A Bio-inspired Software for Segmenting Digital Images

Daniel Díaz-Pernil, Helena Molina-Abril, Pedro Real
Research Group on Computational Topology and Applied Mathematics
Universidad de Sevilla
Spain
sbdani@us.es, habril@us.es, real@us.es

Miguel A. Gutiérrez-Naranjo
Research group on Natural Computing
Universidad de Sevilla
Spain
magutier@us.es

***Abstract*—Segmentation in computer vision refers to the process of partitioning a digital image into multiple segments (sets of pixels). It has several features which make it suitable for techniques inspired by nature. It can be parallelized, locally solved and the input data can be easily encoded by bio-inspired representations. In this paper, we present a new software for performing a segmentation of 2D digital images based on Membrane Computing techniques.**

I. Introduction

Nature is a big inspiration source for designing solutions to a broad panoply of problems. Natural Computing studies computational paradigms inspired from various well known natural phenomena in physics, chemistry and biology[1]. It abstracts the way in which nature acts, conceiving new computing models. The field is growing rapidly and there are many open research lines based on nature. Among them, *Cellular Automata* [2] conceived by Ulam and von Newman as a spatial distribution of cells able to reproduce the behavior of complex systems; *Genetic algorithms* introduced by J. Holland [3] which is inspired by natural evolution and selection in order to find a good solution in a large set of feasible candidate solutions; *Neural Networks* introduced by W.S. McCulloch and W. Pitts [4] it is based on the interconnections of neurons in the brain; *DNA-based* molecular computing, that was born when L. Adleman [5] published a solution to an instance of the Hamiltonian path problem by manipulating DNA strands in a lab; *Swarm Intelligence* [6] based on the behavior and communication of mobile organisms as ants or bees acting in the environment; *Artificial Immune Systems* [7] based on the natural immune system of biological organisms; *Amorphous computing* [8] inspired from the development of morphogenesis in biological organisms or *Membrane Computing* [9], [10] based on the functioning and morphology of living cells and tissues.

All these computational paradigms have in common the use of an alternative way of encoding the information, adapted to the bio-inspired substrate and the use of intrinsic parallelism of natural processes.

In this paper we present a bio-inspired software for solving the Segmentation Problem in Digital Imagery. Segmentation in computer vision (see [11]), refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

Segmentation in Digital Imagery has several features which make it suitable for techniques inspired by nature. One of them is that it can be parallelized and locally solved. Regardless how large is the picture, the segmentation process can be performed in parallel in different local areas of it. Another interesting feature is that the basic necessary information can be easily encoded by bio-inspired representations.

In the literature, one can find several attempts for bridging problems from Digital Imagery with Natural Computing as the works by K.G. Subramanian *et al.* [12], [13] or the work by Chao and Nakayama where Natural Computing and Algebraic Topology are linked by using Neural Networks [14] (extended Kohonen mapping). In this paper, we will use an information encoding and techniques borrowed from 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 are called *P systems* [9]. Roughly speaking, a P system consists of a membrane structure, in the compartments of which one places multi sets of objects which evolve according to given rules. In the most extended model, the rules are applied in a synchronous non-deterministic maximally parallel manner, but some other semantics are being explored [2].

According to their architecture, these models can be split into two sets: P systems such that their membrane structure is a tree-like graph, called *cell-like P systems* and P systems whose membrane structure is a general graph. In this second group we can find *tissue-like P systems* and *spiking neural P systems*. This paper is devoted to the second approach: tissue-like P

[1] An introduction on Natural Computing can be found in [1].

[2] We refer to [10] for basic information in this area, to [15] for a comprehensive presentation and the web site [16] for the up-to-date information.

978-1-4244-6439-5/10/$26.00 ©2010 IEEE

systems. In [17], [18], [19], Christinal *et al.* started a new bio-inspired research line where the power and efficiency of tissue-like P systems [20], [21] were applied to topological processes for 2D and 3D digital images. In this paper, we present a new software tool to segment 2D digital images based in the works of Christinal *et al.* in [17], [19]. This tool simulates the behavior of the tissue-like P systems described in [17] and allows us to work with images in JPG format.

Simulation of different variants of P systems have been widely studied in the last years. Since there do not exist implementations of P systems *in vivo* nor *in vitro*, the natural way to explore the behavior of designed P systems is to simulate it in conventional computers. A short description of some of these simulators can be found in [22], [23]. In [24], a first simulator for tissue-like P systems was presented. Currently, a big effort is being developed in the *P-lingua project* [25], by combining an efficient simulation engine with an *ad-hoc* programming language.

The paper is organized as follows: firstly, we present our bio-inspired formal framework. Next, we present the family of tissue-like P systems used to obtain a segmentation of a 2D digital image. In Section IV we introduce our software tool and illustrate its use with some examples. Finally, some conclusions are presented.

II. Formal framework: tissue-like P systems

Tissue-like P systems were presented by Martín–Vide *et al.* in [26]. They have two biological inspirations (see [27]): intercellular communication and cooperation between neurons. The common mathematical model of these two mechanisms is a network of processors dealing with symbols and communicating these symbols along channels specified in advance.

The main features of this model, from the computational point of view, are that cells have not polarization (the contrary holds in the cell-like model of P systems, see [10]) and the membrane structure is a general graph.

Formally, a *tissue-like P system* with input of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, \ldots, w_q, \mathcal{R}, i_\Pi, o_\Pi),$$

where

1) Γ is a finite *alphabet*, whose symbols will be called *objects*, $\Sigma (\subset \Gamma)$ is the input alphabet, $\mathcal{E} \subseteq \Gamma$ (the objects in the environment),
2) $w_1, \ldots, w_q$ are strings over Γ representing the multi sets of objects associated with the cells at the initial configuration,
3) $\mathcal{R}$ is a finite set of communication rules of the following form: $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \ldots, q\}, i \neq j$, $u, v \in \Gamma^*$,
4) $i_\Pi, o_\Pi \in \{0, 1, 2, \ldots, q\}$.

A tissue-like P system of degree $q \geq 1$ can be seen as a set of q cells (each one consisting of an elementary membrane) labelled by $1, 2, \ldots, q$. We will use 0 to refer to the label of the environment, i_Π denotes the input region and o_Π denotes the output region (which can be the region inside a cell or the environment).

The strings $w_1, \ldots, w_q$ describe the multi sets of objects placed in the q cells of the system. We interpret that $\mathcal{E} \subseteq \Gamma$ is the set of objects placed in the environment, each one of them available in an arbitrary large amount of copies.

The communication rule $(i, u/v, j)$ can be applied over two cells labelled by i and j such that u is contained in cell i and v is contained in cell j. The application of this rule means that the objects of the multi sets represented by u and v are interchanged between the two cells. Note that if either $i = 0$ or $j = 0$ then the objects are interchanged between a cell and the environment.

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 (non-deterministically chosen when there are several possibilities), 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.

III. Segmenting digital images in constant time

In this section, we segment images based on edge-based segmentation. Edge-based segmentation finds boundaries of regions which are sufficiently different from each other. We define two family of tissue-like P systems for edge-based segmentation, one of them to segment 2D images and after, we adapt these systems to segment 3D images.

There exist different techniques to segment an image. Some of them are clustering [28], histogram-based methods [29], watershed transformation methods [30], graph partitioning [31] and image pyramids methods [32]. Some of the practical applications of image segmentation are medical imaging [28], objects classification and face recognition [33].

A. A family of tissue-like P systems for a 2D segmentation

We can divide the image in multiple pixels forming a network of points of $\mathbb{N}^2$. Let $\mathcal{C} \subseteq \mathbb{N}$ be the set of all colors in the given 2D image and they are in a certain order. Moreover, we will suppose each pixel is associated with a color of the image. Then we can codify the pixel (i, j) with associated color $a \in \mathcal{C}$ by the object a_{ij}.

The following question is how to decide which pixel is adjacent to a given one. We have decided to use in this paper the 4-adjacency [34], [35]. In this case, each pixel has four (horizontal and vertical) neighbors. The extension to different adjacencies is straightforward.

At this point, we want to find the border cells of the different color regions that are within the image. Then, for each image with $n \times m$ pixels ($n, m \in \mathbb{N}$) we will construct a tissue-like P system whose input is given by the objects a_{ij} codifying a pixel, with $a \in \mathcal{C}$. The output of the system is given by the objects that appear in the output cell when the system stops.

Based on that, we can define a family of tissue-like P systems to perform an edge-based segmentation to a 2D image.

For each $n, m \in \mathbb{N}$ we consider the tissue-like P system $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$, defined as follows:

(a) $\Gamma = \Sigma \cup \{\bar{a}_{ij} : 1 \leq i \leq n,\ 1 \leq j \leq m\} \cup \{A_{ij} : 1 \leq i \leq n,\ 1 \leq j \leq m,\ A \in \mathcal{C}\}$, $\Sigma = \{a_{ij} : a \in \mathcal{C},\ 1 \leq i \leq n,\ 1 \leq j \leq m\}$, $\mathcal{E} = \Gamma - \Sigma$,

(b) $w_1 = w_2 = \emptyset$,

(c) R is the following set of communication rules:

1. $(1, a_{ij}b_{kl}/\bar{a}_{ij}A_{ij}b_{kl}, 0)$, for $a, b \in \mathcal{C},\ a < b,\ 1 \leq i, k \leq n$ and $1 \leq j, l \leq m$.
   These rules are used when the image has two adjacent pixels with different associated colors (border pixels). Then, the pixel with less associated color is marked and the system brings from the environment an object representing this marked pixel (edge pixel).

2. $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i+1j+1}b_{i+1j}\ /\ \bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i+1j+1}$ $b_{i+1j}, 0)$ for $a, b \in \mathcal{C}, a < b,\ 1 \leq i \leq n-1,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{i-1j}\bar{a}_{i-1j+1}b_{ij+1}/\bar{a}_{ij}\bar{a}_{i-1j}A_{i-1j}\bar{a}_{i-1j+1}$ $b_{ij+1}, 0)$ for $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i-1j+1}b_{i-1j}/\bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i-1j+1}$ $b_{i-1j}, 0)$ for $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{i+1j}\bar{a}_{i+1j+1}b_{ij+1}/\bar{a}_{ij}\bar{a}_{i+1j}A_{i+1j}\bar{a}_{i+1j+1}$ $b_{ij+1}, 0)$ for $a, b \in \mathcal{C}, a < b, 1 \leq i \leq n-1,\ 1 \leq j \leq m-1$.
   These rules mark with a bar the pixels which are adjacent to two pixels of the same color which were marked before, but with the condition that the marked objects are adjacent to another pixel with a different color. Moreover, an edge object representing the last marked pixel is brought from the environment.

3. $(1, A_{ij}/\lambda, 2)$, for $1 \leq i \leq n,\ 1 \leq j \leq m$.
   This rule is used to send the edge pixels to the output cell.

d) $i_\Pi = 1$, $o_\Pi = 2$.

An overview of the Computation: Rules of type 1, in a parallel manner, identify the border pixels and bring the edge pixels from the environment. These rules need 4 steps to mark all the border pixels. From the second step, the rules of type 2 can be used with the first rules at the same time. So, in 4 more steps we can bring from the environment the edge pixels adjacent to two border pixels (as explained above). The P system can apply the first two types of rules simultaneously in some configurations, but it always applies the same number of these two types of rules because this number is given by the edge pixels (we consider 4-adjacency). Finally, the third type of rules are applied in the following step on the edge pixels appearing in the cell. So, with one step more we will have all the edge pixels in the output cells. Thus, we need only 9 steps to obtain an edge-based segmentation for an $n \times m$ image. Therefore, we can conclude that the problem of edge-segmentation in 2D images is solved in constant time with respect to the number of steps of any computation.

IV. A Software Tool

In [17], preliminary segmentation results were obtained using the *tissue simulator* developed in [24]. Such a *tissue simulator* follows one of the common features of the first generation of simulators for cell–like P systems, that is the lack of efficiency in favor of expressivity. Therefore, experiments performed using this tool were extremely slow, and could only use synthetic images of at most 30×30 pixels.

In order to perform experiments with bigger images, a new software tool has been developed. This software makes possible the obtention of segmented real images that have been partitioned following a membrane computing approach.

For optimal effectiveness and flexibility, the object oriented C++ programming language has been used in the implementation.

The software input consists of a digital 2D image. The image format can be any of the most common raster image formats (jpg, png, gif,...). Such image is provided to the P system as a set of objects a_{ij} where (i, j) covers the $n \times m$ array of pixels and a belongs to $\mathcal{C}$, the set of colors.

At the beginning, the input cell contains objects a_{ij} codifying the colored pixels from an 2D image (where a is the color value of the pixel, and i, j its coordinates).

As an output, the software provides a black image (with the same format as the input image), where the detected border pixels are white. In other words, pixels belonging to the output cell of the system will be considered white, and printed out in the output image.

Some experiments and results are shown in Figure 1 and Figure 2. Figure 1 shows a geometrical 340×340 picture together with the output image of the software. Due to the sharp boundaries within the image, a precise segmentation result is obtained. In Figure 2 a more complex image is shown. It is a 600×600 medical image that corresponds to Computed Tomography (CT) human lungs. In that case, the software tool needs to preprocess the initial image before segment it. This fact is due to the simplicity of the segmentation algorithm implemented here, and future improvements of it are planned. In this case, the preprocessing transforms an image with many gray levels into a black and white image (middle image in Figure 2). The output of the system is show on the right picture. The segmentation process took 0.33 sec.

V. Conclusions and Future Work

Segmentation has features which makes it suitable for techniques from Natural Computing. Local solution or the parallelism of the process can be studied from a theoretical point of view, but for an effective application of these techniques to the real world, it is necessary to have an appropriate software tool.

This paper represents a new step, since it represents a big improvement with respect to the simulator presented in

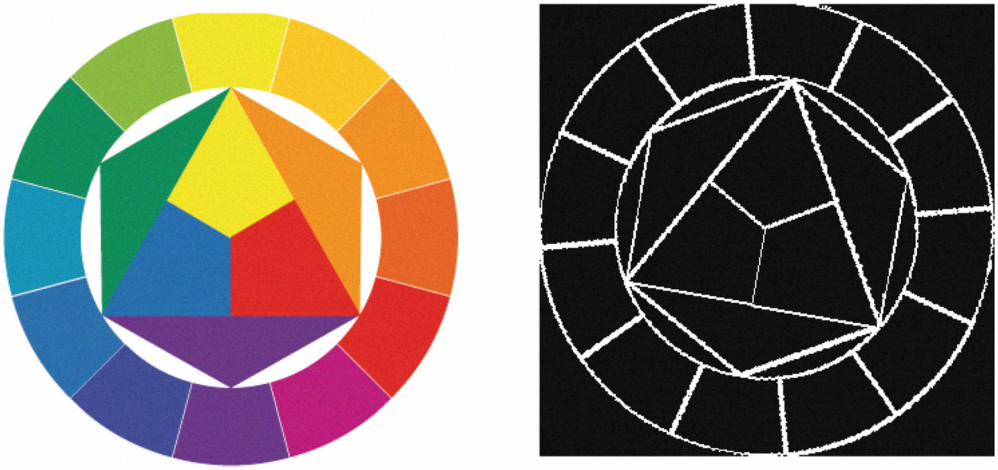

Fig. 1. Segmentation result of a 340 × 340 pixels image computed in 0.105 s.

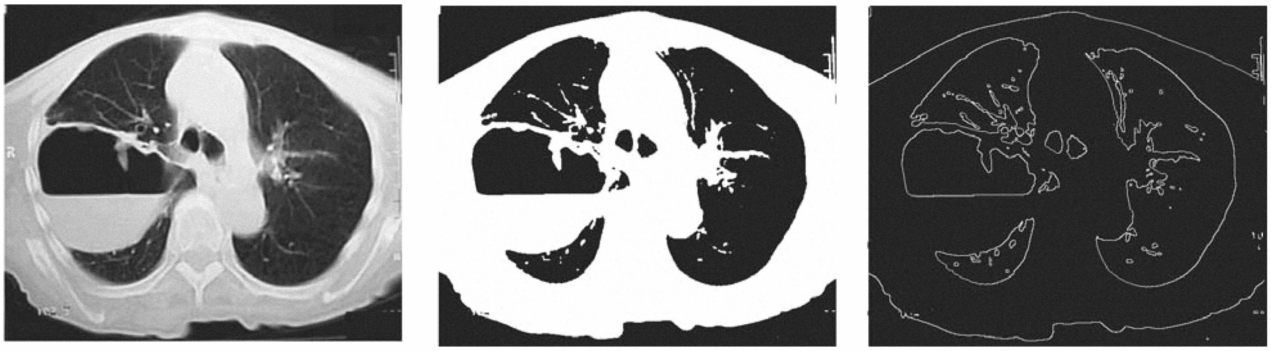

Fig. 2. Segmentation result of a 600 × 600 CT image of human lungs, computed in 0.33 s. On the left, the initial image, on the middle the binarized image, and on the right the obtained segmentation result.

[24]. The presented software tool is able to deal with images of reasonable size and can become a helping tool for the treatment of digital images, as the second example of a medical image shows.

The software can be improved and several research lines are open. One of them is to study the influence of the type of adjacency (4 or 8) on the result. The possibility of including preprocessing features in our tool be also studied. As a final remark, we will consider to adapt this software to a parallel hardware architecture and exploit in a realistic way the intrinsic parallelism of membrane computing methods.

Acknowledgement

DDP and MAGN 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. PR and HMA 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] L. Kari and G. Rozenberg, "The many facets of natural computing," *Communications of the ACM*, vol. 51, no. 10, pp. 72–83, 2008.

[2] J. von Neumann, *Theory of Self-Reproducing Automata*. Urbana, IL: Univ. of Illinois Press, 1966.

[3] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.

[4] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Neurocomputing: foundations of research*, pp. 15–27, 1988.

[5] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.

[6] A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Wiley and Sons, 2005.

[7] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, 2002.

[8] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. K. Jr., R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss, "Amorphous computing," *Communications of the ACM*, vol. 43, no. 5, pp. 74–82, 2000.

[9] Gh. Păun, "Computing with membranes," *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.

[10] ——, *Membrane Computing. An Introduction*. Berlin, Germany: Springer-Verlag, 2002.

[11] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[12] R. Ceterchi, M. Mutyam, Gh. Păun, and K. G. Subramanian, "Array-rewriting P systems," *Natural Computing*, vol. 2, no. 3, pp. 229–249, 2003.

[13] R. Ceterchi, R. Gramatovici, N. Jonoska, and K. G. Subramanian,

"Tissue-like P systems with active membranes for picture generation," *Fundamenta Informaticae*, vol. 56, no. 4, pp. 311–328, 2003.

[14] J. Chao and J. Nakayama, "Cubical singular simplex model for 3D objects and fast computation of homology groups," in *13th International Conference on Pattern Recognition (ICPR'96)*, vol. IV. Los Alamitos, CA, USA: IEEE Computer Society, 1996, pp. 190–194.

[15] Gh. Păun, G. Rozenberg, and A. Salomaa, Eds., *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.

[16] "P system web page. http://ppage.psystems.eu/."

[17] H. A. Christinal, D. Díaz-Pernil, and P. Real, "Segmentation in 2D and 3D image using tissue-like P system," in *CIARP*, ser. Lecture Notes in Computer Science, E. Bayro-Corrochano and J.-O. Eklundh, Eds., vol. 5856. Springer, 2009, pp. 169–176.

[18] ——, "Using membrane computing for obtaining homology groups of binary 2D digital images," in *IWCIA*, ser. Lecture Notes in Computer Science, P. Wiederhold and R. P. Barneva, Eds., vol. 5852. Springer, 2009, pp. 383–396.

[19] ——, "P systems and computational algebraic topology," *Journal of Mathematical and Computer Modelling*, 2010, in press.

[20] D. Díaz-Pernil, M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "A uniform family of tissue P systems with cell division solving 3-COL in a linear time," *Theoretical Computer Science*, vol. 404, no. 1-2, pp. 76–87, 2008.

[21] D. Díaz-Pernil, M. J. Pérez-Jiménez, and A. Romero, "Efficient simulation of tissue-like P systems by transition cell-like P systems," *Natural Computing*, vol. 8, pp. 797–806, 2009.

[22] M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Available membrane computing software," in *Applications of Membrane Computing*, ser. Natural Computing Series, G. Ciobanu, M. J. Pérez-Jiménez, and Gh. Păun, Eds. Springer, 2006, pp. 411–436.

[23] D. Díaz-Pernil, C. Graciani, M. A. Gutiérrez-Naranjo, I. Pérez-Hurtado, and M. Mario J. Pérez-Jiménez, "Software for P systems," in [15], pp. 437–454.

[24] R. Borrego-Ropero, D. Díaz-Pernil, and M. J. Pérez-Jiménez, "Tissue simulator: A graphical tool for tissue P systems," in *Proceedings of the International Workshop Automata for Cellular and Molecular Computing*, G. Vaszil, Ed. Budapest, Hungary: MTA SZTAKI, August 2007, pp. 23–34, satellite of the 16th International Symposium on Fundamentals of Computational Theory.

[25] D. Díaz-Pernil, I. Pérez-Hurtado, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "A P-lingua programming environment for membrane computing," in *Workshop on Membrane Computing*, ser. Lecture Notes in Computer Science, D. W. Corne, P. Frisco, Gh. Păun, G. Rozenberg, and A. Salomaa, Eds., vol. 5391. Springer, 2008, pp. 187–203.

[26] C. Martín-Vide, J. Pazos, Gh. Păun, and A. Rodríguez-Patón, "A new class of symbolic abstract neural nets: Tissue P systems," in *COCOON '02: Proceedings of the 8th Annual International Conference on Computing and Combinatorics*. London, UK: Springer-Verlag, 2002, pp. 290–299.

[27] C. Martín-Vide, Gh. Păun, J. Pazos, and A. Rodríguez-Patón, "Tissue P systems," *Theoretical Computer Science*, vol. 296, no. 2, pp. 295–326, 2003.

[28] D. Wang, H. Lu, J. Zhang, and J. Z. Liang, "A knowledge-based fuzzy clustering method with adaptation penalty for bone segmentation of ct images," in *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 6488–6491.

[29] O. J. Tobias and R. Seara, "Image segmentation by histogram thresholding using fuzzy sets," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1457–1465, 2002.

[30] H. Yazid and H. Arof, "Image segmentation using watershed transformation for facial expression recognition," in *IFMBE Proceedings, 4th Kuala Lumpur International Conference on Biomedical Engineering*, 2008, pp. 575–578.

[31] X. Yuan, N. Situ, and G. Zouridakis, "A narrow band graph partitioning method for skin lesion segmentation," *Pattern Recognition*, vol. 42, no. 6, pp. 1017–1028, 2009.

[32] W. G. Kropatsch, Y. Haxhimusa, and A. Ion, "Multiresolution image segmentations in graph pyramids," in *Applied Graph Theory in Computer Vision and Pattern Recognition*, ser. Studies in Computational Intelligence, A. Kandel, H. Bunke, and M. Last, Eds. Springer, 2007, vol. 52, pp. 3–41.

[33] S.-H. Kim, H.-G. Kim, and K.-H. Tchah, "Object oriented face detection using colour transformation and range segmentation," *Electronics Letters, IEEE*, no. 34, pp. 979–980, 1998.

[34] A. Rosenfeld, "Digital topology," *American Mathematical Monthly*, vol. 86, no. 8, pp. 621–630, October 1979.

[35] ——, "Connectivity in digital pictures," *Journal for Association of Computing Machinery*, vol. 17, no. 1, pp. 146–160, 1970.