Optimal multi-level thresholding with membrane computing

Hong Peng [a,*], Jun Wang [b], Mario J. Pérez-Jiménez [c]

[a] Center for Radio Administration and Technology Development, Xihua University, Chengdu, Sichuan, 610039, China
[b] School of Electrical and Information Engineering, Xihua University, Chengdu, Sichuan, 610039, China
[c] Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, University of Seville, Sevilla, 41012, Spain

Keywords:

Membrane computing
Cell-like P systems
Image segmentation
Multi-level thresholding
Histogram

A B S T R A C T

The conventional methods are not effective and efficient for image multi-level thresholding due to time-consuming and expensive computation cost. The multi-level thresholding problem can be posed as an optimization problem, optimizing some thresholding criterion. In this paper, membrane computing is introduced to propose an efficient and robust multi-level thresholding method, where a cell-like P system with the nested structure of three layers is designed as its computing framework. Moreover, an improved velocity-position model is developed to evolve the objects in membranes based on the special membrane structure and communication mechanism of objects. Under the control of evolution-communication mechanism of objects, the cell-like P system can efficiently exploit the best multi-level thresholds for an image. Simulation experiments on nine standard images compare the proposed multi-level thresholding method with several state-of-the-art multi-level thresholding methods and demonstrate its superiority.

1. Introduction

Image segmentation is one of the most important tasks in computer vision and video applications. Thresholding has been widely used as a popular image segmentation technique [1]. The goal of thresholding is to separate objects from background image or discriminate objects from objects that have distinct gray levels. The existing thresholding methods can be roughly classified as two categories: bi-level thresholding and multi-level thresholding [2–4]. Bi-level thresholding segments an image into two different regions. The pixels with gray values greater than a certain threshold are classified into object, and those with gray values lower than the threshold are regarded as background. Thresholding problem can be posed as an optimization problem. Otsu's method [5] and Kapur's method [6] are simple and effective bi-level thresholding, which maximize the between-class variance of gray levels and the entropy of the histogram to optimize single threshold for an image respectively. Multi-level thresholding determines more than one threshold for an image and segments the image into several distinct regions, which correspond to one background and several objects. The Otsu's and Kapur's methods can be extendable to multi-level thresholding, however, they are inefficient because gray level histograms of most of the real-life images are multimodal. Thus, multi-level thresholding has received much attention in recent years. In order to solve the multi-level thresholding problem, some natural computing methods have been applied to solve the multi-level thresholding problem, for example, genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), differential evolution (DE), artificial bee colony (ABC), and bacterial foraging (BF) algorithm. Tao et al. [7] presented a three-level thresholding method that used the GA to find the best thresholds. Hammouche et al. [8] proposed a multi-level thresholding method, which allowed the determination of the appropriate number of thresholds as well as the adequate thresholds. However, GA has several shortcomings, for example, slow convergence rate and premature convergence to local minima. Thus, some PSO-based multi-level thresholding methods have been developed [9–11]. In addition, Tao et al. [12] used the ACO to obtain the best parameters of the presented entropy-based object segmentation method, while Sathya et al. [13] proposed a multi-level thresholding method using the bacterial foraging algorithm. Akay et al. [14] presented a study on PSO and ABC algorithms for multilevel thresholding. Agrawal et al. [15] presented an optimal multi-level thresholding method using cuckoo search algorithm. Osuna-Enciso et al. [16] reported a comparison study of PSO, ABC and DE for multi-threshold image segmentation. Fan et al. [17] proposed a molecular kinetic theory optimization algorithm (MKTOA) to solve the multi-level thresholding problem. Yin et al. [18] proposed a multilevel image segmentation through fuzzy entropy maximization and graph cut optimization.

Membrane computing initiated by Gh. Păun [19], as a new branch of natural computing, is inspired from the structure and functioning of living cells as well as interaction of living cells in

* Corresponding author.
E-mail address: ph.xhu@hotmail.com (H. Peng).

tissues and organs. Membrane computing is a novel class of distributed parallel computing models, known as P systems [20]. In the past years, a variety of variants of P systems have been proposed [21–27], including membrane algorithm of solving global optimization problems [28]. The research results on a variety of optimization problems have indicated that compared to the existing evolutionary algorithms, membrane algorithm offers a more competitive method due to its three advantages: better convergence, stronger robustness and better balance between exploration and exploitation [29–31].

Based on the above consideration, this paper introduces membrane computing to deal with multi-level thresholding problem and proposes a novel multi-level thresholding method. A cell-like P system with the nested structure of three layers, including several evolution membranes, several local store membranes and a global store membrane, is considered as its optimization framework to exploit the best thresholds for an image. Moreover, based on the special membrane structure and communication mechanism of objects, an improved velocity-position model is developed to evolve the objects in the system. In recent, Peng et al. [32] presented a multi-level thresholding method based on tissue-like P systems, where fuzzy entropy is used as the objective function to optimize the thresholds. However, there are three differences with Peng's method [32]: (1) this paper uses the between-class variance criterion and entropy criterion as objective functions respectively, and the existing works have indicated that they are two most effective measures in histogram-based thresholding; (2) a variant with a special membrane structure, namely, a cell-like P system with the nested structure of three layers, is considered in this paper, so the proposed method is inspired from the different mechanism from Peng's method; (3) the external best objects are used to guide the evolution of objects in Peng's method, which can cause the degradation of the objects when initial objects in evolution membranes are very close to each others in solution space.

The rest of this paper is organized as follows. Section 2 reviews two multi-level thresholding problems to be solved, which use the between-class variance criterion and entropy criterion as objective functions, respectively. Section 3 describes the proposed multi-level thresholding method based on cell-like P systems. Experimental results are provided in Section 4, and conclusions are discussed in Section 5.

2. Problem statement

Assume that a given image I has L gray levels, $\{1, 2, \ldots, L\}$. Let h_i denotes the number of pixels with gray level i, thus total number of pixels equals $N = h_1 + h_2 + \ldots + h_L$. The occurrence probability of gray level i is given by

$$p_i = \frac{h_i}{N}, \qquad p_i \geq 0, \qquad \sum_{i=1}^{L} p_i = 1. \tag{1}$$

For the image I, a multi-level thresholding method determines m thresholds, $(t_1, t_2, \ldots, t_m)$, and divides the image into $m+1$ classes: C_0 for $[1, \ldots, t_1]$, C_1 for $[t_1 + 1, \ldots, t_2]$, ..., and C_m for $[t_m + 1, \ldots, L]$. Therefore, the gray level probability distributions for the $m+1$ classes are as follows:

$$C_0\left(\frac{p_1}{\omega_0}, \ldots, \frac{p_{t_1}}{\omega_0}\right), \qquad C_1\left(\frac{p_{t_1+1}}{\omega_1}, \ldots, \frac{p_{t_2}}{\omega_1}\right), \quad \ldots,$$
$$C_m\left(\frac{p_{t_m+1}}{\omega_m}, \ldots, \frac{p_L}{\omega_m}\right) \tag{2}$$

where

$$\omega_0 = \sum_{i=1}^{t_1} p_i, \qquad \omega_1 = \sum_{i=t_1+1}^{t_2} p_i, \quad \ldots, \quad \omega_m = \sum_{i=t_m+1}^{L} p_i. \tag{3}$$

Mean levels for the $m+1$ classes, $\mu_0, \mu_1, \ldots, \mu_m$, respectively, are

$$\mu_0 = \sum_{i=1}^{t_1} \frac{ip_i}{\omega_0}, \qquad \mu_1 = \sum_{i=t_1+1}^{t_2} \frac{ip_i}{\omega_1}, \quad \ldots, \quad \mu_m = \sum_{i=t_m+1}^{L} \frac{ip_i}{\omega_m}. \tag{4}$$

Let μ_T be the mean intensity for whole image. Thus, we have

$$\mu_T = \omega_0\mu_0 + \omega_1\mu_1 + \cdots + \omega_m\mu_m, \quad \omega_0 + \omega_1 + \cdots + \omega_m = 1. \tag{5}$$

Multi-level thresholding can be posed as an optimization problem, which optimizes the m thresholds by maximizing some thresholding criterion (objective function). Usually, there are two thresholding criterions broadly used in literature, between-class variance criterion and entropy criterion, which can be used as the objective function of the optimization problem.

2.1. Case 1: between-class variance criterion

The between-class variance criterion is firstly used in Otsu's bi-level thresholding method [33,34], and then is extended to multi-level thresholding. For the m thresholds, the between-class variance of the image I can be defined by

$$\sigma_B^2 = \sigma_0 + \sigma_1 + \cdots + \sigma_m \tag{6}$$

where $\sigma_0 = \omega_0(\mu_0 - \mu_T)^2, \sigma_1 = \omega_1(\mu_1 - \mu_T)^2, \ldots, \sigma_m = \omega_m(\mu_m - \mu_T)^2$. Thus, the multi-level thresholding problem can be configured as the following optimization problem:

$$\max_{1 \leq t_1 \leq \cdots \leq t_m \leq L} J_1(t_1, t_2, \ldots, t_m) = \max_{1 \leq t_1 \leq \cdots \leq t_m \leq L} \sigma_B^2(t_1, t_2, \ldots, t_m) \tag{7}$$

where $t_1, t_2, \ldots, t_m$ are m parameters (thresholds) to be optimized.

2.2. Case 2: entropy criterion

The entropy criterion has been developed by Kapur in bi-level thresholding [6,35], and has been extended to multi-level thresholding. For the m thresholds, the entropy criterion can be defined as follows:

$$H_e = H_0 + H_1 + \cdots + H_m \tag{8}$$

where

$$H_0 = -\sum_{i=1}^{t_1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=1}^{t_1} p_i,$$
$$H_1 = -\sum_{i=t_1+1}^{t_2} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=t_1+1}^{t_2} p_i,$$
$$\cdots\cdots \qquad \cdots\cdots \qquad \cdots\cdots$$
$$H_m = -\sum_{i=t_m+1}^{L} \frac{p_i}{\omega_m} \ln \frac{p_i}{\omega_m}, \quad \omega_m = \sum_{i=t_m+1}^{L} p_i.$$

Based on the entropy criterion, the multi-level thresholding problem can be configured as the following optimization problem:

$$\max_{1 \leq t_1 \leq \cdots \leq t_m \leq L} J_2(t_1, t_2, \ldots, t_m) = \max_{1 \leq t_1 \leq \cdots \leq t_m \leq L} H_e(t_1, t_2, \ldots, t_m) \tag{9}$$

where $t_1, t_2, \ldots, t_m$ are m parameters (thresholds) to be optimized.

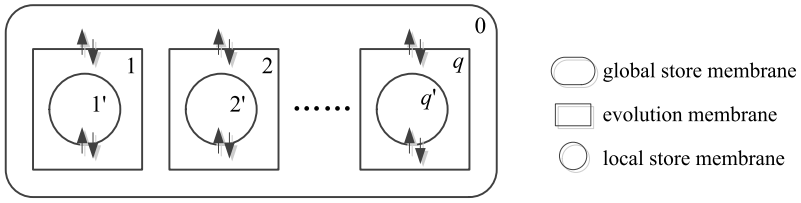

Fig. 1. The cell-like P system with the nested structure of three layers, where arrows denote the communication relations of objects.

3. Proposed multi-level thresholding method

The proposed multi-level thresholding method uses a cell-like P system with the nested structure of three layers as its optimization framework. The role of the cell-like P system is to optimize the m thresholds for an image. Fig. 1 shows the cell-like P system with the nested structure of three layers, consisting of $(2q+1)$ membranes: a skin (the outermost) membrane, q middle membranes and (the innermost) elementary membranes. The q middle membranes are labeled by $1, 2, \ldots, q$ respectively, called evolution membranes, role of which is to evolve the objects of the system. Each evolution membrane contains an elementary membrane, called local store membrane, role of which is to store the best object found so far in the evolution membrane. The local store membranes are labeled by $1', 2', \ldots, q'$, respectively. The skin membrane labeled by 0 is also called global store membrane, role of which is to store the best object found so far in whole system. Note that there are no evolution rules in the global store membrane and the local store membranes.

3.1. Object representation

In the designed cell-like P system, each evolution membrane contains n objects, and each of them expresses a set of m thresholds to be optimized. Thus, each object is a m-dimensional vector of the form

$$X = (t_1, t_2, \ldots, t_m) \tag{10}$$

where $t_1, t_2, \ldots, t_m$ are m thresholds to be optimized. The cell-like P system will find the best thresholds for an image by evolution of objects. Each local store membrane contains only an object, called local best object, which is the best object found so far in the corresponding evolution membrane. The global store membrane also contains an object, called global best object, which is the best object found so far in whole system.

3.2. Object evolution

Object evolution is realized by evolution rules. In this work, an improved position-velocity model is developed based on inherent communication mechanism of cell-like P systems, which can be regarded as a variant of the velocity-position model in particle swarm optimization (PSO) [36,37]. For jth object in kth evolution membrane, $X_{k,j}$, the extended velocity-position model can be described as follows:

$$\begin{aligned} V_{k,j} &= \omega \cdot X_{k,j} + c_1 r_1 (P_{k,j} - X_{k,j}) + c_2 r_2 (G_k - X_{k,j}) \\ &\quad + c_3 r_3 (G_* - X_{k,j}), \\ X_{k,j} &= \mathrm{floor}(X_{k,j} + V_{k,j}), \quad j = 1, 2, \ldots, n, \ k = 1, 2, \ldots, q, \end{aligned} \tag{11}$$

where ω is an inertia weight, c_1, c_2 and c_3 are learning factors, r_1, r_2, r_3 are three random numbers in $(0, 1)$, $\mathrm{floor}(\cdot)$ is a rounded function, $V_{k,j}$ is the velocity vector associated with $X_{k,j}$, and $P_{k,j}$ is the found best position of object $X_{k,j}$ so far. Different from the classical velocity-position model, the extended velocity-position model uses two best objects to guide the evolution of objects in each evolution membrane: one is the best object in current evolution membrane, G_k, and another is the found global best object in all evolution membranes so far, G_*. The consideration can bring two benefits: (1) the mechanism that the global best object involves the co-evolution of objects in multiple evolution membranes can speed up the convergence of the system; (2) since the two best objects are from different sources, this mechanism can better improve the diversity of objects in the system, thus it can avoid premature convergence to local optima.

3.3. Object communication

Another mechanism of the cell-like P system is communication of objects between each evolution membrane and the corresponding local store membrane or between each evolution membrane and the global store membrane. The arrows in Fig. 1 show the communication relations of objects. The communication often happens after the evolution of objects in evolution membranes. After all objects are evolved, each evolution membrane automatically communicates its best object into the corresponding local store membrane and the global store membrane to update the corresponding local best object and the global best object, respectively. The update strategy used is as follows: if fitness value of the communicated object is higher than that of the existing object in local store membranes or global store membrane, it will replace the existing object; otherwise, it will be discarded.

3.4. Halting and output

As usual in P systems, all of the evolution membranes as parallel computing units work in a maximally parallel way (a universal clock is considered here). Moreover, the global store membrane is assigned as output region of whole system. For simplicity, a maximum execution step number given in advance is considered as halting condition. The cell-like P system will continue to run under the control of its evolution–communication mechanism until the halting condition is reached. When the system halts, the global best object in the global store membrane is regarded as the output of whole system, namely, m best thresholds.

Based on the cell-like P system, the proposed multi-level thresholding method is summarized in Table 1. In the following, we briefly discuss its computational complexity. We consider its two versions: parallel version and non-parallel version. The thresholding method consists of three main steps: initialization, object evolution and halting. From Table 1, it can be observed that initialization step contains double loop (q and n times, respectively), so its time complexity is $O(qn)$. For object evolution step, there are triple loop (q, n, and S times, respectively), therefore, its time complexity is $O(qnS)$. For halting step, its time complexity is $O(1)$. Therefore, for non-parallel version, its time complexity is $O(qnS)$. However, in the case of parallel version, the proposed multi-level thresholding method has $O(nS)$ complexity since all of the q evolution membranes are parallel computing units.

4. Experimental results and analysis

In experiments, we have implemented non-parallel version of the proposed multi-level thresholding method (due to the limit of series architecture of the used computer) as well as two recently developed multi-level thresholding methods, which are BF-based and PSO-based methods. Table 2 gives the input parameters of the three methods used in experiments. Nine well-known test images with 256 gray levels and size 512×512 are used for conducing our experiments, which are "Lena", "Peppers", "Baboon", "Hunter", "Stanwick", "Living room", "House", "Airplane" and "Butterfly", respectively. Fig. 2 shows these images and their histograms.

Two case studies are considered to evaluate the efficiency and effectiveness of the proposed method, where Ostu's between-class variance and Kapur's entropy are used as objective functions of the optimal multi-level thresholding problem respectively. For each case study, the comparison is considered in terms of quantitative and qualitative results, including best thresholds, objective function value, PSNR (Peak Signal-to-Noise Ratio), computing time and the thresholded image. A higher objective function value indices a better result. A larger PSNR value indicates a better quality of thresholding. The shown objective function values and computing times are the averages over 50 runs, and mean value and standard deviation of PSNR are also provided.

Table 1
The multi-level thresholding method based on cell-like P systems.

Input parameters: Histogram $\{h_i \mid i = 1, 2, \ldots, L\}$, the number of evolution membranes q, the number of objects in each evolution membrane n, maximum execution step number S, learning factors c_1, c_2 and c_3, and inertia weight ω.
Output results: the best thresholds, $G_* = (t_1^*, t_2^*, \ldots, t_m^*)$.
Step 1. **Initialization**
  for $k = 1$ *to* q
    for $j = 1$ *to* n
      Generate j-th initial object for evolution membrane k, X_{kj};
      Compute the fitness value of the object according to Eq. (6) or Eq. (8);
    end for
    Fill local best object G_k using the best initial object in evolution membrane k;
  end for
  Fill global best object G_* using the best of all initial objects;
  Set computing step $s = 0$;
Step 2. **Object evolution in evolution membranes**
  for each evolution membrane k ($k = 1, 2, \ldots, q$) in parallel *do*
    for $j = 1$ *to* n
      Evolve object X_{kj} using the improved velocity-position model (10);
      Compute the fitness value of the object according to Eq. (6) or Eq. (8);
    end for
    Update the local best object G_k in the corresponding local store membrane;
    Update the global best object G_* in the global store membrane;
  end for
Step 3. **Halting condition judgment**
  if $s \leq S$ is satisfied
    $s = s + 1$;
    goto **Step 2**;
  end if
  The system exports the global best object G_* and halts;

4.1. Case study 1

Case study 1 is considered to solve the optimal multi-level thresholding problem (7), in which Ostu's between-class variance is used for its objective function. Table 3 provides the best thresholds, average objective function values of the three methods over 50 runs, in which four number of thresholds are considered, i.e., $m = 2, 3, 4$ and 5. The best thresholds of a method refer to a set of thresholds with the largest objective function value. It can be clear observed from Table 3 that the proposed method achieves the largest objective function value in comparison to other two methods.

Figs. 3–5 show the detailed qualitative segmentation results of P system for the nine images. Table 4 gives mean values and standard deviations of PSNR values obtained by the three methods over 50 runs. It can be seen from Table 4 that average PSNR values of P system are higher than that of PSO and BF. The quantitative and qualitative results illustrate that compared with PSO and BF, P system has better quality of thresholding. Meanwhile, the P system has the lowest standard deviation for each of the nine images. The comparison results demonstrate that P system outperforms other two methods in terms of robustness.

Table 4 also provides the comparison results of the three methods in terms of computing time (second). Note that the computing time refers to spending time of a method when it converges to its best objective function value during its a run. It can be seen from Table 4 that BF has the smallest average computing time, second is PSO, while P system is the longest in the three methods. However, an interesting conclusion is observed in the comparison results. According to the parameter configure of the three methods listed in Table 2, the P system has all 250 objects, which apparently corresponds to a PSO with 250 particles or a BF with 250 bacteria. Intuitively, the computing time of the non-parallel version of the P system is several times more than that of PSO or BF. However, its computing time is slightly higher than that of PSO or BF. The results illustrate that the non-parallel version of the P system has also relatively faster convergence.

4.2. Case study 2

Case study 2 is used to test the performance of the three methods for the optimal multi-level thresholding problem (9), where Kapur's entropy is used for its objective function. Table 5 reports the best thresholds, average objective function values of the three methods over 50 runs. The results clear show that the P system can exploit the best thresholds for the nine images.

Table 6 provides the comparison results of PSNR values obtained by the three methods over 50 runs in terms of mean value and standard deviation, respectively. It can be seen from Table 6

Table 2
Input parameters of cell-like P system, BP and PSO.

Cell-like P system		BP		PSO	
Number of membranes (q)	5	Number of bacterium (S)	50	Population size (N)	50
Number of objects (n)	50	Number of chemotactic steps (N_c)	10	Maximum number of iterations	200
Maximum number of iterations	200	Swimming length (N_s)	10	Inertia weight (w_{max}, w_{min})	0.9, 0.4
Inertia weight (w_{max}, w_{min})	0.9, 0.4	Number of reproduction steps (N_{re})	4	Learning factors (c_1, c_2)	2
Learning factors (c_1, c_2, c_3)	1.0	Number of elimination of dispersal event (N_{ed})	2		
		Probability of elimination and dispersal (P_{ed})	0.02		
		Depth of attractant (d_{att})	0.1		
		Width of attract (W_{att})	0.2		
		Height of repellent (h_{rep})	0.1		
		Width of repellent (W_{rep})	10		

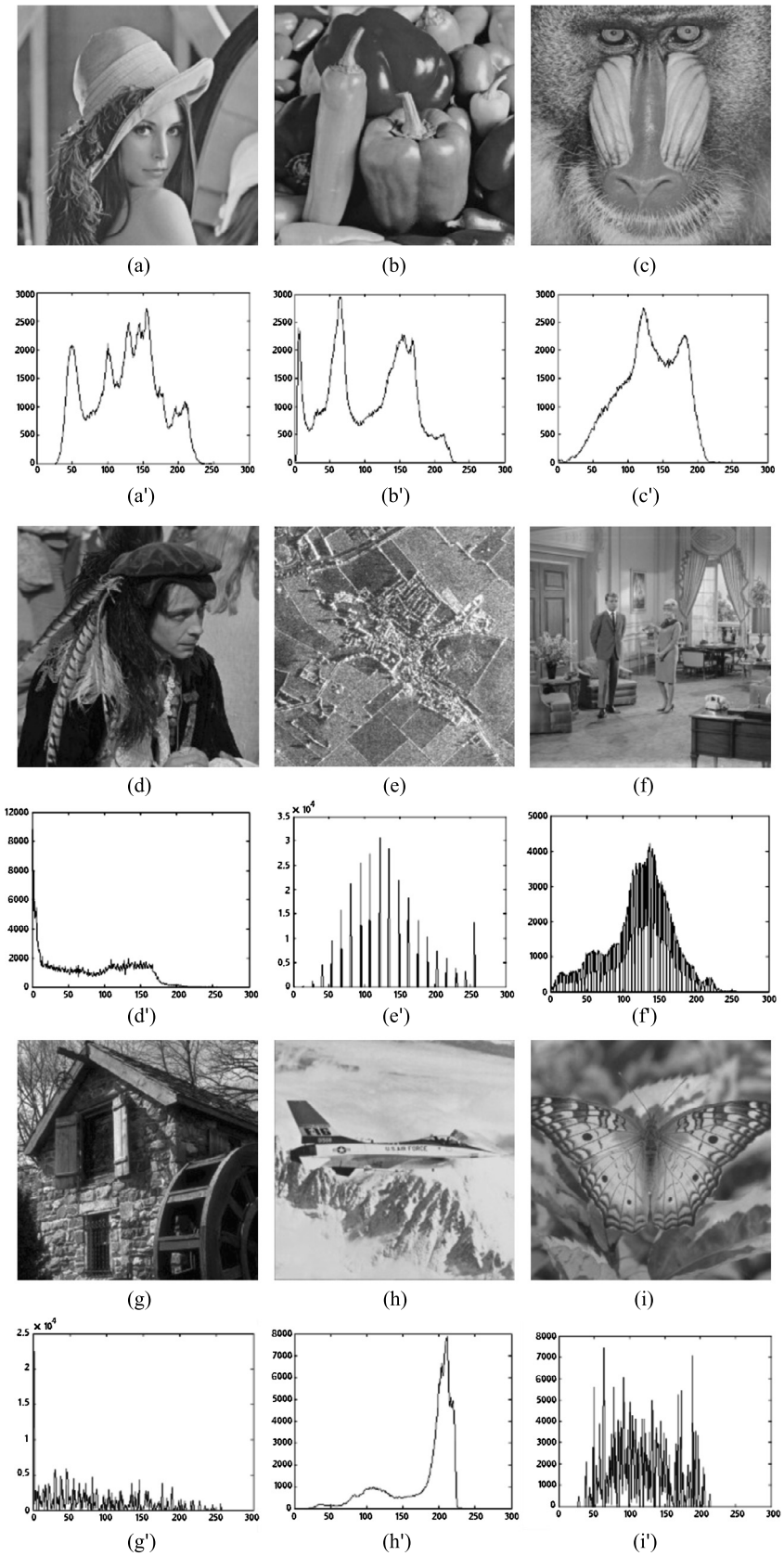

Fig. 2. The tested images and their histograms: (a) Lena, (b) Peppers, (c) Baboon, (d) Hunter, (e) Stanwick, (f) Living room, (g) House, (h) Airplane and (i) Butterfly.

Table 3
Comparison of best thresholds and average objective function values obtained by three methods for case study 1.

Test image	m	Best thresholds			Average objective function value		
		P system	PSO	BF	P system	PSO	BF
Lena	2	93,151	94,152	91,151	**1961.523**	1961.413	1961.555
	3	80,126,171	79,127,170	79,125,170	**2128.126**	2127.776	2128.071
	4	75,114,145,180	78,112,134,175	76,117,151,182	**2191.442**	2180.686	2189.026
	5	73,109,136,160,188	79,110,140,167,188	66,92,122,149,183	**2217.434**	2212.554	2215.609
Peppers	2	49,116	76,144	73,137	**2866.337**	2469.578	2474.808
	3	43,99,153	72,124,171	63,125,174	**3066.248**	2623.273	2625.362
	4	41,89,135,175	57,92,130,172	54,89,128,171	**3151.990**	2695.886	2697.783
	5	39,80,118,150,182	56,84,115,150,179	47,86,123,158,183	**3195.937**	2733.508	2735.644
Baboon	2	97,149	96,149	98,150	**1548.140**	1547.997	1548.012
	3	85,125,161	85,126,166	84,126,159	**1638.320**	1635.362	1637.007
	4	72,106,137,168	79,105,140,174	77,109,139,169	**1692.149**	1684.335	1690.722
	5	67,99,125,149,174	74,104,134,161,180	70,99,127,154,177	**1717.890**	1721.958	1716.727
Hunter	2	51,116	52,116	51,117	**3064.210**	3064.067	3064.118
	3	36,86,135	39,86,135	36,86,135	**3213.445**	3212.058	3213.446
	4	30,72,111,146	36,84,130,157	31,80,120,152	**3269.515**	3257.176	3266.349
	5	22,53,88,122,152	37,85,125,154,177	31,73,109,141,178	**3308.141**	3276.317	3291.132
Stanwick	2	110,183	113,177	109,176	**2331.370**	2340.395	2340.395
	3	99,148,199	81,145,197	98,146,189	**2519.875**	2526.303	2529.934
	4	85,125,169,221	92,133,162,206	88,134,173,222	**2609.491**	2618.489	2621.146
	5	68,114,135,178,228	79,116,139,162,204	80,109,135,165,224	**2658.407**	2665.412	2668.069
Living room	2	88,147	88,145	87,146	**1626.643**	1627.796	1627.824
	3	76,124,164	81,127,165	75,124,164	**1758.850**	1757.466	1759.845
	4	57,99,133,169	69,110,143,178	64,102,134,172	**1827.456**	1822.113	1826.628
	5	50,89,120,146,179	56,98,128,156,190	56,94,125,148,180	**1870.574**	1865.476	1869.996
House	2	55,128	57,127	56,129	**3420.715**	3420.985	3421.282
	3	42,98,162	48,104,165	43,102,165	**3622.756**	3617.983	3622.305
	4	31,75,123,178	40,88,140,194	34,82,135,182	**3725.315**	3702.288	3711.701
	5	24,56,92,130,178	32,74,129,158,188	33,80,124,170,212	**3785.194**	3752.146	3759.015
Airplane	2	116,174	117,174	117,175	**1837.797**	1837.723	1837.751
	3	95,146,191	99,158,193	91,147,190	**1669.278**	1665.758	1667.289
	4	88,132,173,204	84,125,168,201	84,127,169,202	**1955.048**	1953.886	1954.248
	5	71,108,143,179,204	60,101,138,177,204	71,110,138,175,203	**1979.959**	1977.874	1978.433
Butterfly	2	98,151	99,150	99,151	**1553.073**	1553.067	1553.072
	3	82,118,160	79,119,164	78,117,162	**1669.278**	1665.756	1667.289
	4	71,99,126,161	80,113,145,177	75,105,135,165	**1711.219**	1702.905	1707.098
	5	71,99,125,153,179	75,106,129,157,180	76,104,129,154,180	**1736.657**	1730.786	1733.031

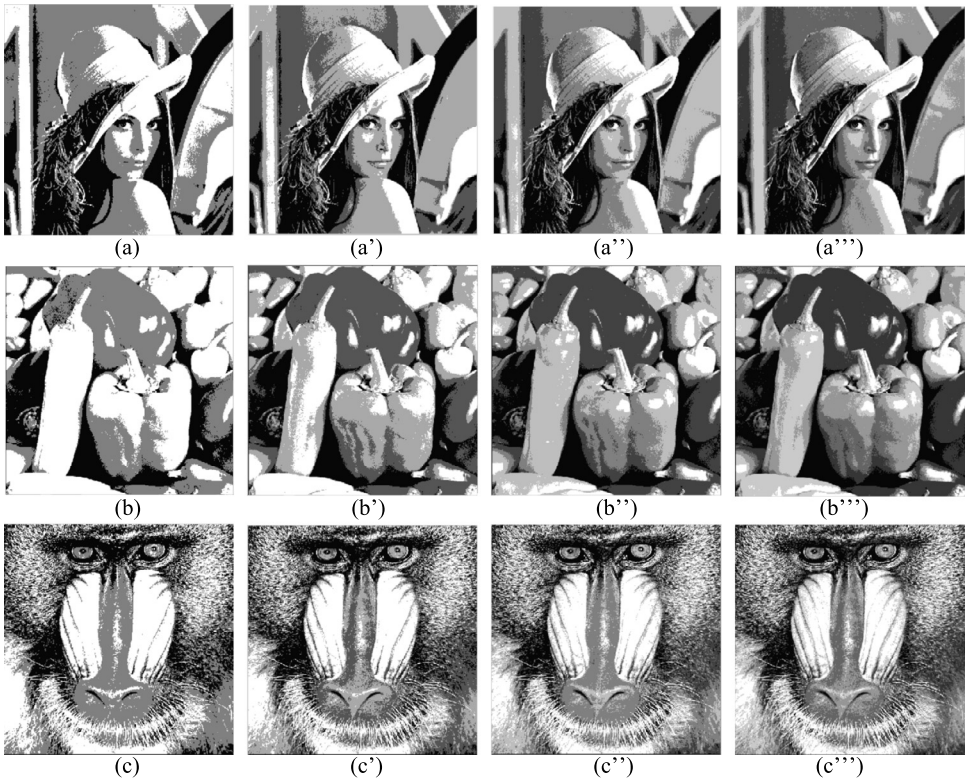

Fig. 3. The thresholded images generated by P system in case study 1 for Lena, peppers and baboon: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

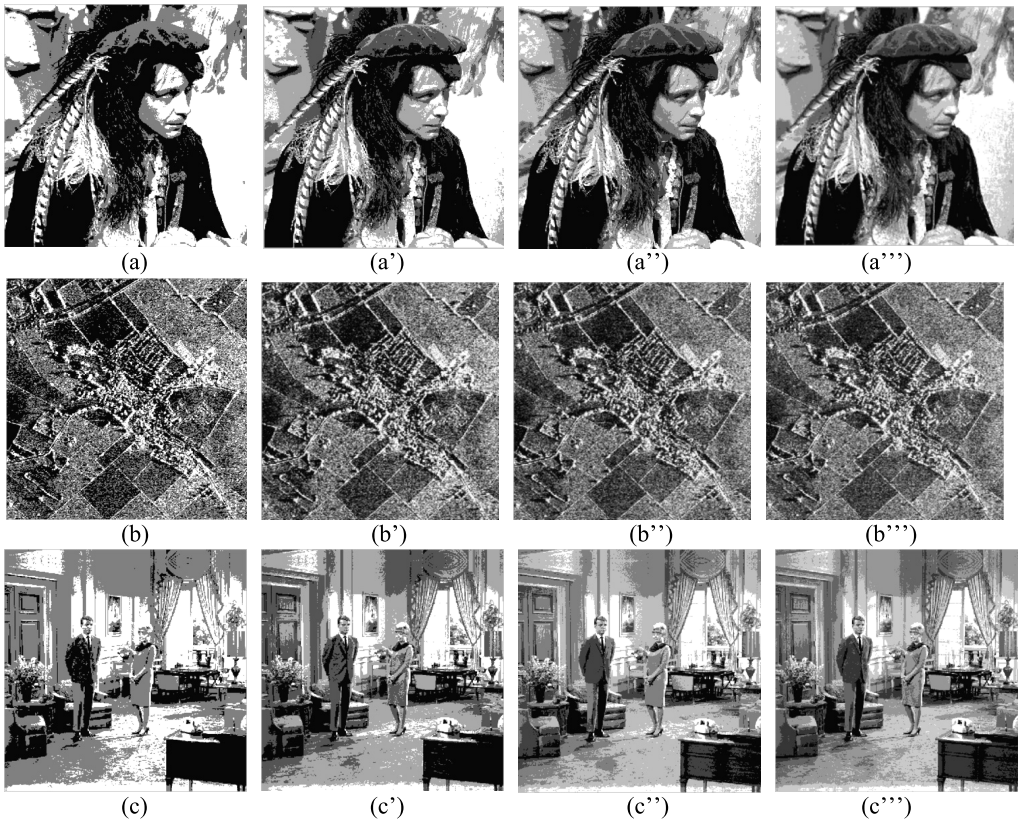

Fig. 4. The thresholded images generated by P system in case study 1 for hunter, Stanwick and living room: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

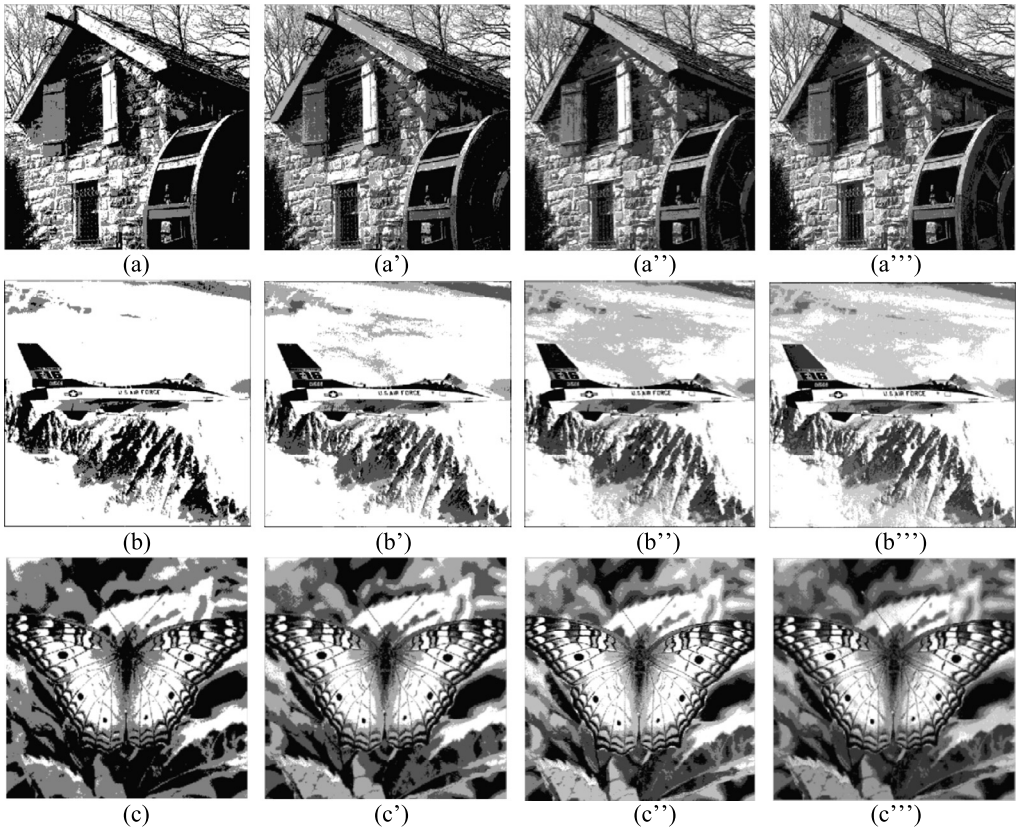

Fig. 5. The thresholded images generated by P system in case study 1 for house, airplane and butterfly: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

Table 4
Comparison of PSNR value and computing time over 50 runs for case study 1.

Test images	m	PSNR						Average computing time (second)		
		Mean value			Standard deviation					
		P system	PSO	BF	P system	PSO	BF	P system	PSO	BF
Lena	2	**17.536**	14.873	15.163	**0.000**	0.000	0.000	7.128	6.749	**6.353**
	3	**20.532**	16.753	17.338	**0.000**	2.265e−6	3.718e−6	8.993	8.503	**7.791**
	4	**21.521**	18.832	19.397	**3.162e−12**	3.561e−5	2.526e−6	9.271	8.925	**8.382**
	5	**22.372**	19.446	20.562	**5.336e−9**	6.173e−5	4.772e−6	10.387	9.836	**8.832**
Peppers	2	**18.645**	12.821	12.598	**0.000**	0.000	0.000	7.238	6.538	**5.823**
	3	**19.479**	16.592	15.961	**0.000**	4.123e−6	8.462e−6	8.362	7.491	**6.358**
	4	**21.516**	19.126	16.639	**5.323e−12**	2.735e−5	7.636e−5	8.992	7.875	**6.782**
	5	**22.478**	20.365	20.113	**6.721e−7**	3.142e−4	6.537e−4	10.529	9.016	**6.981**
Baboon	2	**18.327**	13.032	13.009	**0.000**	0.000	0.000	7.437	6.792	**6.331**
	3	**19.492**	16.981	17.982	**2.321e−14**	3.571e−6	1.432e−5	8.325	7.442	**6.825**
	4	**20.335**	17.028	18.325	**1.992e−9**	4.713e−6	8.546e−6	8.581	8.035	**7.339**
	5	**20.975**	18.115	18.693	**5.237e−7**	1.115e−5	2.159e−5	9.416	8.521	**7.956**
Hunter	2	**15.836**	11.037	11.316	**0.000**	0.000	0.000	7.151	6.921	**6.353**
	3	**17.358**	14.382	14.539	**7.457e−10**	2.251e−6	5.773e−7	7.892	7.585	**7.082**
	4	**18.117**	15.291	16.194	**2.842e−8**	5.912e−6	3.162e−6	8.452	8.126	**7.391**
	5	**18.673**	16.448	17.319	**5.383e−7**	2.364e−4	6.921e−5	9.117	8.614	**7.938**
Stanwick	2	**17.523**	13.665	14.217	**0.000**	0.000	0.000	7.353	6.237	**5.523**
	3	**18.281**	14.361	15.372	**3.352e−8**	3.118e−5	1.982e−6	7.894	6.731	**6.415**
	4	**19.795**	15.722	16.553	**2.916e−7**	2.261e−5	5.337e−6	8.392	7.535	**6.821**
	5	**21.316**	16.628	18.449	**4.615e−5**	5.712e−4	3.321e−5	9.476	8.256	**7.135**
Living room	2	**14.323**	13.015	12.917	**0.000**	0.000	0.000	7.137	6.438	**6.129**
	3	**17.982**	17.108	17.035	**0.000**	8.113e−6	2.662e−6	7.831	7.225	**6.743**
	4	**19.447**	19.216	19.118	**4.227e−9**	9.847e−6	5.831e−6	8.328	7.689	**7.208**
	5	**21.152**	21.147	20.836	**6.883e−8**	9.558e−5	5.554e−5	9.336	8.872	**8.131**
House	2	**17.821**	12.583	12.681	**0.000**	0.000	0.000	7.215	6.735	**6.295**
	3	**19.582**	13.772	13.989	**4.112e−8**	4.568e−5	2.769e−6	7.669	7.421	**6.887**
	4	**21.733**	14.821	15.832	**2.834e−7**	8.883e−5	4.651e−6	8.095	7.883	**7.513**
	5	**22.812**	16.135	16.669	**5.064e−7**	9.647e−5	8.482e−5	9.572	8.982	**8.142**
Airplane	2	**16.445**	13.447	13.631	**0.000**	0.000	0.000	7.359	6.532	**6.149**
	3	**19.063**	15.472	15.693	**3.236e−8**	4.335e−6	3.448e−7	8.314	7.681	**6.975**
	4	**21.261**	15.533	16.181	**8.495e−8**	3.821e−6	6.425e−7	9.217	7.983	**7.236**
	5	**22.519**	17.465	17.592	**2.532e−7**	5.484e−5	6.119e−6	9.976	8.523	**7.952**
Butterfly	2	**16.345**	13.027	13.042	**0.000**	0.000	0.000	7.431	6.638	**6.447**
	3	**19.021**	16.558	17.115	**5.882e−15**	6.335e−5	3.116e−6	7.982	7.318	**6.759**
	4	**21.283**	18.735	19.425	**6.421e−13**	7.223e−4	2.625e−5	8.689	7.956	**7.365**
	5	**22.891**	21.298	21.733	**3.836e−7**	1.363e−3	7.353e−5	9.894	8.526	**8.237**

that the P system has the highest average PSNR value for each image in comparison to BF and PSO. Figs. 6–8 show the thresholded images generated by the P system. The quantitative and qualitative results demonstrate that the P system can achieve a better quality of thresholding for nine images. Meanwhile, the comparison results in Table 6 show that the P system have the lowest standard deviation in the three methods, which illustrates the P system is robust for the optimal multi-level thresholding problem (9).

The comparison results of the three methods in terms of computing time are provided in Table 6. Although the computing time of the P system is slightly larger than that of PSO and BF, however, the comparison results indicate that its non-parallel version has also relatively faster convergence based on above analysis.

4.3. Statistical significance test

A nonparametric statistical significance test, Wilcoxon's rank sum test, is conducted at the 5% significance level in the experiments. We create three groups for each case study, which are corresponding to the three methods (P system, BF and PSO) respectively. Each group consists of best objective function values and RMSE values produced by 50 consecutive runs of the corresponding methods for the nine images, respectively. Table 7 and Table 8 give the p-values of two groups (one group corresponding to P system and another group corresponding to some other method) for the two case studies, respectively. The shown results indicate that all p-values are less than 0.05 (5% significance level). This is a strong evidence for establishing significant superiority of the proposed multi-level thresholding method.

5. Conclusions and further work

This paper has presented a novel method inspired from membrane computing to solve the optimal multi-level thresholding problem. A cell-like P system with the nested structure of three layers has been considered as its computing framework designed, and an extended velocity-position model has been developed to evolve the objects of the system. Based on inherent evolution-communication mechanism, the cell-like P system can effectively and efficiently exploit the best thresholds for an image. Moreover, the mechanism can also accelerate the convergence of the proposed thresholding method and enhance the diversity of objects in the system. The proposed multi-level thresholding method has been tested on nine standard images and compared with two recently developed thresholding methods, PSO-based and BF-based multi-level thresholding methods. Simulation results of both qualitative and quantitative comparisons for the three multi-level thresholding methods demonstrate the proposed multi-level thresholding method has a better quality, robustness and computation efficiency.

The parallel computing is one of advantages of the cell-like P system, however, the parallel computing has not been realized in the simulation due to the limit of series architecture of the computer used in the experiments. In order to overcome the shortcoming, our further work is to consider the realization of its high

Table 5
Comparison of best thresholds and average objective function values obtained by three methods for case study 2.

Test image	m	Best thresholds			Average objective function value		
		P system	PSO	BF	P system	PSO	BF
Lena	2	97,164	99,165	97,164	**12.346**	12.345	12.347
	3	82,126,175	86,151,180	88,142,188	**15.317**	15.132	15.226
	4	64,97,138,179	92,129,162,191	74,114,149,184	**18.012**	17.837	17.933
	5	63,94,128,163,194	74,115,145,170,197	64,95,128,163,194	**20.611**	20.442	20.607
Peppers	2	93,177	79,146	79,149	**12.553**	12.516	12.518
	3	73,126,178	104,141,180	69,100,155	**15.824**	15.093	15.399
	4	46,84,130,179	57,110,162,199	63,109,144,178	**18.733**	18.096	18.268
	5	43,76,111,144,181	70,116,138,166,200	54,89,131,164,197	**21.402**	20.732	20.997
Baboon	2	79,143	76,144	81,144	**12.216**	12.213	12.216
	3	44,98,152	72,130,181	53,112,150	**15.281**	15.008	15.211
	4	33,74,114,159	65,121,153,180	39,90,131,168	**18.129**	17.574	17.999
	5	33,70,105,139,173	73,110,142,166,192	38,79,113,148,180	**20.789**	20.224	20.719
Hunter	2	92,179	83,179	85,179	**12.376**	12.369	12.373
	3	59,117,179	85,128,166	57,104,175	**15.613**	15.128	15.552
	4	46,90,133,179	74,131,174,200	50,98,139,180	**18.526**	18.041	18.381
	5	46,90,133,179,222	90,120,164,190,219	49,93,137,179,222	**21.265**	20.533	21.256
Stanwick	2	97,183	97,181	95,181	**4.978**	4.978	4.978
	3	68,145,198	74,140,181	78,125,189	**5.571**	5.503	5.551
	4	56,121,168,203	92,128,152,207	73,132,174,206	**5.932**	5.689	5.892
	5	30,79,133,174,207	66,109,121,150,195	64,106,135,170,210	**6.161**	5.916	6.066
Living room	2	93,174	86,175	89,170	**11.912**	12.401	12.405
	3	48,105,176	73,158,187	71,124,173	**14.906**	15.212	15.407
	4	48,100,150,198	59,124,172,202	60,104,147,189	**17.637**	18.141	18.318
	5	43,87,126,165,200	72,97,119,158,197	47,94,134,169,200	**20.158**	20.675	21.119
House	2	71,144	81,144	65,144	**10.849**	10.832	10.847
	3	47,96,154	81,116,155	56,110,172	**13.383**	13.101	13.265
	4	47,81,121,163	75,123,154,193	47,87,131,167	**15.631**	15.102	15.607
	5	39,71,107,143,189	48,97,139,159,189	41,73,112,144,176	**17.667**	17.251	17.626
Airplane	2	76,174	80,175	76,173	**12.176**	12.149	12.175
	3	73,128,182	72,121,191	66,124,186	**15.432**	15.292	15.359
	4	68,106,144,184	74,129,162,188	71,113,149,185	**18.205**	18.031	18.177
	5	66,96,127,157,187	81,118,144,167,192	68,98,131,161,187	**20.768**	20.396	20.751
Butterfly	2	96,144	95,141	97,144	**10.476**	10.474	10.475
	3	83,118,152	63,126,172	75,109,154	**12.812**	12.313	12.754
	4	77,105,133,164	71,113,162,184	73,97,127,157	**14.921**	14.231	14.876
	5	73,96,120,144,164	92,116,142,157,182	74,97,120,144,167	**16.837**	16.337	16.828

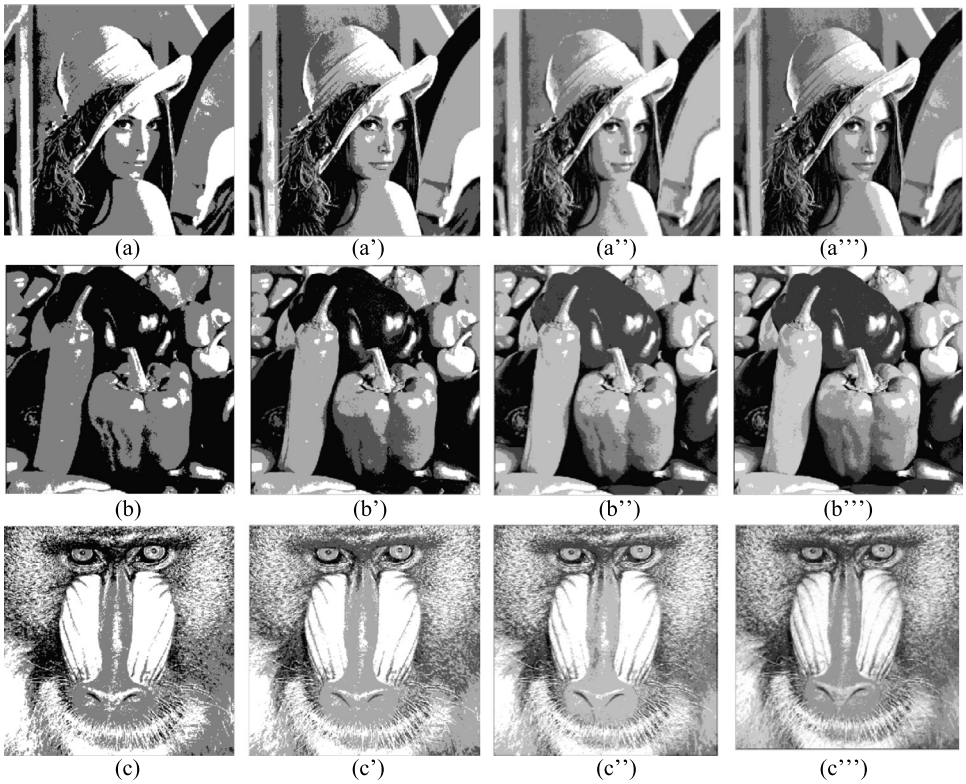

Fig. 6. The thresholded images generated by P system in case study 2 for Lena, peppers and baboon: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

Table 6
Comparison of PSNR value and computing time over 50 runs for case study 2.

Test images	m	PSNR						Average computing time (second)		
		Mean value			Standard deviation					
		P system	PSO	BF	P system	PSO	BF	P system	PSO	BF
Lena	2	**17.931**	15.591	15.637	**0.000**	0.000	0.000	12.352	11.873	**11.285**
	3	**20.895**	17.284	17.528	**0.000**	2.137e−6	2.371e−6	12.847	12.426	**11.712**
	4	**21.874**	19.195	19.816	**2.352e−12**	2.632e−5	1.892e−6	14.415	13.225	**12.615**
	5	**22.798**	19.937	20.985	**4.285e−9**	5.217e−5	3.538e−6	14.863	13.576	**12.928**
Peppers	2	**19.136**	13.251	12.832	**0.000**	0.000	0.000	12.361	11.226	**10.634**
	3	**19.981**	16.869	16.373	**0.000**	3.247e−6	7.153e−6	12.983	11.792	**10.957**
	4	**22.084**	19.463	17.295	**3.984e−12**	1.895e−5	6.582e−5	13.421	12.253	**11.528**
	5	**22.892**	20.801	20.346	**5.426e−7**	2.226e−4	5.326e−4	13.792	12.609	**11.931**
Baboon	2	**18.854**	13.284	13.521	**0.000**	0.000	0.000	13.252	12.126	**11.827**
	3	**19.935**	17.452	18.438	**2.148e−14**	2.483e−6	1.314e−5	13.876	12.792	**12.336**
	4	**20.792**	17.385	18.825	**1.782e−9**	3.652e−6	7.448e−6	14.338	13.237	**12.841**
	5	**21.532**	18.327	19.083	**4.739e−7**	1.043e−5	1.835e−5	14.927	13.831	**13.215**
Hunter	2	**16.415**	11.351	11.694	**0.000**	0.000	0.000	12.588	12.109	**11.462**
	3	**17.635**	14.672	14.835	**6.873e−10**	1.836e−6	4.339e−7	13.054	12.783	**12.338**
	4	**18.608**	15.448	16.382	**2.526e−8**	5.512e−6	2.835e−6	14.179	13.215	**12.874**
	5	**19.236**	16.923	17.535	**4.681e−7**	2.127e−4	6.431e−5	15.326	14.196	**13.691**
Stanwick	2	**17.832**	13.882	14.682	**0.000**	0.000	0.000	13.398	12.146	**11.379**
	3	**18.503**	14.745	15.593	**3.283e−8**	2.835e−5	1.768e−6	13.526	12.583	**12.531**
	4	**20.215**	16.135	17.016	**2.754e−7**	2.097e−5	5.241e−6	14.478	13.462	**12.794**
	5	**21.765**	16.916	18.895	**4.496e−5**	4.565e−4	2.895e−5	15.395	14.391	**13.263**
Living room	2	**14.698**	13.472	13.352	**0.000**	0.000	0.000	12.315	11.625	**10.865**
	3	**18.431**	17.395	17.416	**0.000**	7.594e−6	2.436e−6	14.179	12.839	**11.576**
	4	**19.889**	19.516	19.524	**3.518e−9**	9.682e−6	5.691e−6	14.516	13.234	**12.693**
	5	**21.634**	21.445	21.361	**5.623e−8**	9.732e−5	5.468e−5	15.374	14.236	**13.451**
House	2	**18.275**	12.893	12.892	**0.000**	0.000	0.000	13.236	12.135	**11.412**
	3	**19.923**	14.193	14.327	**3.885e−8**	4.253e−5	2.519e−6	13.523	12.362	**11.825**
	4	**22.252**	15.339	16.269	**2.624e−7**	8.651e−5	4.375e−6	14.137	12.931	**12.314**
	5	**23.365**	16.426	17.136	**4.379e−7**	8.492e−5	7.975e−5	15.982	13.758	**12.893**
Airplane	2	**16.791**	13.926	14.095	**0.000**	0.000	0.000	12.438	11.986	**11.358**
	3	**19.452**	15.841	15.894	**2.851e−8**	4.235e−6	3.192e−7	13.835	13.062	**12.371**
	4	**21.385**	15.908	16.525	**7.518e−8**	3.674e−6	6.335e−7	14.579	13.357	**12.635**
	5	**22.907**	17.763	17.876	**2.375e−7**	5.293e−5	5.869e−6	15.218	14.054	**13.184**
Butterfly	2	**16.573**	13.526	13.352	**0.000**	0.000	0.000	12.463	11.825	**11.232**
	3	**19.361**	16.994	17.524	**5.392e−15**	6.281e−5	2.874e−6	13.392	12.476	**11.905**
	4	**21.495**	19.021	19.831	**6.236e−13**	6.905e−4	2.438e−5	14.326	13.115	**12.523**
	5	**23.214**	21.475	22.436	**3.724e−7**	1.292e−3	6.925e−5	14.618	13.462	**12.741**

Table 7
The results of p-values of Wilcoxon's rank sum test for case study 1.

Test images	Between-class variance		PSNR	
	P system vs. PSO	P system vs. BF	P system vs. PSO	P system vs. BF
Lena	2.543e−4	1.823e−3	2.513e−3	0.0287
Peppers	3.117e−4	1.664e−3	2.678e−3	0.0274
Baboon	2.825e−4	1.903e−3	2.719e−3	0.0283
Hunter	2.799e−4	1.885e−3	2.645e−3	0.0312
Stanwick	2.692e−4	1.754e−3	2.582e−3	0.0307
Living room	2.875e−4	1.923e−3	2.816e−3	0.0285
House	2.917e−4	1.879e−3	2.596e−3	0.0292
Airplane	2.796e−4	1.851e−3	2.651e−3	0.0325
Butterfly	2.823e−4	1.866e−3	2.743e−3	0.0338

Table 8
The results of p-values of Wilcoxon's rank sum test for case study 2.

Test images	Entropy		PSNR	
	P system vs. PSO	P system vs. BF	P system vs. PSO	P system vs. BF
Lena	2.477e−4	1.534e−3	2.013e−3	0.0186
Peppers	2.521e−4	1.612e−3	2.196e−3	0.0227
Baboon	2.724e−4	1.596e−3	1.995e−3	0.0193
Hunter	2.619e−4	1.599e−3	2.531e−3	0.0231
Stanwick	2.588e−4	1.647e−3	2.016e−3	0.0198
Living room	2.593e−4	1.694e−3	2.526e−3	0.0226
House	2.731e−4	1.705e−3	2.113e−3	0.0205
Airplane	2.635e−4	1.615e−3	2.418e−3	0.0235
Butterfly	2.643e−4	1.662e−3	2.335e−3	0.0218

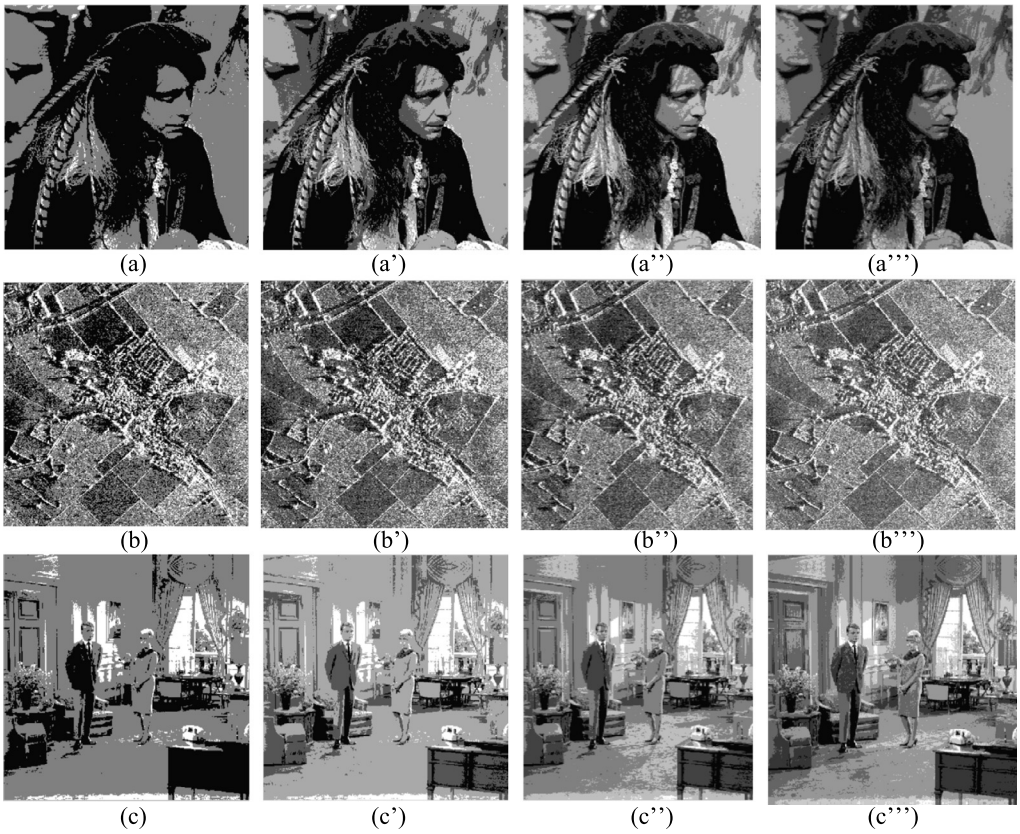

Fig. 7. The thresholded images generated by P system in case study 2 for hunter, Stanwick and living room: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

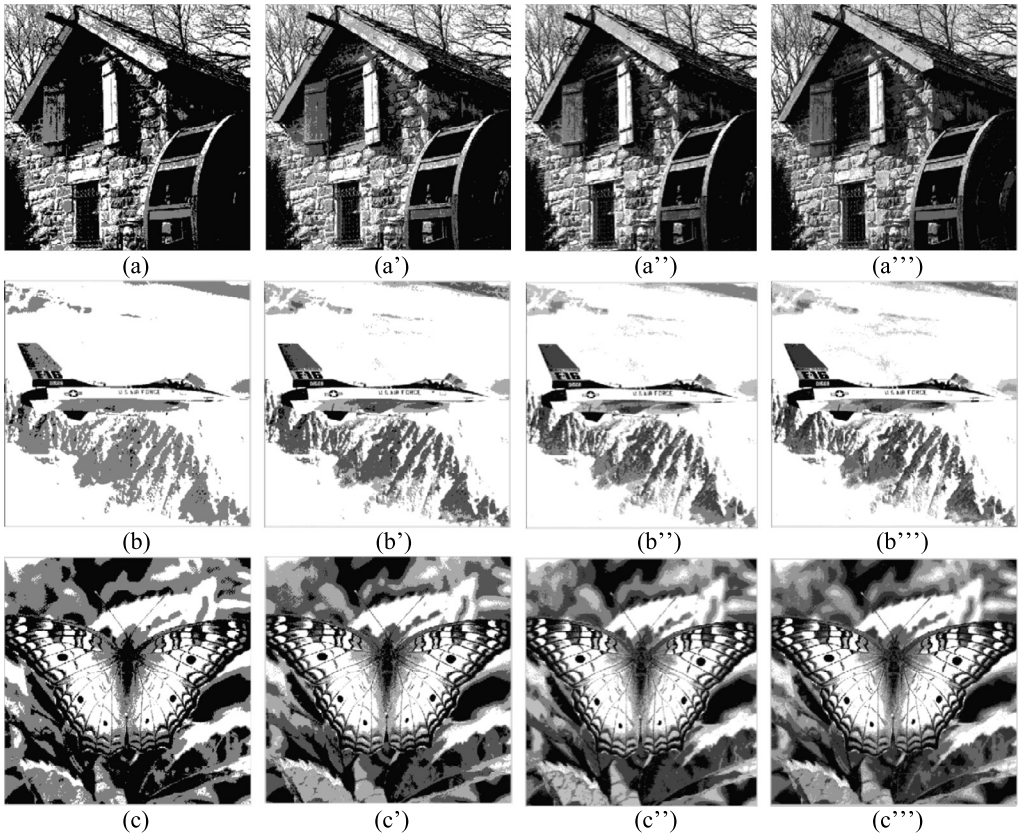

Fig. 8. The thresholded images generated by P system in case study 2 for house, airplane and butterfly: (a)–(c) represent 2-level thresholding; (a')–(c') represent 3-level thresholding; (a")–(c") represent 4-level thresholding; (a"')–(c"') represent 5-level thresholding.

performance computing, for example, on GPUs (graphics processing units) or multicore CPU system.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61170030), Chunhui Project Foundation of the Education Department of China (No. Z2012025, No. Z2012031), Research Fund of Sichuan Key Technology Research and Development Program (No. 2013GZX0155), Open Research Funds of Key Laboratory of High Performance Scientific Computing (No. SZJJ2012-002) and Intelligent Network Information Processing (No. SZJJ2012-030), China.

References

[1] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, J. Electron. Imaging 13 (1) (2004) 146–165.
[2] A. Pikaz, A. Averbuch, Digital image thresholding based on topological stable state, Pattern Recognit. 29 (5) (1996) 829–843.
[3] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Anal. Chim. Acta 59 (2004) 187–195.
[4] H. Wang, H. Peng, J. Shao, A thresholding method based on P systems for image segmentation, ICIC Express Lett. 6 (1) (2012) 221–227.
[5] N. Otsu, A threshold selection method from gray level histograms, IEEE Trans. Syst. Man Cybern. SMC-9 (1979) 62–66.
[6] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Graph. Image Process. 29 (3) (1985) 273–285.
[7] W.B. Tao, J.W. Tian, J. Liu, Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm, Pattern Recognit. Lett. 24 (2003) 3069–3078.
[8] K. Hammouche, M. Diaf, P. Siarry, A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation, Comput. Vis. Image Underst. 109 (2008) 163–175.
[9] H. Gao, W.B. Xu, J. Sun, Y.L. Tang, Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm, IEEE Trans. Instrum. Meas. 59 (4) (2010) 934–946.
[10] M. Maitra, A. Chatterjee, A hybrid cooperative comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding, Expert Syst. Appl. 34 (2008) 1341–1350.
[11] E. Zahara, S.-K.S. Fan, D.M. Tsai, Optimal multithresholding using a hybrid optimization approach, Pattern Recognit. Lett. 26 (2005) 1085–1095.
[12] W.B. Tao, H. Jin, L.M. Liu, Object segmentation using ant colony optimization algorithm and fuzzy entropy, Pattern Recognit. Lett. 28 (7) (2008) 788–796.
[13] P.D. Sathya, R. Kayalvizhi, Optimal multilevel thresholding using bacterial foraging algorithm, Expert Syst. Appl. 38 (2011) 15549–15564.
[14] B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. 13 (6) (2013) 3066–3091.
[15] S. Agrawal, R. Panda, S. Bhuyan, B.K. Panigrahi, Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, Swarm Evol. Comput. 11 (2013) 16–30.
[16] V. Osuna-Enciso, E. Cuevas, H. Sossa, A comparison of nature inspired algorithms for multi-threshold image segmentation, Expert Syst. Appl. 40 (4) (2013) 1213–1219.
[17] C. Fan, H. Ouyang, Y. Zhang, L. Xiao, Optimal multilevel thresholding using molecular kinetic theory optimization algorithm, Appl. Math. Comput. 239 (15) (2014) 391–408.
[18] S. Yin, X. Zhao, W. Wang, M. Gong, Efficient multilevel image segmentation through fuzzy entropy maximization and graph cut optimization, Pattern Recognit. 47 (9) (2014) 2894–2907.
[19] Gh. Păun, Computing with membranes, J. Comput. Syst. Sci. 61 (1) (2000) 108–143.
[20] Gh. Păun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, New York, 2010.
[21] M. Ionescu, Gh. Păun, T. Yokomori, Spiking neural P systems, Fundam. Inform. 71 (2–3) (2006) 279–308.
[22] J. Wang, L. Zhou, H. Peng, G.X. Zhang, An extended spiking neural P system for fuzzy knowledge representation, Int. J. Innov. Comput. Inf. Control 7 (7A) (2011) 3709–3724.
[23] R. Freund, Gh. Păun, M.J. Pérez-Jiménez, Tissue-like P systems with channel-states, Theor. Comput. Sci. 330 (1) (2005) 101–116.
[24] Gh. Păun, M.J. Pérez-Jiménez, Membrane computing: brief introduction, recent results and applications, BioSystem 85 (1) (2006) 11–22.
[25] H. Peng, J. Wang, M.J. Pérez-Jiménez, H. Wang, J. Shao, T. Wang, Fuzzy reasoning spiking neural P system for fault diagnosis, Inf. Sci. 235 (2013) 106–116.
[26] J. Wang, P. Shi, H. Peng, Mario J. Pérez-Jiménez, Tao Wang, Weighted fuzzy spiking neural P systems, IEEE Trans. Fuzzy Syst. 21 (2) (2013) 209–220.
[27] J. Wang, H. Peng, Adaptive fuzzy spiking neural P systems for fuzzy inference and learning, Int. J. Comput. Math. 90 (4) (2013) 857–868.
[28] T.Y. Nishida, An approximate algorithm for NP-complete optimization problems exploiting P systems, in: Proceedings of the Workshop on Uncertainty in Membrane Computing, Palma de Mallorca, 2004, pp. 185–192.
[29] L. Huang, I. Suh, A. Abraham, Dynamic mul-objective optimization based on membrane computing for control of time-varying unstable plants, Inf. Sci. 181 (11) (2011) 2370–2391.
[30] G. Zhang, J. Cheng, M. Gheorghe, Q. Meng, A hybrid approach based on different evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, Appl. Soft Comput. 13 (3) (2013) 1528–1542.
[31] H. Peng, J. Wang, M.J. Pérez-Jiménez, A. Riscos-Núñez, The framework of P systems applied to solve optimal watermarking problem, Signal Process. 101 (2014) 256–265.
[32] H. Peng, J. Wang, M.J. Pérez-Jiménez, P. Shi, A novel image thresholding method based on membrane computing and fuzzy entropy, J. Intell. Fuzzy Syst. 24 (2) (2013) 229–237.
[33] D. Huang, C. Wang, Optimal multi-level thresholding using two-stage Otsu optimization approach, Pattern Recognit. Lett. 30 (2009) 275–284.
[34] P.K. Sahoo, S. Soltani, A.K.C. Wong, A survey of thresholding techniques, IEEE Trans. Comput. Vis. Graph. Image Process. 41 (2) (1988) 233–260.
[35] R.P. Nikhil, K.P. Sankar, Entropic thresholding, IEEE Trans. Signal Process. 16 (1989) 97–108.
[36] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. IEEE Int. Conf. Neural Networks, Perth, Australia, vol. 4, 1995, pp. 1942–1948.
[37] R.C. Eberhart, J. Kennedy, Y.H. Shi, Swarm Intelligence, Morgan Kaufmann, San Mateo, CA, 2001.

Hong Peng received the B.Sc. degree and the M.E. degree in Mathematics from Sichuan Normal University, Chengdu, China in 1987 and 1990, and the Ph.D. degree in Signal and Information Processing from University of Electronic Science and Technology of China, Chengdu, China in 2010.

He was a lecturer in the Sichuan College of Science and Technology, China (1990–1999) and an associate professor in Xihua University, China (2000–2004). He was a visiting scholar in Research Group of Natural Computing, University of Seville, Spain (2011.09–2012.08). He is currently a professor in the Center for Radio Administration and Technology Development, Xihua University, China since 2005. His research interests include membrane computing, image processing, signal processing and kernel methods.

Jun Wang received the B.Sc. degree and the M.E. degree in Industry Automation from Chongqing University, China in 1988 and 1991, respectively; the Ph.D. degree in Electrical Engineering from the Southwest Jiaotong University, China in 2006.

She was a lecturer in the Sichuan College of Science and Technology, China (1991–2003) and an associate professor in Xihua University, China (1998–2003). She is currently a professor in the School of Electrical and Information Engineering, Xihua University, China since 2004. Her research interests include electrical automation, intelligent control, and membrane computing.

Mario J. Pérez-Jiménez received B.Sc. degree in Mathematics from Barcelona University, Spain in 1971 and Ph.D. degree in Mathematics from the University of Seville, Spain in 1992.

He was assistant professor in the University of Barcelona (1971–1983), and Guest Professor of the Huazhong University of Science and Technology, Wuhan, China (from 2005). Currently, he is a member of the Academia Europaea (The Academy of Europe), and a full professor in the Department of Computer Science and Artificial Intelligence, University of Seville, Spain, where he is the head of the Research Group on Natural Computing. His research interests include theory of computation, computational complexity theory, natural computing (DNA computing and membrane computing), bioinformatics, and computational modelling for systems biology and population biology. He has published twelve books in computer science and mathematics, and over 250 scientific papers in international journals (collaborating with many researchers worldwide).