A new social and momentum component adaptive PSO algorithm for image segmentation

Akhilesh Chandler a, b, Amitava Chatterjee c, Patrick Siarry a,∗

a Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, EA 3956), Université Paris XII Val de Marne, 61 avenue du Général de Gaulle, 94010 Créteil, France
b Department of Electronics and Computers, Indian Institute of Technology, Roorkee 247667, India
c Jadavpur University, Electrical Engineering Department, Kolkata 700 032, India

Article info
Keywords:
Image segmentation
Multilevel thresholding
PSO
'Social' and 'momentum' component
Iterative scheme

Abstract
In this paper, we present a new variant of Particle Swarm Optimization (PSO) for image segmentation using optimal multi-level thresholding. Some objective functions which are very efficient for bi-level thresholding purpose are not suitable for multi-level thresholding due to the exponential growth of computational complexity. The present paper also proposes an iterative scheme that is practically more suitable for obtaining initial values of candidate multilevel thresholds. This self iterative scheme is proposed to find the suitable number of thresholds that should be used to segment an image. This iterative scheme is based on the well known Otsu’s method, which shows a linear growth of computational complexity. The thresholds resulting from the iterative scheme are taken as initial thresholds and the particles are created randomly around these thresholds, for the proposed PSO variant. The proposed PSO algorithm makes a new contribution in adapting ‘social’ and ‘momentum’ components of the velocity equation for particle move updates. The proposed segmentation method is employed for four benchmark images and the performances obtained outperform results obtained with well known methods, like Gaussian-smoothing method (Lim, Y. K., & Lee, S. U. (1990). On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. Pattern Recognition, 23, 935–952; Tsai, D. M. (1995). A fast thresholding selection procedure for multimodal and unimodal histograms. Pattern Recognition Letters, 16, 653–666), Symmetry-duality method (Yin, P. Y., & Chen, L. H. (1993). New method for multilevel thresholding using the symmetry and duality of the histogram. Journal of Electronics and Imaging, 2, 337–344), GA-based algorithm (Yin, P. -Y. (1999). A fast scheme for optimal thresholding using genetic algorithms. Signal Processing, 72, 85–95) and the basic PSO variant employing linearly decreasing inertia weight factor.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Image segmentation is considered as an important basic operation for meaningful analysis and interpretation of image acquired. It is useful in separating objects from background, or discriminating objects from objects that have distinct gray-levels. Image thresholding is widely used for image segmentation. It can be classified as bi-level thresholding and multilevel thresholding. Bi-level thresholding classifies the pixels into two groups, one group includes those pixels with gray levels above a certain threshold and the other group includes the rest. Multilevel thresholding divides the pixels into several groups and the pixels belonging to a group, having intensity values within a specific range, are assigned a single intensity value. Image gray-level histograms are considered as efficient tools for development of image segmentation algorithms. The objective is to find the gray level values, called thresholds, on the basis of which the classifications are made. The simplest is bi-level thresholding problem in which only one gray value is to be found. However the problem gets more and more complex when we try to achieve segmentation with greater detail by employing multilevel thresholding, in which more than one threshold are to be found. Usually it is not simple to determine exact locations of distinct valleys in a multimodal histogram of an image, that can segment the image efficiently and hence the problem of multilevel thresholding is regarded as an important area of research interest among the research communities worldwide.

Thresholding techniques can be classified into two types: optimal thresholding methods (Kapur, Sahoo, & Wong, 1985; Kittler & Illingworth, 1986; Otsu, 1979; Pun, 1980, 1981) and property-based thresholding methods (Lim & Lee, 1990; Tsai, 1995; Yin & Chen, 1993). Optimal thresholding methods search for the optimal thresholds which make the thresholded classes on the histogram
reach the desired characteristics. Usually, it is made by optimizing an objective function. Property-based thresholding methods detect the thresholds by measuring some property of the histogram. Property-based thresholding methods are fast and suitable for the case of multilevel thresholding, while the number of thresholds is hard to determine and should be specified in advance.

Several algorithms have so far been proposed in literatures that have addressed the issue of optimal thresholding (Brink, 1995; Cheng, Chen, & Li, 1998; Hu, Hou, & Nowinski, 2006; Huang & Wang, 1995; Kapur et al., 1985; Li, Zhao, & Cheng, 1995; Otsu, 1979; Pun, 1980; Saha & Udupa, 2001; Tobias & Seara, 2002; Yin, 1999). While many of them address the issue of bi-level thresholding, others have considered the multilevel problem. Many such schemes attempt to achieve optimal thresholding such that the thresholded classes achieve some desired characteristic. Now, several such methods of designing the desired characteristic to judge the quality of segmentation algorithms have been proposed. Methods based on optimizing an objective function include maximization of posterior entropy to measure homogeneity of segmented classes (Brink, 1995; Cheng et al., 1998; Kapur et al., 1985; Pun, 1981; Saha & Udupa, 2001), maximization of the measure of separability on the basis of between-class variance (Otsu, 1979), thresholding based on index of fuzziness and fuzzy similarity measure (Huang & Wang, 1995; Li et al., 1995), minimization of Bayesian error (Kittler & Illingworth, 1986), etc. Several such methods have originally been developed for bi-level thresholding and later extended to multilevel thresholding, e.g. methods described in Kapur et al. (1985) and Otsu (1979). All of these methods have a common problem, that the computational complexity rises exponentially when extended to multilevel thresholding due to the exhaustive search employed. In this paper we use bi-level Otsu thresholding method for multi-level thresholding with less computational complexity. An iterative procedure is proposed in this work which efficiently determines the number of thresholds and the positions of these thresholds, which are used as initial values for a proposed variant of PSO based stochastic algorithm, utilized for gray image segmentation purpose.

Particle Swarm Optimization (PSO) is an evolutionary computation technique introduced by Eberhart and Kennedy (1995), Eberhart, Simpson, and Dobbins (1996) and Kennedy and Eberhart (1995). The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm (Millonas, 1994). Since the introduction of the particle swarm optimizer by James Kennedy and Russ Eberhart in 1995, numerous variations of the basic algorithm have been developed in the literature. In this paper, a new variant of PSO is presented which shows better results than the previous variants on all the problems that we have experimented.

The rest of the paper is organized as follows. Section 2 presents the iterative scheme employed to find the multiple thresholds and the suitable number of thresholds to threshold an image. Section 3 presents the proposed new variant of PSO algorithm for multilevel thresholding. Section 4 presents the entropy criterion based measure employed for optimal thresholding. Performance evaluation is presented in detail in Section 5. Conclusions are presented in Section 6.

2. The proposed iterative scheme

In bi-level thresholding, both Kapur and Otsu methods can find the optimal threshold efficiently. Kapur and Otsu also extended their methods to multilevel thresholding. However, both these methods have the disadvantage that the computational time is very expensive due to the exhaustive search. Let us assume that we wish to search $c$ optimal thresholds $[t_1, t_2, ..., t_c]$. Both Kapur and Otsu methods must exhaustively search for all possible values of $[t_1, t_2, ..., t_c]$. This will result in a complexity of $O(c^c)$, which grows exponentially with the number of thresholds. Thus, in practical applications, these methods are not suitable for multilevel thresholding. In this section, we propose an iterative procedure to find multiple thresholds using Otsu’s method for bi-level thresholding. A self-iterative procedure is also mentioned to find number of thresholds used for segmentation, where a maximum number of thresholds allowed is given a priori.

As will be seen in the experiments, the proposed algorithm can exhibit significant saving of the computational time, in case of multilevel thresholding.

The proposed iterative scheme uses the uniformity measure as the criterion on the basis of which the suitable number of thresholds and the values of these thresholds are determined. The quality of the thresholded images is also evaluated by the uniformity measure, a popular measure extensively used in the literature, in the past. This is given by:

$$U = 1 - 2c \sum_{i=0}^{c} \frac{\sum_{j \in R_i} (f_i - \mu_j)^2}{N_{\text{max}} - N_{\text{min}}}$$

where

- $c$: number of thresholds
- $R_i$: $i$th segmented region
- $N$: total number of pixels in the given image
- $f_i$: gray level of pixel $i$
- $\mu_j$: mean gray level of pixels in $j$th region
- $f_{\text{max}}$: maximum gray level of pixels in the given image
- $f_{\text{min}}$: minimum gray level of pixels in the given image.

The proposed algorithm starts with the bi-level thresholding. The higher level thresholds can be obtained by applying the bi-level Otsu method each time in a suitable thresholded class to maximize the value of objective function (i.e. the uniformity measure), as much as possible. The algorithm is detailed below, step-by-step:

Iterative scheme:

Step 1: Apply bi-level thresholding Otsu’s method to the whole histogram.
Step 2: Set $n = 1$
Step 3: Apply Otsu’s method in every thresholded class $[t_i, t_i+1]$ separately and compute the value of objective function in each case.
Step 4: Determine the class, with new threshold, which produced highest value of the objective function, in Step 2.
Step 5: Store this new threshold value and move to the next iteration.
Step 6: Set $n = n + 1$
Step 7: If $n$ is less than number of thresholds to be created, go to Step 3. Else stop.

The self-iterative scheme to find the number of thresholds to segment an image is same as the above procedure until Step 6. The rest of the steps are given below:

Step 7: Record the value of uniformity.
Step 8: If $n$ is less than the maximum number of thresholds, go to Step 3. Otherwise, go to Step 9.
Step 9: Output that level of thresholds for which highest uniformity measure was obtained.

Let us assume that we wish to search for $c$ optimal thresholds $[t_1, t_2, ..., t_c]$. If we search for all possible values of $[t_1, t_2, ..., t_c]$
exhaustively using Otsu’s method, it will result in a complexity of the order of $O(f^c)$, which grows exponentially with the number of thresholds. In our proposed scheme, for every new threshold to be created, all the values in $[0, L-1]$ are checked, which takes some order of time complexity, as in bi-level thresholding. It means that, according to our proposed algorithm, for determining $c$ thresholds, we need to execute the bi-level thresholding procedure for $c$ times. Hence, in this case time complexity is $O(t c)$, which grows linearly with number of thresholds. In Fig. 1 we present the results of CPU times consumed for multilevel thresholds, with $c = 1, 2, 3, 4, 5$, for Lena and Pepper images and compare them with Gaussian-smoothing method, Symmetry-duality method and GA learning-Otsu method (Yin, 1999). The corresponding threshold values and mean uniformity measures are also accordingly shown. For our method, the CPU-time results are better in every case, when compared with Gaussian-smoothing and Symmetry-duality methods and are better than GA learning-Otsu for thresholds greater than three. The uniformity measure results of the proposed method are much higher than those of the other methods.

3. Proposed variant of PSO

Particle Swarm Optimization (PSO) is an evolutionary computation technique introduced by Eberhart and Kennedy (1995), Eberhart et al. (1996) and Kennedy and Eberhart (1995). In PSO, each individual of the population, called a particle, has an adaptable velocity, according to which it moves over the search space. Each particle keeps track of its coordinates in hyperspace, which are associated with the solution (fitness) it has achieved so far. This value is called $p_{best}$. Fitness of a particle is the value of objective function at present coordinates of the particle. Particles hopefully move to the optimal solution of this objective function. The best performance among all particles obtained so far is stored as $g_{best}$. Suppose that the search space is $D$-dimensional, then the $i$th particle of the swarm can be represented by a $D$-dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots , x_{iD})$. The velocity of this particle can be represented by another $D$-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots , v_{iD})$. The best previously visited position of the $i$th particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots , p_{iD})$. Defining $g$ as the index of the best particle in the swarm, the velocity of $i$th particle, in $d$th dimension, i.e. $v_{id}$ is determined by the additive influence of $v_{gd}$ in the previous iteration (called the ‘momentum’ component) and individual stochastic weighting of its distance from $p_{id}$ (called the ‘cognitive’ component) and its distance from $g_{id}$ (called the ‘social’ component). $c_1$ and $c_2$ are the acceleration coefficients which determine the extent of stochastic weighting employed for the cognitive and the social components, individually. Therefore:

$$v_{id} = v_{id} + c_1 \times \text{rand}(\cdot) \times (p_{id} - x_{id}) + c_2 \times \text{rand}(\cdot) \times (g_{id} - x_{id}) \tag{2}$$

$$x_{id} = x_{id} + v_{id} \tag{3}$$

where $c_1$ and $c_2$ are two positive constants, called acceleration coefficients, and $\text{rand}(\cdot)$ generates pseudo-random numbers, uniformly distributed in $[0, 1]$. To obtain a judicious mixture of global exploration (during initial stages of the optimization process) and local exploitation (during later stages of the optimization process) capabilities, the “momentum” component contains the linearly varying inertia weight ($w$). This $w$ starts with a high value and decreases linearly, as described in Eberhart and Shi (1997).

To improve the performance of PSO, we modified this variant of PSO given by Eberhart and Shi and proposed a new variant. The salient feature of the proposed modifications is that if a particle moves to a position with better fitness value in an iteration, it implies that the particle is moving towards the optimal solution. Then in the next iteration, we suggest that more weight be given to the ‘momentum’ part and less weight to the ‘social’ part. Hence the acceleration coefficient, $c_2$, is no more kept constant and is varied according to the situation. If the particle moves to a position with worse fitness in an iteration, then in the next iteration more weight is let to the ‘social’ part and less weight to the ‘momentum’ part, assuming that the particle moved in a direction which does not lead to the optimal solution. If $w$ is a linearly decreasing variable, the adapted new coefficients of the ‘momentum’ part and the ‘social’ parts are taken as $w$ and $2w$ respectively. If there is no improvement from the fitness value in previous iteration or $2w$ and $w$ respectively if there is an improvement in the fitness value from the previous iteration. According to the proposed algorithm, gradually the acceleration coefficient $c_2$ is decreasing with iterations and hence local exploration (cognitive component) dominates during later stages of optimization process, which strengthens the original idea of Eberhart and Shi. To make our ideas fruitful, a random factor is also included in momentum part. The proposed algorithm is given as follows: If $i$th particle has improved its fitness from previous iteration:

<table>
<thead>
<tr>
<th>Images</th>
<th>$c$</th>
<th>Thresholds (Proposed method)</th>
<th>Proposed method</th>
<th>CPU time (s)</th>
<th>Gaussian-smoothing method</th>
<th>Symmetry-duality method</th>
<th>GA-based method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>1</td>
<td>116</td>
<td>0.0313</td>
<td>0.5775</td>
<td>0.286</td>
<td>0.0132</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>116,166</td>
<td>0.0771</td>
<td>0.6050</td>
<td>0.284</td>
<td>0.0352</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>116,166,190</td>
<td>0.1563</td>
<td>0.5556</td>
<td>0.284</td>
<td>0.0132</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>116,140,166,190</td>
<td>0.2544</td>
<td>0.1616</td>
<td>0.284</td>
<td>0.2169</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>75,116,140,166,190</td>
<td>0.3969</td>
<td>0.6055</td>
<td>0.284</td>
<td>0.7149</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>1</td>
<td>119</td>
<td>0.0313</td>
<td>0.6235</td>
<td>0.297</td>
<td>0.0099</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>119,165</td>
<td>0.0938</td>
<td>0.7700</td>
<td>0.297</td>
<td>0.0731</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>62,119,165</td>
<td>0.1406</td>
<td>0.6882</td>
<td>0.297</td>
<td>0.0144</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>62,119,165,187</td>
<td>0.2188</td>
<td>0.6642</td>
<td>0.297</td>
<td>0.2950</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>62,119,143,165,187</td>
<td>0.3500</td>
<td>0.6815</td>
<td>0.297</td>
<td>0.7020</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1a. Time complexities of the proposed iterative scheme, Gaussian-smoothing method, Symmetry-duality method and GA-based method.
$w_1 = 2 \times w; \quad w_2 = w$

Else

$w_1 = w; \quad w_2 = 2 \times w$

$$v_{id} = w_1 \times \text{rand}(i) \times v_{id} + c_1 \times \text{rand}(i) \times (p_{id} - x_{id}) + w_2 \times \text{rand}(i) \times (x_{id} - x_d)$$

(6)

$x_{id} = x_{id} + v_{id}$

(7)

where $w$ is a linearly decreasing inertia weight, originally proposed in Eberhart and Shi (1997).

The thresholds resulted from procedure mentioned in Section 2 are utilized as initial thresholds, and the particles for PSO are randomly distributed around these initial thresholds. The intention of using initial thresholds is to find the optimal solution more efficiently, i.e. using less number of particles and iterations. For image segmentation problems, various entropy criterion measures are used as objective functions. Otsu’s multi-threshold entropy measure is used as objective function here in PSO. The details of this entropy measure are described in Section 4.

4. Entropy criterion based fitness measure

Otsu introduced a method which maximizes the between class variance to separate the segmented classes as farther as possible, to find the optimal threshold for segmentation. This between class variance measure is called as Otsu’s multi-threshold entropy measure and is used as the objective function in the PSO. It is formulated as follows.

Let the gray levels of a given image range over $[0, L-1]$ and $h(i)$ denote the occurrence of gray-level $i$.

Let

$$N = \sum_{i=0}^{L-1} h(i); \quad P_i = h(i)/N; \quad \text{for } 0 \leq i \leq L - 1$$

Maximize

$$f(t) = w_0 w_1 (\mu_0 - \mu_1)^2$$

(8)

where

$$w_0 = \sum_{i=0}^{L-1} P_i; \quad \mu_0 = \sum_{i=0}^{L-1} i \times P_i / w_0$$

(9)

$$w_1 = \sum_{i=1}^{L-1} P_i; \quad \mu_1 = \sum_{i=1}^{L-1} i \times P_i / w_0$$

(10)

and the optimal threshold is the gray level that maximizes (8).

(8) can also be written as:

$$f(t) = \sigma_0^2 - w_0 \times \sigma_0^2 - w_1 \times \sigma_1^2$$

(11)

where $w_0, w_1, \mu_0$ and $\mu_1$ are the same as given in Eqs. (9) and (10), and

$$\sigma_0 = \sum_{i=0}^{L-1} (i - \mu_0)^2 \times P_i / w_0$$

$$\sigma_1 = \sum_{i=0}^{L-1} (i - \mu_1)^2 \times P_i / w_1$$

$$\sigma = \sum_{i=0}^{L-1} (i - \mu)^2 \times P_i / w$$

$$w = \sum_{i=0}^{L-1} P_i; \quad \mu = \sum_{i=0}^{L-1} i \times P_i / w$$

Hence, maximizing (11) is same as minimizing $f_1(t) = w_0 \times \sigma_0^2 + w_1 \times \sigma_1^2$.

Expanding this logic to multi-level thresholding,

$$f_1(t) = w_0 \times \sigma_0^2 + w_1 \times \sigma_1^2 + w_2 \times \sigma_2^2 + \ldots \ldots + w_c \times \sigma_c^2$$

where $c$ is the number of thresholds.

$$f_1(t) = \sum_{i=0}^{c} w_i \times \sigma_i^2$$

is used as the objective function for the proposed PSO based procedure which is to be optimized (minimized).

A close look into this equation will show it is very similar to the expression for uniformity measure.

5. Performance evaluation

The performance of the proposed scheme was compared with the results of Genetic Algorithm based multiple thresholding (Yin, 1999) (named GA-learning-Otsu algorithm in Yin (1999)), Gaussian-smoothing method (Lim & Lee, 1990; Tsai, 1995) and Symmetry-duality method (Yin & Chen, 1993). As these methods were demonstrated by considering two popular images, that of Lena and Pepper (each of 512 × 512 size), we also chose these two images to present comparative results. The results of two additional images, House and Elaine, were compared with the results of basic PSO employing linearly decreasing inertia weight.

The performance of PSO depends on many factors, like acceleration coefficients, inertia weight factor, maximum allowed velocity ($V_{\text{max}}$), maximum allowed position ($X_{\text{max}}$), number of particles, number of iterations. Here, $w$ was initially taken to be 0.9, which linearly decreases with iterations. $V_{\text{max}}$ is equal to $X_{\text{max}}$ which is the maximum gray value, i.e. 255. Similarly, $V_{\text{min}} + X_{\text{min}} = 0$. The acceleration coefficient $c_1$ was kept constant at 2.4. Fig. 2 shows the performance of PSO for varied number of particles from 10 to 30, for a fixed number of 100 iterations. The method was tested for $c = 3$. The graph in Fig. 3 shows that PSO is at its best with 25 particles for two images and near to best for the other two. Fig. 2 next shows the performance for varying number of iterations, with number of particles kept fixed at 25. This shows that PSO performs better with 100 iterations. Hence all the remaining simulations were carried out with 25
particles and 100 iterations. The results of the proposed method on images Lena and Pepper are presented in Fig. 4, which are compared with the results of GA-learning-Otsu algorithm, Gaussian-smoothing method and Symmetry-duality method. The method is also implemented for House and Elaine images and the results are compared with the basic PSO employing linear inertia weight. The values of inertia weights in basic PSO and proposed PSO are kept same for proper comparisons. The acceleration coefficients $c_1$ and $c_2$ are maintained at two in basic PSO. Each PSO based algorithm is repeated ten times each and the results reported are the average results, obtained over ten simulations.

Each time the self-iterative scheme was run to determine the number and values of initial thresholds and then the proposed

<table>
<thead>
<tr>
<th>Iterations</th>
<th>$c$</th>
<th>No. of particles</th>
<th>Lena image</th>
<th>House image</th>
<th>Elaine image</th>
<th>Pepper image</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>3</td>
<td>10</td>
<td>0.9679</td>
<td>0.9791</td>
<td>0.9710</td>
<td>0.9743</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>15</td>
<td>0.9701</td>
<td>0.9793</td>
<td>0.9727</td>
<td>0.9750</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>20</td>
<td>0.9704</td>
<td>0.9794</td>
<td>0.9733</td>
<td>0.9750</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>25</td>
<td>0.9721</td>
<td>0.9793</td>
<td>0.9744</td>
<td>0.9759</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>30</td>
<td>0.9705</td>
<td>0.9798</td>
<td>0.9744</td>
<td>0.9756</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of particles</th>
<th>$c$</th>
<th>Iterations</th>
<th>Lena image</th>
<th>House image</th>
<th>Elaine image</th>
<th>Pepper image</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>3</td>
<td>50</td>
<td>0.9712</td>
<td>0.9793</td>
<td>0.9766</td>
<td>0.9759</td>
</tr>
<tr>
<td>75</td>
<td>3</td>
<td>75</td>
<td>0.9713</td>
<td>0.9798</td>
<td>0.9741</td>
<td>0.9757</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>100</td>
<td>0.9732</td>
<td>0.9795</td>
<td>0.9752</td>
<td>0.9756</td>
</tr>
<tr>
<td>125</td>
<td>3</td>
<td>125</td>
<td>0.9729</td>
<td>0.9797</td>
<td>0.9749</td>
<td>0.9757</td>
</tr>
<tr>
<td>150</td>
<td>3</td>
<td>150</td>
<td>0.9714</td>
<td>0.9793</td>
<td>0.9738</td>
<td>0.9760</td>
</tr>
</tbody>
</table>

**Fig. 2a.** Uniformity values for $c = 3$ and varying iterations for four benchmark images.

**Fig. 2b.** Uniformity values for $c = 3$ and varying number of particles, for four benchmark images.

**Fig. 3.** Graphical representation of the results given in Fig. 2, for the proposed algorithm: (a) uniformity vs. number of particles and (b) uniformity vs. number of iterations.

**Fig. 4a.** Results of the proposed method for Lena and Pepper images, compared with those of Gaussian, Symmetry-duality and GA-based methods.
PSO was applied for four images. The results and images are shown in Figs. 5–7. For the self iteration scheme, the maximum number of thresholds was set as eight. For the proposed PSO, 25 particles and 100 iterations were taken.

### Table 1: Uniformity values and threshold values for proposed and basic PSO methods

<table>
<thead>
<tr>
<th>Images</th>
<th>c</th>
<th>Uniformity values</th>
<th>Threshold values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed method</td>
<td>Basic PSO</td>
</tr>
<tr>
<td>House</td>
<td>2</td>
<td>0.9765</td>
<td>0.9738</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.9796</td>
<td>0.9734</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.9786</td>
<td>0.9712</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.9753</td>
<td>0.9725</td>
</tr>
<tr>
<td>Elaine</td>
<td>2</td>
<td>0.9777</td>
<td>0.9750</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.9741</td>
<td>0.9679</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.9750</td>
<td>0.9737</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.9823</td>
<td>0.9740</td>
</tr>
</tbody>
</table>

### Fig. 4b. Results of the proposed method for House and Elaine images, compared with those of basic PSO method.

### Fig. 5. Results of the proposed PSO method employing self-iterative scheme to find number and values of initial thresholds.

### Fig. 6. Gray images and the corresponding thresholded images of Lena (c = 5 thresholds) and Pepper (c = 6 thresholds) using the proposed method.

### Fig. 7. Gray images and the corresponding thresholded images of Elaine (c = 6 thresholds) and House (c = 3 thresholds) using the proposed method.

### 6. Conclusion

In this paper, we have described an optimal multilevel thresholding algorithm employing a new proposed variant of PSO. The paper also proposes an iterative scheme to obtain initial thresholds. This can be useful for practical situations because here the computational complexity grows linearly with the number of thresholds. This is more suitable compared to those methods where there is an exponential growth of computational complexity. The present paper also proposes an iterative scheme that is practically more suitable for obtaining initial values of these candidate multilevel thresholds, using Otsu’s method. The proposed PSO algorithm makes a new contribution in adapting ‘social’ and ‘momentum’ components of the velocity equation for PSO updates. It has been shown in the previous sections, with the help of results obtained for four benchmark images, that the proposed method...
could produce better results than several well known methods, previously proposed.

References


