Multilevel Thresholding Image Segmentation based on Autonomous Groups Particle Swarm Optimization

Murinto

1Informatic of Engineering, Faculty of Information Technology, Universitas Ahmad Dahlan Yogyakarta, Indonesia

murintokusno@tif.uad.ac.id

Abstract. Multilevel thresholding problems based on Otsu criteria are discussed in this paper. One weakness of the Otsu method is that computational time increases exponentially according to the number of thresholding dimensions. In this paper, a modified Particle Swarm Optimization (PSO) algorithm called Autonomous Groups Particles Swarm Optimization (AGPSO) is proposed to reduce two problems trapped in local minima and a slow convergence rate in solving high-dimensional problems. AGPSO is used for multilevel thresholding image segmentation. The performance of AGPSO is compared with standard PSO on three natural images. The parameters used to compare the performance of AGPSO and PSO are SSIM, PSNR, Computation Time, optimal threshold obtained from each algorithm. From the experimental results show that AGPSO is better when compared to PSO in image segmentation, from the resulting fitness value and higher SSIM and PNSR values.

1. Introduction

There are various methods of image segmentation that currently exist in the literature. One of the most commonly used segmentation methods is thresholding [1]. Among the several thresholding methods available, the Otsu method that deals with optimization issues has received much attention in the research that has been conducted [2]. Although the performance of the thresholding method is very good in bilevel thresholding, in multilevel thresholding this will require a lot of calculation time so it is not effective to overcome optimization problems [3]. On the other hand, there are various bioinspired algorithms that have advantages in their ability to learn to overcome optimization problems, some of which are genetic algorithm (GA) and particle swarm optimization (PSO). Both of these algorithms are widely used in optimization problems [4]. From previous studies in [5] [6] [7], PSO is more efficient and appropriate when used to find the optimal threshold of more than two. However, it is well known that one of the disadvantages of PSO is trapped in optimal local and premature convergence in handling complex optimization problems [8].

In this paper, a modified Particle Swarm Optimization (PSO) algorithm called Autonomous Groups Particles Swarm Optimization (AGPSO) is proposed to reduce two problems trapped in local minima and a slow convergence rate in solving high dimensional problems. The main idea of the AGPSO algorithm is inspired by the diversity of individuals in groups of birds or insects. Different functions with various slopes, arcs, and interception points are used to adjust the social and cognitive parameters of the PSO algorithm to provide different behavioural particles as in natural colonies. This paper is organized into several sections. Part 1 is an introduction section that contains background problems and proposed solutions. In Section 2, the conventional PSO algorithm and the proposed AGPSO algorithm are briefly explained. Section 3 provides a description of the multilevel thresholding problem based on Otsu criteria. Section 4 presents the results of the experiments to evaluate the performance of the proposed algorithm. Finally, conclusions are given in Section 5.
2. Autonomous Group Particle Swarm Optimization

The particle swarm optimization (PSO) technique was first introduced by Kennedy and Eberhart (1995) [9], is a stochastic optimization technique similar to the behaviour of bird flocks or sociological behaviour of a group of humans. The basic idea of PSO is to involve a scenario where a flock of birds in the search for food in an area. All birds do not know exactly where the food is, but with each iteration, they will know how far the food will be found. The best strategy will be followed by birds close to food and also from the previous best position achieved. PSO was built with the concept of optimization through a swarm particle. The PSO algorithm is a multiagent parallel search technique that maintains a particle swarm and each particle represents a potential solution in the swarm. All particles fly through the multi-dimensional search space by adjusting their position based on their own experience and from their neighbours.

The original PSO algorithm is written in the form of velocity updated and position updated (Kennedy and Eberhart, 1995) as shown in equations (1) and equation (2) respectively.

\[
v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \cdot r_1 \cdot (p_{id}^t - x_{id}^t) + c_2 \cdot r_2(p_{gd} - x_{id}^t) \\
x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}
\]

where \( c_1 \) and \( c_2 \) are positive constants, called the acceleration coefficient. \( r_1 \) and \( r_2 \) are two random numbers with values within range \([0,1]\). \( w \) is the weight of inertia (inertia weight). A large value of inertia weight will facilitate a global exploration while a small inertia weight facilitates local exploitation. The \( i \) particle is represented as \( X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \). The best previous position of particle \( i \) is stored and represented as \( P_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \). The position provides the best fitness value (the best fitness value). The index of the best particles among all the particles in the population is represented by the symbol \( g \). The rate of change of position (velocity) for particle \( i \) is represented as \( V_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \). During the update process, the speed of each dimension of a particle is limited to \( V_{\text{max}} \). \( D \) is the dimension of each search space.

The concept of autonomous groups is inspired by the diversity of individuals in a group of animals or a group of insects. In some cases, individuals are not quite similar in terms of intelligence and ability, but they all carry out their respective duties as members of the group. Each individual is able to work in any situation. In standard PSO, all particles have the same behaviour in terms of local and global search (local and global search), so the particles are considered as a group with the same strategy. However, using autonomous groups with a common goal in any population-based optimization algorithm theoretically can produce more randomized and directed searches simultaneously. In the autonomous mathematical model, groups consist of different strategies for updating \( c_1 \) and \( c_2 \). The renewal strategy of autonomous groups can be applied with various continuous functions at intervals \([0,1]\). Some functions can be used to update cognitive factors (\( c_1 \)) and social factors (\( c_2 \)). These functions consist of ascending or descending linear and polynomial, as well as exponential and logarithmic functions. In Figure 1 the \( c_1 \) and \( c_2 \) updates are shown, where the blue curve is for \( c_1 \) update while the red curve is for \( c_2 \) update. From the picture, it can be seen that \( c_1 \) decreases according to the iteration, whereas \( c_2 \) increases. From this, it is clear that the particles tend to have higher local search capabilities when \( c_1 \) is higher than \( c_2 \). On the other hand, the search for the search space particle is more global when \( c_2 \) is higher than \( c_1 \). In autonomous groups how to find a balance between \( c_1 \) and \( c_2 \) as dynamic coefficients core of this algorithm [10].

![Figure 1](image-url)
3. Multilevel Thresholding Multispectral Image Segmentation

The Otsu method is a non-parametric method for segmentation that divides the entire image into classes so that the variance of the different classes is maximum. In his research, Otsu (1979) defines variance between classes as a sum of the Sigma functions of each class, which are written as equations:

\[ g(t) = \sigma_0 + \sigma_1 \]  
\[ \sigma_0 = w_0 + (\mu_0 - \mu_A)^2 \]  
\[ \sigma_1 = w_1 + (\mu_1 - \mu_A)^2 \]  

Where \( \mu_T \) is intensity mean of origin image.

In bi-level thresholding, mean level each of levels (\( \mu_i \)) can be written as:

\[ \mu_0 = \sum_{i=0}^{T-1} \frac{|p_i|}{w_0} \]  
\[ \mu_1 = \sum_{i=T}^{L-1} \frac{|p_i|}{w_1} \]  

The optimal threshold is obtained from the maximization function between class variances, which is written as an equation:

\[ T^* = \arg\max g(t) \]  

As for the multi thresholding problem, the Otsu method can be written as:

\[ \sigma_0 = w_0 + (\mu_0 - \mu_A)^2, \sigma_1 = w_1 + (\mu_1 - \mu_A)^2, \sigma_j = w_j + (\mu_j - \mu_A)^2, \sigma_m = w_m + (\mu_m - \mu_A)^2 \]  

In multilevel thresholding, the mean level for each of the classes (\( \mu_i \)) can be written as:

\[ \mu_0 = \sum_{i=0}^{T_1-1} \frac{|p_i|}{w_0}, \mu_1 = \sum_{i=T_1}^{T_2-1} \frac{|p_i|}{w_1}, \mu_j = \sum_{i=T_j}^{T_{j+1}-1} \frac{|p_i|}{\mu_j}, \mu_m = \sum_{i=T_n}^{L-1} \frac{|p_i|}{\mu_n} \]  

The optimal multilevel threshold is obtained by maximizing the variance between classes of functions, which is written as an equation:

\[ T^* = \arg\max (\sum_{i=0}^{n} \sigma_i) \]  

Threshold method is a method that is generally used in image segmentation to separate important targets from the background image. This is the basic principle for dividing image pixels into different regions through determining threshold features. A common threshold segmentation can be explained as follows: Assume there is a certain threshold and the image is divided into two groups of pixels with, one (target) with a threshold greater than the other (background) with a lower threshold value (T). There are two types of threshold segmentation methods: single and multi-threshold.

At the bi-level thresholding, the image pixels with grey levels below the T value will be assigned to one group \( K_0 \), while m pixels t with grey levels above T will then be included in another group, \( K_2 \). At the end of the thresholding process, an image I will be grouped into two groups. If an image I is assumed to be represented through the gray level L scale, then bilevel thresholding can be defined as an equation:

\[ K_0 = \{f(x,y) \in I|0 \leq f(x,y) \leq T - 1\} \]  
\[ K_1 = \{f(x,y) \in I|0 \leq f(x,y) \leq T - 1\} \]  

Multilevel thresholding uses more than one threshold value and produces an image with several groups.

\[ K_0 = \{f(x,y) \in I|0 \leq f(x,y) \leq T_1 - 1\} \]  
\[ K_1 = \{f(x,y) \in I|T_1 \leq f(x,y) \leq T_2 - 1\} \]  
\[ K_i = \{f(x,y) \in I|T_i \leq f(x,y) \leq T_{i+1} - 1\} \]  
\[ K_n = \{f(x,y) \in I|T_n \leq f(x,y) \leq L - 1\} \]  

Where \( T_1 = 1,2,\ldots,n \) are threshold values and \( n \) are number of thresholds.
4. Results and Discussion

In this research, the proposed strategy is applied to multilevel thresholding image segmentation which was tested on 3 test images, namely house image, mountain image and zebra image. Each test image has a size of 256 x 256 pixels, 8-bit gray level. Before the experiment, the parameters of AGPSO and PSO and other image segmentation parameters are determined first. The simulation is done using MATLAB R.2016b programming on an Intel Core i5-4210U 2.40 GHz processor with 4 GB RAM. The effectiveness and feasibility of the proposed approach for multilevel thresholding is carried out by maximizing between class variances (Otsu method), with the aim of selecting the best thresholding value and higher objective function values with rapid convergence. In Figure 2, three natural images and corresponding histogram images are shown. The AGPSO algorithm used in this study is compared with the standard PSO algorithm.

![Figure 2: Natural Images and their histogram (a) house (b) zebra (c) mountain](image)

Table 1 shows the results of the application of the PSO and AGPSO algorithms to the mountain multilevel thresholding image segmentation, where the level applied in this study is the level of m = 2,3,4,5.

| Threshold (m) | PSO method | APSO method |
|--------------|------------|-------------|
| m=2          | Origin Image | Histogram | Result Image | Histogram |
| m=3          | Origin Image | Histogram | Result Image | Histogram |
Optimal threshold values obtained using PSO and AGPSO for 1500 iterations, fitness values, CPU processing time, and PSNR, SSE, SSIM values are shown sequentially in Table 2, Table 3, Table 4 and Table 5.

**Table 2.** Comparison of optimal threshold values of the Otsu method using the PSO and AGPSO optimization algorithms

| Image  | \(m\) | PSO | AGPSO |
|--------|-------|-----|-------|
| House  | 2     | 61  | 61    |
|        | 3     | 35  | 35    |
|        | 4     | 28  | 54    |
|        | 5     | 21  | 39    |
| Zebra  | 2     | 117 | 118   |
|        | 3     | 62  | 130   |
|        | 4     | 58  | 105   |
|        | 5     | 51  | 74    |
| Mountain | 2   | 115 | 116   |
|         | 3    | 72  | 129   |
|         | 4    | 72  | 122   |
|         | 5    | 46  | 76    |

**Table 3.** Comparison of Fitness Otsu methods using PSO and AGPSO optimization algorithms

| Citra  | \(m\) | PSO | AGPSO | Deviation Standard |
|--------|-------|-----|-------|--------------------|
| House  | 2     | 810.6528 | 810.6528 | 0 | 0 |
|        | 3     | 958.4464 | 958.4464 | 28.2843 | 28.2843 |
|        | 4     | 1.0085e+03 | 1.0085e+03 | 29.0517 | 29.0517 |
|        | 5     | 1.0292e+03 | 1.0292e+03 | 29.3201 | 29.3201 |
| Zebra  | 2     | 1.4861e+03 | 1.4861e+03 | 0 | 0 |
|        | 3     | 1.7017e+03 | 1.7017e+03 | 48.0833 | 48.0833 |
|        | 4     | 1.7992e+03 | 1.7992e+03 | 56.2406 | 56.2406 |
|        | 5     | 1.8450e+03 | 1.8449e+03 | 55.3918 | 55.3918 |
| Mountain | 2   | 1.7068e+03 | 1.7068e+03 | 0 | 0 |
|         | 3    | 1.9285e+03 | 1.9285e+03 | 40.3051 | 40.3051 |
|         | 4    | 2.0404e+03 | 2.0404e+03 | 55.0757 | 55.0757 |
|         | 5    | 2.0907e+03 | 2.0907e+03 | 59.4306 | 59.4306 |
Table 4. Comparison of Processing Times (in seconds) the Otsu method uses the PSO and AGPSO optimization algorithms

| Citra  | m | PSO       | AGPSO      |
|--------|---|-----------|------------|
| House  | 2 | 4.134520  | 4.885279   |
|        | 3 | 6.456734  | 6.746695   |
|        | 4 | 8.387650  | 8.652968   |
|        | 5 | 9.345620  | 10.582818  |
| Zebra  | 2 | 4.097349  | 4.533585   |
|        | 3 | 6.283603  | 6.608194   |
|        | 4 | 7.649079  | 8.950233   |
|        | 5 | 9.373399  | 10.953475  |
| Mountain | 2 | 4.074655  | 4.557945   |
|        | 3 | 5.867928  | 6.518023   |
|        | 4 | 7.639760  | 9.288086   |
|        | 5 | 9.485907  | 10.171466  |

Table 5. Comparison of the Otsu method PSNR, SSE, SSIM using the PSO and AGPSO optimization algorithms

| Citra  | m | PSO       | AGPSO     |
|--------|---|-----------|-----------|
|        |   | PSNR (dB) | MSE       | SSIM     | PSNR (dB) | MSE       | SSIM     |
| House  | 2 | 17.3799   | 1.1887e+03| 0.2017  | 17.3562   | 1.1953e+03| 0.2666  |
|        | 3 | 21.1764   | 495.9554  | 0.5246  | 21.1764   | 485.9554  | 0.5426  |
|        | 4 | 23.6710   | 279.2392  | 0.6469  | 23.6710   | 279.2392  | 0.6469  |
|        | 5 | 25.7122   | 174.5272  | 0.7581  | 25.7122   | 174.5272  | 0.7581  |
| Zebra  | 2 | 12.2313   | 3.8900e+03| 0.1841  | 12.2315   | 3.8898e+03| 0.1837  |
|        | 3 | 15.8922   | 1.6744e+03| 0.4402  | 15.8922   | 1.6744e+03| 0.4402  |
|        | 4 | 17.3867   | 1.1869e+03| 0.5063  | 17.3867   | 1.1869e+03| 0.5063  |
|        | 5 | 19.2580   | 771.4082  | 0.6357  | 19.4757   | 733.69314 | 0.6480  |
| Mountain | 2 | 11.7879   | 4.3081e+03| 0.1699  | 11.7884   | 4.3077e+03| 0.1692  |
|        | 3 | 14.5919   | 2.2589e+03| 0.3309  | 14.5919   | 2.2589e+03| 0.3309  |
|        | 4 | 15.0430   | 2.0360e+03| 0.3371  | 15.0430   | 2.0360e+03| 0.3371  |
|        | 5 | 20.8853   | 530.3386  | 0.7002  | 20.8853   | 530.3886  | 0.7002  |

5. Conclusion
In this paper, a modified particle swarm optimization algorithm, AGPSO, is proposed to overcome two problems that arise in solving high dimensions. Both problems are trapped in the local minimum and the slow rate of convergence. The results show that PSO with autonomous particle groups outperform conventionally and some recent modifications of PSO in terms of escaping from the local minimum and convergence speed. The performance of AGPSO is compared to the PSO standard on three natural images. The parameters used to compare the performance of AGPSO and PSO are SSIM, PSNR, Computation Time, optimal threshold obtained from each algorithm. From the experimental results show that AGPSO is better when compared to PSO in image segmentation, from the resulting fitness value and higher SSIM and PNSR values.

Acknowledgement
This research is funded by The Research Institution of Universitas Ahmad Dahlan (LPPM) through the Competitive Research Grant scheme in 2019. Thanks to LPPM.

References
[1] Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic imaging, 13(1), 146-166. Another reference
[2] Vala, H. J., & Baxi, A. (2013). A review on Otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(2), 387-389.

[3] Otsu, N. (1979). A Threshold Selection Method for Grey Level Histograms. *IEEE Transactions on System, Man and Cybernetics* 9(1), 62–66.

[4] Paulinas, M., & Ušinskas, A. (2007). A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control*, 36(3).

[5] Murinto, M., Astuti, N. R. D. P., & Mardhia, M. M. (2019). Multilevel thresholding hyperspectral image segmentation based on independent component analysis and swarm optimization methods. *International Journal of Advances in Intelligent Informatics*, 5(1), 66-75.

[6] Sathya, P. D., & Kayalvizhi, R. (2010). PSO-based Tsallis thresholding selection procedure for image segmentation. *International Journal of Computer Applications*, 5(4), 39-46.

[7] Maitra, M., & Chatterjee, A. (2008). A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Systems with Applications*, 34(2), 1341-1350.

[8] Gao, H., Xu, W., Sun, J., & Tang, Y. (2009). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(4), 934-946.

[9] Mirjalili, S., Lewis, A., & Sadiq, A. S. (2014). Autonomous particles groups for particle swarm optimization. *Arabian Journal for Science and Engineering*, 39(6), 4683-4697.

[10] Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942-1948).