Is the colony of ants able to recognize graphic objects?

Dawid Połap[1], Marcin Woźniak[1],
Christian Napoli[2], Emiliano Tramontana[2], Robertas Damaševičius[3] .

[1]Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-101 Gliwice, Poland
dawid.polap@gmail.com, marcin.wozniak@polsl.pl
[2]Department of Mathematics and Informatics, University of Catania,
Viale A. Doria 6, 95125 Catania, Italy
napoli@dmi.unict.it, tramontana@dmi.unict.it
[3]Software Engineering Department, Kaunas University of Technology,
Studentu 50, Kaunas, Lithuania
robertas.damasevicius@ktu.lt

Abstract. This paper is to discuss a matter of ants swarm intelligence for 2D input images recognition systems. In the following sections we try to analyze possibility of using artificial ants colony algorithm to analyze input images. Experiments have been performed on a set of test images, to present and prove efficacy and precision of recognition.

Keywords: image processing, evolutionary computation, swarm algorithm, object classifier

1 Introduction

Computer Science gives many dedicated methods or tailored solutions to solve classification of various objects. EC methods assist in complex problems solving: dynamic systems positioning [28], queueing systems modeling ([10], [27], [31], [19], [30], [29], [6]), benchmark tests [34] or heat transfer optimization ([12], [13]). This knowledge is proper to be used by Decision Support System (DSS) ([11], [5], [8], [17]).

Image recognition is one of the processes, where dedicated computer systems may help to classify input objects. For this non-trivial operation an efficacy is of a paramount importance. In the following sections we try to analyze efficacy of selected Evolutionary Computation (EC) method, in particular Artificial Ant Colony Algorithm (AACA) combined with a dedicated filtering to create object recognition system.

1.1 Related Works

EC methods are proven to efficiently assist in image preprocessing. Firefly Algorithm (FA) was used in image compression [14], gray-scale image watermarking

[18] and key-point recognition [32]. Cuckoo Search Algorithm (CSA) is efficient in intelligent video target racking [26], satellite image segmentation [4] and image recognition [33]. Therefore, in the following sections we discuss a novel approach to object classification based on Artificial Ant Colony Algorithm (AACA). Ant-based optimization algorithms have already been used for edge detection and shape recognition before. For example [22] have employed ACS combined with MRF (Markov Random Fields) for detection of edges. [15] have used a cellular automata based method similar to the ant-based techniques for segmentation of images. [3] have employed the algorithm based on the social behavior of ants when cleaning their nest for image segmentation. [35] have proposed the Ant Colony System (ACS)-based image feature extraction algorithm for edge extraction. [16] has employed ant-based swarm intelligence for detection of centers of objects in an image. [23] have described the ACS-based algorithm to separate background and foreground components of an image. [9] presents a technique inspired by swarm methodologies such as ant colony algorithms for feature extraction for edge detection and segmentation. [2] presents an efficient ant-based edge detector based on the distribution of ants using a state transition function. [24] establish a pheromone matrix that represents the edge presented at each pixel position of the image, according to the movements of ants driven by the local variation of the image's intensity values.

2 Toward novel image recognition

Image recognition is important problem, which solution can find wide range of applications in various DSS. Common graphics DSS are based on classic methods like SIFT or SURF ([33],[25]), which unfortunately demand complex mathematical operations to determine position of most important features in images. To simplify this process we can use EC methods. First experiments on EC application in image processing were presented in [32] and [33]. This article is devoted to improvement in recognition by further simplification of calculation. Moreover we also present a tailored filtering that together with AACA compose DSS.

2.1 Image Filtering

In the proposed solution, a sobel based filtering is obtained by an edge detection approach using derivative preprocessing. Input images are filter using differential operator ([7], [1]), which approximates two dimensional gradient of a luminance function by a convolution with an integer filter applied along the axial directions. In order to detect the edges contained in a gray scale image, the sobel operator makes use of the second order differential operator. For a continuous function $f : \mathbb{R}^2 \to \mathbb{R}$ and a given point $\mathbf{x_i} = (x_{i,1}, x_{i,2})$, the gradient $\boldsymbol{\nabla} f$ is computed using partial derivatives $\partial_1 f$ and $\partial_2 f$ calculated at each $\mathbf{x_i}$ pixel. An approximation is a convolution of kernels for a small area of neighbor pixels, where $\partial_1 f$ and $\partial_2 f$

use a separate kernel each

$$S1 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} , \quad S2 = \begin{pmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} . \tag{1}$$

combined into a gradient operator. One for a maximum response for the vertical edge and the other for a maximum response for the horizontal edge of the input object. As a result it gives an edge amplitude image. To obtain the edges an associated luminance intensity matrix I is used to compute for every pixel $\mathbf{x_i} = (x_{i,1}, x_{i,2}) \in \mathcal{I}$ the following functions

$$\begin{aligned} g_1(\mathbf{x_i}) &= g_1(x_{i,1}, x_{i,2}) = \sum_{m=1}^{3} \sum_{n=1}^{3} S1_{mn} \cdot I(x_{i,1} + m - 2, x_{i,2} + n - 2), \\ g_2(\mathbf{x_i}) &= g_2(x_{i,1}, x_{i,2}) = \sum_{m=1}^{3} \sum_{n=1}^{3} S2_{mn} \cdot I(x_{i,1} + m - 2, x_{i,2} + n - 2), \end{aligned} \tag{2}$$

$$g(\mathbf{x_i}) = g(x_{i,1}, x_{i,2}) = g_1^2(\mathbf{x_i}) + g_2^2(\mathbf{x_i}).$$

Once the values of the functions in (2) are obtained for each pixel, the edges are defined as the pixels in the subset of points $\mathcal{E} \subset \mathcal{I}$ that

$$\forall \mathbf{x_i} \in \mathcal{E} \Rightarrow \begin{cases} g(\mathbf{x_i}) > 4 \left\langle g^2 \right\rangle \\ g_1(\mathbf{x_i}) > g_2(\mathbf{x_i}) \\ g(\mathbf{x_i}) \geq g(x_{i,1}, x_{i,2} - 1) \\ g(\mathbf{x_i}) \geq g(x_{i,1}, x_{i,2} + 1) \end{cases} \vee \begin{cases} g(\mathbf{x_i}) > 4 \left\langle g^2 \right\rangle \\ g_1(\mathbf{x_i}) < g_2(\mathbf{x_i}) \\ g(\mathbf{x_i}) \leq g(x_{i,1} - 1, x_{i,2}) \\ g(\mathbf{x_i}) \leq g(x_{i,1} + 1, x_{i,2}) \end{cases} , \tag{3}$$

basing on the g_1, g_2 and g computed for each pixel $\mathbf{x_i} \in \mathcal{I}$ as in (2). When a sobel filter is applied to 2D input image $\mathcal{I}$, a new indexed image is depicted as a plain representation of the points of $\mathcal{E} \subset \mathcal{I}$, as defined in (3). The indexed image coordinate system is a representation of $\mathcal{I}$, whereby all the values are zeros, except for the coordinates of the points in $\mathcal{E}$ which are ones. Finally the image is then processed to recognize the objects, where the bright patterns can become the shapes of object easy to recognize by AACA. Applied in shape detection DSS filtering is presented in Algorithm 1.

2.2 Artificial Ant Colony Algorithm

Artificial Ant Colony Algorithm (AACA) simulates behavior of ants when collecting food. Each ant moves in a random way. In case, when the ant finds food, it goes home leaving a trail of the pheromones. This pheromone help another ants to follow the trail. However, over the time the pheromones evaporates. Therefore i.e. short trails ensure that the potency of pheromones will be greater. This process is modeled to form an EC optimization algorithm.

Pheromones levels are updated in AACA according to

$$f^{t+1}(\mathbf{x_i}, \mathbf{x_j}) = (1 - \rho) f^t(\mathbf{x_i}, \mathbf{x_j}) + \mathit{\Gamma}_i^t, \tag{4}$$

Algorithm 1: Applied image filtering

```
Start,
Import image 𝓘 to Im,
Calculate the number of pixels columns and rows in 2D input image Im,
Create the 3 × 3 filters S1 and S2 using (1),
while n ≤ rows do
    while m ≤ columns do
        Grays[m][n] = ccvmean(Im[m][n]),
        Compute g, g1 and g2 on Grays[m][n] using (2),
    end
end
Save g as a bitmap grayscale image GIm,
Stop.
```

where ρ is evaporation rate, t means number of iteration and n is the number of ants in population that must traveled to worker $\mathbf{x_i}$ over Γ_i^t distance. This distance, that each ant must travel, is modeled in equation

$$\Gamma_i^t = \sum_{i=1}^{n} \frac{1}{L_{ij}^t}, \tag{5}$$

where L_{ij}^t means the length of traveled path by ant i to j. The length L_{ij} between any two ants i and j in the population of workers situated at points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the image is defined using Cartesian metric

$$L_{ij}^t = \|\mathbf{x}_i^t - \mathbf{x}_j^t\| = \sqrt{\sum_{k=1}^{2}(x_{i,k}^t - x_{j,k}^t)^2}, \tag{6}$$

where notations in t iteration are: $\mathbf{x_i}^t$, $\mathbf{x_j}^t$–points in $R \times R$ space, $x_{i,k}^t$, $x_{k,j}^t$–k-th components of the spatial coordinates $\mathbf{x_i}^t$ and $\mathbf{x_j}^t$ that describe image points in the space. The probability of choosing path from point $\mathbf{x_i}$ to $\mathbf{x_j}$ by worker ant determines

$$p^t(\mathbf{x_i}, \mathbf{x_j}) = \frac{[f^t(\mathbf{x_i}, \mathbf{x_j})]^\alpha \left[\frac{1}{L_{ij}^t}\right]^\beta}{\sum_{\alpha \in N_i^k}\left([f^t(\mathbf{x_i}, \mathbf{x}_\alpha)]^\alpha \left[\frac{1}{L_{i\alpha}^t}\right]\right)}, \tag{7}$$

where α is the impact of left pheromones, L_{ij} is the distance between i and j, N_i^k is a set of locations, which ant k has not visited and which leads from i.

Having all these informations population of worker ants can move over the image. This movement is based on the path choosing probability from point $\mathbf{x_i}$ to $\mathbf{x_j}$ by worker ant

$$\mathbf{x_i^{t+1}} = \mathbf{x_i^t} + sign(\mathbf{x_i^t}(ind(t)) - \mathbf{x_i^t}), \tag{8}$$

here $ind(t)$ is an array of neighbor indices after sort. How does the movement is done? The worker ant which is situated at the certain image point has eight

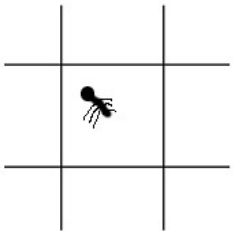

Fig. 1. Worker ant moves possibility in each iteration

possible directions (locations) to move, see Fig. 1. In each t iteration the ant can move only one location (pixel) further toward the strongest pheromone source. The ant uses information about best location in each iteration, this information is calculated in (7), and moves toward this direction. This process is a simulation of the search that worker ants do while searching for food using pheromones left by other workers from the colony. Entire AACA is presented in Algorithm 2.

Algorithm 2: AACA to recognize 2D images

Start,
Define all coefficients: n size of workers population, α impact of left pheromones, ρ evaporation rate, number of *generations*,
Define fitness function for the algorithm using (9),
Create a random initial population P of *ants* in 2D image,
$t = 0$,
while $t \leq generation$ **do**
    Update pheromone values using (4),
    Calculate distances between worker ants (6),
    Calculate possible path to follow by worker i to location j $p^t(\mathbf{x_i}, \mathbf{x_j})$ using (7),
    Evaluate population using (9),
    Determine the best position to follow,
    Move population of workers using (8),
    Next generation t++,
end
Values from last population with best fitness are the solution,
Stop.

2.3 Proposition of image recognition method based on dedicated filtering and AACA

Our research aim at finding a simple and efficient method for 2D image classification. First attempt to this task was presented in [32]. In this article we present novel approach to image classification. The novel approach is combined of dedicated filtering and bio-inspired solution. Addition of filtering helps to increase efficiency for recognition of shapes. To preform classification on filtered images we use AACA defined in section 2.2. Our attempt is based on nature. If we look closely at the colony of ants, we can see that there are roles that each ant has. Some of them carry of the queen, soldiers protect the anthill and others search for food. Worker ants have become the prototype of our method. In nature they search for food in the forest and bring it home. However they also provide an information about the source of food to other workers. This information is provided by the use of pheromones, which guide other workers to food. This process is modeled in (4). Each of the workers leave trace of pheromones that is recognized by others. After comparing pheromones intensity ants follow direction of the most strongest trace. This is modeled with probability (7). The source that the ants must find in the image are filtered images of various objects. After filtering, in the picture stay only most important features describing depicted objects. Among them colony of ants choose points of greater importance. These points are key for object recognition.

In the AACA worker ant is representing a single pixel of the image. A population of workers is simulated in order to move over the image to search for specific areas crucial for automatic recognition process. Using filtering we extract the borders of input objects, which are marked in white on a dark background what is perfect input for AACA recognition. Workers validate pixels using simplified fitness function that reflects the brightness of filtered points

$$\Phi(\mathbf{x_i}) = \begin{cases} 0.1\ldots1 & \text{saturation} \\ 0 & \text{other} \end{cases}, \tag{9}$$

where $\Phi(\mathbf{x_i})$ is the quality of the evaluated pixel reflected by a value in the scale from 0.0 to 1.0, where color saturation changes from black to white.

3 Experiments

In the research we have examine proposed solution to classify images. The recognition system was first filtering input images using method presented in section 2.1. These images, after filtering were passed to next component. AACA method from section 2.2 have been implemented on filtered images. Artificial ant colony was moved over the image to search for important features. The evolutionary method was performed with coefficients $\rho = 0.1$, $\beta = 0.3$.

In the experiments we were trying to localize objects like cars, architecture construction (houses and their elements, bridges), nature elements and so on. Fig. 2 to Fig. 6 present classification. Leftmost top image presents the original input, going to the right we can see the image after first filtering and AACA recognition (in red). The second row shows the image after second filtering, AACA recognition (in red) and main classification result. Proposed novel recognition system

Fig. 2. 2D input image recognition process over single objects

can recognize objects. If we look at first AACA recognition without prefiltering, in each figure, we can see that the system is just covering objects in the picture with points. In this way we can only show location of the objects. However if we compare this with recognition after first and second filtering, we can see that AACA over filtered images is able not only to show place of the objects we are looking for but also show most important points of the shape. Sometimes AACA recognition over once filtered image has enough precision. However sometimes second filtering can help if we are classifying images where object have many details.

3.1 Conclusions

Novel approach to image recognition helps to simplify this process. The knowledge is useful for DSS. Because of flexibility of the proposed solution we can use the recognition in two ways. Location of classified points (recognition without filtering) can be used to help DSS in decision about size of the object, location or surroundings. If the filtering is applied, proposed classifier show less points however these points are better for shape recognition. As now they are classified

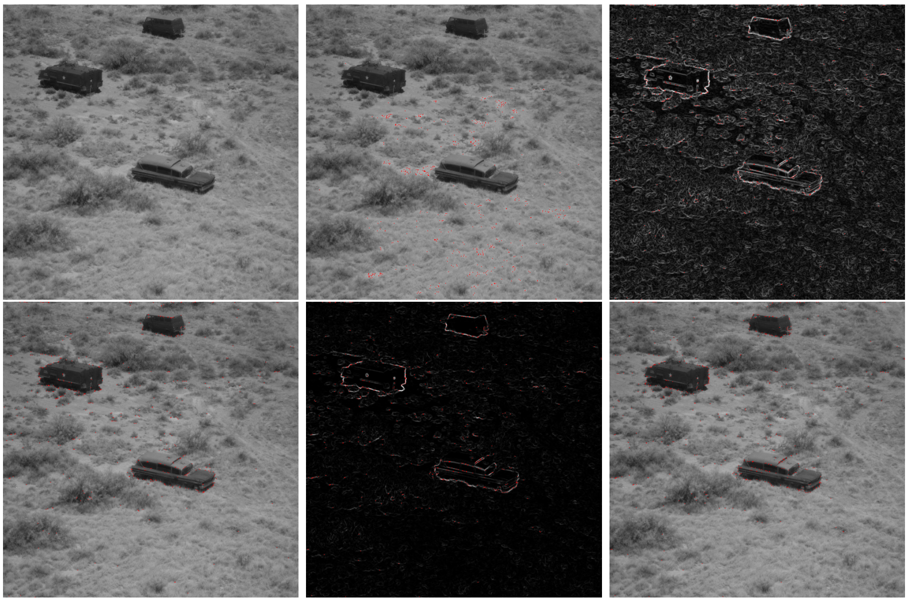

Fig. 3. 2D input image recognition process over landscape objects

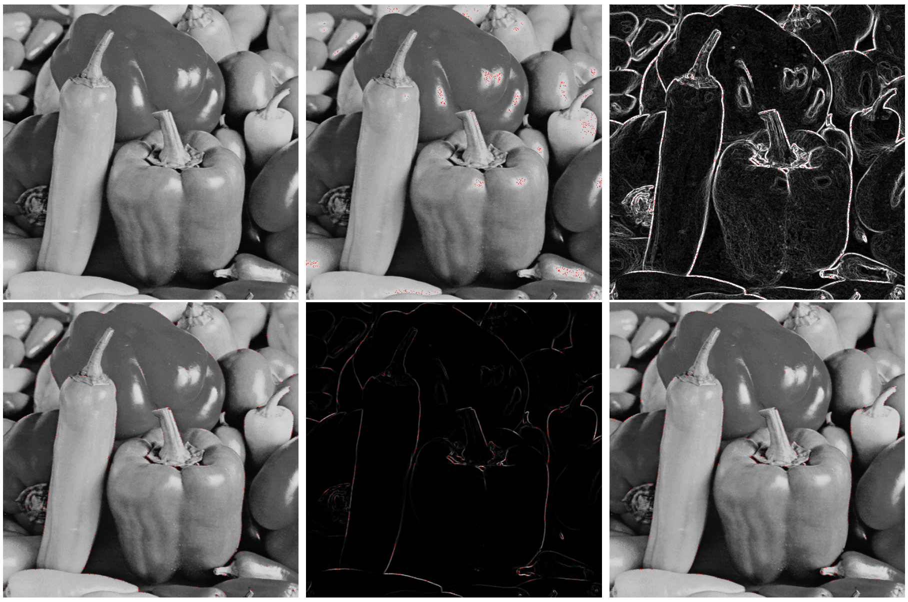

Fig. 4. 2D input image recognition process over still nature

in most important points of input images DSS can use their location to calculate features important for shape recognition, etc. To classify objects using shape

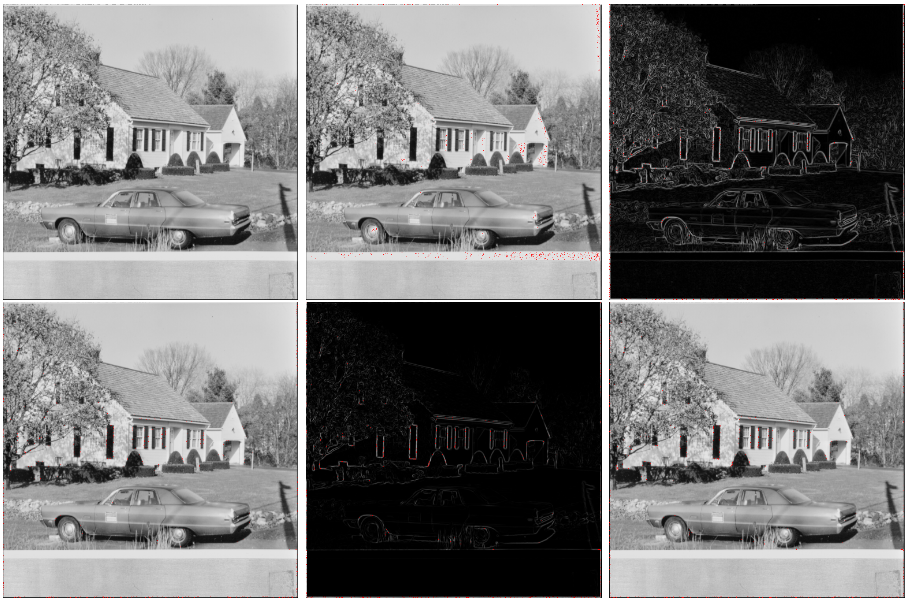

Fig. 5. 2D input image recognition process over architecture

Fig. 6. 2D input image recognition process over nature objects

points we can use some rough and fuzzy methods ([20], [21]) or neural approach ([34], [8]).

4 Final Remarks

Novel classification based on filtering and AACA is precise, easy to implement and fast. The solution can be used in two ways, with prefiltering or without it. Both recognitions are proper for various DSS. This makes it a very promising tool for Artificial Intelligence systems. The solution can work as a part of sophisticated image classifiers, where one needs to detect shape of objects or just simple detection systems where decision is taken only using the location of the recognized object. In both cases presented method give proper results and recognition process is performed ad-hoc without complicated mathematical methods. This makes it very important for DSS where we use simple computation devices without efficient processing units. The precision of AACA in recognition of objects is possible to increase, i.e. when we use larger population of ants or use some fast aggregation methods. This improvements will be considered in the future work.

References

1. G. Anusha, T. Prasad, and D. Narayana. Implementation of sobel edge detection on FPGA. *International Journal of Computer Trends and Technology*, 3(3):472–475, 2012.
2. D. Aydin. An efficient ant-based edge detector. *T. Computational Collective Intelligence*, 1:39–55, 2010.
3. K. Benatcha, M. Koudil, N. Benkhelat, and Y. Boukir. ISA an algorithm for image segmentation using ants. In *Proceedings of IEEE International Symposium on Industrial Electronics*, pages 2503–2507. IEEE, 2008.
4. A.K. Bhandari, V.K. Singh, A.Kumar, and G.K. Singh. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapurs entropy. *Expert Systems with Applications*, 41(7):3538–3560, 2014.
5. F. Bonanno, G. Capizzi, G. Lo Sciuto, C. Napoli, G. Pappalardo, and E. Tramontana. A cascade neural network architecture investigating surface plasmon polaritons propagation for thin metals in openmp. *Lecture Notes in Artificial Intelligence - ICAISC'2014*, 8468, PART I:22–33, 2014.
6. G. Borowik, M. Woźniak, A. Fornaia, R. Giunta, C. Napoli, G. Pappalardo, and E. Tramontana. A software architecture assisting workflow executions on cloud resources. *International Journal of Electronics and Telecommunications*, 61(1):17–23, 2015. DOI: 10.1515/eletel-2015-0002.
7. J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.
8. R. Damaševičius. Structural analysis of regulatory DNA sequences using grammar inference and support vector machine. *Neurocomputing*, 73(4-6):633–638, 2010.
9. S. Etemad and T. White. An ant-inspired algorithm for detection of image edge features. *Electronic Letters on Computer Vision and Image Analysis*, 11(8):4883–4893, 2011.
10. M. Gabryel, R. K. Nowicki, M. Woźniak, and W. M. Kempa. Genetic cost optimization of the $GI/M/1/N$ finite-buffer queue with a single vacation policy. *Lecture Notes in Artificial Intelligence - ICAISC'2013*, 7895:12–23, 2013. DOI: 10.1007/978-3-642-38610-7_2.

11. M. Gabryel, M. Woźniak, and R. K. Nowicki. Creating learning sets for control systems using an evolutionary method. *Lecture Notes in Computer Science - ICAISC'2012*, 7269:206–213, 2012. DOI: 10.1007/978-3-642-29353-5_24.
12. E. Hetmaniok, I. Nowak, D. Słota, and A. Zielonka. Determination of optimal parameters for the immune algorithm used for solving inverse heat conduction problems with and without a phase change. *Numer. Heat Transfer B*, 62:462–478, 2012.
13. E. Hetmaniok, D. Słota, and A. Zielonka. Experimental verification of immune recruitment mechanism and clonal selection algorithm applied for solving the inverse problems of pure metal solidification. *Int. Comm. Heat & Mass Transf.*, 47:7–14, 2013.
14. M. H. Horng. Vector quantization using the firefly algorithm for image compression. *Expert Systems with Applications*, 39(1):1078–1091, 2012.
15. F. Keshtkar and W. Gueaieb. Segmentation of dental radiographs using a swarm intelligence approach. In *Proceedings of Canadian Conf. Electrical and Computer Engineering*, pages 328–331, 2006.
16. E. Lakehal. A swarm intelligence based approach for image feature extraction. In *Proceedings of International Conference on Multimedia Computing and Systems*, pages 31–35, 2009.
17. I. Martisius, D. Birvinskas, R. Damaševičius, and V. Jusas. EEG dataset reduction and classification using wave atom transform. *Lecture Notes in Computer Science - ICANN'2013*, 8131:208–215, 2013.
18. A. Mishra, C. Agarwal, A. Sharma, and P. Bedi. Optimized gray-scale image watermarking using DWT SVD and firefly algorithm. *Expert Systems with Applications*, 41(17):7858–7867, 2014.
19. C. Napoli, G. Papplardo, and E. Tramontana. Improving files availability for bittorrent using a diffusion model. In *IEEE 23nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE 2014*, pages 191–196, June 2014.
20. A. Niewiadomski. Imprecision Measures for Type-2 Fuzzy Sets. Applications to Linguistic Summarization of Databases. *Lecture Notes in Artificial Intelligence*, 5097:285–294, 2008.
21. R. K. Nowicki, B. Nowak, and M. Woźniak. Rough k nearest neighbours for classification in the case of missing input data. In George A. Papadopoulos, editor, *Proceedings of the 9th International Conference on Knowledge, Information and Creativity Support Systems*, pages 196–207. University of Cyprus Press, 6-8 November, Limassol, Cyprus, 2014.
22. S. Ouadfel and M. Batouche. MRF-based image segmentation using ant colony system. *Electronic Letters on Computer Vision and Image Analysis*, 2(2):12–24, 2013.
23. Y. Wang Q. Wan. Detecting moving objects by ant colony system in a MAP-MRF framework. In *Proceedings of International Conference on E-Product E-Service and E-Entertainment*, pages 1-4, 2010.
24. J. Tian, W. Yu, L. Chen, and L. Ma. Image edge detection using variation-adaptive ant colony optimization. *Lecture Notes in Computer Science*, 6910:27–40, 2011.
25. T. Uktveris. Efficiency analysis of object position and orientation detection algorithms. *Communications in Computer and Information Science - ICIST'2014*, 465:302–311, 2014.
26. G. S. Walia and R. Kapoor. Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search. *Expert Systems with Applications*, 41(14):6315–6326, 2014.

27. M. Woźniak. On applying cuckoo search algorithm to positioning $GI/M/1/N$ finite-buffer queue with a single vacation policy. In *Proceedings of the 12th Mexican International Conference on Artificial Intelligence - MICAI'2013*, pages 59–64, 24-30 November, Mexico City, Mexico, 2013. IEEE. DOI: 10.1109/MICAI.2013.12.
28. M. Woźniak. Fitness function for evolutionary computation applied in dynamic object simulation and positioning. In *Proceedings of the IEEE Symposium Series on Computational Intelligence - SSCI'2014 : 2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems - CIVTS'2014*, pages 108–114, 9-12 December, Orlando, Florida, USA, 2014. IEEE. DOI: 10.1109/CIVTS.2014.7009485.
29. M. Woźniak. On positioning traffic in nosql database systems by the use of particle swarm algorithm. In *Proceedings of XV Workshop DAGLI OGGETTI AGLI AGENTI - WOA'2014*, page paper 5, 25-26 September, Catania, Italy, 2014. CEUR Workshop Proceedings (CEUR-WS.org), RWTH Aachen University.
30. M. Woźniak, W. M. Kempa, M. Gabryel, and R. K. Nowicki. A finite-buffer queue with single vacation policy - analytical study with evolutionary positioning. *International Journal of Applied Mathematics and Computer Science*, 24(4):887–900, 2014. DOI: 10.2478/amcs-2014-0065.
31. M. Woźniak, W. M. Kempa, M. Gabryel, R. K. Nowicki, and Z. Shao. On applying evolutionary computation methods to optimization of vacation cycle costs in finite-buffer queue. *Lecture Notes in Artificial Intelligence - ICAISC'2014*, 8467:480–491, 2014. DOI: 10.1007/978-3-319-07173-2_41.
32. M. Woźniak and Z. Marszałek. An idea to apply firefly algorithm in 2D images key-points search. *Communications in Computer and Information Science - ICIST'2014*, 465:312–323, 2014. DOI: 10.1007/978-3-319-11958-8_25.
33. M. Woźniak and D. Połap. Basic concept of cuckoo search algorithm for 2D images processing with some research results. In *Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications - SIGMAP'2014*, pages 164–173, 28-30 August, Vienna, Austria, 2014. SciTePress - INSTICC. DOI: 10.5220/0005015801570164.
34. M. Woźniak and D. Połap. On some aspects of genetic and evolutionary methods for optimization purposes. *International Journal of Electronics and Telecommunications*, 61(1):7–16, 2015. DOI: 10.1515/eletel-2015-0001.
35. X. Zhuang, G. Yang, and H. Zhu. A model of image feature extraction inspired by ant swarm system. In *Proceedings of International Conference on Natural Computation*, pages 553–557, 2008.