

# A Novel Method for Segmentation of Remote Sensing Images based on Hybrid GA-PSO

Pedram Ghamisi

Research Scholar

Geodesy and Geomatics Engineering Faculty,

K.N.Toosi University of Technology,  
No. 1346, Vali\_Asr St., Tehran, Iran

## ABSTRACT

Image segmentation is defined as the process of dividing an image into disjoint homogenous regions and it could be regarded as the fundamental step in various image processing applications. In this paper, a novel multilevel thresholding segmentation method is proposed for grouping the pixels of remote sensing (RS) images into different homogenous regions. In this way, Hybrid Genetic Algorithm-Particle Swarm Optimization (HGAPSO) is used for finding the optimal set of threshold values. The new method is tested on two different study areas and results are compared with PSO-based image segmentation comprehensively. Results show HGAPSO based image segmentation performs better than PSO-based method in different points of view.

**Keywords:** Segmentation, Hybrid GA-PSO, Multilevel thresholding method

## 1. INTRODUCTION

Segmentation refers to the process of partitioning a digital image into multiple objects. In other words, image segmentation could assign a label to each pixel in the image such that pixels with the same label share certain visual characteristics. These objects are more meaningful than each pixel. Image segmentation plays an important role in analyzing and understanding images [1]. The segmentation of images into meaningful objects could be useful for further image processing purposes such as classification and object recognition.

There are a few methods for image segmentation such as texture analysis based, histogram thresholding based, clustering based and region based split and merging methods [2].

One of the most famous methods for image segmentation is thresholding method, which is commonly used for segmentation of an image into two or more clusters [3]. Otsu introduced a nonparametric method for automatic threshold selection based on histogram evaluation for segmentation of images [4]. Otsu also introduced three objective functions to evaluate threshold value at the level  $t$ , as follows:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \quad (1)$$

Where  $\sigma_W^2$  is the within-class variance,  $\sigma_B^2$  is the between-class variance and  $\sigma_T^2$  is the total variance.

The problem of bi-level thresholding is reduced to an optimization problem to probe for the threshold  $t$  that maximizing the  $\sigma_B^2$  and minimizing  $\sigma_W^2$  [3]. For two level thresholding, the problem is solved by finding  $T^*$  which satisfies  $\max(\sigma_B^2(T^*))$  where  $0 \leq T^* < L$  and  $L$  is the maximum intensity value. This problem could be extended to  $n$ -level thresholding through satisfying  $\max(\sigma_B^2(T^*_1, T^*_2, \dots, T^*_n))$  that  $0 \leq T^*_1 < T^*_2 < \dots < T^*_n < L$ . One way for finding the optimal set of thresholds is the exhaustive search method. The exhaustive search method based on the Otsu criterion is simple, but it has a disadvantage that it is computationally expensive [3]. Exhaustive search for  $n - 1$  optimal thresholds involves evaluations of fitness of  $n(L-n+1)^{n-1}$  combinations of thresholds [3] so this method isn't suitable in computational cost's point of view.

The task of determining  $n - 1$  optimal thresholds for  $n$ -level image thresholding could be formulated as a multidimensional optimization problem. Some methods were proposed for solving the Otsu threshold selection such as finding the optimum set of thresholds by PSO and GA [5], [6], [1], [3] and [7]. In the follow, a brief description of GA and PSO are brought and strengths and weaknesses of these methods are perused.

### 1.1 Genetic Algorithm

Genetic algorithm (GA) is an adaptive method that can be used to solve search and optimization purposes [8]. GA is based on the genetic process of biological organisms. In comparison to other techniques, GA can emphasize much stronger on global, as opposed to local search and optimization [9]. Furthermore, GA is able to find an optimal solution without having to explore the whole (often vast) search space.

GA starts optimization with several solutions. Each of these solutions is called chromosome or individual. Each chromosome consists of several genes which can have different values. These genes carry the attributes of each individual. Set of The chromosomes constitute a population. Each chromosome receives a fitness value based on its genes. The fitter chromosomes are selected for generation. For generation phase, two fitter chromosomes are selected and their chromosomes are recombined to make a new offspring (or solution). The act of combination is done by crossover. After crossover, mutation is applied on each child individually [8]. This cycle is repeated until a termination criterion is met [10].

For solving the multilevel thresholding problem, some techniques using GA have been introduced [5], [6]. Segmentation methods

based on GA perform well for complex optimization problems particularly for problems in which variables are highly correlated [1].

## 1.2 Particle Swarm Optimization (PSO)

PSO is a very promising evolutionary computation technique that has been developed recently due to research on bird flock simulation by Kennedy and Eberhart [11]. The main specification of PSO is simplicity and velocity. The PSO algorithm acts such as genetic algorithms and, additionally, it can be implemented much easier than GA and has fewer parameters to adjust. PSO consists of the set of solution, which is called population. Each solution consists of the set of parameters and represents a point in multidimensional space. Each parameter could be a solution for our problem and calls particle. Group of particles (population) is called swarm. Particles move through the search space with a specified velocity for finding the optimal solution. Each particle keeps a memory which helps it in keeping the track of its previous best position. The positions of the particles are distinguished as personal best and global best. Particles' velocities are adjusted according to the historical behavior of each particle and its neighbors while they fly through the search space. Each move of particles is deeply affected by its current positions and its memory of previous useful parameters, and by the cooperation and group knowledge of the swarm [11]. Therefore, the particles have a tendency to fly towards the better and better search area over the search process course. The Velocity of  $i$ -th particles in  $k$ -th iteration determined as:

$$V_{id}^{k+1} = WV_{id}^k + C_1r_1(pb_{id}^k - x_{id}^k) + C_2r_2(gb_{id}^k - x_{id}^k) \quad (2)$$

$C_1$  and  $C_2$  are acceleration constant.  $C_1$  guides each particle towards local best position whereas  $C_2$  called social parameter and guides the particle towards global best position.  $r_1$  and  $r_2$  are random values in range of 0 and 1.  $W$  is the inertia weight and is predefined by user.  $X_{id}^k$  shows The position of each particle in  $d$ -dimensional search space.  $pb_{id}^k$  is the best previous position of each particles which is called particle best position.  $gb_{id}^k$  is the best position of the all particles and is called global best particle. The  $i$ -th particle position is updated by:

$$x_{id}^{k+1} = x_{id}^k + V_{id}^k \quad (3)$$

PSO is widely used for image segmentation and solving the multilevel thresholding problem by [1], [3] and [7].

## 1.3 Hybrid GA-PSO (HGAPSO)

Hybrid GA-PSO combines the standard velocity and updates rules of PSO with the ideas of selection, crossover and mutation from GAs.

One drawback of PSO is the swarm may prematurely converge. The main cause of this problem is that, particles try to converge to a single point, which is on the line between the global best and the personal best positions. This point is not guaranteed for a local optimum [12]. Another reason for this problem could be the fast rate of information flow between particles. In this manner, similar particles are created with a loss in diversity and the possibility of being trapped in local optima is increased [13].

Another problem in PSO is the different parameter settings for a stochastic search algorithm cause the high performance variances [13]. Generally, there isn't any specific parameter setting could be applied to all problems. Increasing the inertia weight ( $W$ ) will increase the speed of the particles and cause more exploration (global search) and less exploitation (local search) [13]. So, finding the best set of parameters isn't a simple task, and it might be different from one problem to another [13].

The main reason of using the PSO is its simple conceptual and could be implemented in a few lines of code. Further reason is PSO also have memory. In a GA if a chromosome is not selected, the information contained by that individual is lost. However, without a selection operator which has been used in GA, PSO may waste resources on inferior individuals [13]. A PSO's behavior enhances the search for an optimal solution, however a GA has problem for finding an exact solution and are best at reaching a global region [14]. In general, Hybrid GA-PSO is used to overcome these problems. Matthew Settles and Terence Soule suggest that the correct combination of GA and PSO has the potential to achieve better results faster and to work effective across a wide variety of problems. Both Angeline [15] and Eberhart [14] have suggested that a hybrid combination of the GA and PSO models could make a very efficient search strategy.

The Hybrid GA-PSO is made for optimization of problems in continuous, multidimensional search spaces [15].

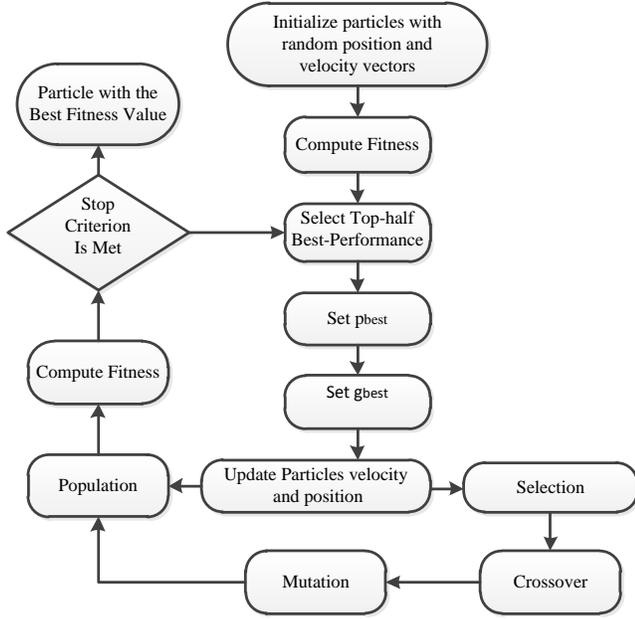
P.D. Sathya and R. Kayalvizhi proposed a multilevel thresholding method based on PSO and compared their method with GA-based thresholding method. Comparison showed PSO-based image segmentation runs faster, and more stable than GA [14]. Therefore, one goal of this paper is comparison of a proposed method with PSO-based one.

In this paper, a novel method for segmentation of RS images based on HGAPSO are introduced. This method can solve the Otsu problem for delineating multilevel threshold values and has a great potential to segment of images more efficiently than PSO. Result shows HGAPSO-based method is more stable and efficient than PSO-based method, in terms of finding the optimal set of thresholds for segmentation of RS images.

This paper is organized as follows. In Section 2, the proposed segmentation method, its structures and specifications, are presented. Section 3 is devoted to the experimental results and comparison of the new methods with the PSO-based one. Finally, Section 4 presents the concluding remarks and future works.

## 2. PROPOSED METHOD

In this research study, a novel Hybrid GA-PSO (HGAPSO) based method is proposed for segmentation of remote sensing images. Fig. 1 shows general engine of HGAPSO.



**Fig. 1: General engine of HGAPSO**

At first, some random individuals are produced. These individuals may be regarded as chromosomes in terms of GA, or as particles in terms of PSO. Then, new individuals on the next generation are created by enhancement, crossover and mutation operations.

**Enhancement:** In each generation, after calculating of all individuals' fitness functions, the top-half best-performing individuals are opted. These individuals are regarded as elites. Instead of reproducing the individuals directly by GA, elites are enhanced by PSO at first. Each elite is regarded as a particle and the total number of elites illustrates a swarm based on PSO, (2) and (3) are applied to the elites. By applying PSO on the elites, the search ability would increase. Half of the population in the next generation is produced by the enhanced individuals, and the remainder is occupied by crossover and mutation operations [13].

**Crossover:** The tournament-selection scheme is used for selection step. For producing the individuals with better performance, crossover only is done on individuals, which were enhanced by PSO. Two enhanced elites are selected in the random way, and their fitness values are compared to each other. Then, the elite with better fitness is opted. Further the other parent is selected in the same manner. One-point crossover operation is used for producing the new offsprings [13].

**Mutation:** In this research work, uniform mutation with a constant probability equal to 0.1 is used [13].

The inertia weight ( $W$ ) is an important factor for the PSO's or HGAPSO's convergence. It is used to control the impact of previous history of velocities on the current velocity. A large inertia weight factor facilitates global exploration (i.e., searching of a new area) while small weight factor facilitates local exploration. Therefore, it is better to choose large weight factor for initial iterations and gradually reduce weight factor in successive iterations. This can be done by using:

$$W = W_{max} - (W_{max} - W_{min}) \times Iter / Iter_{max} \quad (4)$$

Where  $W_{max}$  and  $W_{min}$  are initial and final weight respectively.  $Iter$  is the current iteration number and  $Iter_{max}$  is the maximum iteration number. The pseudo code for the HGAPSO for maximization of a fitness function is given in Algorithm 1. Also between-class variance ( $\sigma_B^2$ ) is selected as the fitness function and calculated as follows:

$$\sigma_B^2 = n_b(T)n_o(T) + (\mu_b - \mu_o)^2 \quad (5)$$

Where

$$n_b(T) = \sum_{i=0}^T p(i) \quad (6)$$

$$n_o(T) = \sum_{i=T+1}^N p(i) \quad (7)$$

$\mu_b$  and  $\mu_o$  are the mean level of class background and class foreground respectively.

The solution with the best fitness function is selected as the optimal set of image thresholds. It should be noticed that, the fitness value of each solution is calculated as follows:

$$F(Xi) = \sigma_B^2(Xi) \quad (4)$$

The best solution with the max fitness or  $X_{gbest}$  is declared by (5):

$$X_{gbest} = \text{Max } \sigma_B^2(T^*_1, T^*_2, \dots, T^*_n) \quad (5)$$

that  $1 \leq T^*_1 < T^*_2 < \dots < T^*_n < L$

**Algorithm 1: pseudo code of HGAPSO**

Initialize the number of population ( $N$ )  
 Initialize the number of Iteration ( $Iter$ )  
 Initialize  $C_1, C_2, W, X_{max}, X_{min}, V_{max}, V_{min}$   
 Initialize the Crossover and Mutation percent  
 Initialize the random  $X_{gbest}$

**FOR** each Population  $i$  **DO**  
**FOR** each variable  $v$  **DO**  
 Initialize the random  $X_{iv}$  between  $X_{max}$  and  $X_{min}$   
 Initialize the random  $V_{iv}$  between  $V_{max}$  and  $V_{min}$   
**ENDFOR**  
 Compute  $Fitness(X_i)$   
**IF**  $Fitness(X_i) > Fitness(X_{gbest})$  **THEN**  
 $X_{gbest} = X_{iv}$   
**ENDIF**  
**ENDFOR**

Sort  $X_i$  based on  $Fitness(X_i)$   
**While** ( $Iter < Iter_{max}$ ) **Do**  
 Select top-half best-performing  $X$  as elite  
**FOR** each elite  $i$  **DO**  
**FOR** each variable  $v$  **DO**  
 Compute  $V_{iv}$  by (3)  
 restrict  $V_{iv}$  between  $V_{max}$  and  $V_{min}$   
 Compute  $X_{iv}$  by (4)  
 restrict  $X_{iv}$  between  $X_{max}$  and  $X_{min}$   
**ENDFOR**  
**IF**  $Fitness(X_i) > Fitness(X_{pbest(i)})$  **THEN**  
**FOR** each variable  $v$  **DO**  
 $X_{pbest} = X_{iv}$   
**ENDFOR**

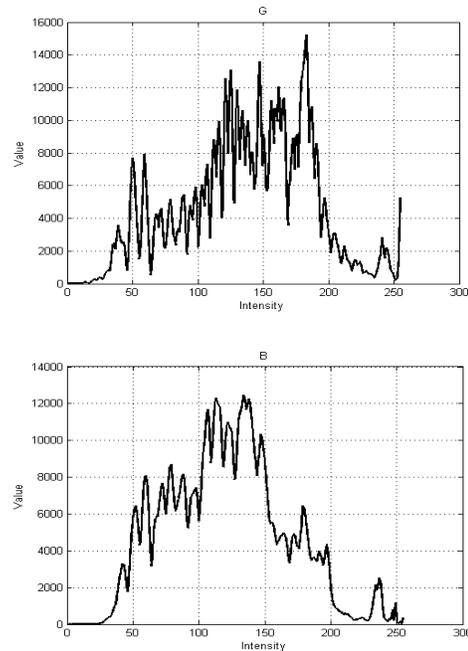
```

ENDIF
IF  $Cost(X_i) < Cost(X_{g_{best}})$  THEN
FOR each variable  $v$  DO
 $X_{g_{best} v} = X_{iv}$ 
ENDFOR
ENDIF
    Make new Population (1: N/2) with X
    Select pairs to mate from X
    Apply crossover operator
    Apply mutation operator
    Make new Population ((N/2) + 1: N) with X
FOR each Population  $i$  DO
FOR each variable  $v$  DO
        Initialize the random  $X_{iv}$  between  $X_{max}$  and  $X_{min}$ 
        Initialize the random  $V_{iv}$  between  $V_{max}$  and  $V_{min}$ 
ENDFOR
    Compute  $Cost(X_i)$ 
ENDFOR
    Iteration = iteration + 1
ENDFOR
ENDWhile
    
```

### 3. EXPERIMENTAL RESULT

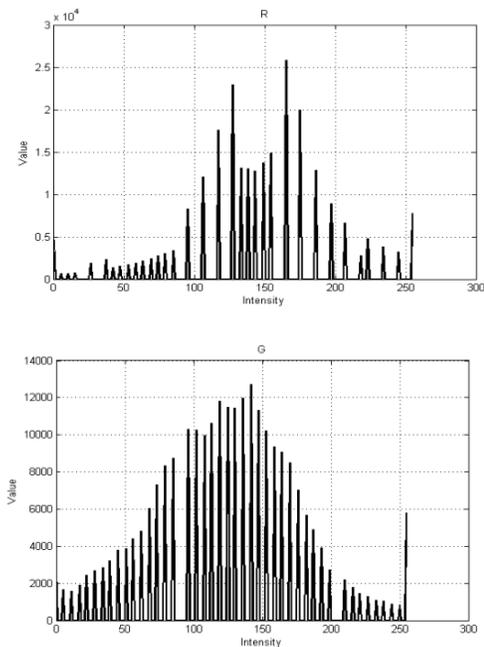
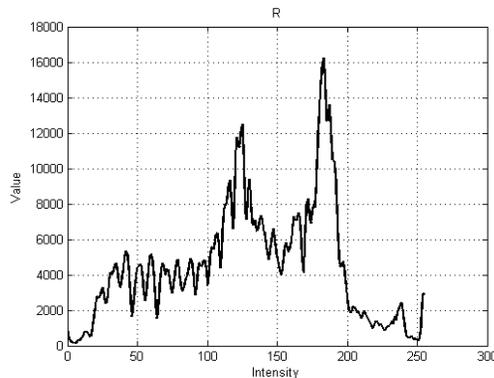
PSO- and HGAPSO-based image segmentation which are proposed in this paper were programmed in MATLAB on a computer having Intel Core 2 Duo E7500 processor and 2GB of memory.

Two RS images are used for the experiments which were captured by different sensors with different spectral and spatial resolutions. The first data set has a wide variety of intensity values. However, the second one is scarce and has limited digital numbers. In the follow, these data sets and the corresponding results will be described. The first test case is a RGB image which was captured from Atlanta by GeoEye Inc. with 1200×950 dimension size. Fig. 4 (a) shows this image and Fig. 2 illustrates the histogram of R, G and B respectively. The second test case is a false color RGB image from Canon city, Colorado, which was captured by Landsat Thematic Mapper (TM) sensor which has about 30 meter resolution. In this data set, band 4, 3 and 2 are used for showing R, G, and B respectively. Fig. 5 (a) shows this data set and Fig.3 illustrates the histogram of each band.



**Fig. 2: Histograms of R, G, and B for the first test case**

For comparing algorithms in computational time point of view, CPU process time for segmentation of each test case in different level is calculated and results are presented in table 2. The results show that the PSO has fewer computational time. The reason of higher computational time in HGAPSO is three extra processes (Selection, Crossover and Mutation) which must be applied. The difference between two methods' computational time is so low and could be eliminated.



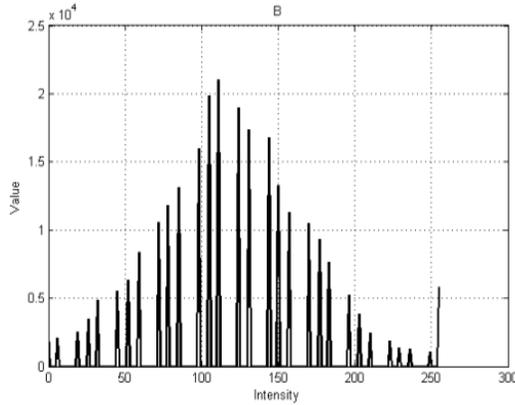


Fig. 3: Histograms of R, G, and B for the second test case

The proposed multilevel thresholding technique based on HGAPSO is implemented with the following parameters.

Table 1: Initial parameters of HGAPSO

Parameter	Value
Population	30
Number of Iteration	First test case:25 Second test case:15
$C_1$	2
$C_2$	2
$W_{max}$	0.4
$W_{min}$	1
$V_{max}$	10
$V_{min}$	-10
$X_{max}$	255
$X_{min}$	0
Crossover probability	0.8
Mutation probability	0.1

Since all the optimization algorithms are organized based on random search and stochastic way for solving problems, the results of experiments are not completely the same in each iteration of the algorithm. Hence, it is necessary to analyze the stability of all the algorithms [14]. This comparison is used for finding which algorithm is more stable than others. In [14]

standard deviation is used for showing the stability of the algorithm. Standard deviation is calculated as follows:

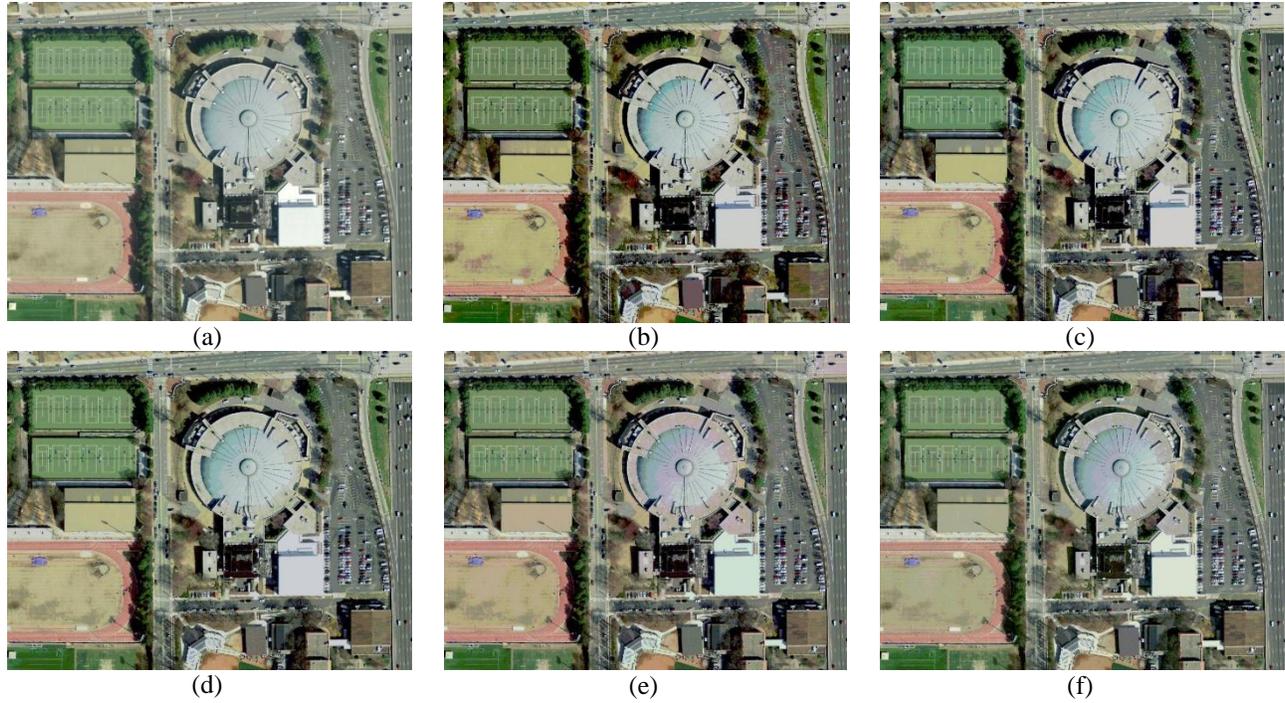
$$\text{Standard deviation} = \sqrt{\frac{\sum_{i=1}^{Ite} (\sigma_i - \mu)^2}{Ite}}$$

$Ite$  is the number of iteration of the algorithm.  $\sigma_i$  is the best fitness value in the  $i$ -th iteration of the algorithm and  $\mu$  is the mean value of  $\sigma$ .

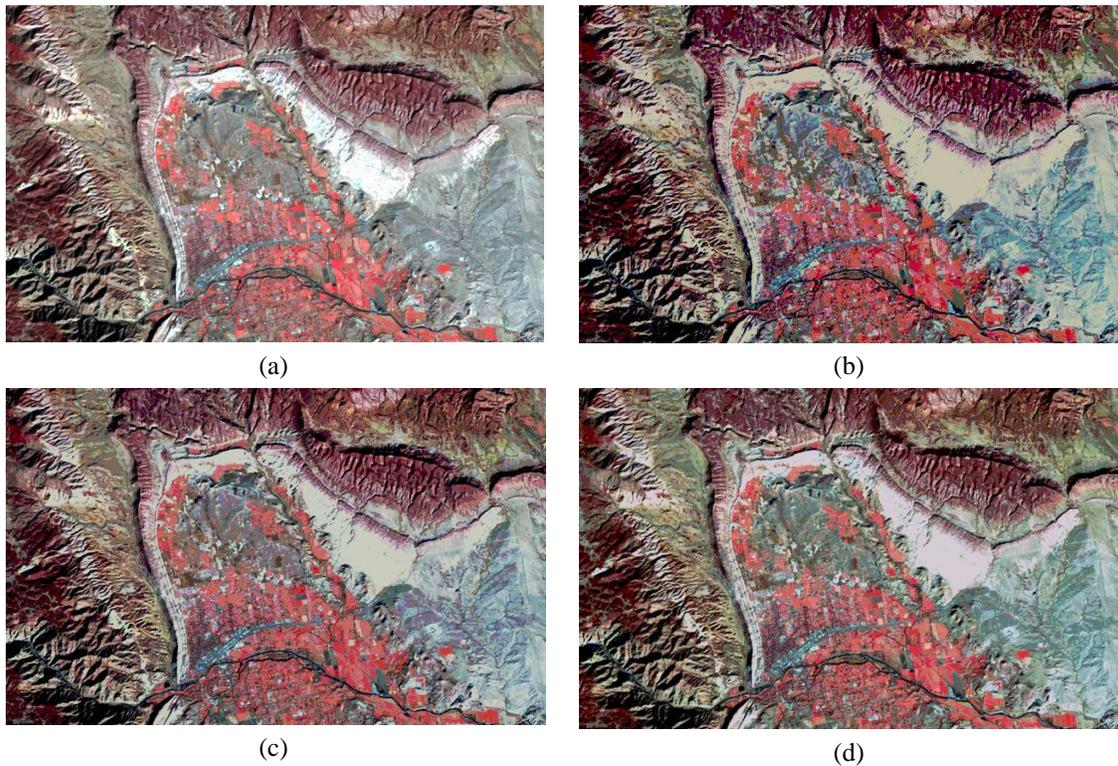
Table 2: standard deviation and CPU time for different levels of the first and second test cases

Test Images	Level	Standard Deviation		CPU time	
		PSO	HGAPSO	PSO	HGAPSO
Test case 1	6	0.4854	<b>0.4566</b>	1.0017	2.0561
		1.1877	<b>0.3521</b>		
		2.9401	<b>0.4878</b>		
	7	0.6443	<b>0.5314</b>	1.1571	2.2216
0.7459		<b>0.4625</b>			
1.1403		<b>0.9216</b>			
8	<b>0.6163</b>	0.7355	1.3128	2.3868	
	1.0756	<b>0.5422</b>			
	1.8001	<b>0.6939</b>			
9	1.2430	<b>0.4402</b>	1.4799	2.5510	
	<b>0.7180</b>	0.8056			
	1.7377	<b>0.8699</b>			
Test case 2	4	1.2985	<b>0.8429</b>	0.5029	1.5429
		<b>0.1390</b>	0.4829		
		0.6166	<b>0.5984</b>		
	5	1.0945	<b>0.7581</b>	0.2933	1.3424
		2.6909	<b>0.9437</b>		
		2.9892	<b>1.5501</b>		
6	<b>0.5557</b>	0.8278	0.3487	1.4050	
	<b>1.0633</b>	1.5730			
	1.0453	<b>0.5567</b>			

The result of standard deviation is shown in table 2. As can be seen standard deviation in almost all experimental results based on HGAPSO is fewer than PSO-based segmentation. So, HGAPSO-based segmentation is more stable than PSO-based one. As can be seen in previous section, the HGAPSO could be converged in fewer iteration than other methods such as GA and PSO. As a result,  $\sigma_i$  remains same in more iteration. Subsequently, the difference between  $\sigma_i$  and  $\mu$  decreases and consequently the standard deviation value which is calculated by (6) is decreased.



**Fig. 4: The First test case. (a) Input image, (b) 5 levels segmented image, (c) 6 levels segmented image, (d) 7 levels segmented image, (e) 8 levels segmented image, (f) 9 levels segmented image**



**Fig. 5: The First test case. (a) Input image, (b) 4 levels segmented image, (c) 5 levels segmented image, (d) 6 levels segmented image**

**Table 3: Optimal threshold value and fitness value for different levels of the first and second test cases**

Test Images	Level	Optimal Threshold Value					Fitness value	
		PSO					PSO	HGAPSO
Test case 1	6	57 98 136 168 211	58 100 136 169 205	2.9013e+003	<b>2.9020e+003</b>			
		73 107 140 170 208	73 110 139 168 210	<b>2.2600e+003</b>	2.2599e+003			
		62 93 122 162 205	69 99 128 160 206	1.7938e+003	<b>1.7998e+003</b>			
	7	47 79 111 140 166 205	47 79 107 136 168 208	2.9281e+003	<b>2.9283e+003</b>			
63 90 119 143 173 212		65 99 128 152 178 218 67	<b>2.2811e+003</b>	2.2810e+003				
70 94 119 143 167 209		93 119 143 169 205	1.8224e+003	<b>1.8231e+003</b>				
8	40 71 102 128 147 172 208	43 74 102 128 152 175 206	2.9444e+003	<b>2.9467e+003</b>				
	56 83 108 132 152 177 209	64 93 118 140 165 187 222	2.2958e+003	<b>2.2975e+003</b>				
	60 81 102 123 144 169 213	69 86 107 129 153 177 206	<b>1.8351e+003</b>	1.8334e+003				
9	47 75 104 124 147 168 185	47 73 102 130 150 174 194	2.9571e+003	<b>2.9577e+003</b>				
	217	220	2.3077e+003	<b>2.3096e+003</b>				
	53 78 108 132 153 169 187	60 89 114 132 149 169 194	<b>1.8378e+003</b>	1.8346e+003				
207	45 66 95 117 135 147 172	6 70 92 113 131 149 171						
	207	209						
Test case 2	4	86 148 198	85 146 201	2.3464e+003	2.3464e+003			
		82 137 194	80 141 198	2.5629e+003	2.5629e+003			
		81 134 183	84 139 189	2.4364e+003	2.4364e+003			
	5	63 120 161 198	66 119 160 199	2.4539e+003	2.4539e+003			
		59 104 148 202	57 104 151 208	2.6902e+003	2.6902e+003			
		49 89 138 185	47 91 132 189	2.5507e+003	2.5507e+003			
6	44 95 132 155 202	42 103 140 172 211	2.5071e+003	<b>2.5109e+003</b>				
	48 85 120 154 209	49 90 121 153 208	2.7620e+003	2.7620e+003				
	48 91 126 169 203	49 85 125 167 221	2.6229e+003	<b>2.6249e+003</b>				

Table 3 shows the optimal threshold value for each level and corresponding fitness value. Both of fitness value and optimal threshold value are calculated for R, G, and B. In the first test case, most of the time, HGAPSO acts more significant than PSO. HGAPSO has a great potential for finding the optimal set of thresholds in complicated images with the wide variety of intensities.

In the second test case mostly, two methods act in the same way when the level of the segmentation is low. In this situation, population has the low number of variables, so the crossover couldn't act well and as a result, the output of the HGAPSO is related to the enhance elites with the PSO and mutation only and consequently HGAPSO- and PSO-based method could has the same result.

In segmentation with high levels, when the input image has wide variety of intensities, it is recommended to use HGAPSO for image segmentation. For segmentation of image with low levels, there isn't much difference between results of PSO- and HGAPSO-based methods. Thus it can be concluded that HGAPSO-based segmentation is more efficient than PSO-based one in general.

#### 4. CONCLUSION AND FUTURE WORK

In this study, a new method for segmentation of RS images is introduced. This method is based on HGAPSO and could be used for segmentation of RGB or grayscale images. HGAPSO is proposed for solving the Otsu problem for delineating multilevel threshold values and overcome the disadvantages of GA and PSO based methods. In this case, the performance of HGAPSO and PSO is compared from different points of view such as CPU time, Standard deviation, optimal threshold value and corresponding

fitness value. Results indicate that HGAPSO is more stable than PSO and has higher potential for finding the optimal set of thresholds with better fitness value than another method, especially when the level of segmentation is high and image has a wide variety of intensities.

#### 5. REFERENCES

- [1] D. B. Fogel, "Evolutionary computation: Toward a new philosophy of machine intelligence, Second edition, Piscataway, NJ: IEEE Press, 2000.
- [2] A.D. Brink, "Minimum spatial entropy threshold selection", IEE Proceedings on Vision Image and Signal Processing 142 (1995), 128-132.
- [3] R. V. Kulkarni and G. K. Venayagamoorthy Bio-Inspired Algorithms for Autonomous Deployment and Localization of Sensor, IEEE trans. On systems, vol. 40, no. 6, 2010. 663-675.
- [4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [5] C. C. Lai, D. C. Tseng, "A hybrid approach using Gaussian smoothing and genetic algorithm for multilevel thresholding", *International Journal of Hybrid Intelligent Systems* 1(3) (2004),143-152.
- [6] P. Y. Yin, "A fast scheme for optimal thresholding using genetic algorithms", *Signal processing* 72 (1999), 85–95.

- [7] Yi-Tung Kao, Erwie Zahara and I-Wei Kao, “ A hybridized approach to data clustering,” *Expert Systems with Applications* 34 (2008), 1754-1762.
- [8] D. Beasley, D. Bull, and R. Martin, An overview of genetic algorithms. *Univ. Computing* 15 (2), 1993, no. 58-69, part 1.
- [9] M.Frieke, B. V. Coillie, P.Lieven, C. Verbeke, and R.Robert, “selection by genetic algorithms in object-based classification of ikonos imagery for forest mapping in flanders,” *Remote Sensing of Environment*, vol. 110, no. 7, pp. 476–487, Jul 2007.
- [10] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Massachusetts, USA: Addison-Wesley Longman., 1989
- [11] J. Kennedy, and R. Eberhart, “Particle Swarm Optimization”, *Proceedings of the IEEE conference on neural networks ICNN’95, Perth, Australia* 4 (1995), 1942-1948.
- [12] P. Y. Yin, “A fast scheme for optimal thresholding using genetic algorithms”, *Signal processing* 72 (1999), 85–95.
- [13] C. F. Lu and C. F. Juang., “Evolutionary fuzzy control of flexible ac transmission system,” *IEE Proc.-Gener. Transm. Distrib.*, vol. 152, no. 4, pp. 441–448, 2005.
- [14] P.D. Sathya and R. Kayalvizhi, PSO based tsallis tresholding selection procedure for image segmentation, *International Journal of Computer Applications*, Vol.5, No.4, pp. 39-46, 2010