) of the input image and $sigm(x)$ is defined as

$$sigm(x) = \frac{1}{1+e^{-x}} \tag{7}$$

and B and k control the threshold and rate of enhancement, respectively. ($0 < B < 1$, $B \in \Re$; $k \in \aleph$). The transformation function (5) relative to the original image pixel values is shown in Figure 2. It can be observed that $G(I)$ is continuous and monotonically increasing; therefore, the enhancement will not introduce new discontinuities into the reconstructed image.

4.2. Ant System algorithm for edge detection

The generic Ant System algorithm described in Section 3 was used as a base for the proposed edge detector. In digital images, discrete environment in which the ants move is defined by pixels, i.e. their gray-intensity values, $0 \leq I(i,j) \leq I_{max}$, i = 1,2,...,N; j = 1,2,...,M. Possible ant's moves to the neighboring pixels are shown in Figure 3.

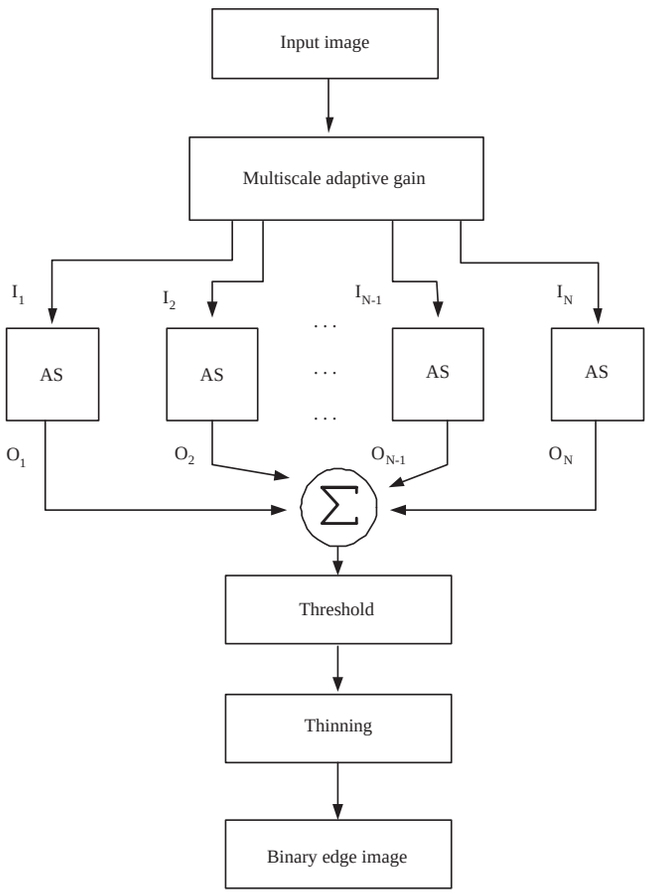
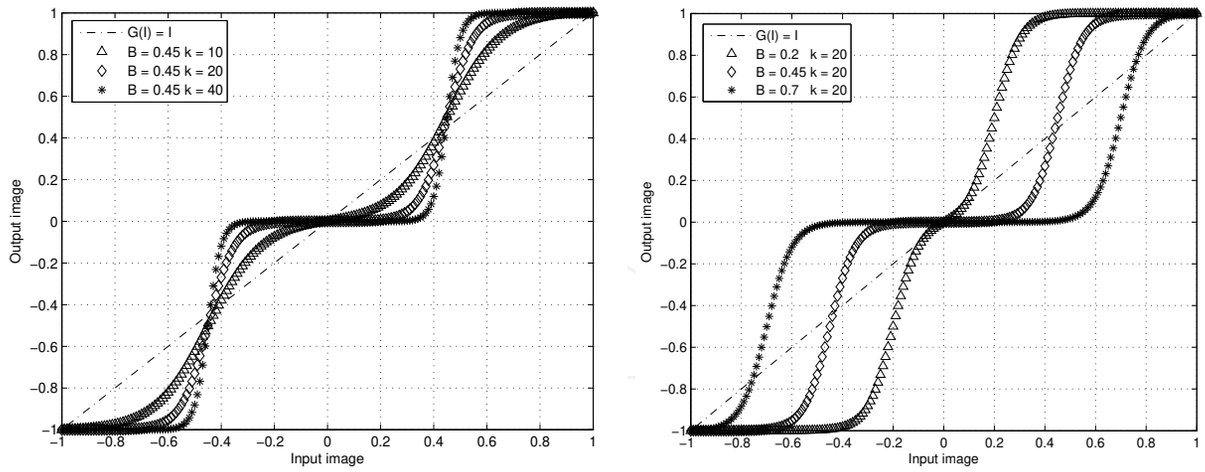


Figure 1. Block diagram of the proposed edge detection method



(a) (b)

Figure 2. Transformation function $G(I)$ in respect to the original image pixel values: (a) B = 0.45; k = 10, 20 and 40; (b) B = 0.2, 0.45 and 0.7; k = 20.

Unlike the cities' visibility in the TSP, the visibility of the pixel at (i, j) is defined as follows:

$$\eta_{ij} = \frac{1}{I_{max}} \cdot max \begin{bmatrix} |I(i-1,j-1) - I(i+1,j+1)|, \\ |I(i-1,j+1) - I(i+1,j-1)|, \\ |I(i,j-1) - I(i,j+1)|, \\ |I(i-1,j) - I(i+1,j)| \end{bmatrix} \tag{8}$$

where I_{max} is the maximum gray-intensity value in the image ($0 \leq I_{max} \leq 255$). For the pixels in regions of distinct gray-intensity changes the higher visibility values are obtained, which makes those pixels more attractive to ants.

The AS algorithm is an iterative process which includes the following steps:

1. Initialization: the number of ants proportional to $\sqrt{N \cdot M}$ is randomly distributed on the pixels in the image. Only one ant is allowed to reside on a pixel within the same iteration. Initial non-zero pheromone trail value, τ_0, is assigned to each pixel, otherwise the ants would never start the search.

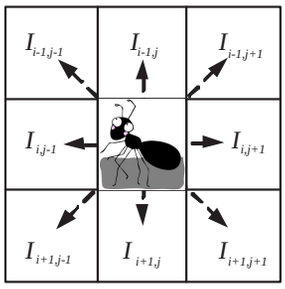


Figure 3. Proposed pixel transition model

2. Pixel transition rule: Unlike their biological counterparts, artificial ants have memory. Tabu_k represents the list of pixels that the kth ant has already visited. If ant is found surrounded by the pixels that are either in the tabu list or occupied by other ants, it is randomly displaced to another unoccupied pixel that is not in the tabu list. Otherwise, the displacement probability of the kth ant to a neighboring pixel (i, j) is given by:

$$p_{(i,j)}^{k} = \begin{cases} \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_u \sum_v (\tau_{uv})^{\alpha}(\eta_{uv})^{\beta}} & \textit{(i,j)} \text{ and } \textit{(u,v)} \text{ are allowed nodes} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where τ_{ij} and η_{ij} are the intensity of the pheromone trail and the visibility of the pixel at (i, j), respectively, and α and β are control parameters ($\alpha, \beta > 0$; $\alpha, \beta \in \Re$).

3. Pheromone update rule: Negative feedback is implemented through pheromone evaporation according to:

$$\tau_{ij(new)} = (1 - \rho)\tau_{ij(old)} + \Delta\tau_{ij} \tag{10}$$

where

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{11}$$

and

$$\Delta\tau_{ij}^{k} = \begin{cases} \eta_{ij} \text{ if } \eta_{ij} \geq T \text{ and } k\text{th ant displaces to pixel } \textit{(i,j)} \\ 0 \quad \text{otherwise.} \end{cases} \tag{12}$$

T is a threshold value which prevents ants from staying on the background pixels hence enforcing the search for the true edges. The existence of the pheromone evaporation rate, ρ, prevents the algorithm stagnation. Pheromone trail evaporates exponentially from the repeatedly not-visited pixels.

4. Stopping criterion: The steps 2 and 3 are repeated in a loop and algorithm stops executing when the maximum number of iterations is reached.

4.3. Simulation results and discussion

The proposed method was tested on four different grayscale images of 256×256 pixels resolution: "Cameraman", "Lena", "House" and "Peppers". As seen from the block diagram in Figure 1, first the Multiscale Adaptive Gain defined in (5) is applied to the input image: $0 \leq I(i, j) \leq I_{max}$, $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, M$. ($N = M = 256$.) The values of B and k were varied to obtain a set of nine enhanced images: $B = \{0.2, 0.45, 0.7\}$; $k = \{10, 20, 40\}$.



(a) (b) (c) (d)

Figure 4. Effects of the transformation function $G(I)$; "Cameraman", 256 × 256 pixels: (a) original image; (b) $B = 0.2$, $k = 10$; (c) $B = 0.45$, $k = 20$; (d) $B = 0.7$, $k = 40$.

The effects of the transformation function on the image "Cameraman" are shown in Figure 4. It can be observed that, by changing the transformation function's parameters, some features in the image become highlighted while others get attenuated.

Afterwards, the AS-based edge detector is applied to each of the nine enhanced images. The algorithm's parameters are set as proposed in [22]: $\tau_0 = 0.01$, $\alpha = 1$, $\beta = 10$, $\rho = 0.05$ and $T = 0.08$. The number of ants equal to $\sqrt{N \cdot M} = 256$ was randomly distributed over the pixels in the image with the condition that no two ants were placed on the same pixel. The memory (tabu list) length for each ant was set to 10. The algorithm was stopped after 100 iterations generating a pheromone-trail matrix of the same resolution as the original image. After each of the nine enhanced images was processed, the sum of the pheromone-trail matrices produced the final pheromone-trail image (Figure 5(e)–(h)). The parameters values such as the number of ants, the memory length, and the number of iterations were obtained as a result of trial and error and their further optimization will be a part of future work.

The effectiveness of the proposed method was compared with the ant-based edge detectors proposed by Tian *et al.* [32] and Nezamabadi-pour *et al.* [22], and the results are shown in Figure 6. To provide a fair comparison, the threshold and morphological edge-thinning operations are neglected. The simulation results show that the proposed method outperforms the other two methods in terms of visual quality of the extracted edge information and sensitivity to weaker edges. The qualitative results of the edge detector proposed in [8] were presented after applying the thinning step, hence a fair comparison with the here-presented results could not be made. It is worth mentioning that the number of iterations used in experiments in [8] was much higher (1000 iterations) than required by our algorithm. The performance evaluation is given in Subsection 4.3.1.

The main contribution of the proposed edge-detection method is the preprocessing step and the parallel execution of the Ant System-based edge detector on a set of images that finally produce the output edge image. The execution time of the proposed method is high for real-time image processing, which would require additional algorithm's code optimization in a different programming environment. The presented experiments were performed in Matlab software that offers an easy high-level implementation but is ineffective in terms of speed.

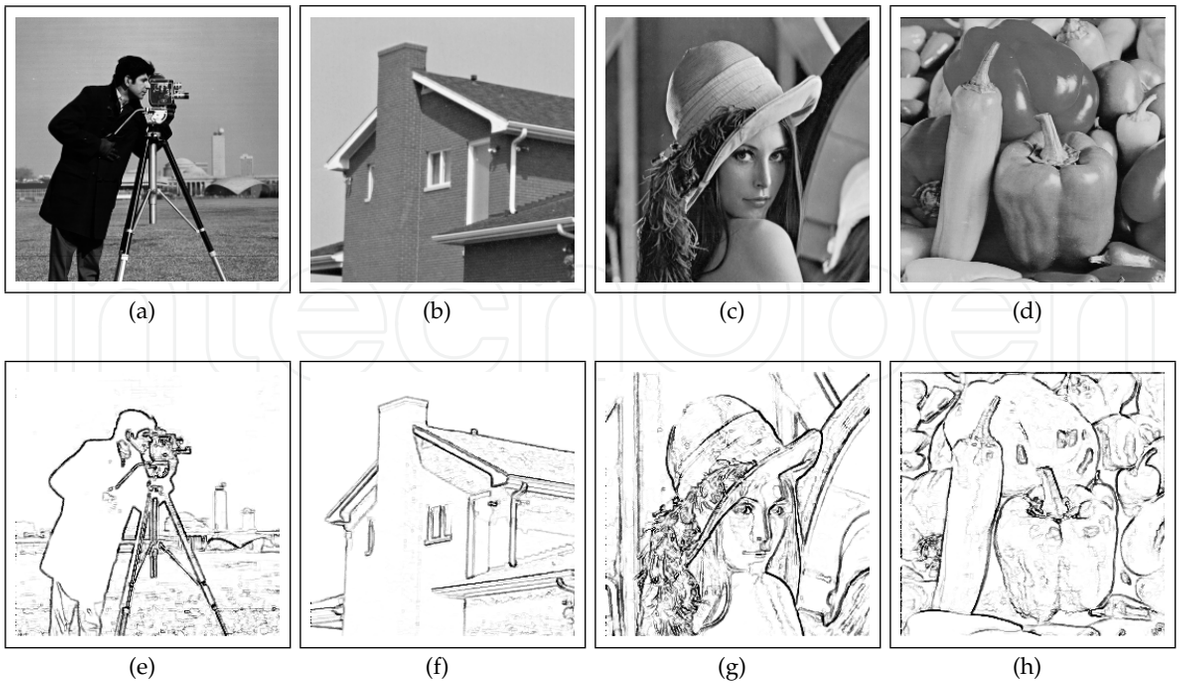


Figure 5. Qualitative results of the proposed method, 256 × 256 pixel images: (a) "Cameraman" original image; (b) "House" original image; (c) "Lena" original image; (d) "Peppers" original image; (e) "Cameraman" pheromone trail image; (f) "House" pheromone trail image; (g) "Lena" pheromone trail image; (h) "Peppers" pheromone trail image.

4.3.1. Performance evaluation

In the complexity-performance trade-off, it was found that varying the values of algorithm's parameters can affect its performance. A set of experiments was performed on a synthetic test image (Figure 7) to show how the number of ants and iterations will be related to the number of detected edge points. The results of this analysis are shown in Figure 8. The number of ants is proportional to the square root of the image resolution $n = \sqrt{N \cdot M}$. The number of edge points was 780.

It can be observed that when the number of ants was increased, the required number of iterations was reduced to achieve a similar performance. Figure 8 shows that the algorithm needs more than 130 iterations to reach good performance when the number of ants was

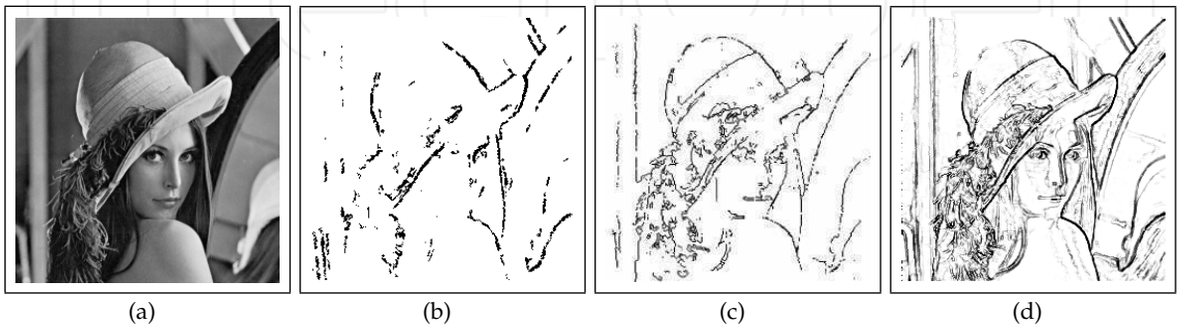


Figure 6. Comparative results with other ant-based edge detectors, "Lena" 256 × 256 pixels: (a) original image; (b) Tian *et al.*; (c) Nezamabadi-pour *et al.*; (d) the proposed method.

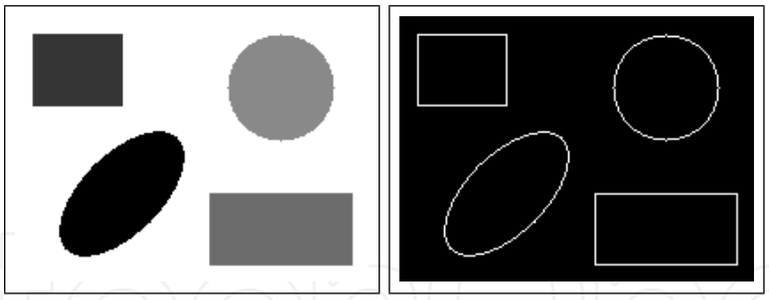


Figure 7. Test image, 256 × 256 pixels: (a) original image; (b) ground-truth edge image.

set to $1 \cdot n$. However, the results not presented here showed that the algorithm was able to detect the maximal number of edge pixels after 400 iterations. Future work may include the optimization of parameters with respect to the computation time.

5. Ant System-based broken-edge linking algorithm

Conventional image edge detection always results in missing edge segments. Broken-edge linking is an improvement technique that is complementary to edge detection. It is used to connect the broken edges in order to form the closed contours that separate the regions of interest. The detection of the missing edge segments is a challenging task. A missing segment is sought between two endpoints where the edge is broken. The noise that is present in the original image may limit the performance of edge-linking algorithms.

Many broken-edge linking techniques have been proposed to compensate the edges that are not fully connected by the conventional edge detectors. [16] applied morphological image enhancement techniques to detect and preserve thin-edge features in the low contrast regions of an image. [33] applied Sequential Edge-Linking (SEL) algorithm that provided full connectivity of the edges but for a rather simplified two-region edge-detection problem.

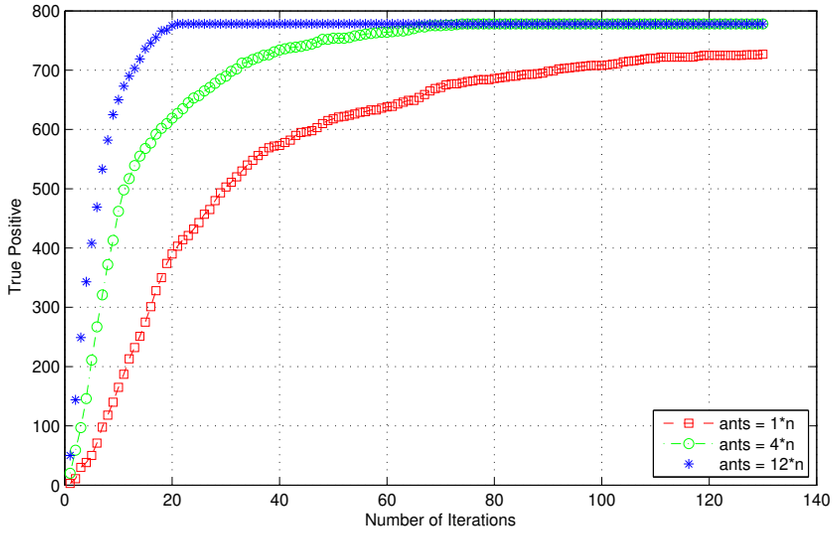


Figure 8. Extracted features vs. number of iterations for different ant-colony size.

Authors proposed this method to extract the contour of a breast as the region of interest in mammogram. [29] applied adaptive structuring elements to dilate the broken edges along their slope directions. [20] proposed improvement to the traditional Ant Colony Optimization (ACO) based method for broken-edge linking to reduce the computational cost.

In this section, the Ant System-based broken-edge linking algorithm proposed by [13] is presented. As inputs are used: the Sobel edge image and the original grayscale image. The Sobel edge image is a binary image obtained after applying the Sobel edge operator [31] to the original grayscale image. From this image the endpoints are extracted that will be used afterwards as the starting pixels for the ants' routes.

The original image is used to produce the *grayscale visibility* matrix, which for the pixel at (i, j) is calculated as follows:

$$\xi_{ij} = \frac{1}{I_{max}} \cdot max \begin{bmatrix} |I(i-1, j-1) - I(i+1, j+1)|, \\ |I(i-1, j+1) - I(i+1, j-1)|, \\ |I(i, j-1) - I(i, j+1)|, \\ |I(i-1, j) - I(i+1, j)| \end{bmatrix} \tag{13}$$

where I_{max} is the maximum gray value in the image, so ξ_{ij} is normalized ($0 \leq \xi_{ij} \leq 1$). For the pixels in regions of distinct gray intensity changes the higher values are obtained. The matrix of *grayscale visibility* will be the initial pheromone trail matrix. It is also used to calculate the fitness value of a route chosen by ant. The resulting image will contain the routes (connecting edges) with the highest fitness values found as optimal routes between the endpoints. In order to discard non-optimal routes, a fitness threshold is applied. Finally, the output image is the improved image that is a sum of the Sobel edge image and the connecting edges. The block diagram of the proposed method is shown in Figure 9.

The proposed AS-based algorithm for broken-edge linking includes the following steps:

1. Initialization: The number of ants equals the number of endpoints found in the Sobel edge image, and each endpoint will be a starting pixel of a different ant. Initial pheromone trail for each pixel is set to its *grayscale visibility* value.
2. Pixel transition rule: Possible ant's transitions to the neighboring pixels are defined by 8-connection pixel transition model shown in Figure 3. The admissible neighboring pixels for the kth ant to move to are the ones not in the tabu_k list. The probability for the kth ant to move from pixel (r, s) to pixel (i, j) is calculated as follows:

$$p^k_{(r,s)(i,j)} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_u \sum_v (\tau_{uv})^\alpha (\eta_{uv})^\beta} & \text{if}(i, j) \text{ and } (u, v) \quad \notin \quad \text{tabu}_k \\ & r - 1 \leq i, u \leq r + 1, \\ & s - 1 \leq j, v \leq s + 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

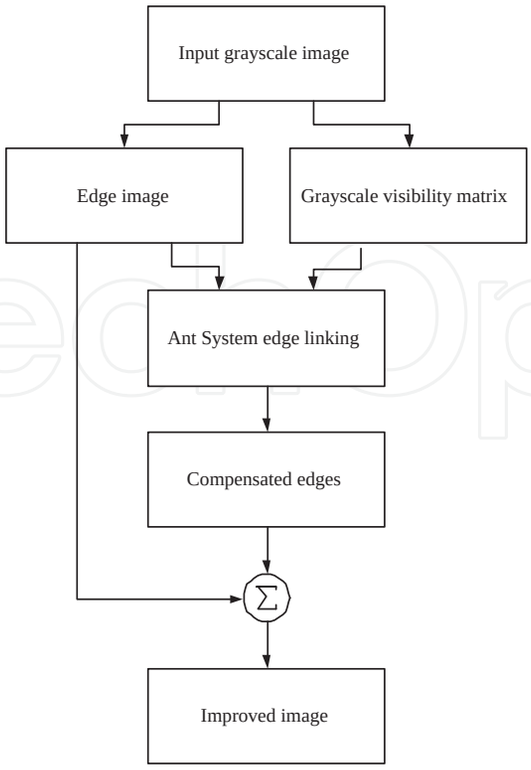


Figure 9. Block diagram of the proposed edge linking method

where τ_{ij} and η_{ij} are the intensity of the pheromone trail and the visibility of the pixel at (i,j), respectively, and α and β are control parameters ($\alpha,\beta > 0$; $\alpha,\beta \in \Re$). The visibility of a pixel should not be misinterpreted as its *grayscale visibility*, and for the pixel at (i,j) it is defined as:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{15}$$

where d_{ij} is the Euclidean distance of the pixel at (i,j) from the closest endpoint.

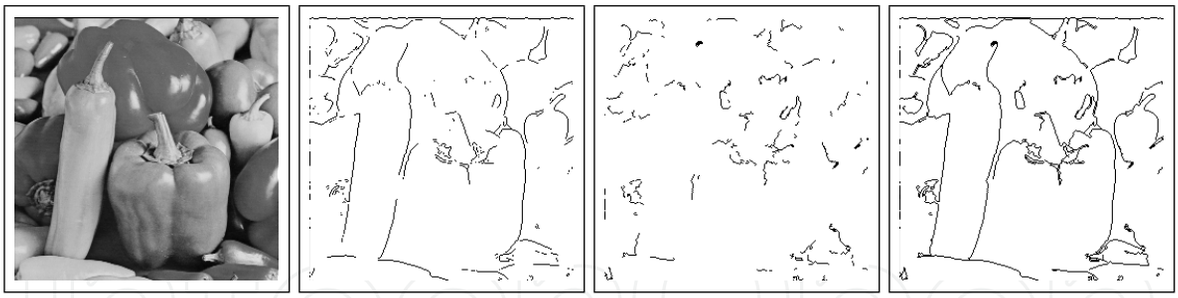
3. Pheromone update rule: Negative feedback is demonstrated through the pheromone trails evaporation according to:

$$\tau_{ij(new)} = (1-\rho)\tau_{ij(old)} + \Delta\tau_{ij} \tag{16}$$

where ρ is the pheromone evaporation rate ($0 < \rho < 1$; $\rho \in \Re$), and

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{17}$$

where



(a) (b) (c) (d)

Figure 10. Qualitative results of the proposed edge-linking method, "Peppers" 256 × 256 pixels: (a) original image; (b) Sobel edge image; (c) resulting image of the proposed method; (d) improved edge image.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{f_k}{Q} & \text{if} \quad k\text{th} \quad \text{ant displaces to pixel} \quad (i,j) \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

The fitness value of a pixel, f_k, is equal to the fitness value of the route it belongs to. The proposed fitness function is given by:

$$f_k = \frac{\bar{\xi}}{\sigma_\xi \cdot N_p} \tag{19}$$

where $\bar{\xi}$ and σ_ξ are the mean value and the standard deviation of the *grayscale visibility* of the pixels in the route, and N_p is the total number of pixels belonging to that route. Pheromone evaporation prevents algorithm stagnation. From the repeatedly not-visited pixels the pheromone trail evaporates exponentially.

4. Stopping criterion: The steps 2 and 3 are repeated in a loop and algorithm stops executing when the maximum number of iterations is reached. An iteration ends when all the ants finish the search for the endpoints, by either finding one or getting stuck and being unable to advance to any adjacent pixel.

5.1. Simulation results and discussion

The simulation results of the proposed algorithm applied to the "Peppers" image of 256 × 256 pixels are shown in Figure 10. The algorithm detects the missing edge segments (Figure 10(c)) as the optimal routes consisted of the edge pixels. The initial pheromone trail for each pixel was set to its *grayscale visibility* value. In this manner, the pixels belonging to true edges have a higher probability of being chosen by ants on their initial routes, which shortens the time needed to find a satisfactory solution, or improves the solution found for a fixed number of iterations. The results were obtained after 100 iterations; this number was chosen on trial and error basis.

Designated values $\alpha = 10$ and $\beta = 1$ were determined on trial and error basis. A large α/β ratio forces the ants to choose the strongest edges. The existence of the control parameter β is important since it inclines the ant's route towards the closest endpoint. Experimental

results showed that, by setting the β value to zero, it took more steps for the ants to find the endpoints which made the computation time longer. In some cases, ants were not even able to find the satisfactory solution for a reasonable number of steps, or they just got stuck between already visited pixels.

The effect of the α/β parameter ratio on the resulting image is best presented in Figure 11. It can be observed that the endpoint in the upper-left corner of the ROI image (Figure 11(c)–(e)) was not connected to any of the closer endpoints, and that the ants successfully found the more remote endpoint which was the correct one. The existence of the β parameter keeps the ants away from the low-contrast regions, such as the region of low gray-intensity pixels between two closer endpoints.

The ant's memory, i.e. the length of the tabu list, was set to 10. Larger ant's memory values would improve the quality of the resulting binary image but would as well lead to the prolonged computation time. The designated value was large enough to keep the ants from being stuck in small pixel circles.

The fitness value of a route is dependent on the mean value and the standard deviation of the grayscale visibility of the pixels in the route, and the total number of pixels belonging to that route, as defined in (19). The routes that have higher grayscale visibility mean value are the stronger edges as the gray level contrast of their adjacent pixels is higher. The smaller standard deviation of the grayscale visibility of the pixels in the route results in a higher fitness value. By this, more importance is given to the routes consisted of pixels belonging to the same edge, thus avoiding the ants crossing between the edges and leaving pheromone trails on non-edge pixels. Finally, the shorter routes are more favorable as a solution, therefore by keeping the total number of pixels in the route smaller, the higher fitness values are obtained.

The number of iterations was set to 100, which gave satisfactory results within an acceptable computational time of execution. The lower resolution images, for example 128×128 pixels, allowed a larger number of iterations to be used, since a smaller number of ants was processed for a smaller number of relatively closer endpoints. The execution time of the algorithm was not optimal, and it was measured in minutes. One of the reasons is that the algorithm code was not written in an optimal manner since Matlab as a programming environment is not intended for a fast code execution, but rather for an easy high-level algorithm implementation.

In order to test the proposed method on different input images, simulations were performed on "House", "Lena" and "Cameraman" images of size 256×256 pixels. The results confirm the effectiveness of the method, as shown in Figure 12. It can be noticed that the found edge segments are often not unidirectional, which indicates that the fitness function was adequately defined and the ants found the true edges. The main contribution of the proposed broken-edge-linking method is in using a bottom-up approach that avoids using a global threshold to find the missing segments.

6. Adaptability of the proposed edge detector

The adaptability and the ability to learn are important features of autonomous systems. In ant colonies, natural and artificial, learning consists in changing the environment by laying the pheromone trails while searching for food. The structures that emerge from the

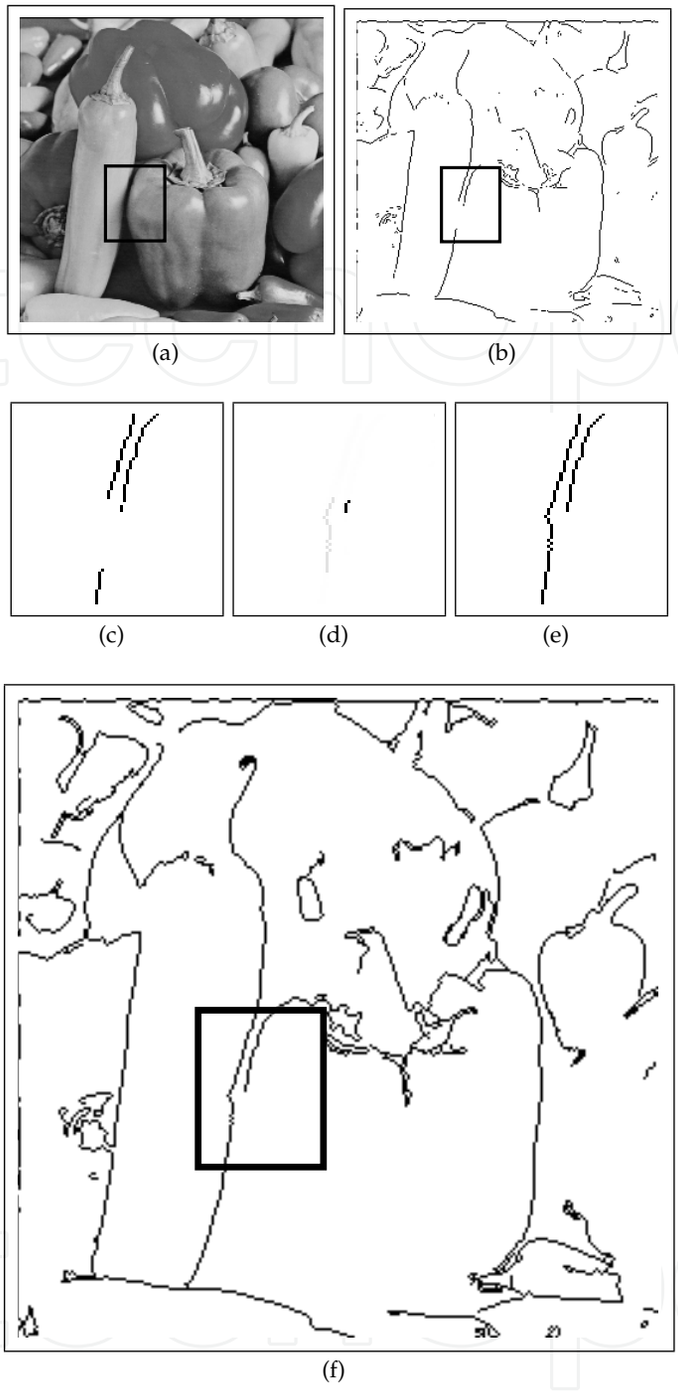


Figure 11. Effect of the control parameters on correct connection of the endpoints: "Peppers" 256 × 256 image: (a) original image with marked region of interest (ROI); (b) Sobel edge image with marked ROI; (c) enlarged ROI: Sobel edge image; (d) enlarged ROI: pheromone trails image; (e) enlarged ROI: improved edge image; (f) improved edge image with marked ROI.

accumulated pheromone represent the stored information about the environment that can be used by any member of the swarm. Although a single ant has no knowledge of the global pattern, the designer of such a swarm-based system is a privileged observer of the emergence that comes as a result of the cooperative behavior.

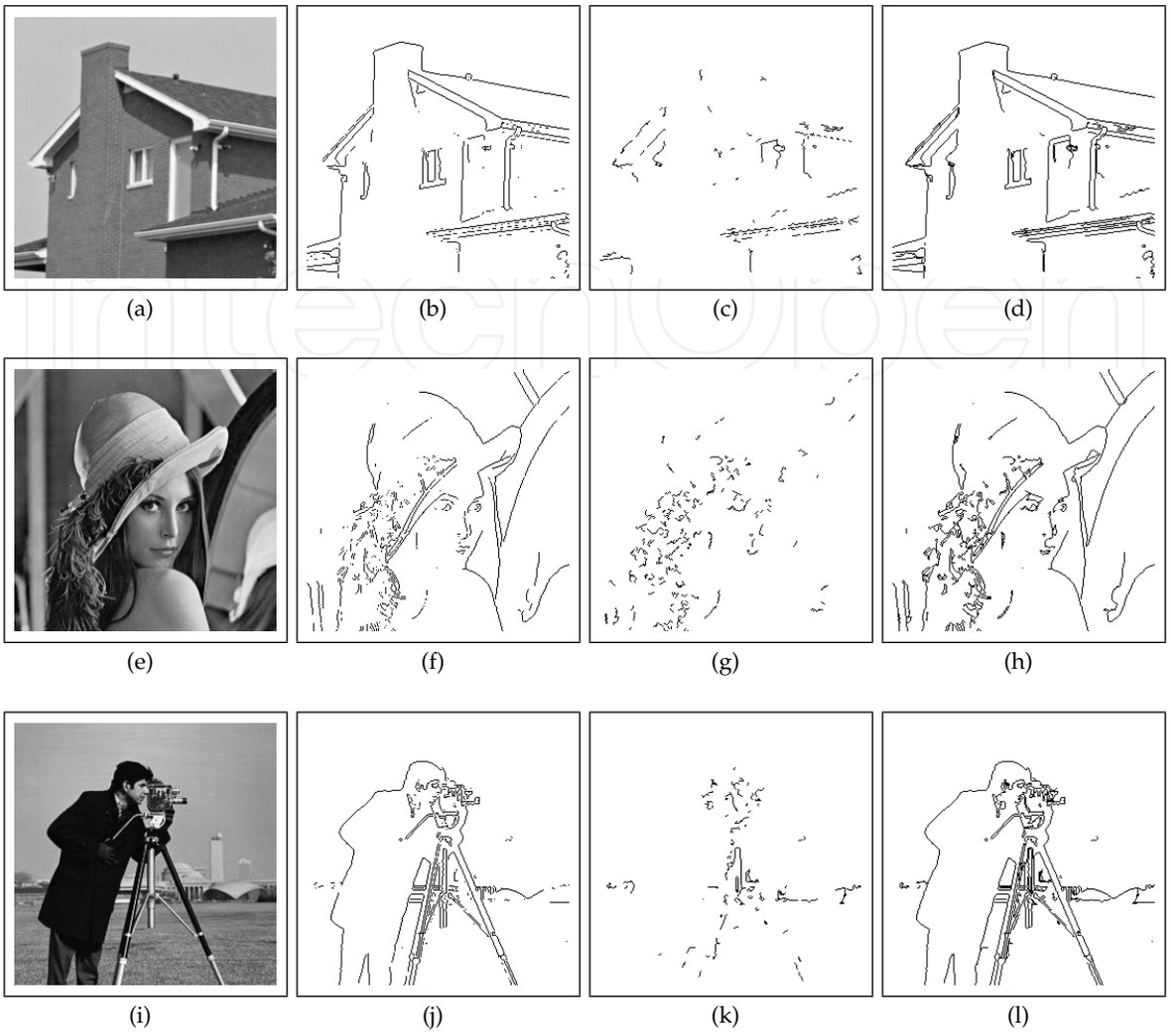


Figure 12. Qualitative results of the proposed method, 256 × 256-pixel images: (a) "House" original image; (b) "House": Sobel edge image; (c) "House": result of the proposed method; (d) "House": improved edge image; (e) "Lena": original image; (f) "Lena": Sobel edge image; (g) "Lena": result of the proposed method; (h) "Lena": improved edge image; (i) "Cameraman": original image; (j) "Cameraman": Sobel edge image; (k) "Cameraman": result of the proposed method; (l) "Cameraman": improved edge image.

The resulting mass behavior in swarms is hard to predict. Although the adaptability can be demonstrated on a variety of applications such as in image segmentation [27], a general theoretical framework on design and control of swarms does not exist. Artificial swarms use bottom-up approach, meaning that the designer of such distributed multi-agent system needs to set the rules for local interactions between the agents themselves and, if required, between the agents and the environment. The indirect communication via environment is referred to as *stigmergy*, and in case of ant colonies, it consists in pheromone-laying and pheromone-following. For each specific application, the food that ants search for must also be defined.

This section presents a study on the adaptability of the algorithm proposed in Section 4 [12]. Experiments with two different sets of grayscale images were performed. In the first experimental setup, a set of three different grayscale images was used to test the adaptability of the proposed AS-based edge detector. The images were obtained by applying

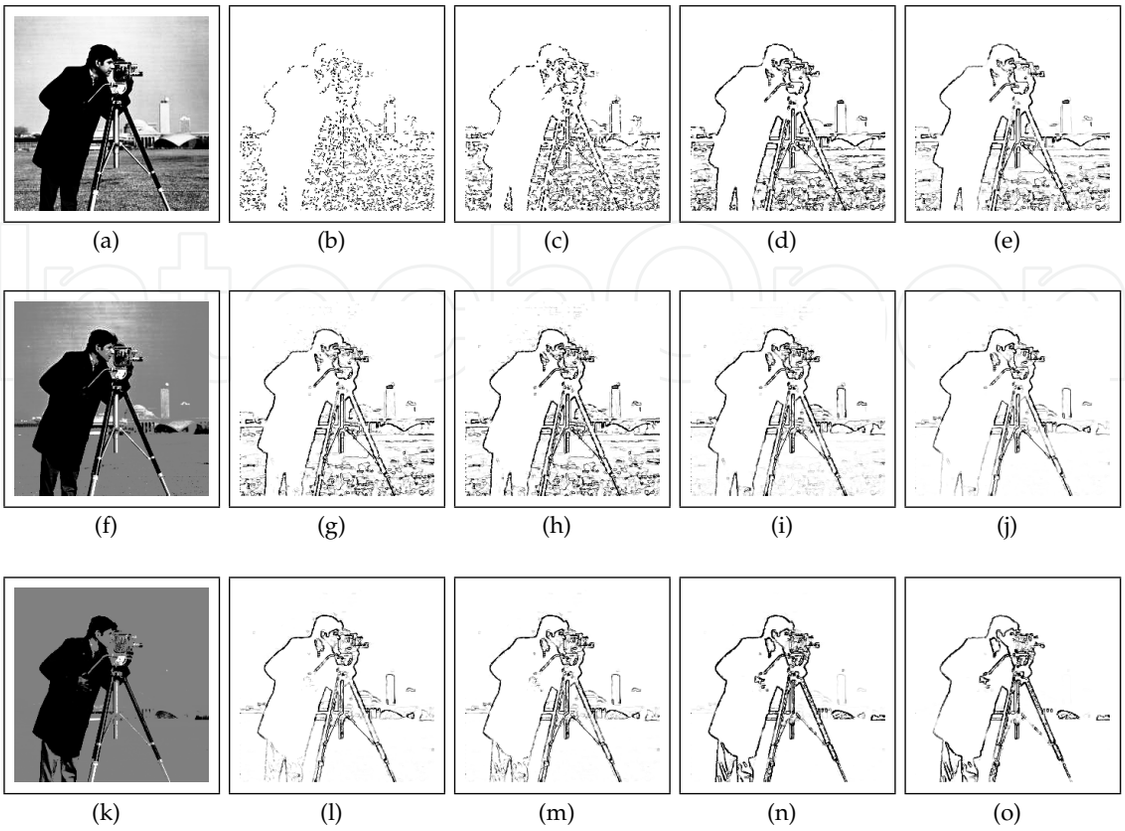


Figure 13. Adaptive edge detection on enhanced "Cameraman" images, 256 × 256 pixels: (a) enhanced image 1; (b) t=5 iterations; (c) t=10 iterations; (d) t=50 iterations; (e) t=100 iterations; (f) enhanced image 2; (g) t=105 iterations; (h) t=110 iterations; (i) t=150 iterations; (j) t=200 iterations; (k) enhanced image 3; (l) t=205 iterations; (m) t=210 iterations; (n) t=250 iterations; (o) t=300 iterations.

a Multiscale Adaptive Gain contrast enhancement to the 256 × 256 pixel "Cameraman" image (see Figure 4). Every $N_i = 100$ iterations one image from the set was replaced by another. The response of the artificial ant colony to the change in the environment was a different distribution of pheromone trails. The number of 100 iterations per image was enough for the new pheromone structure to be established. The algorithm parameters used in the experiments were determined empirically: $\tau_0 = 0.01$, $\rho = 0.5$, $\alpha = 1$, $\beta = 10$, $T = 0.08$ and the tabu list length was set to 10. Parameters could be optimized for a better edge detection, but it is of no importance for this study. It would not affect the adaptability of the algorithm since every image change would result in a change of the pheromone trail structure. Simulation results are shown in Figure 13.

The results show that the Ant System-based edge detector was capable of detecting the changes that occurred as a result of replacing one image from the set with another. The experiments were repeated for a set of four widely used test grayscale images: "Cameraman", "Lena", "House", and "Peppers". The images were used as inputs to the algorithm in that order. Every $N_i = 100$ iterations one image was replaced by the next one from the set. Again, the change in the environment produced by the change of input image resulted in different pheromone patterns, which is shown in Figure 14.

It can be observed that the new pheromone trails accumulated on the pixels belonging to the newly-emerged edges, while the pheromone trails where the edges were no longer present

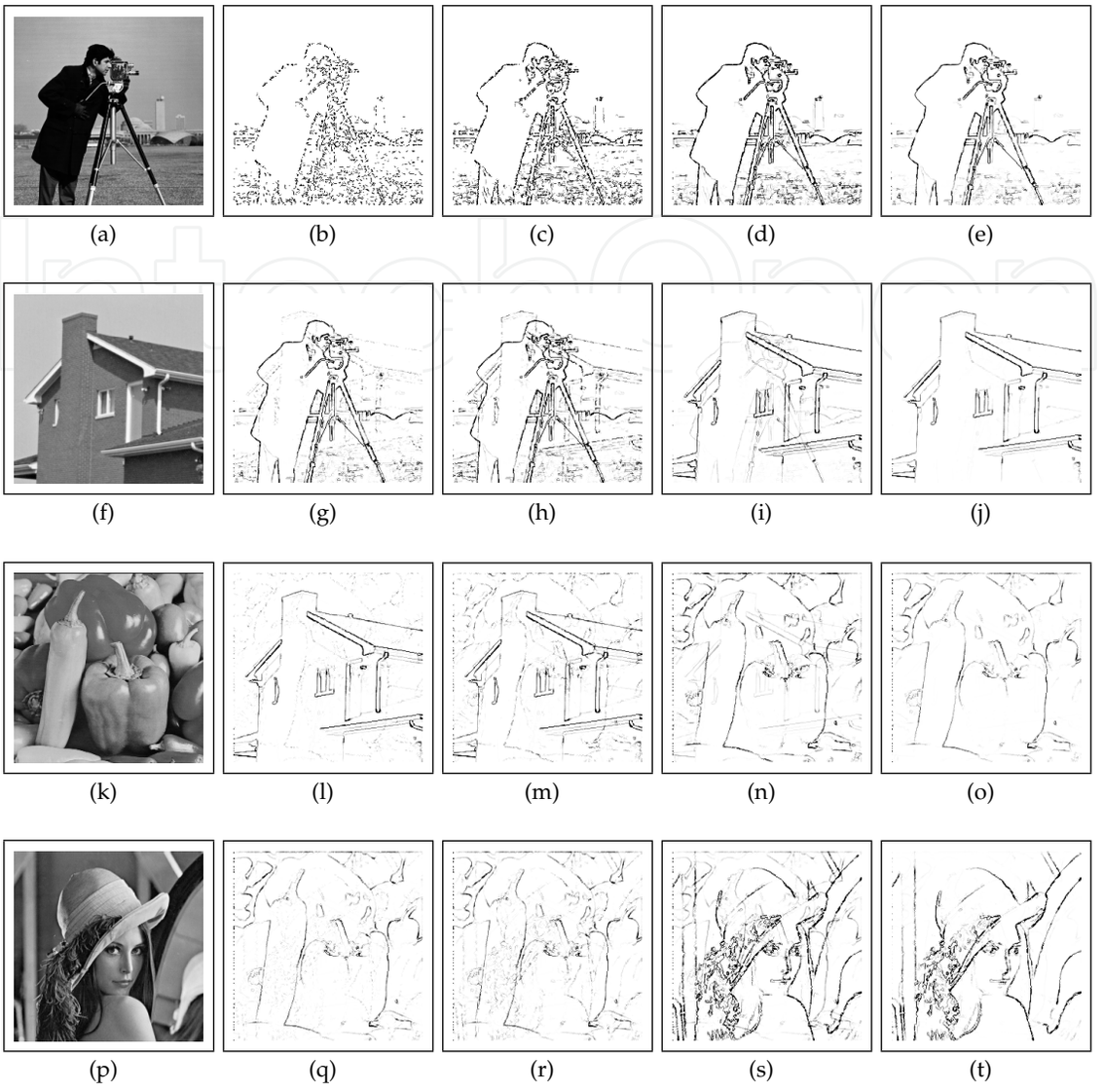


Figure 14. Adaptive edge detection on four test images, 256 × 256 pixels: (a) "Cameraman"; (b) t=5 iterations; (c) t=10 iterations; (d) t=50 iterations; (e) t=100 iterations; (f) "House"; (g) t=105 iterations; (h) t=110 iterations; (i) t=150 iterations; (j) t=200 iterations; (k) "Peppers"; (l) t=205 iterations; (m) t=210 iterations; (n) t=250 iterations; (o) t=300 iterations; (p) "Lena"; (q) t=305 iterations; (r) t=310 iterations; (s) t=350 iterations; (t) t=400 iterations.

gradually disappeared. In order to obtain a quicker transition between different pheromone distributions, the evaporation rate ρ was set to a higher value than for the edge-detection simulations ($\rho = 0.5$). This resulted in disappearing of the "weakest" edges and introduced slightly poorer overall performance of the proposed edge detector. The experimental results show that the algorithm is able to adapt to a dynamically changing environment resulting in different pheromone trail patterns. Even though the images were used in the experiments, the study could be extended to any other type of digital habitat which can lead to a new set of applications for the adaptive artificial ant colonies.

One of the possible applications for the adaptive edge detector could be real-time image processing where online image preprocessing could be used to obtain better image segmentation. By applying various image enhancement techniques, such as contrast

enhancement, certain features in the image could be amplified while others could be reduced or even removed. This would enable easier detection of the regions of interest in the image.

7. Conclusions

Two edge-detection methods inspired by the ants foraging behavior were proposed. The first method uses a grayscale image as input and as output produces a pheromone image marking the location of the edge pixels. The second method finds the missing edge segments after the edge detection was applied and can be used as a complementary tool to any edge detector. In our work, the Sobel edge detector was used to produce the binary edge image.

The first method combines a nonlinear contrast enhancement technique, Multiscale Adaptive Gain, and the Ant System algorithm inspired by the ants foraging behavior. The set of enhanced images was obtained after applying the Multiscale Adaptive Gain and the Ant System algorithm generated pheromone patterns where the true edges were found. The experiments showed that our method outperformed other ACO-based edge detectors in terms of visual quality of the extracted edge information and sensitivity in finding weaker edges. The quantitative analysis showed that the performance could further be optimized by varying the number of ants and iterations.

The adaptability of the proposed edge detector was demonstrated in a dynamically changing environment made of a set of digital grayscale images. The algorithm responded to the changes by generating pheromone patterns according to the distribution of the newly-created edges. It also proved to be robust since even an ant colony of a smaller size could detect the edges, even though the number of detected edge pixels was reduced.

The second proposed method uses the ant colony search for the edge segments that connect pairs of endpoints. A novel fitness function was proposed to evaluate the found segments. It depends on two variables: the pixels grayscale visibility and the edge-segment length. The fitness function produces higher values for the segments that consisted of smaller number of pixels, which had grayscale visibility of a higher mean value and a lower variance. Another novelty was to apply the grayscale visibility matrix as the initial pheromone trails matrix so that the pixels belonging to true edges have a higher probability of being chosen by ants on their initial routes, which reduced the computational load. The proposed broken-edge linking method was tested as a complementary tool for the Sobel edge detector, and it significantly improved the output edge image.

Future research will include optimization and automatic detection of the proposed methods' parameters for an improved edge detection results. Until now these parameters were experimentally obtained. An exhaustive analysis of the edge detection method's adaptability will be performed in order to apply it to other digital habitats. Also, the methods optimization for faster execution would make them suitable for real-time image processing.

Acknowledgements

This work has been financed by the EU-funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTeractive RObotics research network), grant agreement no.: 238486.

8. Acronyms

ACO Ant Colony Optimization
ACS Ant Colony System
AS Ant System
BA Bees Algorithm
CLF Coordinate Logic Filters
CLO Coordinate Logic Operations
CLT Coordinate Logic Transforms
LSE Least Square Error
PLIP Parameterized Logarithmic Image Processing
PSO Particle Swarm Optimization
ROI Region Of Interest
SEL Sequential Edge-Linking
TSP Traveling Salesman Problem

Author details

Aleksandar Jevtić[1] and Bo Li[2]

* Address all correspondence to: aleksandar.jevtic@robosoft.fr; bo.li@tfe.umu.se

1 Robosoft, Bidart, France
2 Dept. of Applied Physics and Electronics, Umeå University, Umeå, Sweden

References

[1] Baterina, A. V. & Oppus, C. [2010]. Image edge detection using ant colony optimization, *WSEAS Transactions on Signal Processing* 6(2): 58–67.

[2] Bonabeau, E., Dorigo, M. & Theraulaz, G. [1999]. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

[3] Canny, J. [1986]. A computational approach to edge detection, *IEEE Transactions on Pattern Analysis & Machine Intelligence* 8: 679–714.

[4] Danahy, E. E., Panetta, K. A. & Agaian, S. S. [2007]. Coordinate logic transforms and their use in the detection of edges within binary and grayscale images, *IEEE International Conference on Image Processing, 2007. ICIP 2007.*, Vol. 3, pp. III–53–III–56.

[5] Dorigo, M., Maniezzo, V. & Colorni, A. [1996]. Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics - Part B* 26(1): 29–41.

[6] Dorigo, M. & Stützle, T. [2004]. *Ant colony optimization*, MIT Press, Cambridge.

[7] Engelbrecht, A. P. [2005]. *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, Ltd, Chichester, UK.

[8] Etemad, S. A. & White, T. [2011]. An ant-inspired algorithm for detection of image edge features, *Applied Soft Computing* 11(8): 4883–4893.

[9] Gonzalez, R. C. & Woods, R. E. [2008]. *Digital Image Processing*, 3rd edn, Prentice Hall, New Jersey, USA.

[10] He, X., Jia, W., Hur, N., Wu, Q., Kim, J. & Hintz, T. [2006]. Bilateral edge detection on a virtual hexagonal structure, *in* G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros, H. Theisel & T. Malzbender (eds), *Advances in Visual Computing*, Vol. 4292 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 176–185.

[11] Huang, P., Cao, H. & Luo, S. [2008]. An artificial ant colonies approach to medical image segmentation, *Computer Methods & Programs in Biomedicine* 92: 267–273.

[12] Jevtić, A. & Andina, D. [2010]. Adaptive artificial ant colonies for edge detection in digital images, *Proceedings of the 36th Annual Conference on IEEE Industrial Electronics Society, IECON 2010*, pp. 2813–2816.

[13] Jevtić, A., Melgar, I. & Andina, D. [2009]. Ant based edge linking algorithm, *Proceedings of 35th Annual Conference of the IEEE Industrial Electronics Society (IECON 2009)*, Porto, Portugal, pp. 3353–3358.

[14] Jevtić, A., Quintanilla-Dominguez, J., Barrón-Adame, J.-M. & Andina, D. [2011]. Image segmentation using ant system-based clustering algorithm, *in* E. Corchado, V. Snasel, J. Sedano, A. E. Hassanien, J. L. Calvo & D. Slezak (eds), *SOCO 2011 - 6th International Conference on Soft Computing Models in Industrial and Environmental Applications*, Vol. 87, Springer Berlin / Heidelberg, pp. 35–45.

[15] Jevtić, A., Quintanilla-Domínguez, J., Cortina-Januchs, M. G. & Andina, D. [2009]. Edge detection using ant colony search algorithm and multiscale contrast enhancement, *Proceedings of 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2009)*, San Antonio, TX, USA, pp. 2193–2198.

[16] Jiang, J. A., Chuang, C. L., Lu, Y. L. & Fahn, C. S. [2007]. Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions, *IET Image Processing* 1(3): 269–277.

[17] Kennedy, J. & Eberhart, R. C. [1995]. Particle swarm optimisation, *Proceedings of IEEE International Conference on Neural Networks Vol. IV*, IEEE service center, Piscataway, NJ, USA, pp. 1942–1948.

[18] Khajehpour, P., Lucas, C. & Araabi, B. N. [2005]. Hierarchical image segmentation using ant colony and chemical computing approach, *in* L. Wang, K. Chen & Y. S. Ong (eds), *Advances in Natural Computation*, Vol. 3611 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1250–1258.

[19] Laine, A. F., Schuler, S., Fan, J. & Huda, W. [1994]. Mammographic feature enhancement by multiscale analysis, *IEEE Transactions on Medical Imaging* 13(4): 725–740.

[20] Lu, D. S. & Chen, C. C. [2008]. Edge detection improvement by ant colony optimization, *Pattern Recognition Letters* 29(4): 416–425.

[21] Mertzios, B. G. & Tsirikolias, K. [2001]. Applications of coordinate logic filters in image analysis and pattern recognition, *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, 2001. ISPA 2001.*, pp. 125–130.

[22] Nezamabadi-pour, H., Saryazdi, S. & Rashedi, E. [2006]. Edge detection using ant algorithms, *Soft Computing* 10(7): 623–628.

[23] Panetta, K. A., Wharton, E. J. & Agaian, S. S. [2008]. Logarithmic edge detection with applications, *Journal of Computers* 3(9): 11–19.

[24] Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. & Zaidi, M. [2006]. The bees algorithm - a novel tool for complex optimisation problems, *Proceedings of IPROMS 2006 Conference*, Cardiff, UK, pp. 454–461.

[25] Prewitt, J. M. S. [1970]. *Object Enhancement and Extraction in Picture Processing and Psychopictorics*, Academic Press, New York.

[26] Rahebi, J. & Tajik, H. R. [2011]. Biomedical image edge detection using an ant colony optimization based on artificial neural networks, *International Journal of Engineering Science and Technology (IJEST)* 3(12): 8211–8218.

[27] Ramos, V. & Almeida, F. [2000]. Artificial ant colonies in digital image habitats - a mass behavior effect study on pattern recognition, *Proceedings of ANTS'2000 - 2nd International Workshop on Ant Algorithms (From Ant Colonies to Artificial Ants), M. Dorigo, M. Middendorf, T. Stüzle (Eds.)*, Brussels, Belgium, pp. 113–116.

[28] Ramos, V., Muge, F. & Pina, P. [2002]. Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies, *in* J. Ruiz-del Solar, A. Abraham & K. M. (eds), *Frontiers in Artificial Intelligence and Applications, Soft Computing Systems - Design, Management and Applications*, Vol. 87 of *2nd International Conference on Hybrid Intelligent Systems*, Springer Berlin / Heidelberg, Santiago, Chile, pp. 500–509.

[29] Shih, F. Y. & Cheng, S. [2004]. Adaptive mathematical morphology for edge linking, *Information Sciences?Informatics and Computer Science: An International Journal* 167(1-4): 9–21.

[30] Shin, M. C., Goldgof, D. B. & Bowyer, K. W. [2001]. Comparison of edge detector performance through use in an object recognition task, *Computer Vision & Image Understanding* 84(1): 160–178.

[31] Sobel, I. & Feldman, G. [1968]. A 3×3 isotropic gradient operator for image processing, *presented at a talk at the Stanford Artificial Project, unpublished but often cited* .

[32] Tian, J., Yu, W. & Xie, S. [2008]. An Ant Colony Optimization algorithm for image edge detection, *Proceedings of the IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, pp. 751–756.

[33] Wei, L., Sheng, L., Yi, R. X. & Peng, D. [2008]. A new contour detection in mammogram using sequential edge linking, *2008 Second International Symposium on Intelligent Information Technology Application (IITA '08)*, Vol. 1, Hong Kong, China, pp. 197–200.