CLPSO-based Fuzzy Color Image Segmentation

A. Borji
School of Cognitive Sciences
Institute for Studies in Theoretical Physics and Mathematics
Tehran, IRAN
borji@ipm.ir

M. Hamidi
Department of Electrical and Computer Engineering
Azad university of Qazvin
Qazvin, IRAN
mandana.hamidi@gmail.com

A. M. Eftekhari moghadam
Department of Electrical and Computer Engineering
Azad university of Qazvin
Qazvin, IRAN
eftekhari@itrc.ac.ir

Abstract - **A new method for color image segmentation using fuzzy logic is proposed in this paper. Our aim here is to automatically produce a fuzzy system for color classification and image segmentation with least number of rules and minimum error rate. Particle swarm optimization is a sub class of evolutionary algorithms that has been inspired from social behavior of fishes, bees, birds, etc, that live together in colonies. We use comprehensive learning particle swarm optimization (CLPSO) technique to find optimal fuzzy rules and membership functions because it discourages premature convergence. Here each particle of the swarm codes a set of fuzzy rules. During evolution, a population member tries to maximize a fitness criterion which is here high classification rate and small number of rules. Finally, particle with the highest fitness value is selected as the best set of fuzzy rules for image segmentation. Our results, using this method for soccer field image segmentation in Robocop contests shows 89% performance. Less computational load is needed when using this method compared with other methods like ANFIS, because it generates a smaller number of fuzzy rules. Large train dataset and its variety, makes the proposed method invariant to illumination noise.**

I. INTRODUCTION

The process of partitioning a digital image into multiple regions (sets of pixels) is called image segmentation. Actually, partitions are different objects in image which have the same texture or color. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics. Some of practical applications of image segmentation are: image processing, computer vision, face recognition, medical imaging, digital libraries, image and video retrieval, etc [1].

Image segmentation methods fall into five categories: Pixel based segmentation [2], Region based segmentation [3], Edge based segmentation [4], [5], Edge and region Hybrid segmentation [6] and Clustering based segmentation [7], [8], [9], [10], [11]. Color image segmentation using fuzzy classification is a pixel based segmentation method. A pixel is assigned a specific color by the fuzzy system. One approach in designing such a fuzzy system is an expert to look at training data and try to manually develop a set of fuzzy rules. Two drawbacks with such method are that first it is very cumbersome and time consuming and second there is no guarantee that the produced fuzzy rules are the best possible ones. So, there is need for a method which could produce fuzzy rules and membership functions automatically. Several methods have been proposed in literature for automatic production of fuzzy rules like genetic algorithms and ANFIS [12], [13], [14], [15]. One problem with these methods is that they generate a large number of fuzzy rules which causes slow classification and processing speed [8]. In this paper, we propose a method which produces a smaller number of rules while preserving low error rate. To that aim we use CLPSO to search for a set of such fuzzy rules.

The rest of this paper is organized as follows: in section two, fuzzy color classification is explained. An introduction to comprehensive learning particle swarm optimization is illustrated in section three. In section four we give the details of our proposed method for fuzzy color image segmentation using CLPSO. Our experimental setup and results are shown in section five. Finally, conclusions and discussions come in section six.

II. FUZZY COLOR CLASSIFICATION

Fuzzy color classification is a supervised learning method for segmentation of color images. This method assigns a color class to each pixel of an input image by applying a set of fuzzy rules on it. A set of training image pixels, for which the colors are known are used to train the fuzzy system. The trained fuzzy system will be later evaluated on test images.

Different color spaces like HSL, RGB, YIQ, etc, have been suggested in image processing, each suitable for different domains [1]. In this study, we chose HSL color space because a color in this space is represented in three dimensions: one which codes the color itself (H) and another two which explain details of the color, saturation (S) and lightness (L). Because of this segregation of color components, this color space is most suitable for our purpose [1].

As it can be seen in Fig. 1, H dimension is shown in a circle with colors occupying a range of degrees around it. Instead of assigning a specific hue value to each color around this circle, a fuzzy membership function can code for a color by giving it a range of hues each with different membership value. As an example, H dimension in Fig. 2 is partitioned into ten trapezoidal membership functions each one coding a different color.

1-4244-1214-5/07/$25.00 ©2007 IEEE

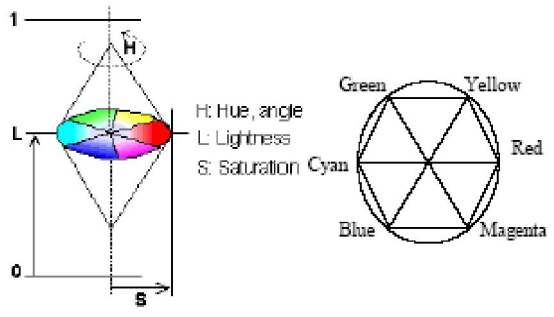

Fig. 1 HSL color space.

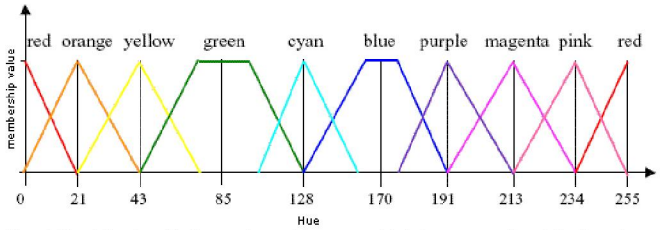

Fig. 2 Partitioning H dimension with trapezoidal fuzzy membership functions

Trapezoidal membership function showed in Fig. 3 needs four parameters to be specified [1].

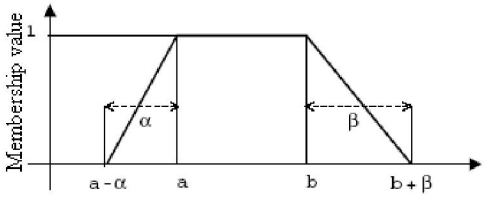

Fig. 3 trapezoidal fuzzy subset

To represent two remaining dimensions of a color, because of their less importance for determining a color compared with Hue dimension, we divide each dimension to three parts: weak, medium and strong. Combining these two dimensions we come to nine regions for representing a color shown in Fig. 4.

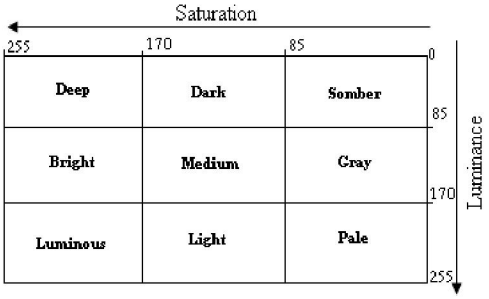

Fig. 4 Color representation on S and L dimensions

A two dimensional membership function is then placed on each region. In order to generate two dimensional membership functions, three 1D trapezoidal membership functions is placed over each dimension and then by multiplying these functions a set of nine 2D membership functions is generated. Following figure illustrates above concept.

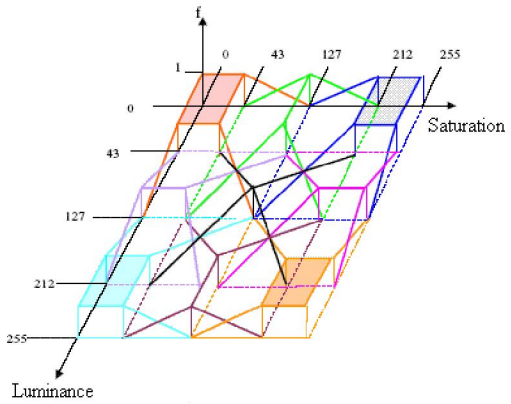

Fig. 5 Fuzzy memberships on S and L

Each fuzzy rule is represented as follows:

$$\begin{aligned} &j-th\ rule: \\ &if\ x_1\ is\ A_{j1}\ and\ x_2\ is\ A_{j2}\ and\ ...\ and\ x_m\ is\ A_{jm} \\ &then\ \underline{x}=(x_1,x_2,....,x_m)\ belongs\ to\ class\ H_j \\ &with\ CF=CF_j \qquad j=1,2,..,R \end{aligned} \tag{1}$$

in which R is the number of fuzzy rules, m is the dimensionality of input vector, $H_j \in \{1,2,...,M\}$ is output of the jth rule, M is the number of color classes, $CF_j \in [0, 1]$ is the certainty factor of jth rule. We tried three types of membership functions in our experiments. Equation (2), gives an example of how we code membership functions on a Gaussian membership function.

$$\begin{aligned} &\mu_{A_{ji}}(m_{(ji,1)},m_{(ji,2)},m_{(ji,3)};x_i) \\ &= \begin{cases} \exp\left(-\left(\dfrac{x_i-m_{(ji,1)}}{m_{(ji,2)}}\right)^2\right), if\ x_i \le m_{(ji,1)} \\ \exp\left(-\left(\dfrac{x_i-m_{(ji,1)}}{m_{(ji,3)}}\right)^2\right), if\ x_i > m_{(ji,1)} \end{cases} \end{aligned} \tag{2}$$

in above equation, $\underline{m}_{ji}=[m_{(ji,1)},m_{(ji,2)},...,m_{(ji,p)}]$ are the parameters of the ith membership function of jth rule. P is the number of parameters. $\underline{r}_j=[\underline{m}_{j1},\underline{m}_{j2},...,\underline{m}_{jM}]$ is the jth rule and $r=[\underline{r}_1,\underline{r}_2,...,\underline{r}_R]$ represents the whole fuzzy rule base. $\underline{a}=[H_1,CF_1,H_2,CF_2,....H_R,CF_R]$ is the output of the fuzzy rule base. When an input vector $x=(x_1,x_2,...,x_m)$ is presented to the system, output of the system is calculated as:

$$\begin{aligned} &q_j(x)=\prod_{i=1}^{M}\mu_{A_{ji}}(x_i) \\ &y=\arg\max_{j=1}^{R} q_j(x).CF_j \end{aligned} \tag{3}$$

Classification performance of a fuzzy system is very much dependent on its parameters. We are going to use a global search method (Comprehensive learning particle swarm optimization) to find a rule base with both minimum number of rules and minimum error rate [15].

III. COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a recently proposed algorithm by James Kennedy and R. C. Eberhart in 1995, motivated by social behaviour of organisms such as bird flocking and fish schooling [17], [18]. PSO as an optimization tool provides a population-based search procedure in which individuals called particles change their position (state) with time [19]. In PSO, individuals represent points in the n-dimensional search space [20]. A particle represents a potential solution. The velocity V_i^d and X_i^d position of the *d*th dimension of the *i*th particle are updated as follows:

$$V_i^d \leftarrow V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) \quad (4)$$

$$X_i^d \leftarrow X_i^d + V_i^d \quad (5)$$

where $X_i = (X_i^1, X_i^2, ..., X_i^D)$ is the position and $V_i = (V_i^1, V_i^2, ..., V_i^D)$ is the velocity of particle i [21], [22]. $pbest_i = (pbest_i^1, pbest_i^2, ..., pbest_i^D)$ is the best previous position yielding the best fitness value for the *i*th particle and $gbest = (gbest^1, gbest^2, ..., gbest^D)$ is the best position discovered by the whole population. c_1 and c_2 and are the acceleration constants reflecting the weighting of stochastic acceleration terms that pull each particle toward *pbest* and gbest positions respectively [23]. $rand1_i^d$ and $rand2_i^d$ are two random numbers in the range [0, 1]. A particle's velocity on each dimension is clamped to a maximum magnitude $V_{\max}$. If $|V_i^d|$ exceeds a positive constant value $V_{\max}^d$ specified by the user, then the velocity of that dimension is assigned to $sign(|V_i^d|)V_{\max}^d$.

Although there are numerous variants for the PSO, premature convergence when solving multimodal problems is still the main deficiency of the PSO. In the original PSO, each particle learns from its *pbest* and *gbest* simultaneously. Restricting the social learning aspect to only the *gbest* makes the original PSO converge fast. However, because all particles in the swarm learn from the *gbest* even if the current *gbest* is far from the global optimum, particles may easily be attracted to the *gbest* region and get trapped in a local optimum if the search environment is complex with numerous local solutions. As $f(x) = f([x^1, x^2, ..., x^D])$, the fitness value of a particle is possibly determined by values of all D parameters, and a particle that has discovered the region corresponding to the global optimum in some dimensions may have a low fitness value because of the poor solutions in the other dimensions.

CLPSO is a novel learning strategy that improves the original PSO, all particles' *pbest* are used to update the velocity of any one particle. CLPSO ensures that the diversity of the swarm is preserved to discourage premature convergence. In this new learning strategy, the following velocity updating equation is used:

$$V_i^d \leftarrow \omega * V_i^d + c * rand_i^d * (pbest_{f_i(d)}^d - X_i^d) \quad (6)$$

Where $f_i = [f_i(1), f_i(2), ..., f_i(D)]$ defines which particles' *pbests* the particle i should follow. $pbest_{f_i(d)}^d$ can be the corresponding dimension of any particle's including its own *pbest*, and the decision depends on probability Pc, referred to learning probability, which can take different values for different particles. For each dimension of particle i, we generate a random number. If this random number is larger than Pc_i, the corresponding dimension will learn from its own *pbest*; otherwise it will learn from another particle's *pbest*. We employ the tournament selection procedure when the particle's dimension learns from another particle's *pbest* as follows:

1) Randomly choose two particles out of the population which excludes the particle whose velocity is updated.

2) Compare the fitness values of these two particles' *pbest*s and select the better one.

3) Use the winner's *pbest* as the exemplar to learn from for that dimension. If all exemplars of a particle are its own *pbest*, we will randomly choose one dimension to learn from another particle's *pbest*'s corresponding dimension.

All these $pbest_{f_i}$ can generate new positions in the search space using the information derived from different particles' historical best positions [24].

IV. CLPSO BASED FUZZY CLASSIFICATION

Population P with L particles is represented as a vector:

$$P = \begin{bmatrix} \underline{p}_1 \\ \underline{p}_2 \\ . \\ \underline{p}_h \\ . \\ \underline{p}_L \end{bmatrix} = \begin{bmatrix} \underline{r}^1 & \underline{g}_1 \\ \underline{r}_2 & \underline{g}_2 \\ . & . \\ \underline{r}_h & g_h \\ . & . \\ \underline{r}_L & g_L \end{bmatrix} \quad (7)$$

$\underline{p}_h = [\underline{r}_h \ \underline{g}_h]$ is a member of population, and exemplify a fuzzy rule base[2]. The parameter vector $\underline{r}_h = [\underline{r}_1^h \ \underline{r}_2^h \underline{r}_3^h ... \underline{r}_j^h ... \underline{r}_B^h]$ consists of the premise parameters of the candidate fuzzy rules, where B is a user-defined positive integer to decide the maximum number of fuzzy rules. $\underline{r}_j^h = [\underline{m}_{j1}^h \underline{m}_{j2}^h ... \underline{m}_{ji}^h ... \underline{m}_M^h]$ is the membership functions of the jth rule in which M codes for the number of inputs of the system. $\underline{m}^h{}_{ji} = [\underline{m}_{(ji,1)}^h, \underline{m}_{(ji,2)}^h, ..., \underline{m}_{(ji,p)}^h]$ is the *i*th membership function of the *j*th rule in which p is the number of parameters of each membership function.

In order to be able to reduce the number of rules we used vector $g_h = [g_1^h \ g_2^h \ ... g_j^h \ ... g_B^h]$. In fact for each rule $\underline{r}_j^h$ there exists an element g_j^h. If $g_j^h \geq 0.5$ then the corresponding rule is considered eligible for adding to the rule base. Let r_h be the number of acceptable rules, then index of these rules and their

representation are: $I_r^h \in \{1,2,...,B\},\ r=1,2,...,r_h$, $\{\underline{r}_{I_1^h}^h, \underline{r}_{I_2^h}^h, ..., \underline{r}_{I_r^h}^h, ... \underline{r}_{I_{rh}^h}^h\}$ respectively.

Consequently, the rule base of the generated fuzzy classification system is described as follows:

r − th rule:

$$\begin{aligned} &if\ x_1 is A_{I_r^h 1}^h\ and\ x_2\ is A_{I_r^h 2}^h\ and ... and\ x_m\ is A_{I_r^h m}^h \\ &then \underline{x} = (x_1, x_2, ..., x_m)\ belongs\ to\ class\ H_r \\ &with\ CF = CF_r \quad r = 1,2,..,r_k \end{aligned} \tag{8}$$

where $A_{I_r^h i}^h,\ i=1,2,...,m$ are the fuzzy sets of the *r*th generated fuzzy rule.

To completely define each rule, certainty factor CF and output H must be determined [2]. To that purpose, we use a set of training data. There are N training data patterns, each belonging to one of M colors. Each training pattern is represented by a vector $(\underline{x}_n, y_n)$, $n=1,2,...,N$, in which $\underline{x}_n = (x_{n1}, x_{n2}, ..., x_{nm})$ and $y_n \in \{1,2,...,M\}$ are input and output of *n*th sample respectively.

For the *r*th rule, CF and H are determined as follows:

$$1)\quad \theta_t = \sum_{\underline{x}_p \in Class\ t} q_r(\underline{x}_p), t = 1,2,....,M \tag{9}$$

$$2)\quad H_r = \arg\max_{t=1}^{M} \theta_t \tag{10}$$

3) Determine the grade of certainty CF_r of the r*th* fuzzy rule by:

$$CF_r = \frac{\theta_{H_r} - \theta}{\sum_{t=1}^{M} \theta_t} \tag{11}$$

$$where,\ \theta = \sum_{\substack{t=1 \\ t \neq H_r}}^{M} \frac{\theta_t}{M-1} \tag{12}$$

Fitness of each particle should satisfy two criteria when selecting rule bases. A rule base is better if it has the less number of rules and minimum error rate. An example of such a fitness function is:

$$f_h = fit(\underline{p}_h) = g_1(\underline{p}_h) g_2(\underline{p}_h) \tag{13}$$

in above equation, functions g_1 and g_2 serve for the first and second criteria correspondingly.

$$g_1(\underline{p}_h) = \exp\left(\frac{-NICP(p_h)}{\sigma_e}\right) \tag{14}$$

$$g_2(\underline{p}_h) = \exp\left(\frac{-r_h}{\sigma_r}\right) \tag{15}$$

$NICP(p_h)$ is the number of false classified patterns. Gradually, in the course of running algorithm, particles try to find rule bases which maximize the fitness function. To put all things together, we describe the method in an algorithmic step-wise manner as follows:

Step 1) Initialize the CLPSO-based method. General parameters of the algorithm are listed in Table. I.

TABLE I
METHOD PARAMETERS

Parameter name	Symbol	Initialization value
Population size	L	Variable
Max number of rules	B	90
Max number of iterations	K	400
Constants of fitness function	σ_e, σ_r	5,10
Constants of CLPSO	(ψ, d_1, d_2, c)	(1,1,1,0.7)

Each member of the population $p_h = [r_h\ g_h]$ is randomly initialized as follows:

$$\begin{aligned} &r_h = [m_{11}^h\ m_{12}^h\ ...m_{1M}^h\ ...m_{j1}^h\ m_{j2}^h\ ...m_{jM}^h ...m_{B1}^h \\ &m_{B2}^h\ ...m_{BM}^h\ ]\ and\ m_{ji}^h = [m_{(ji,1)}^h, m_{(ji,2)}^h, ...m_{(ji,p)}^h] \\ &and\ g_h = [g_1^h\ g_2^h\ ...g_j^h\ ...g_B^h].\ m_{(ji,k)}^h,\ j \in \{1,2,...,B\}, \\ &i \in \{1,2,...,M\},\ k \in \{1,2,3\} \end{aligned} \tag{16}$$

is randomly generated as follows:

$$m_{(ji,k)}^h = m_{(ji,k)}^{\max} + (m_{(ji,k)}^{\max} - m_{(ji,k)}^{\min}).rand() \tag{17}$$

where the range of the parameter $m_{(ji,k)}^h$ is defined as $[m_{(ji,k)}^{\min}\ m_{(ji,k)}^{\max}]$ which is determined from the minimum and maximum of membership functions in each dimension.

$$g_j^h = rand(). \tag{18}$$

(rand() is a uniformly distributed random number in [0,1]).

Velocity vector $\underline{v}_h, h=1,2,..,L$ of a particle is initialized randomly as: $\underline{v}_h = [\underline{\alpha}_h, \underline{\beta}_h]$

$$\begin{aligned} \alpha_h = [&\alpha_{11}^h\ \alpha_{12}^h\ ...\alpha_{1M}^h\ ...\alpha_{j1}^h\ \alpha_{j2}^h\ ...\alpha_{jM}^h \\ &...\alpha_{B1}^h\ \alpha_{B2}^h\ ...\alpha_{BM}^h\ ] \end{aligned} \tag{19}$$

$$\alpha_{ji}^h = [\alpha_{(ji,1)}^h, \alpha_{(ji,2)}^h, ...\alpha_{(ji,p)}^h]$$

$$\beta_h = [\beta_1^h\ \beta_2^h\ ...\beta_j^h\ ...\beta_B^h] \tag{20}$$

in above formulas α_{ji}^h , β_j^h are calculated as follows:

$$\alpha_{(ji,k)}^h = \frac{m_{(ji,k)}^{\max} - m_{(ji,k)}^{\min}}{20}.rand() \tag{21}$$

$$\beta_j^h = \frac{rand()}{20} \tag{22}$$

Step 2) Select particle's neighbors. For each particle generate $f_i = [f_i(1), f_i(2), ...f_i(D)]$ as follows: Select two

particles $f1_i^d$ and $f2_i^d$ randomly, If Fitness[pbest($f1_i^d$)] > Fitness[pbest($f2_i^d$)] then $f_i(d) = f1_i^d$ else $f_i(d) = f2_i^d$

Step 3) Update $g_h = [g_1^h \ g_2^h \ ...g_j^h \ ...g_B^h]$:

$$\begin{aligned} &if\ \psi \geq rand() \quad then \\ &g_{j^*}^h = 1 - g_{j^*}^h,\ j^* = rand(B\ rand() + 0.5) \end{aligned} \tag{23}$$

Step 4) Update *Pbests* and *gbest*. If Fitness(X_i) > Fitness($pbest_i$) then $pbest_i = X_i$. If Fitness(X_i) > Fitness($gbest_i$) then $gbest_i = X_i$

Step 5) Update velocity and position of each particle according to (6) and (5).

Step 6) Decay Velocities:

$$\begin{aligned} &v_h = v_h.d_1, h = 1,2,...,L, \ \ d_1 \in [0,1] \\ &\psi = \psi * d_2.\ \ d_2 \in [0,1] \end{aligned} \tag{24}$$

Step 7) Check Stopping Criteria:

$$gen = gen + 1 \tag{25}$$

if gen > K or the average velocity of particles surpasses a small threshold near zero then go to next step, else go to step 3.

Step 8) Stop: Report particle with the best fitness value as best rule base.

For a detailed explanation of CLPSO algorithm, reader is referred to [25].

V. Experiments and Results

In this section, details of our implementation and experimental results are presented. To evaluate the efficiency of this method we conducted experiments on segmentation of color images taken from a Robocop football field which is a benchmark problem in robotic contests.

A. Training Data

Color images taken from a small size Robocop soccer field are used as training data. Because we were to build a segmentation system invariant to illumination, we tried to choose images taken at different lightening situations. Each training point belongs to one of ten color classes (Red1, Orange, Yellow, Green, Cyan, Blue, Purple, Magenta, Pink, Red2). (Red1 and Red2 correspond to first and last membership function in Hue dimension). A total number of 9,000 training data and 3,000 test data were used. From 1,200 data points available for each class, 900 was used for training and remained 300 for testing.

B. Fuzzy system Structure

We used a Sugeno-type Fuzzy inference system with three inputs consisting dimensions of HSL color space. Ten membership functions for H dimension and three for each S and L dimensions was used. Ten constant membership functions were put on output variable. Three types of membership functions trapezoidal, Gaussian and Bell-shaped were investigated for the purpose of their comparison over this problem. Maximum and minimum of membership functions are determined according to [2]. Our results rank membership functions as Bell-shaped, Gaussian and trapezoidal respectively according to their performance.

C. Algorithm Parameters

We manually set values for parameters based on previous studies and problem characteristics [24]. Those values are listed in Table I.

D. Initial Population Generation

Proper initialization of particles plays an important role in algorithm efficiency. We compared two initialization methods, randomly generating rules and rule generation using neurofuzzy method. In order to generate fuzzy rules, we used ANFIS toolbox provided with MATLAB® programming environment.

TABLE II
ANFIS PERFORMANCE

Trapezoidal membership function		Gaussian membership function		Bell-shaped membership function	
Number of rules	Accuracy	Number of rules	Accuracy	Number of rules	Accuracy
90	60%	90	81.49%	90	77.63%

TABLE III
CLPSO-BASED FUZZY PERFORMANCE

Population Initialization method	Trapezoidal membership function		Gaussian membership function		Bell-shaped membership function	
	Number of rules	Accuracy	Number of rules	Accuracy	Number of rules	Accuracy
Random	40	60.32%	42	51.84%	38	53.63%
ANFIS	47	60.82%	39	73.34%	36	89%

As it is shown in Table. III, a higher performance is achievable with ANFIS initialization compared with random method. The reason might be that we constrain the maximum number of iterations which is needed for algorithm to be computationally tractable. The total number of rules generated with this method is about one third of the average number of rules generated with ANFIS.

An example of running our method to segment a soccer field image is shown in Fig. 6. Top row shows the performance of our method with ANFIS initialization and Bell-shaped membership functions. Bottom row illustrates the performance of ANFIS method. Images at the left are originals and those at the right are segmented ones.

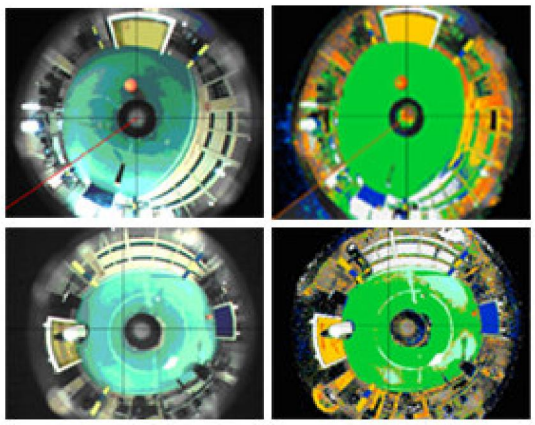

Fig. 6 Comparison of CLPSO and PSO on image segmentation. Top is CLPSO, Bottom is ANFIS.

VI. Conclusions

A new method for segmentation of color images using a CLPSO-based fuzzy system was introduced in this paper. A particle of the swarm codes for a set of fuzzy rules. A fitness function rates for the Optimality of each particle. During iterations, particles try to maximize fitness function by cooperatively working on search space. This process is continued until either maximum number of iteration is met or average velocity approaches zero. Finally, the rule base represented by the best particle is used for the task of image segmentation. Because of smaller number of rules generated by this method, it shows higher computation speed. Using CLPSO for searching in the space of all possible solutions we achieved a better solution in terms of small size and high performance compared with ANFIS. Because we used a large training set including images taken at different illuminations, this method is robust to variations in illumination.

References

[1] A.A. Younes, I. Truck, and H. Akdaj, "Color Image Profiling Using Fuzzy Sets," *Turk J Elec Engin,* vol.13, no.3, 2005.

[2] G.A. Ruz, P.A. Estévez, and C.A. Perez, "A Neurofuzzy Color Image Segmentation Method for Wood Surface Defect Detection," *Forest Prod. J.* 55 (4), 52-58, 2005.

[3] A. Moghaddamzadeh and N. Bourbakis, "A Fuzzy Region Growing Approach for Segmentation of Color Images," *Pattern Recognition*, 30(6):867-881, 1997.

[4] G.S. Robinson, "Color Edge Detection," *Opt. Eng*, 16(5): 479-484, 1977.

[5] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell*, 8(6): 679-698, 1986.

[6] A. Shiji and N. Hamada, "Color Image Segmentation Method Using Watershed Algorithm and Contour Information," *Proc. Inter. Conf. on Image Processing*, 4:305-309, 1999.

[7] B. Zhang, "Generalized K-harmonic Means–boosting in Unsupervised Learning," Technical report (HPL-2000–137), Hewlett-Packard Labs, 2000.

[8] J. Bezdek, "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," *IEEE Trans Pattern Anal Mach Intell,* 2:1-8, 1980.

[9] G. Ball and D. Hall, "A Clustering Technique for Summarizing Multivariate Data" *Behav Sci,* 12:153–155, 1967.

[10] K. Huang, "A Synergistic Automatic Clustering Technique (Syneract) for Multispectral Image Analysis," *Photogrammetric Eng Remote Sens,* 1(1):33–40, 2002.

[11] M.G.H. Omran, A. Salman, and A.P. Engelbrecht, "Dynamic Clustering Using Particle Swarm Optimization with Application in Image Segmentation," *Pattern Anal Applic,* 8: 332–344, 2005.

[12] J.S. Jang, C.T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New Jersey, U.S.A., 1997.

[13] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive Fuzzy Rule-Based Classification Systems," *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 3, pp. 238-50, Aug 1996.

[14] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms," *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 260-270,1995.

[15] K.S. Deshmukh, G.N. Shinde, "An Adaptive Color Image Segmentation," *Electronic Letters on Computer Vision and Image Analysis* 5(4):12-23, 2005.

[16] J.S. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference systems," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 23, pp. 665-685, 1993.

[17] C.S. Wallace, "An Improved Program for Classification," Technical report, no. 47, Department of Computer Science, Monash University, Australia, 1984.

[18] R.H. Turi, "Clustering-based Colour Image Segmentation," PhD Thesis, Monash University, Australia, 2001.

[19] C.S. Wallace and D.M. Boulton, "An Information Measure for Classification," *Comput J,* 11:185-194, 1968.

[20] Y. SM and R. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization," P*roc. Congress on Evolutionary Computation,* Seoul, S. Korea, 2001.

[21] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New-York, 1973.

[22] M. Omran, A. Engelbrecht, and A. Salman, "Particle Swarm Optimization Method for Image Clustering," *Int J Pattern Recogn Artif Intell,* 19(3):297-322, 2005.

[23] R Eberhart and Y. Shi, "Comparision between Genetic Algorithms and Particle Swarm Opimization," *Proc. of the Seventh Annual Conference on Evolutionary Programming*, pp. 611-619. Springer-Verlag, 1998.

[24] C.C. Chen, "Design of PSO-based Fuzzy Classification Systems," *Tamkang Journal of Science and Engineering*, vol. 9, no. 1, pp. 63-70, 2006.

[25] J.J. Liang, A.K. Qin, and S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of multimodal Functions," *IEEE trans. Evolutionary Computation*, vol. 10, no. 3, June, 2006.