

## Thresholding 2D Images with Cell-like P Systems

Hepzibah A. CHRISTINAL<sup>1,2</sup>, Daniel DÍAZ-PERNIL<sup>2</sup>,  
Miguel A. GUTIÉRREZ-NARANJO<sup>3</sup>, Mario J. PÉREZ-JIMÉNEZ<sup>3</sup>

<sup>1</sup>Karunya University, Coimbatore, Tamilnadu, India

<sup>2</sup>Research Group on Computational Topology and Applied Mathematics  
Department of Applied Mathematics, University of Sevilla, Spain

<sup>3</sup>Research Group on Natural Computing, Department of Computer Science  
and Artificial Intelligence, University of Sevilla, Spain

E-mail: [hepzi@yahoo.com](mailto:hepzi@yahoo.com), [sbdani@us.es](mailto:sbdani@us.es)  
[magutier@us.es](mailto:magutier@us.es), [marper@us.es](mailto:marper@us.es)

**Abstract.** Thresholding is the process of splitting a digital image into sets of pixels in order to make it easier to analyze. Pixels are ordered according to a scale of one of their features as brightness or color and the final image is obtained by comparing the measure of the feature with some *thresholds*. In this paper we present a family of cell-like P systems which solve the thresholding problem in linear time on the number of pixels.

### 1. Introduction

In Computer Vision [10], segmentation is the process of splitting a digital image into sets of pixels in order to make it simpler and easier to analyze. One of its main uses is the localization of objects and boundaries. Technically, the process consists on assigning a label to each pixel, in such way that pixels with the same label form a meaningful region. Among the applications of segmentation in Digital Imagery we can find the face recognition [14] or location of objects in satellite images (roads, forests, etc.) [2], but probably its main application area is medical imaging [1]. Segmentation has shown its utility in bordering tumors and other pathologies, computer-guided surgery or the study of anatomical structure, but also in techniques which are not thought to produce images but it produces positional information as electroencephalography (EEG), or electrocardiography (EKG). There exist different

techniques to segment an image, such as *clustering methods* [4], *histogram-based methods* [12], *Watershed transformation methods* [11] or *graph partitioning methods* [13].

Thresholding is a method of image segmentation. Its basic aim is to obtain a binary image from a grayscale one. The idea is to split the set of pixels into two sets (black and white) depending on its brightness and a fixed value, the *threshold*. If the brightness of the pixel is greater than the threshold, then the pixel is labeled as *object*. Otherwise, it is labeled as *background*. After labeling, a new binary image is created by coloring each pixel white or black, depending on the label.

The basic thresholding method can be generalized in a natural way. Instead of getting a binary image by labeling the original set of pixels by  $\{0, 1\}$ , we can consider a larger set of labels,  $\{1, \dots, k\}$  so we obtain a final image with  $k$  levels in a grayscale. Another natural generalization is to replace the grayscale by another scale on the features of the pixel (brightness, intensity, color, etc.).

In order to solve the thresholding problem we present a solution in the framework of Membrane Computing. Membrane Computing is a theoretical model of computation inspired by the structure and functioning of cells as living organisms able to process and generate information. The computational devices in Membrane Computing are called *P systems* [6]. Roughly speaking, a P system consists of a membrane structure, in whose compartments one places multisets of objects which evolve according to given rules. In the most investigated model, the rules are applied in a synchronous non-deterministic maximally parallel manner, but some other semantics are being explored<sup>1</sup>.

The paper is structured as follows: Next we give a formal description of the thresholding problem and present our solution in the framework of Membrane Computing. In Section 3 we first give a simple example and then we show different solutions obtained by thresholding a  $30 \times 30$  image. Finally, final remarks are given in the last section.

## 2. Formal Framework

Let us consider a set of  $n$  pixels  $A = \{a_1, \dots, a_n\}$ . An *image* on  $A$  with colors in the finite set  $H$  is a mapping  $I : A \rightarrow H$ . As usual, such image can be written as a set of pairs  $(a_i, x)$  where  $i \in \{1, \dots, n\}$  and  $x = I(a_i)$ .

The thresholding problem can be settled as follows: Given a set of pixels  $A = \{a_1, \dots, a_n\}$ , a set of input colors  $H_1 = \{h_1, \dots, h_t\}$ , an image  $I_1 : A \rightarrow H_1$ , a set of output colors  $H_2 = \{c_1, \dots, c_k\}$  with  $k \leq t$  and a mapping  $\psi : H_1 \rightarrow H_2$ , the thresholding problem consists of obtaining an image  $I_2 : A \rightarrow H_2$  such that for all  $a \in A$ ,  $I_2(a) = \psi(I_1(a))$ .

The intuition is that if two pixels  $a_1$  and  $a_2$  have colors  $h_1$  and  $h_2$  in the initial image and  $\psi(h_1) = \psi(h_2) = c_*$ , then  $a_1$  and  $a_2$  will have the same color  $c_*$  in the output image.

Without loss of generality, we will consider an enumeration of  $H_1 = \{h_1, \dots, h_t\}$ ,

---

<sup>1</sup>We refer to [7] for basic information in this area, to [9] for a comprehensive presentation and the web site [15] for the up-to-date information.

an enumeration of  $H_2 = \{c_1, \dots, c_k\}$  (such enumerations can be seen as a formalizations of the corresponding scale), a mapping  $\psi : H_1 \rightarrow H_2$  and a sequence of natural numbers (called *thresholds*  $b_1, \dots, b_{k+1}$  with  $b_1 = 1 < b_2 < \dots < b_{k+1} = t+1$  such that  $\psi^{-1}(c_1) = \{h_1, \dots, h_{b_2-1}\}$ ,  $\psi^{-1}(c_2) = \{h_{b_2}, \dots, h_{b_3-1}\}, \dots, \psi^{-1}(c_k) = \{h_{b_k}, \dots, h_t\}$ . In other words, the set of colors of  $H_1$  associated with  $c_j \in H_2$  will be a finite sequence of colors where the lower color has the index  $b_j$  and the upper one the index  $b_{j+1} - 1$ . By using this notation, we will define the length of  $c_j$  as  $f_j = b_{j+1} - b_j$ . We will use these notations in the description of our solution.

For example, let us consider the set of colors  $H_1 = \{h_1, \dots, h_{20}\}$  and  $H_2 = \{c_1, \dots, c_4\}$ . Let  $\psi$  be the mapping  $\psi : H_1 \rightarrow H_2$  such that  $\psi^{-1}(c_1) = \{h_1, h_2, h_3\}$ ,  $\psi^{-1}(c_2) = \{h_4, h_5, \dots, h_{12}\}$ ,  $\psi^{-1}(c_3) = \{h_{13}, \dots, h_{19}\}$  and  $\psi^{-1}(c_4) = \{h_{20}\}$ . In this case  $b_1 = 1$ ,  $b_2 = 4$ ,  $b_3 = 13$ ,  $b_4 = 20$  and  $b_5 = 21$  with the lengths  $f_1 = 3$ ,  $f_2 = 9$ ,  $f_3 = 7$ ,  $f_4 = 1$ .

In order to design our solution, we will consider a cell-like P system model with two polarizations and dissolution. The semantics is slightly different from that of P systems with active membranes, where polarizations are also used. The first difference is that the right-hand-side of a dissolution rule can be the empty set. This is not a big difference, since it could also be a waste object, but we prefer not to add useless objects. The main difference regards the communication rules (send-in and send-out). In our model, several objects can cross out one membrane simultaneously if they verify the left-hand-side of the rule. If all the applied rules which send objects across the same membrane keep the polarization of the membrane, then this polarization will remain unchanged. If at least one of the applied rules requires the change of polarization, then the polarization must be changed.

The membrane structure does not change along the computation and it has  $n+k+1$  membranes: the skin and  $n+k$  elementary membranes, where  $n$  is the number of pixels and  $k$  is the number of output colors. All the  $n$  membranes associated with pixels have the same label  $p$  and the membranes associated with output color will have labels  $1, \dots, k$ . The skin has label  $s$ . At the beginning all membranes have polarization 0. Formally, a P system for solving a image thresholding problem as described above is a tuple

$$\Pi = (\Gamma, E, \mu, \mathcal{E}, w_1^p, \dots, w_n^p, w_1, \dots, w_k, \mathcal{R})$$

where

1.  $\Gamma$  is the finite *alphabet of objects*,  $\Gamma = \{a_1, \dots, a_n, h_1, \dots, h_t, z_1, \dots, z_t\}$
2.  $E = \{0, +\}$  is the set of electrical charges.
3.  $\mu$  is the membrane structure,  $\mu = [[\ ]_p^0 \dots [\ ]_p^0 [\ ]_1^0 \dots [\ ]_k^0 ]_s^0$ , where we have  $n$  membranes with label  $p$ .
4.  $w_1^p, \dots, w_n^p$  are strings over  $\Gamma$  representing the multisets of objects associated with the membranes of label  $p$  at the initial configuration. For each  $i \in \{1, \dots, n\}$ ,  $w_i^p$  has only two objects,  $a_i I_1(a_i)$ , where  $a_i$  is an object which represents a pixel and  $I_1(a_i) \in \{h_1, \dots, h_t\}$  represents the color of the pixel in the image  $I_1$ .

5.  $w_1, \dots, w_k$  are strings over  $\Gamma$  representing the multisets of objects associated with the membranes of labels  $1, \dots, k$  at the initial configuration. For each  $j \in \{1, \dots, k\}$ ,  $w_j = z_{b_j}$ .
6.  $\mathcal{R}$  is a finite set of rules of the following form:

- $[z_{i+1} \rightarrow z_i]_j^0$  for  $i \in \{2, \dots, t\}$  and  $j \in \{1, \dots, k\} \cup \{s\}$   
 $[h_{i+1} \rightarrow h_i]_p^0$  for  $i \in \{2, \dots, h\}$

The sets of objects  $h_i$  and  $z_i$  can be seen as counters. These evolution rules decrease the counters in one unit in each step of computation.

- $z_1 [ ]_j^+ \rightarrow [ \lambda ]_j^0$  for  $j \in \{1, \dots, k\}$   
 $a_i [ ]_j^+ \rightarrow [ a_i ]_j^+$  for  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, k\}$

These rules send objects into the corresponding membranes. In the first set, object  $z_i$  is transformed into the empty multiset  $\lambda$  and the polarization of the membrane changes. In the second set of rules, the object  $a_i$  and the polarization remain unchanged.

- $[z_1]_j^0 \rightarrow z_{f_j} [ ]_j^+$  for  $j \in \{1, \dots, k\}$

Object  $z_1$  is sent out of the membrane by changing the polarization. The object is transformed into  $z_{f_j}$  where  $f_j$  is the length associated with the index  $j$ .

- $[h_1]_p^0 \rightarrow \lambda$

When the counter  $h_i$  reaches  $h_1$  then the membrane is dissolved.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). In one step, each object in a membrane can only be used for one rule, but any object which can participate in a rule of any form must do it, i.e., in each step we apply a maximal set of rules as described above.

## 2.1. Remarks on the Computation

The key point of the computation is the use of the sets  $z_i$  and  $h_i$  as counters. The target is a final configuration where an object  $a_i$  is placed on a membrane with label  $j$  if and only if  $I_2(a_i) = c_j$ .

Firstly, let us focus on membranes with label  $p$ . Each of them represents a pixel  $a_i$  with initial color  $h_v$ . Due to the rules  $[h_{i+1} \rightarrow h_i]_p^0$  and  $[h_1]_p^0 \rightarrow \lambda$ , the object  $a_i$  appears in the skin exactly in the  $v$ -th configuration. According to the rules  $a_i [ ]_j^+ \rightarrow [ a_i ]_j^+$ , in the next step, the object  $a_i$  is sent to a membrane  $j$  with positive polarization. It only remains to check that in the  $v$ -th configuration, there exists only one membrane  $j_0$  with positive polarization and such  $j_0$  verifies  $I_2(a_i) = c_{j_0}$ .

Each membrane  $j$  starts with an object  $z_{b_j}$ . By the application of the rules  $[z_{i+1} \rightarrow z_i]_j^0$  and  $[z_1]_j^0 \rightarrow z_{f_j} [ ]_j^+$ , membrane  $j$  changes its polarization to positive exactly in the  $b_j$ -th step. This means that in the  $b_j$ -th configuration, membrane  $j$  is positively charged and we have an object  $z_{f_j}$  in the skin.

Due to the rules  $[z_{i+1} \rightarrow z_i]_s^0$  and  $z_1 [ ]_j^+ \rightarrow [\lambda]_j^0$ , membrane  $j$  changes again its polarization at the step  $b_j + f_j = b_j + b_{j+1} - b_j = b_{j+1}$ . This means that the membrane  $j$  has positive polarizations exactly in the configurations  $\{b_j, \dots, b_{j+1} - 1\}$ . By definition of the thresholds  $\{b_1, \dots, b_{k+1}\}$ , we have that there exists exactly one  $j \in \{1, \dots, k\}$  such that  $c_j$  has positive polarization in each configuration.

Finally, if  $I_1(a_i) = h_v$  and  $h_v \in \{b_j, \dots, b_{j+1} - 1\} = \psi^{-1}(c_{j_0})$  then  $a_i$  is placed in the skin in the  $v$ -th configuration and the unique positively charged membrane in this configuration is  $j$ . In the next step,  $a_i$  is sent to membrane  $j_0$ .

### 2.2. Complexity Aspects

As seen above, membrane  $j$  turns back to polarization 0 at the step  $b_{j+1}$ . Since  $j$  belongs to  $\{1, \dots, k\}$ , the last elementary membrane to get polarization 0 is  $k$  in the step  $b_{k+1} = t + 1$ . Since the index of objects  $h_i$  is at most  $n$ , then the configuration  $C_{t+1}$  is a halting one and the computation takes exactly  $t + 1$  steps.

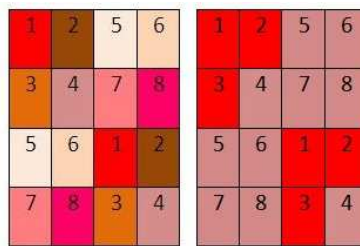
The P system uses an alphabet of  $n + 2t$  objects,  $tk + nk + t + h + k - 1$  rules and the membrane structure has  $n + k + 1$  membranes. In the worst case, we have as many output colors as input ones and as many input colors as pixels, *i.e.*,  $k = t = n$ . This means that the number of steps and the alphabet is linear and the number of rules is quadratic in the number of pixels.

### 3. Examples

In this section we will show several examples of our solution to the thresholding problem. We start with an example with 16 pixels.

Let us consider the image in Fig. 1. The number of pixels in the input image is  $n = 16$ , the number of initial colors is  $t = 8$ ,  $H_1 = \{h_1, \dots, h_8\}$  and the number of final colors is  $k = 2$ ,  $H_2 = \{c_1, c_2\}$ . We will also consider the mapping  $\psi : H_1 \rightarrow H_2$  with the threshold 4

$$\psi(h_i) = \begin{cases} c_1 & \text{if } i < 4 \\ c_2 & \text{if } 4 \leq i \end{cases}$$



**Fig. 1.** The input and output images of the first example in Section 3.

In the initial configuration (see Fig. 2), there are  $n + k = 16 + 2$  membranes inside the skin membrane. Each of the sixteen membranes with label  $p$  contains an object

codifying a pixel  $a_i$  and another object codifying the color  $h_j$  of this pixel. Moreover, we have other two membranes encoding the output colors. The first one has the label 1 and contains an object  $z_1$ , representing the first object of the first interval of the two by which the alphabet of colors have been segmented by the thresholds. The second membrane has the label 2 and contains object  $z_4$  codifying the minimal value of the second interval of the input colors.

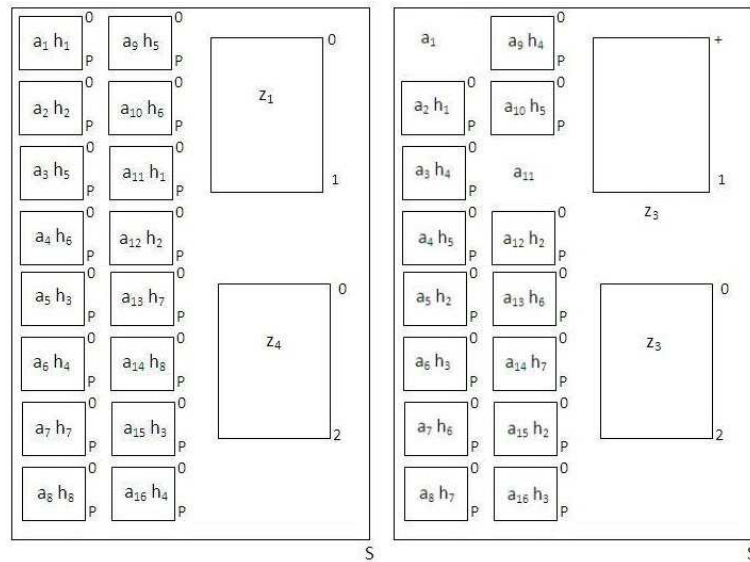


Fig. 2. Configuration  $C_0$  and  $C_1$  of the example.

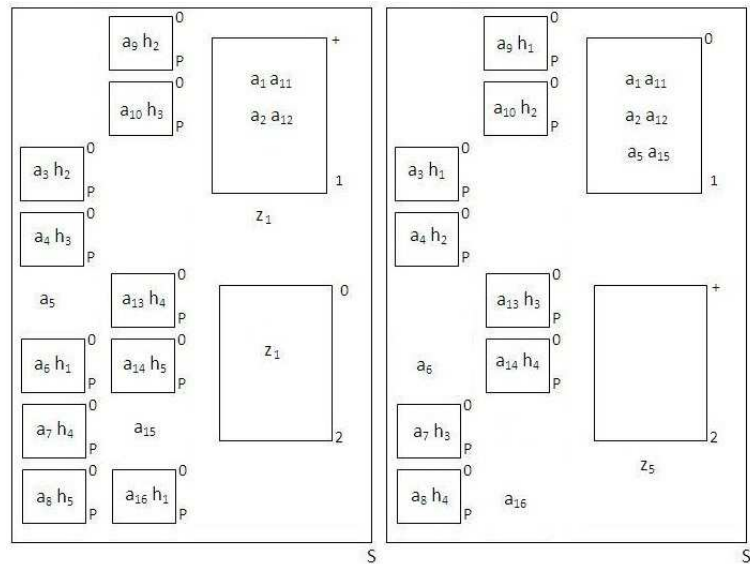


Fig. 3. Configurations  $C_3$  and  $C_4$ .

In the first step, two of the sixteen membranes got dissolved by sending the objects  $a_1, a_{11}$  to the skin membrane. In the membrane with label 1,  $z_3$  is sent out to the skin membrane where the index 3 represents the length of the first interval  $\{h_1, h_2, h_3\}$  and its charge of this membrane has been changed to positive.

In the following step, membranes with objects  $a_2$  and  $a_{12}$  got dissolved and send the objects  $a_2$  and  $a_{12}$  to the skin membrane. The objects  $a_1$  and  $a_{11}$  in the skin membrane are sent into the membrane having label 1 with charge positive. This encodes that pixels  $a_2$  and  $a_{12}$  will have color  $c_1$  in the final image.

The computation goes on and in configuration 3, the counter  $z_i$  in the skin arrives to  $z_1$ . In the next step, this object goes into membrane 1 by changing its configuration. Simultaneously, object  $z_1$  goes out from membrane 2 changing its polarization into positive (see Fig. 3).

Finally, the output configuration is obtained after  $t + 1 = 8 + 1$  steps. In the last step, objects  $a_8$  and  $a_{14}$  are sent into the membrane with label 2 and the membrane returns to polarization 0. In the halting configuration we have two segmentations which are given by membrane with label 1 and membrane with label 2. Membrane with label 1 gives the objects codifying pixels having the colors from the first interval and membrane with label 2 gives the objects codifying the pixels with colors from second interval.

### 3.1. A more complex example

In this section, we show the results obtained by application of our systems with an image of size  $30 \times 30$  (see Fig. 4) with input colors  $H_1 = \{h_1, \dots, h_{30}\}$ . From the same image and set of output colors we can get different outputs by changing the thresholds and output colors.



Fig. 4. An example with  $30 \times 30$  pixels.



In Fig. 5 we show the output image where  $H_2 = \{c_1, \dots, c_6\}$  and  $\psi : H_1 \rightarrow H_2$  such that  $\psi^{-1}(c_i) = \{h_{5(i-1)+1}, \dots, h_{5i}\}$ , for  $i = 1, \dots, 6$ .

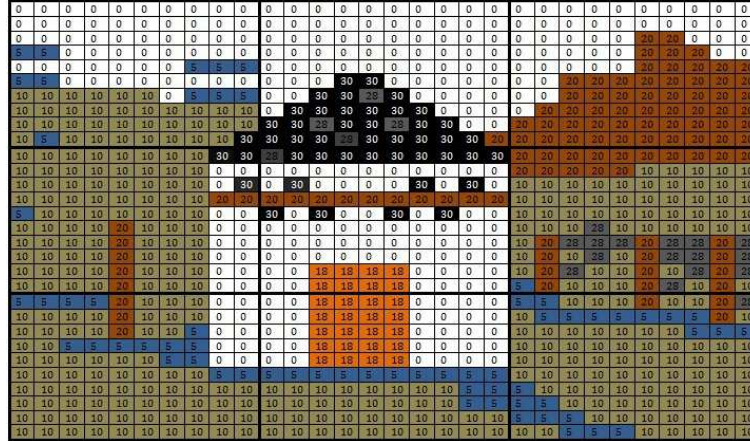


Fig. 5. Thresholding of the image in Fig. 4 with intervals of length 5.

In the second output image (Fig. 6) we have taken  $H_2 = \{c_1, \dots, c_3\}$  and  $\psi : H_1 \rightarrow H_2$  such that  $\psi^{-1}(c_i) = \{h_{10(i-1)+1}, \dots, h_{10i}\}$ , for  $i = 1, \dots, 3$ .

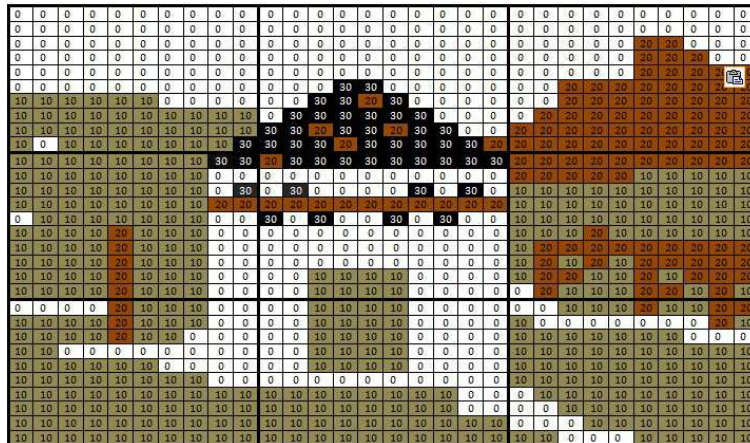


Fig. 6. Thresholding of the image in Fig. 4 with intervals of length 10.

### 4. Final Remarks

Segmentation in Computer Vision in general and the thresholding method in particular are research areas where the advances in new knowledge can be rapidly transferred to real-life improvements. Some of its features these problems is the high degree of parallelization and the possibility of represent the information in a simple way, for



example, as multisets. This fact opens a door to apply in this area a broad set of techniques from Natural Computing.

In this paper we present an algorithm for the thresholding problem in the framework of Membrane Computing and have shown that by using the massive parallelism, the solution can be reached in linear time in the number of pixels of the input. Undoubtedly, the next step is to bring this theoretical result into realistic parallel implementations. Since P systems have not been implemented *in vivo* nor *in vitro* yet, the next step is to consider one of the current research lines for implementing P systems with parallel hardware as [3] or [5]. A realistic implementation of Membrane Computing techniques in these architectures will represent a new horizon for many problems in Digital Imagery.

**Acknowledgements.** DDP, MAGN and MJPJ acknowledge the support of the projects TIN2008-04487-E and TIN-2009-13192 of the Ministerio de Ciencia e Innovación of Spain and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200. HAC acknowledge the support of the project MTM2009-12716 of the Ministerio español de Educación y Ciencia, the project PO6-TIC-02268 of Excellence of Junta de Andalucía, and the “Computational Topology and Applied Mathematics” PAICYT research group FQM-296.

## References

- [1] CAMPADDELLI P., CASIRAGHI E., ESPOSITO A., *Liver Segmentation from Computed Tomography Scans: A Survey and a new Algorithm*, Artificial Intelligence in Medicine, **45**, 2–3 (2009), pp. 185–196.
- [2] GAMANYA R., MAEYER P. D., DAPPER M. D., *An Automated Satellite Image Classification Design Using Object-oriented Segmentation Algorithms: A Move Towards Standardization*, Expert Systems with Applications, **32**, 2 (2007), pp. 616–624.
- [3] GARCÍA-QUISMONDO M., GUTIÉRREZ-ESCUADERO R., PÉREZ-HURTADO I., PÉREZ-JIMÉNEZ M. J., RISCOS-NÚÑEZ A., *An Overview of P-lingua 2.0.*, in Păun et al. [8], pp. 264–288.
- [4] LI X., ZHANG T., QU Z., *Image Segmentation Using Fuzzy Clustering with Spatial Constraints Based on Markov Random Field via Bayesian Theory*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, **91-A**, 3 (2008), pp. 723–729.
- [5] NGUYEN V., KEARNEY D., GIOIOSA G. *A Region-oriented Hardware Implementation for Membrane Computing Applications*, in Păun et al. [8], pp. 385–409.
- [6] PĂUN GH. *Computing with Membranes*, Journal of Computer and System Sciences, **61**, 1 (2000), pp. 108–143.
- [7] PĂUN, GH. *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, Germany, 2002.
- [8] PĂUN GH., PÉREZ-JIMÉNEZ M. J., RISCOS-NÚÑEZ A., ROZENBERG G., SALOMAA A., Eds., *Membrane Computing, 10th International Workshop, WMC 2009, Curtea*

- de Arges, Romania, August 24-27, 2009. Revised Selected and Invited Papers* (2010), vol. 5957 of *Lecture Notes in Computer Science*, Springer.
- [9] PĂUN GH., ROZENBERG G., SALOMAA A., Eds., *The Oxford Handbook of Membrane Computing*, Oxford University Press, 2010.
  - [10] SHAPIRO L. G., STOCKMAN G. C., *Computer Vision*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
  - [11] TARABALKA Y., CHANUSSOT J., BENEDIKTSSON J. A., *Segmentation and Classification of Hyperspectral Images Using Watershed Transformation*, *Pattern Recognition*, **43**, 7 (2010), pp. 2367–2379.
  - [12] TOBIAS O. J., SEARA R. *Image Segmentation by Histogram Thresholding Using Fuzzy Sets*, *IEEE Transactions on Image Processing*, **11**, 12 (2002), pp. 1457–1465.
  - [13] YUAN X., SITU N., ZOURIDAKIS G., *A Narrow Band Graph Partitioning Method for Skin Lesion Segmentation*, *Pattern Recognition*, **42**, 6 (2009), pp. 1017–1028.
  - [14] ZHAO W., CHELLAPPA R., PHILLIPS P. J., ROSENFELD A., *Face Recognition: A Literature Survey*, *ACM Computing Surveys*, **35**, 4 (2003), pp. 399–458.
  - [15] P system web page: <http://ppage.psyste.ms.eu/>