REVIEW PAPER

Membrane computing and image processing: a short survey

Daniel Díaz-Pernil[1] · Miguel A. Gutiérrez-Naranjo[2] · Hong Peng[3]

Received: 22 June 2018 / Accepted: 14 December 2018 / Published online: 4 February 2019

Abstract

Membrane computing is a well-known research area in computer science inspired by the organization and behavior of live cells and tissues. Their computational devices, called P systems, work in parallel and distributed mode and the information is encoded by multisets in a localized manner. All these features make P systems appropriate for dealing with digital images. In this paper, some of the open research lines in the area are presented, focusing on segmentation problems, skeletonization and algebraic-topological aspects of the images. An extensive bibliography about the application of membrane computing to the study of digital images is also provided.

Keywords Membrane computing · P systems · Image processing

1 Introduction

Computer vision [153] is one of the most promising challenges for computer scientists in the coming years.[1] This research area is placed in the interplay of many disciplines such as artificial intelligence, pattern recognition, signal processing, neurobiology, psychology or image processing, among others. It concerns the automated processing of images from the real world to extract and interpret information on a real-time basis.

Roughly speaking, a digital image is a two-dimensional surface where each point is associated with a set of features as brightness or color. It is natural to consider only a discrete version of the definition, since only a finite amount of pixels placed in a lattice of integer coordinates is usually taken. The set of features can also have a finite amount of values (e.g., values in a range $\{0, \dots, 255\}$ for colors). Such discrete amount of data makes digital images appropriate for dealing with membrane computing techniques, but other features can be also considered. One of them is that the treatment of the image can be parallelized and locally solved. Regardless of how large the picture is, many of the processes can be performed in parallel in different local areas of the image. Another interesting feature is that the local information needed for an image transformation can also be easily encapsulated in a membrane and represented as a multiset of objects (for example, the new color of a pixel only depends, in some cases, on the color of the surrounding pixels and such discrete information can be encoded with membrane computing techniques). Such features, together with the maximal parallelism, have encouraged many researchers to explore the links between membrane computing and digital image processing.

Formally, a 2D digital image $\mathcal{I}$ with size $n \times m$ ($n, m \in \mathbb{N}$) is a rectangular net of objects (i, j) called pixels (voxels in 3D images), with $1 \leq i \leq n$ and $1 \leq j \leq m$ (in general, any subset of the integer plane $\mathbb{Z} \times \mathbb{Z}$ can be chosen). Let $\mathcal{C}$, the alphabet of colors of $\mathcal{I}$, be the ordered set of all colors in $\mathcal{I}$. We define the size of $\mathcal{C}$, $|\mathcal{C}|$, as the number of colors of this alphabet. Moreover, we will assume that each pixel of $\mathcal{I}$ is associated with a color of $\mathcal{C}$. So, we encode the pixel (i, j) with the associated color $a \in \mathcal{C}$ as the object a_{ij}. Therefore, the image I can be codified as the set $\{a_{ij} \ : \ a \in \mathcal{C} \ \wedge \ 1 \leq i \leq n \ \wedge \ 1 \leq j \leq m\}$. Another basic

✉ Miguel A. Gutiérrez-Naranjo
magutier@us.es

Daniel Díaz-Pernil
sbdani@us.es

Hong Peng
ph.xhu@hotmail.com

[1] CATAM Research Group, Department of Applied Mathematics I, University of Seville, Seville, Spain

[2] Department of Computer Science and Artificial Intelligence, University of Seville, Seville, Spain

[3] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China

[1] A preliminary version of this paper can be found at [47].

concept associated with pixels is the adjacency. In such case, a distance is defined and two pixels are adjacent if the distance between them is one. Depending on the chosen distance, we can talk about 4-adjacency or 8-adjacency relation. The different treatments of such mappings (digital images) provide a big amount of current applications in biometrics [1], surveillance [35], medical imaging [6], human fingerprint classification [84], cartography [98], data compression and data storage [71], automated inspection of printed circuit boards [180] or optical character recognition (OCR) [156] among many others.

In the last years, the development of new hardware, mainly graphics processing units (GPU), has made possible the effective implementation of the paralellization intrinsic to P systems. Such GPU are especially well suited to address problems that can be expressed as data-parallel computations and, therefore, they are appropriate for the simulation of membrane computing devices. In fact, we can nowadays find many effective solutions to real-life problems inspired by membrane computing principles[2] (e.g., applications in computational economics [166], engineering [99], self-configurable robots [9] or fault diagnosis of power systems [120]).

In the literature, one can find many examples of the use of bio-inspired techniques for dealing with problems associated with the treatment of digital images. One of the classic examples is the use of cellular automata [145, 151]. Other efforts are related to artificial neural networks as in [53, 187] or, more recently, deep learning (see, e.g., [75, 103, 130]). In this paper, we present some of the main research lines bridging membrane computing and digital images. In Sect. 2, we recall the first attempts of linking both disciplines, mainly based on array grammars and on a graphical interpretation of the information encoded in a P system configuration, which allows to associate a picture to it. Section 3 is devoted to one of the main applications of membrane computing to image processing, namely the segmentation of images. Segmentation is the process of splitting a digital image into sets of pixels to make it simpler and easier to analyze. One of its main uses is the localization of objects and boundaries. Technically, the process consists of assigning a label to each pixel, in such a way that pixels with the same label form a meaningful region. Among the applications of segmentation of digital images, we can find the face recognition [186] or location of objects in satellite images (roads, forests, etc.) [61], but probably its main application area is medical imaging [13]. Section 4 is devoted to the skeletonization of images. Skeletonization is one of the approaches for representing a shape with a small amount of information by converting an image into a more compact representation and keeping the meaningful features. The conversion should remove redundant information, but it should also keep the basic structure. Other of the most promising applications focuses on algebraic-topological aspects of the images (Sect. 5) as those related to homology theory or discrete Morse theory (see, e.g., [31, 137]). The paper ends with an example of application (Sect. 6) and some final conclusions and comments on further research.

2 First steps

One of the first steps by bridging membrane computing and digital images was to consider two-dimensional objects. The objects placed in a membrane are usually zero-dimensional or one-dimensional (in the case of string objects [57, 127]). The first step for considering images in membrane computing is the use of two-dimensional objects, called arrays.[3] Array grammars have been widely studied in the literature. They can be considered as a straightforward extension of string grammars to two-dimensional pictures. Such pictures are sets of symbols placed in the points with integer coordinates of the plane (see, e.g., [36, 59, 144, 171]).

In [22], the model of array-rewriting P systems was presented on the basis of the transition P systems [125]: rules are of type $\mathcal{A} \rightarrow \mathcal{B}(tar)$, where $\mathcal{A}$ is the array to be rewritten, $\mathcal{B}$ is the new one and tar $\in$ {here, in, out} indicates the emplacement of the picture where the substitution has been made. Different approaches can be found in [128, 160]. In a natural way, transition P systems were extended to other P systems models, as in [27], where tissue P systems with arrays are used for dealing with the segmentation of images (see Sect. 3).

For example,[4] let us consider a P system with three nested membranes $[\,[\,[\,]_3\,]_2\,]_1$, an alphabet with two symbols a and # (the blank), an initial configuration with membranes 2 and 3 empty and the array $\begin{array}{c} a \\ a \end{array}$ placed in the membrane 1. Let us consider the sets of rules

$$R_1 = \left\{ \begin{array}{cc} & \# \\ \# & a \end{array} \rightarrow \begin{array}{cc} & a \\ \# & a \end{array} (in) \right\},$$

$$R_2 = \left\{ \begin{array}{cc} a & \# \\ \# & \end{array} \rightarrow \begin{array}{cc} a & a \\ \# & \end{array} (out),\ \begin{array}{ccc} a & \# & \# \\ \# & & \end{array} \rightarrow \begin{array}{ccc} a & a & a \\ \# & & \end{array} (in) \right\},$$

$$R_3 = \emptyset.$$

This P system generates all the L-shaped angles with equal arms, each arm being of length at least three. In the

[2] Some of these applications were collected in the volume [34].

[3] An overview of 2D picture array generating models based on membrane computing can be found in [164].

[4] Adapted from the Example 1 in [22].

literature, there are many approaches setting bridges between array grammars and membrane computing (see, e.g., [5, 20–22, 39, 40, 56, 97, 128, 160, 162, 163]).

Among the recent contributions in this research line, we can cite the so-called array P systems with permitting features [162]. The authors use the technique of regulating rewriting [37], associating permitting symbols with rules in the regions of an array P system [22], so that an array is rewritten by a rule only when the permitting symbols of the used rule are present in the array rewritten. This model incorporates the feature of permitting symbols in the rules. The advantage of this approach is that there is a reduction in the number of membranes in comparison to other array P system models. This idea of permitting features is also used in [86], where the authors bridge these concepts with array P system that uses basic puzzle grammar [159, 161]. Another variant of array P system, called tabled parallel array P system, was introduced in [87] by employing a well-known technique, called tables of rules, of grouping rules, especially used in Lindenmayer systems [146]. This enables a specific collection of rules being used at a time and enhances the generative power.

Another of the first links between P systems and digital images was the generation of graphical representation of branching structures able to simulate the growing of higher plants. The growth of plants, considered as a function of time, has attracted the attention of the scientific community for a very long time. Features such as the bilateral symmetry of leaves and the central symmetry of flowers have been a subject of study for computer scientists, mathematicians and life scientists among others. In 1968, Aristid Lindenmayer presented a theoretical framework for studying the development of simple multicellular organisms. The devices introduced in this framework are known as parallel rewriting systems or L-systems. L-systems were introduced for modeling multicellular organisms in terms of division, growth and death of individual cells [101, 102]. Several years later, the range of applications of L-systems were extended to higher plants and complex branching structures [60].

In [64, 65], a first approach for using P systems to simulate the growth and development of living plants was presented. This approach mixes L-systems and P systems, being in fact an L-system factorized into several units, which are them computed in the compartments delimited by the membranes of the P system.

Later, a new approach [140, 142, 143] was presented. It used the model of P systems with membrane creation [110, 124] where an object can produce a whole membrane via the application of a rule. This kind of rule allows to control the development of the membrane structure of a cell-like P system, which has a natural graphical interpretation as a tree-like graph. The multiset placed inside the membranes can be graphically interpreted in terms of color, length or thickness of the corresponding segment in the branching structure, allowing to provide a more and more realistic appearance. Figure 1, borrowed from [143], shows the graphical representation of four configurations of a P system with membrane creation. The drawn trees reproduce the tree-like structure of the membrane structure of the P systems and the length and thickness of the branches and the corresponding angles are fixed by the multiset of objects placed in the membranes in the corresponding configuration. These ideas of representing branching structures were also considered in [139], where a specific software for this graphical representation was developed. Some examples of polygons, spirals, friezes and plants can be found in this paper. Figure 2 shows some of them.

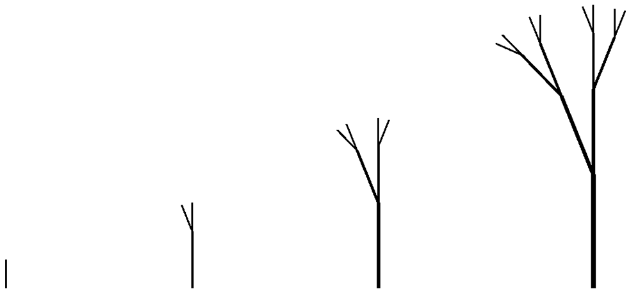

Fig. 1 Graphical interpretation of four configurations of a P system simulating the growing of a plant. Image borrowed from [143]

A close interpretation of the growing of higher plants was made for linking membrane computing and fractals [79]. A fractal [106] is a shape made of parts similar to the whole in some way. This self-similarity occurs over an infinite range of scales in pure mathematical structures, but over a finite range in many natural objects such as clouds, coastlines, surface of tumors or snowflakes. An appropriate use of the creation rules together with the non-determinism intrinsic to P systems and the interpretation of the multisets of objects could be useful in the study of the smooth surface of solid tumors, as pointed out in [80].

3 Segmentation

The study of array grammars or the graphical interpretation of a P system configuration was the first attempt to link membrane computing techniques to the study of images, but they cannot be considered as image processing. In such a way, one of the most studied image processing techniques studied in membrane computing is segmentation.[5]

[5] A recent literature survey devoted exclusively to image segmentation by using membrane computing can be found in [174].

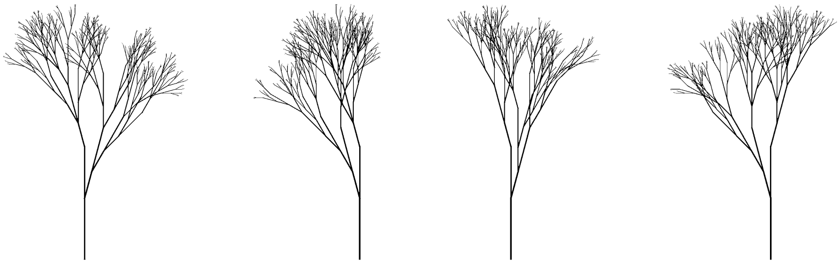

Fig. 2 Graphical representation of four configurations of a P system. Images borrowed from [139]

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. 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 in such a way those pixels with the same label share certain visual characteristics. These regions are mutually disjoint, well defined and have the same properties. The purpose of segmenting an image is to identify regions that are then utilized to recognize and understand the image. In the past decades, a large number of image segmentation algorithms have been developed [25, 104, 115]. These algorithms can be roughly classified into three categories: threshold-based segmentation methods, edge-based segmentation methods and region-based segmentation methods. 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, it produces positional information such as electroencephalography (EEG) or electrocardiography (EKG).

In the literature, there exist different techniques to segment an image. Some of them are clustering methods [100, 169], histogram-based methods [168], watershed transformation methods [167, 179] or graph partitioning methods [181]. Some of the practical applications of image segmentation are medical imaging [169], the study of anatomical structure, location of objects in satellite images (roads, forests, etc.) [154] or face recognition [96] among others.

In the literature, one can find many approaches from membrane computing to the problem of segmenting images. Some of these approaches are summarized in Table 1.

3.1 Threshold-based segmentation methods

Thresholding is widely used as a popular technique in image segmentation [148]. The goal of thresholding is to separate objects from background image or discriminate objects from objects that have distinct gray levels. Its underlying assumption is that an image consists of different regions corresponding to the gray-level ranges. It has been used widely as a tool to segment the gray images, but only a few works on color image segmentation have been reported. The main advantage of this technique lies in its simple computation approach. However, the threshold-based segmentation method ignores the spatial relationship information.

In recent years, P systems have been used to deal with threshold-based segmentation problems. Díaz-Pernil et al. [45] developed an image segmentation method on 2D images using P systems, which was applied to medical image segmentation. Christinal et al. [28] presented an image segmentation method based on tissue-like P systems by using the 4-neighborhood relation of pixels in the 2D image. However, they only addressed the segmentation results of artificial images rather than real-life images. Reina-Molina et al. [134] proposed a thresholding method based on tissue-like P systems with multiple auxiliary cells. Peña-Cantillana et al. [118] presented a method based on thresholding using also 4-adjacency with tissue-like P systems. Christinal et al. [33] have proposed a variant of the P system (tissue-like P system) using the rules to perform a parallel color segmentation of 2D images based on a threshold method. Wang et al. [170] proposed an optimal single-level thresholding method based on P systems. Peng et al. [119] presented a three-level thresholding method based on cell-like P systems for image segmentation. Zhang et al. [185] developed an infrared object segmentation method with membrane computing, which was used to obtain the optimal parameters quickly. Peng et al. [122] proposed a thresholding method based on tissue-like P systems and fuzzy entropy. In [121],

Table 1 The studies of P systems-based segmentation

#	Methods	References
1	Threshold-based segmentation	[28, 33, 45, 118, 119, 121] [122, 134, 170, 185]
2	Region-based segmentation	[32, 88, 123, 172, 178]
3	Edge-based segmentation	[15, 26, 29, 42, 48, 173]
4	Software and hardware implementation	[15, 16, 42, 45, 117, 155]

Peng et al. developed an optimal multi-level thresholding method based on cell-like P system.

3.2 Region-based segmentation methods

There are two key approaches regarding the region-based segmentation method: region growing and splitting–merging. Region growing polymerizes images, pixels or subregions that are considered as seeds into larger regions according to some criteria [23]. The characteristics of pixels and the adjacency of spatial distribution are fully considered in region growing. However, because of its iterative computational process, region growing has a high computing cost. In recent years, P systems have been used to realize several region-based segmentation methods.

Christinal et al. [32] proposed a region-based segmentation method for 2D and 3D images with tissue-like P systems, which was later improved by Carnero et al. [14]. Yang et al. [178] developed a region-based segmentation method with membrane computing, which effectively segmented gray images. However, the method cannot be extended to color images. Thus, Peng et al. [123] presented a region-based method to deal with color image segmentation. Isawasan et al. [88] proposed a region-based segmentation method based on tissue P systems for hexagonal digital images. Yahya et al. [172] proposed in 2015 a region-based method based on tissue P systems for 2D image segmentation. More recently, Yahya et al. have also presented a model for 3D image segmentation by using tissue-like P systems [176] and have used the Jaccard index method [89, 90] to measure the accuracy of the 2D segmentation in [175].

3.3 Edge-based segmentation methods

The edge-based segmentation method is extensively utilized for gray-level image segmentation, which is based on the detection of discontinuity in the gray level. An edge or boundary is a place where there is a more or less abrupt change in the gray level. Among the most used edge detection operators are Roberts operator, Sobel operator, Gauss–Laplace operator and Canny operator. Inspired from the mechanism of P systems, a number of edge-based segmentation methods have been addressed in recent years.

Christinal et al. [29] presented an edge-based segmentation method using tissue-like P systems for 2D and 3D images. Díaz-Pernil et al. [48] proposed an edge-based segmentation method based on tissue-like P systems to obtain homology groups. Díaz-Pernil et al. [42] proposed a parallel implementation of a new algorithm for segmenting images with gradient-based edge detection by using techniques from membrane computing. Carnero et al. [15] used tissue-like P systems to design an edge-based segmentation method. Christinal et al. [26] developed a method to search partially bounded regions with P systems. As a final reference, Yahya et al. presented in [173] a tissue-like P system-based edge-based segmentation method for 2D hexagonal images.

3.4 Software and hardware implementation

Carnero et al. [15, 16] have proposed a new hardware tool including a field-programmable gate array unit (FPGA) to perform segmentation of digital images for solving edge-based detection and noise removal problem. Their system uses membrane computing as well as a hardware programming (VHDL) language to propose an ad hoc processor. In another work, Díaz-Pernil et al. [45] have proposed a new software tool for segmenting 2D digital images on the basis of tissue-like P system, wherein the object oriented C++ programming language has been used in the implementation part. However, they did not provide a clear explanation regarding the technical aspects of developing the proposed tool.

A bio-inspired membrane computing software has been proposed by Peña-Cantillana et al. [117] to solve the threshold problem and it has been implemented in Compute Unified Device Architecture (CUDA$^{\mathrm{TM}}$), an innovative device architecture (see Sect. 7). Sheeba et al. [155] have proposed tissue-like P system to segment medical image, nuclei of the white blood cells (WBCs) of the peripheral blood smear images in morphology segmentation technique. Their algorithm has been implemented using MATLAB software.

In the work of Díaz-Pernil et al. [42], a CUDA$^{\mathrm{TM}}$ has been presented to implement tissue-like P system rules for segmenting images by the use of gradient-based edge detection to enhance the traditional methods of segmenting digital images. In [172], Yahya et al. used the tissue P systems simulator presented in [12] to check the validity of their design. A different approach can be also found in [46].

4 Skeletonization

Skeletonization in image processing is an approach for representing a shape with a small amount of information by converting an image into a more compact representation and keeping the meaningful features. The conversion should remove redundant information, but it should also keep the basic structure. The concept of skeleton was introduced by Blum [10, 11], under the name of medial axis transformation. The skeleton of an image is useful to characterize objects by a compact representation while preserving the connectivity and topological properties of any image. The most important features concerning a shape are its topology (represented by connected components, holes, etc.) and its geometry (elongated parts, ramifications, etc.), and thus they must be preserved.

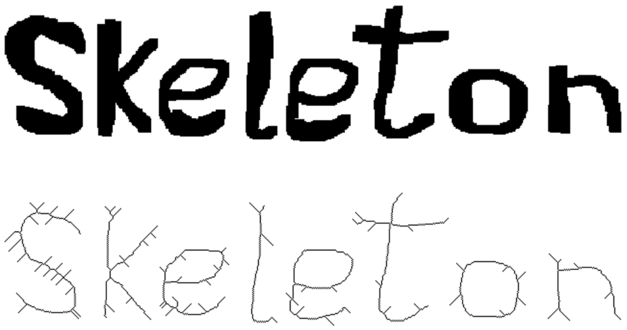

Fig. 3 A hand-written word and its skeletonization. Image borrowed from [49]

Roughly speaking, we can say that the image B is a skeleton of the black and white image A, if the former has fewer black pixels than the latter, preserves its topological properties and, in some sense, keeps its meaning. Figure 3 illustrates this idea. The skeletonized image keeps the meaning of the original one and it uses fewer black pixels. It keeps the basic geometry of the original image and also its topology. Let us remark that the white regions inside the hand-made words are also white regions in the skeletonized one and the connectedness is preserved.

Skeletonization has been found useful for data compression and pattern recognition in a wide range of applications in the industrial and scientific fields. It is usually considered as a pre-processing step in pattern recognition algorithms, but its study is also interesting by itself for the analysis of line-based images such as coronary arteries [52], human fingerprint classification [84], cartography [98], data compression and data storage [71], automated inspection of printed circuit boards [180] or optical character recognition (OCR) [156] among others. In many cases, the transformation of all the pixels can be done in parallel, since the state of a pixel at the step i only depends on the states of a set of pixels at the step $i-1$. Such parallelism in skeletonizing algorithms has been broadly studied (see, e.g., [77, 105, 165, 184]). The development of new hardware architectures has also contributed to new parallel implementations of these algorithms [67, 82, 83].

In [49, 50], Díaz-Pernil et al. presented an implementation of the Guo and Hall algorithm [77, 78] for skeletonizing images by using spiking neural P systems. In this algorithm, the pixels are examined for deletion in an iterative process. First of all, given an $p \times q$ image, it is divided into two subsections. One of the sections is composed of the pixels a_{ij} such that $i+j$ is even. Alternatively, the second sub-section corresponds to the pixels a_{ij} such that $i+j$ is odd. The algorithm consists of two sub-iterations where the removal of redundant pixels from both sub-sections is alternated, i.e., in each step only the pixels of one of the sub-sections are evaluated for its deletion.

The decision is based on a 3×3 neighborhood. Given a pixel $P0$, a clockwise enumeration $P1, \ldots, P8$ of its eight neighbor pixels is considered, as shown in Fig. 4a. As usual, for each $i \in \{1, \ldots, 8\}$, Pi is considered as a Boolean variable, with the truth value 1 if Pi is black and 0 if Pi is white. In each iteration, an evaluated black pixel $P0$ is deleted (changed to white) if and only if a set of conditions are satisfied. The key point in [50] is the use of a compact representation of the neighborhood of a pixel (also used in [147]) and the use of weights associated with the synapses of the SN P system.

A different approach to the skeletonization of images with P systems was presented by Nicolescu [111] where an approach to the problem was presented by using complex objects and actors in the framework of membrane computing.

5 Algebraic-topological aspects

A different approach to computer vision can also be obtained from topology. Topology in computer vision is referred to as connectivity, in a general way. For example, we look for connected components, holes in these connected components, etc., where the compulsory idea is the connection. In particular, algebraic topology [58] provides techniques and algorithms for dealing digital images from a topological point of view.

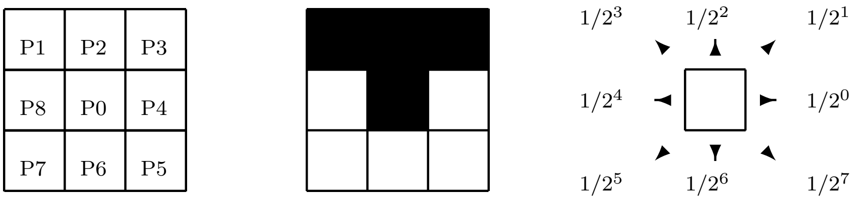

Fig. 4 (Left) enumeration of the pixels in a 3×3 neighborhood. (Center) 3×3 neighborhood with encoding [0, 0, 0, 0, 1, 1, 1, 1, 1], or, shortly, $2^4 + 2^5 + 2^6 + 2^7 + 2^8 = 496$. (Right) scheme of the weights of the synapses. Figure borrowed from [49]

The relationship between algebraic topology and natural computing is not new. In 1996, Chao and Nakayama [24] connected both areas using neural networks by extended Kohonen maps. Some years after, Giavitto et al. studied in [66] the topological structure of the membrane computing and Ceterchi et al. published two works where the digital image was introduced in the framework of the membrane computing [21, 22].

5.1 Effective homology

Recently, the links between algebraic topology and membrane computing have started to be explored via homology theory [30, 31, 48]. In such cases, black and white images are taken and using labeling techniques the number of black connected components and the number of holes[6] of these connected components are calculated. This information is known as the Betti numbers from a 2D picture.

Effective homology [131, 141, 152] is a algebraic-topological theory mainly based on the computational notion of chain homotopy equivalence, a concept which algebraically connects a cell complex or subdivided object with its homology groups. Roughly speaking, a chain homotopy equivalence can be specified by an operator, called chain homotopy operator, working at the level of linear combinations of cells which represents an efficient and non-redundant way of connecting cells. For instance, a chain homotopy operator at the level of cells of dimension 0 of a cell complex K can be completely described by a directed spanning forest (as many trees as connected components the object has) of the graph subcomplex formed by all the cells of K of dimension 0 and 1. Effective homology uses chain homotopy operators for capturing homology information and for representing the object in an algebraic-topological way. In fact, this idea is underlying the Eilenberg–MacLane work [54, 55] for computing the homology of prime spaces in homotopy theory, and it has been recently used in discrete image context. In [74], a method for computing homology aspects (with coefficients in the finite field $\mathbb{Z}/2\mathbb{Z} = \{0, 1\}$) of a three-dimensional digital binary-valued volume V considered over a body-centered-cubic grid is described. The representation used in that paper for a digital image is an algebraic-topological model (AT-model) consisting of two parts:

- *Geometric modeling level* A cell complex $K(V)$ topologically equivalent to the original volume is constructed. A 3D cell complex consists of vertices (0-cells), edges (1-cells), faces (2-cells) and polyhedra (3-cells). In particular, each edge connects two vertices, each face is enclosed by a loop of edges, and each three-cell is enclosed by an envelope of faces.
- *Homology analysis level* Homology information about $K(V)$ is exclusively codified in terms of a chain homotopy operator [72, 73].

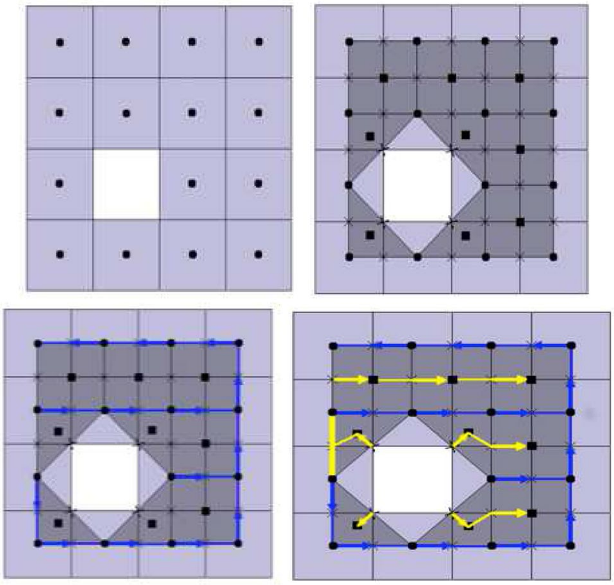

Fig. 5 Example. Pictures **a** left up, **b** right up, **c** left down and **d** right down

This method has recently evolved to a technique for generating $\mathbb{Z}/2\mathbb{Z}$-coefficients. It takes an AT model for a 26-adjacency voxel-based digital binary volume V using a polyhedral cell complex at geometric modeling level [92, 93, 108] and a chain homotopy operator described by a combinatorial vector field (a set of semidirected forests or a discrete differential form) at homology analysis level [132, 133]. For instance, a chain homotopy operator at the level of cells of dimension 0 (vertices) of a cell complex $K(V)$ can be completely described by a semidirected spanning forest of the graph subcomplex formed by all the cells of $K(V)$ of dimension 0 and 1.

In Fig. 5, a pixel-based digital object O (first picture from the left) is analyzed as a cell complex in which the square pixels are the 0-cells. The 1-cells are edges joining 8-neighbor pixels and these 2-cells are triangles or squares formed by three or four mutually (and in a maximal way) 8-adjacent pixels. Picture (b) describes this cell complex (in dark gray) in which the barycenters of the different cells are drawn (solid circles for the 0-cells, crosses for the 1-cells and solid squares for the 2-cells). The subcomplex formed by the 0- and 1-cells can be seen as a subgraph of the 8-adjacency graph of O. In (c), a spanning tree covering all the vertices of the cell complex is specified (in

[6] White connected components surrounded by black connected components.

blue). In fact, we consider a subdivision of this tree, having as 0-cells the vertices of the cell complex and the barycenters of the 1-cells belonging to the tree. An arrow in the tree determines the pairing of the source (0-cell) and sink (1-cell) cells and, consequently, indicating in this way that both are killed in homology group computation. Let us emphasize that only the top left 0-cell of the complex is not paired. It is a representative cycle (critical 0-cell of the homological process determined by the tree) of the unique connected component that the object has. Finally, in (d) we also draw the trees (in yellow) covering the rest of cells. They are semidirected, with arrows from the barycenters of 1-cells to the barycenter of the 2-cells. In terms of a process for computing homology groups, an arrow also means here that its source and sink cells are both killed. There is a edge marked in yellow which is not paired with an arrow. This 1-cell is a representative critical cell of the one-dimensional homology generator that the object has.

Using effective homology theory as the main tool for designing algorithms for computing complexes topological invariants (cohomology ring, (co)homology operations, homotopy groups,...), the problem of decomposing the objects into combinatorial graph-like pieces appears in a natural way. A possible solution to solve the high complexity costs of these processes is provided here by membrane computing. Alsalibi et al. present in [4] a membrane computing software for automatically computing homology groups of 2D digital images in a logarithmic number of steps.

Díaz-Pernil et al. use in [41, 43] a well-known tool from membrane computing, promoters. They are used to speed up the membrane algorithms. In that way, a bigger amount of information is handled. Within the digital imagery setting, we determine here a membrane computing strategy for partially specifying a chain homotopy operator at the level of pixels for a pixel-based digital 2D binary object O. This fundamental data structure in effective homology is obtained in terms of a forest spanning every vertex of its associated adjacency graph. Every tree of this forest determines and localizes the corresponding connected component. In [41], the authors obtain something more than the Betti numbers, they obtain the representative objects of each connected component and the borders of the holes. In ths way, they get the homology groups.

Until here, the efficiency of the membrane models with these kinds of problems is tested. But, we have to think of a bigger problem: what will happen when we want to work with bigger dimensions? We can follow two different ways. We could carry on as before, where the simplicity (from algebraic point of view) enables high efficiency. Or, we can introduce new algebraic-topological concepts where the amount of information to deal with would be increased.

Reina-Molina et al. decided to take this second option in 2012 in [135–137]. They present a simulation of the Morse theory algorithms in a parallel way, getting the homology groups of n-D objects.

We have introduced two different theoretical ways to solve problems from algebraic topology, but software based on these theories has been also developed. On one hand, Peña-Cantillana et al. [41] generate a parallel software using GPUs by CUDA to get the homology groups of 2D shapes based on techniques of spanning trees generated with membrane models where the promoters are compulsory. On the other hand, Reina-Molina et al. use PyCUDA (Python plus CUDA) to solve homological problems in a practical way in [138]. But, the complexity to adapt the Morse theory to parallel algorithms is high. So, this last software works almost completely parallel.

Many open questions arise in the relation of membrane computing and topology. From algebraic topology, we wonder if the membrane computing techniques can help in a better understanding of the problems and the design of more effective solutions. From membrane computing, a deeper study is necessary to explore how specific techniques of the different models can be applied. In particular, the use of priority in the application of rules is a strong requirement. It is worth to study if it can be avoided. From an implementation point of view, the exploration of the different parallel hardware architectures (clusters, grids, FPGA, ...) for the efficient implementation of the algorithms theoretically developed is an open research line.

6 A case study: the parallel segmentation of the optic disc

Image analysis and processing have great significance in the field of medicine, especially in non-invasive treatment and clinical study. However, with the development of new technologies, larger quantity of data, especially high-quality images, is available. Therefore, there is a new necessity of efficient and fast algorithms capable of processing and extracting meaningful features from images in a reasonable time. This is the case of mass screening programs for the early detection of retinal diseases such as glaucoma or diabetic retinopathy. Visual inspection of the large number of images so obtained is a time-consuming task for the medical experts. Moreover, computer-aided diagnosis (CAD) tools based on retinal image processing developed in the past are limited by the balance between accuracy and complexity due to their sequential programming.

In [44], a fully automatized algorithm based on membrane computing techniques for the parallel segmentation of the optic disc in retinal fundus images was presented (see Fig. 6). The optic disc is seen on fundus color photographs as a bright yellowish disc in human retina from where the blood vessels and optic nerves emerge. Its relevance resides

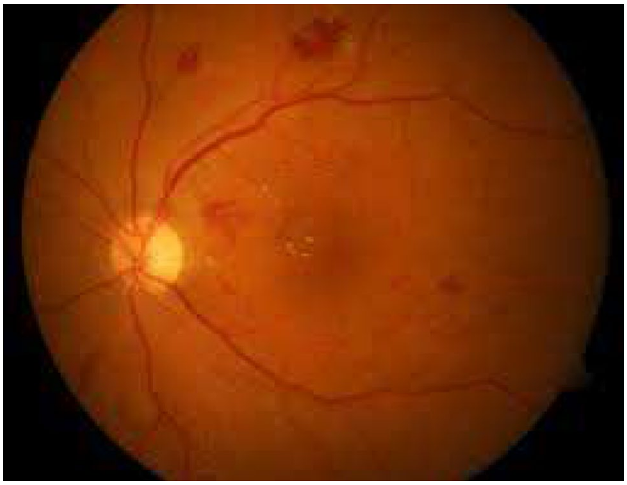

Fig. 6 Retinal image taken from the Standard Diabetic Retinopathy Database (DIARETDB1). The optic disc can be located as a yellow disc inside the image

in the fact that it is a key point for the diagnosis of a wide variety of diseases such as glaucoma or diabetic retinopathy. Moreover, it is usually taken as a base for detecting other anatomical structures (macula, blood vessels) and retinal abnormalities (microaneurysms, hard exudates, drusens, etc.). Most of the methods found in the literature are semi-automatized. This means that the computer treatment is crucial in the localization and detection of the optic disc, but it is the human expert who takes the final decision. In this paper, a fully automatized method is presented where no human expert is necessary for the detection of the optic disc.

Changes in the optic disc can indicate the current state and progression of a certain disease, while its diameter is usually used as a reference for measuring retinal distances and sizes [150]. Therefore, accurate optic disc localization and detection of its boundary is a principal and basic step for automated diagnosis systems [113].

In [44], a new method has been implemented with the GPU technology. Image edges are extracted using a new operator called AGP-color segmentator based on the membrane computing approach presented in [42]. It is a membrane computing implementation on CUDA^{TM} of the 3×3 and 5×5 versions of the Sobel algorithm [157] for edge detection. To choose an appropriate threshold to the binarization, a P system implementation [117] of the Hamadani algorithm [81] is applied. To avoid erroneous results, the obtained image is processed by eliminating the eye border. This is performed by applying a threshold on each color plane of the original image with the algorithm presented in [117]. The circular Hough transform is applied in parallel to the image in an interval of radius wide enough to consider all the possible optic discs.

The Hough transform is a well-known feature extraction technique used in image analysis. The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses [7, 51, 85]. The basic idea behind the Hough transform is to convert the image into a parameter space that is constructed specifically to describe the desired shape analytically.

The reliability of the tool was tested with 129 images from the public databases DRIVE [158] and DIARETDB1 [91] obtaining an average accuracy of 99.6% and a mean consumed time per image of 7.6 and 16.3 s, respectively. A comparison with several state-of-the-art algorithms shows that the algorithm represents a significant improvement in terms of accuracy and efficiency.

7 Final conclusions

Parallelizing classical digital image algorithms is a big challenge in the coming years [38, 116]. Such paralleling is much more complex than the merely simultaneous application of the sequential algorithm to different pieces of the image. The coordination of different simultaneous processes in a whole algorithm is so hard a task that commonly the parallel algorithm needs to be re-designed with only slight references to the classical one. Usually, the design of a new parallel implementation not inspired by the sequential one allows an open-mind vision of the problem and the proposal of new creative solutions. This opens an interesting research line not only for membrane computing researchers, but also for all of them interested in bio-inspired solutions, since the mathematical expression of the classical algorithms may change if we introduce the intrinsic parallelism of natural computing designs. Such new parallel solutions need a strong theoretical support that allows to control, to formalize, to check and to formally verify new algorithms.

As pointed out above, many of the problems in digital images share features very interesting for using these techniques: the information can be split into little pieces and expressed as (multi)sets of objects; the computation steps can be processed by rewriting rules; and the same sequential algorithm must be applied in different regions of the image which are independent and they can be treated locally by a set of processors. All these features lead us to consider membrane computing to deal with digital images.

The key point of paralleling classical sequential algorithms is the search of the efficiency and such efficiency is strongly linked to the development of new parallel hardware architectures which allow a realistic implementation of the theoretical advantages of the parallel processes. Different hardware architectures (clusters, grids, FPGA,...) propose different solutions [94, 95, 112, 149]. One of the most used architectures in the papers presented above has been the Compute Unified Device Architecture, CUDA^{TM}. This is a general-purpose parallel computing architecture that allows the parallel NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a CPU.

GPUs have emerged as general-purpose coprocessors in recent years. Traditionally designed for gaming applications, GPUs offer many computing threads arranged in a single-program multiple-data (SPMD) model. The chosen hardware architecture for our parallel implementation has been the Compute Unified Device Architecture, (CUDA^{TM}), which allows the parallel NVIDIA GPUs [189] to solve many complex computational problems in a more efficient way than on a sequential central processing unit (CPU).[7] This architecture has been widely used in membrane computing also for dealing with images [18, 19].

The choice of this parallel architecture is supported by several reasons. The first one is that the computing language CUDA^{TM} allows programmers a friendly model for implementing easily parallel programs, but the main reason comes from the practical side. In the last few years, there exists an increasing interest in the specialized industry for the development of more and more powerful graphic processing units which can be used for general purposes. This interest leads, on the one hand, to a more economically accessible (and hence, more extended) hardware and, on the other, to the development of more powerful computational units. The use of this new parallel architecture is currently explored as a tool for paralleling the treatment of digital images [17, 109].

With respect to the analysis on the performance of image processing with membrane computing techniques, a deep study is necessary. The authors usually only present a theoretical solution of the problem, some examples or a brief description of the software implementation.[8] Two new research lines are open in the analysis of performance: on the one hand, to study the theoretical properties of the new parallel algorithms from the point of view of the complexity theory and, on the other, to study different implementations in several parallel hardware to extract properties which help to design more efficient algorithms in the future.

Many other problems related to digital images have been addressed with membrane computing techniques. We can cite smoothing [118], which shows how to enhance an image by removing regions that do not provide relevant information, and the approach presented in [183], where membrane computing and quantum-inspired evolutionary algorithms are combined or the search of partially bounded regions [26]. This problem is also related to the HGB2I problem which consists of calculating the number of connected components and the representative curves of the holes of these components.

Detection of more complex structures, as corners in the images, has been explored in [8]. Alsalibi et al. study different bio-inspired approaches for the problem of face recognition in [3] and consider bridging membrane computing and evolutionary computing [182]. The problem of face recognition is also considered in [2] where a bat algorithm [177] under the framework of membrane computing (MC) is employed. Image registration is a research area which aims to find a transformation between two or more images under different conditions [188]. In [62], the authors propose a multi-modal image registration algorithm based on membrane computing. Gimel'farb et al. [68, 70] implemented the symmetric dynamic programming stereo (SDPS) algorithm [69] for stereo matching by using membrane computing techniques.

Many other approaches not cited in this paper have also been presented. The research area is active and provides open lines for future researchers. Among them, we can cite the use of different P system models. In the last years, many different P system models have been presented (mainly by adding or modifying syntactic or semantic features to previous models). It is an open problem to know if any of such new models are more appropriate to deal with problems from digital images. Another research line consists in exploring more problems beyond the ones presented in this paper. Many open questions about new real-life application will need deep studies in the next few years.

[7] For a good overview, the reader can refer to [114].

[8] A detailed description is out of the scope of this paper. An interested reader can consult the bibliography.

References

1. Adeoye OS. A survey of emerging biometric technologies. Int J Comput Appl. 2010;9(10):1–5.
2. Alsalibi B, Venkat I, Al-Betar MA. A membrane-inspired bat algorithm to recognize faces in unconstrained scenarios. Eng Appl Artif Intell. 2017;64:242–60.
3. Alsalibi B, Venkat I, Subramanian K, Lutfi SL, Wilde PD. The impact ofbio-inspired approaches toward the advancement of face recognition. ACM Comput Surv. 2015;48(1):1–33.
4. Alsalibi B, Venkat I, Subramanian KG, Christinal HA. A bio-inspired software for homology groups of 2d digital images. Asian Conf Membr Comput ACMC. 2014;2014:1–4.
5. Annadurai S, Kalyani T, Dare VR, Thomas DG. P systems generating iso-picture languages. Prog Nat Sci. 2008;18(5):617–22.
6. Ayache N. Medical image analysis and simulation. In: Shyamasundar RK, Ueda K, editors. ASIAN. Lecture notes in computer science, vol. 1345. Berlin: Springer; 1997. p. 4–17.
7. Ballard D. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognit. 1981;13(2):111–22.
8. Berciano A, Díaz-Pernil D, Christinal HA, Venkat I, Subramanian KG. First steps for a corner detection using membrane computing. Asian Conf Membr Comput ACMC. 2014;2014:1–6.

9. Bie D, Gutiérrez-Naranjo MA, Zhao J, Zhu, Y. A membrane computing framework for self-reconfigurable robots. Nat Comput. 2018. https://doi.org/10.1007/s11047-018-9702-1.
10. Blum H. An associative machine for dealing with the visual field and some of its biological implications. In: Bernard EE, Kare MR, editors. Proceedings of the 2nd annual bionics symposium, held at Cornell University, 1961. Biological prototypes and synthetic systems, vol. 1. New York: Plenum Press; 1962. p. 244–60.
11. Blum H. An associative machine for dealing with the visual field and some of its biological implications. Computer and Mathematical Sciences Laboratory, Electronics Research Directorate, Air Force Cambridge Research Laboratories, Office of Aerospace Research, United States Air Force; 1962.
12. Borrego-Ropero R, Díaz-Pernil D, Pérez-Jiménez MJ. Tissue simulator: a graphical tool for tissue P systems. In: Vaszil G, editors. Proceedings of the International Workshop Automata for Cellular and Molecular Computing. MTA SZTAKI, Budapest, Hungary. Satellite of the 16th International Symposium on Fundamentals of Computational Theory; 2007. p. 23–34.
13. Campadelli P, Casiraghi E, Esposito A. Liver segmentation from computed tomography scans: a survey and a new algorithm. Artif Intell Med. 2009;45(2–3):185–96.
14. Carnero J, Christinal HA, Díaz-Pernil D, Reina-Molina R, Subathra MSP. Improved parallelization of an image segmentation bio-inspired algorithm. In: Babu BV, Nagar A, Deep K, Pant M, Bansal JC, Ray K, Gupta U, editors. Proceedings of the second international conference on soft computing for problem solving, vol. 236, SocProS 2012, December 28–30, 2012, JK Lakshmipat University (JKLU), Jaipur, India. Advances in intelligent systems and computing. Springer; 2012. p. 75–82.
15. Carnero J, Díaz-Pernil D, Gutiérrez-Naranjo MA. Designing tissue-like P systems for image segmentation on parallel architectures. In: Martínez-del-Amor MA, Păun G, Pérez-Hurtado I, Romero-Campero FJ, Valencia-Cabrera L, editors. Ninth brainstorming week on membrane computing. Sevilla: Fénix Editora; 2011. p. 43–62.
16. Carnero J, Díaz-Pernil D, Molina-Abril H, Real P. Image segmentation inspired by cellular models using hardware programming. Image A Appl Math Image Eng. 2010;1(3):143–50.
17. Carranza C, Murray V, Pattichis M, Barriga ES. Multiscale AM-FM decompositions with GPU acceleration for diabetic retinopathy screening. In: IEEE southwest symposium on image analysis and interpretation (SSIAI); 2012. p. 121–24.
18. Cecilia JM, García JM, Guerrero GD, Martínez-del-Amor MA, Pérez-Hurtado I, Pérez-Jiménez MJ. Simulating a P system based efficient solution to SAT by using GPUs. J Log Algebraic Program. 2010;79(6):317–25.
19. Cecilia JM, García JM, Guerrero GD, Martínez-del-Amor MA, Pérez-Hurtado I, Pérez-Jiménez MJ. Simulation of P systems with active membranes on CUDA. Brief Bioinform. 2010;11(3):313–22.
20. Ceterchi R, Gramatovici R, Jonoska, N. Tiling rectangular pictures with P systems. In: Martín-Vide et al. [107]; 2004. p. 88–103.
21. Ceterchi R, Gramatovici R, Jonoska N, Subramanian KG. Tissue-like P systems with active membranes for picture generation. Fundamenta Informaticae. 2003;56(4):311–28.
22. Ceterchi R, Mutyam M, Păun G, Subramanian KG. Array-rewriting P systems. Nat Comput. 2003;2(3):229–49.
23. Chang Y, Li X. Adaptive image region-growing. IEEE Trans Image Process. 1994;3(6):868–72.
24. Chao J, Nakayama J. Cubical singular simplex model for 3D objects and fast computation of homology groups. In: 13th international conference on pattern recognition (ICPR'96), vol. IV. IEEE Computer Society, Los Alamitos, CA, USA; 1996. p. 190–94.
25. Cheng H, Jiang X, Sun Y, Wang J. Color image segmentation: advances and prospects. Pattern Recognit. 2001;34(12):2259–81.
26. Christinal HA, Berciano A, Díaz-Pernil D, Gutiérrez-Naranjo MA. Searching partially bounded regions with P systems. In: Pant M, Deep K, Nagar A, Bansal JC, editors. Proceedings of the third international conference on soft computing for problem solving: SocProS 2013, vol. 1. New Delhi: Springer; 2014. p. 45–54.
27. Christinal HA, Díaz-Pernil D, Gutiérrez-Naranjo MA, Pérez-Jiménez MJ. Array tissue-like P systems. In: Martínez del Amor MA, Păun G, Pérez Hurtado I, Riscos-Núñez A, editors. Eighth brainstorming week on membrane computing. Sevilla: Fénix Editora; 2010. p. 37–51.
28. Christinal HA, Díaz-Pernil D, Gutiérrez-Naranjo MA, Pérez-Jiménez MJ. Thresholding of 2D images with cell-like P systems. Rom J Inf Sci Technol (ROMJIST). 2010;13(2):131–40.
29. Christinal HA, Díaz-Pernil D, Real P. Segmentation in 2D and 3D image using tissue-like P system. In: Bayro-Corrochano E, Eklundh JO, editors. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15–18, 2009. Proceedings, lecture notes in computer science, vol. 5856. Berlin: Springer; 2009. p. 169–76.
30. Christinal HA, Díaz-Pernil D, Real P. Using membrane computing for obtaining homology groups of binary 2D digital images. In: Wiederhold P, Barneva RP, editors. Combinatorial Image Analysis 13th International Workshop, IWCIA 2009, Playa del Carmen, Mexico, November 24–27, 2009. Proceedings, lecture notes in computer science, vol. 5852. Berlin: Springer; 2009. p. 383–96.
31. Christinal HA, Díaz-Pernil D, Real P. P systems and computational algebraic topology. Mathematical and computer modelling, vol. 52, 1982–1996. The BIC-TA 2009 special issue, international conference on bio-inspired computing: theory and applications; 2010. p. 11–12.
32. Christinal HA, Díaz-Pernil D, Real P. Region-based segmentation of 2D and 3D images with tissue-like P systems. Advances in theory and applications of pattern recognition, image processing and computer vision. Pattern Recognit Lett. 2012;32(16):2206–12.
33. Christinal HA, Díaz-Pernil D, Real JP, Selvan SE. Color segmentation of 2D images with thresholding. In: Mathew J, Patra P, Pradhan DK, Kuttyamma AJ, editors. Proceedings of eco-friendly computing and communication systems: international conference, ICECCS 2012, Kochi, India, August 9–11. Berlin: Springer; 2012. p. 162–69.
34. Ciobanu G, Pérez-Jiménez MJ, Păun G, editors. Applications of membrane computing. Natural computing series. Berlin: Springer; 2006.
35. Collins R, Lipton A, Kanade T. Introduction to the special section on video surveillance. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):745–6.
36. Cook CR, Wang PSP. A Chomsky hierarchy of isotonic array grammars and languages. Comput Graph Image Process. 1978;8(1):144–52.
37. Dassow J, Păun G. Regulated rewriting in formal language theory. 1st ed. New York: Springer Publishing Company Incorporated; 2012.
38. Davies E. Computer and machine vision: theory, algorithms practicalities. Waltham: Elsevier Science; 2012.
39. Dersanambika KS, Krithivasan K. Contextual array P systems. Int J Comput Math. 2004;81(8):955–69.

40. Dersanambika KS, Krithivasan K, Subramanian KG. P systems generating hexagonal picture languages. In: Martín-Vide et al. [107]; 2003. p. 168–80.
41. Díaz-Pernil D, Berciano A, Peña-Cantillana F, Gutiérrez-Naranjo MA. Bio-inspired parallel computing of representative geometrical objects of holes of binary 2D-images. Int J Bioinspired Comput. 2017;9(2):77–92.
42. Díaz-Pernil D, Berciano A, Peña-Cantillana F, Gutiérrez-Naranjo MA. Segmenting images with gradient-based edge detection using membrane computing. Pattern Recognit Lett. 2013;34(8):846–55.
43. Díaz-Pernil D, Christinal HA, Gutiérrez-Naranjo MA, Real P. Using membrane computing for effective homology. Appl Algebra Eng Commun Comput. 2012;23(5–6):233–49.
44. Díaz-Pernil D, Fondón I, Peña-Cantillana F, Gutiérrez-Naranjo MA. Fully automatized parallel segmentation of the optic disc in retinal fundus images. Geometric, topological and harmonic trends to image processing. Pattern Recognit Lett. 2016;83(1):99–107.
45. Díaz-Pernil D, Gutiérrez-Naranjo MA, Molina-Abril H, Real P. A bio-inspired software for segmenting digital images. In: Nagar AK, Thamburaj R, Li K, Tang Z, Li R, editors. Fifth international conference on bio-inspired computing: theories and applications, BIC-TA 2010, University of Hunan, Liverpool Hope University, Liverpool, United Kingdom/Changsha, China, September 8–10 and September 23–26, 2010, vol. 2. Beijing: IEEE Computer Society; 2010. p. 1377–81.
46. Díaz-Pernil D, Gutiérrez-Naranjo MA, Molina-Abril H, Real P. Designing a new software tool for digital imagery based on P systems. Nat Comput. 2011;11(3):381–6.
47. Díaz-Pernil D, Gutiérrez-Naranjo MA, Peng H. Some notes on membrane computing and image processing. Bull Int Membr Comput Soc. 2016;2:103–28.
48. Díaz-Pernil D, Gutiérrez-Naranjo MA, Real P, Sánchez-Canales V. Computing homology groups in binary 2D imagery by tissue-like P systems. Rom J Inf Sci Technol. 2010;13(2):141–52.
49. Díaz-Pernil D, Peña-Cantillana F, Gutiérrez-Naranjo MA. Skeletonizing images by using spiking neural P systems. In: Martínez-del-Amor MA, Păun G, Pérez-Hurtado I, Romero-Campero FJ, editors. Tenth brainstorming week on membrane computing, vol. I. Sevilla: Fénix Editora; 2012. p. 91–110.
50. Díaz-Pernil D, Peña-Cantillana F, Gutiérrez-Naranjo MA. A parallel algorithm for skeletonizing images by using spiking neural P systems. Neurocomputing. 2013;115:81–91.
51. Duda RO, Hart PE. Use of the Hough transformation to detect lines and curves in pictures. Commun ACM. 1972;15(1):11–5.
52. Dufresne TE, Sarwal A, Dhawan AP. A gray-level thinning method for delineation and representation of arteries. Comput Med Imaging Graph. 1994;18(5):343–55.
53. Egmont-Petersen M, de Ridder D, Handels H. Image processing with neural networks—a review. Pattern Recognit. 2002;35(10):2279–301.
54. Eilenberg S, MacLane S. Relations between homology and homotopy groups of spaces. Ann Math. 1945;46(3):480–509.
55. Eilenberg S, MacLane S. Relations between homology and homotopy groups of spaces. II. Ann Math. 1950;51(3):514–33.
56. Fernau H, Freund R, Schmid ML, Subramanian KG, Wiederhold P. Contextual array grammars and array P systems. Ann Math Artif Intell. 2015;75(1–2):5–26.
57. Ferretti C, Mauri G, Zandron C. P systems with string objects. In: Păun et al. [129]; 2010. p. 168–97.
58. Freedman D, Chen C. Algebraic topology for computer vision. In: Science and technology; 2009. p. 239–68.
59. Freund R. Array grammars. Tech. Rep. 15/00, Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona; 2000.
60. Frijters D, Lindenmayer A. A model for the growth and flowering of aster novae-angliae on the basis of table< 1, 0> L-systems. In: Rozenberg G, Salomaa A, editors. L systems, most of the papers were presented at a conference in Aarhus, Denmark, January 14–25, 1974, lecture notes in computer science, vol. 15. Springer; 1974. p. 24–52.
61. Gamanya R, Maeyer PD, Dapper MD. An automated satellite image classification design using object-oriented segmentation algorithms: a move towards standardization. Expert Syst Appl. 2007;32(2):616–24.
62. Gao Z, Zhang C. MCIR: a multi-modal image registration algorithm based on membrane computing. In: 2017 international conference on computing intelligence and information system (CIIS); 2017. p. 263–69.
63. García-Quismondo M, Macías-Ramos LF, Păun G, Valencia-Cabrera L, editors. Tenth brainstorming week on membrane computing, vol. II. Sevilla: Fénix Editora; 2012.
64. Georgiou A, Gheorghe M. Generative devices used in graphics. In: Alhazov A, Martín-Vide C, Păun G, editors. Preproceedings of the workshop on membrane computing. Technical Report; 28/03, . Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, Spain; 2003. p. 266–72.
65. Georgiou A, Gheorghe M, Bernardini F. Membrane-based devices used in computer graphics. In: Ciobanu G, Păun G, Pérez-Jiménez MJ, editors. Applications of membrane computing, natural computing series. Berlin: Springer; 2006. p. 253–81.
66. Giavitto JL, Michel O. The topological structures of membrane computing. Fundamenta Informaticae. 2002;49(1–3):123–45.
67. Gil Montoya M, Garcia I. Implementation of parallel thinning algorithms on multicomputers: analysis of the work load balance. In: Proceedings of the sixth Euromicro workshop on parallel and distributed processing, PDP '98; 1998. p. 257–63.
68. Gimel'farb G, Nicolescu R, Ragavan S. P systems in stereo matching. In: Real P, Díaz-Pernil D, Molina-Abril H, Berciano A, Kropatsch W, editors. Computer analysis of images and patterns. Lecture notes in computer science, vol. 6855. Berlin: Springer; 2011. p. 285–92.
69. Gimel'farb GL. Probabilistic regularisation and symmetry in binocular dynamic programming stereo. Pattern Recognit Lett. 2002;23(4):431–42.
70. Gimel'farb GL, Nicolescu R, Ragavan S. P system implementation of dynamic programming stereo. J Math Imaging Vis. 2013;47(1–2):13–26.
71. González RC, Woods RE. Digital image processing. Upper Saddle River: Pearson/Prentice Hall; 2008.
72. González-Díaz R, Jiménez MJ, Medrano B, Molina-Abril H, Real P. Integral operators for computing homology generators at any dimension. In: Ruiz-Shulcloper J, Kropatsch WG, editors. CIARP. Lecture notes in computer science, vol. 5197. Berlin Heidelberg: Springer; 2008. p. 356–63.
73. González-Díaz R, Jiménez MJ, Medrano B, Real P. Chain homotopies for object topological representations. Discret Appl Math. 2009;157(3):490–9.
74. González-Díaz R, Real P. On the cohomology of 3D digital images. Discret Appl Math. 2005;147(2–3):245–63.
75. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press; 2016.
76. Graciani C, Păun G, Romero-Jiménez A, Sancho-Caparrini F, editors. Fourth brainstorming week on membrane computing, vol. II. Sevilla: Fénix Editora; 2006.

77. Guo Z, Hall RW. Parallel thinning with two-subiteration algorithms. Commun ACM. 1989;32:359–73.
78. Guo Z, Hall RW. Fast fully parallel thinning algorithms. CVGIP Image Underst. 1992;55:317–28.
79. Gutiérrez-Naranjo MA, Pérez-Jiménez, MJ. Fractals and P systems. In: Graciani et al. [76]; 2006. p. 65–86.
80. Gutiérrez-Naranjo MA, Pérez-Jiménez MJ, Riscos-Núñez A, Romero-Campero FJ. How to express tumours using membrane systems. Prog Nat Sci. 2007;17(4):449–57.
81. Hamadani, N. Automatic target cueing in IR imagery. Master's thesis, Air Force Institute of Technology, WAFP; 1981.
82. Heydorn S, Weidner P. Optimization and performance analysis of thinning algorithms on parallel computers. Parallel Comput. 1991;17(1):17–27.
83. Holt C, Stewart A. A parallel thinning algorithm with fine grain subtasking. Parallel Comput. 1989;10(3):329–34.
84. Hongbin P, Junali C, Yashe Z. Fingerprint thinning algorithm based on mathematical morphology. In: 8th international conference on electronic measurement and instruments. ICEMI '07; 2007. p. 618–21.
85. Hough PVC. Machine analysis of bubble chamber pictures. In: International conference on high energy accelerators and instrumentation. CERN; 1959. p. 554–58.
86. Isawasan P, Muniyandi RC, Venkat I, Subramanian KG. Array-rewriting P systems with basic puzzle grammar rules and permitting features. In: Leporati A, Rozenberg G, Salomaa A, Zandron C, editors. Membrane computing—17th international conference, CMC 2016, Milan, Italy, July 25–29, 2016, revised selected papers. Lecture notes in computer science, vol. 10105. Springer; 2016. p. 272–85.
87. Isawasan P, Venkat I, Muniyandi RC, Subramanian, KG. A membrane computing model for generation of picture arrays. In: Zaman HB, Robinson P, Smeaton AF, Shih TK, Velastin SA, Jaafar A, Ali NM, editors. Proceedings of advances in visual informatics—4th international visual informatics conference, IVIC 2015, Bangi, Malaysia, November 17–19. Lecture notes in computer science, vol. 9429. Springer; 2015. p. 155–65.
88. Isawasan P, Venkat I, Subramanian KG, Khader AT, Osman O, Christinal HA. Region-based segmentation of hexagonal digital images using membrane computing. In: IEEE 2014 Asian conference on membrane computing (ACMC); 2014. p. 1–4.
89. Ivchenko GI, Honov SA. On the Jaccard similarity test. J Math Sci. 1998;88(6):789–94.
90. Jaccard P. Nouvelles recherches sur la distribution florale. Bulletin de la Sociète Vaudense des Sciences Naturelles. 1908;44:223–70.
91. Kauppi T, Kalesnykiene V, Kamarainen JK, Lensu L, Sorri I, Raninen A, Voutilainen R, Uusitalo H, Kälviäinen H, Pietilä J. The diaretdb1 diabetic retinopathy database and evaluation protocol. In: Proceedings of the British machine vision conference 2007, University of Warwick, UK, September 10–13. British Machine Vision Association; 2007. p. 252–61.
92. Kenmochi Y, Imiya A, Ichikawa A. Discrete combinatorial geometry. Pattern Recognit. 1997;30(10):1719–28.
93. Kenmochi Y, Imiya A, Ichikawa A. Boundary extraction of discrete objects. Comput Vis Image Underst. 1998;71(3):281–93.
94. Khalid NEA, Ahmad SA, Noor NM, Fadzil AFA, Taib MN. Analysis of parallel multicore performance on Sobel edge detector. In: Proceedings of the 15th WSEAS international conference on computers. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA; 2011. p. 313–18.
95. Khalid NEA, Ahmad SA, Noor NM, Fadzil AFA, Taib MN. Parallel approach of Sobel edge detector on multicore platform. Int J Comput Commun. 2011;5:236–44.
96. Kim SH, Kim HG, Tchah KH. Object oriented face detection using colour transformation and range segmentation. IEEE Electron Lett. 1998;34:979–80.
97. Krishna SN, Rama R, Krithivasan K. P systems with picture objects. Acta Cybernetica. 2001;15(1):53–74.
98. Lee KH, Cho SB, Choy YC. Automated vectorization of cartographic maps by a knowledge-based system. Eng Appl Artif Intell. 2000;13(2):165–78.
99. Lefticaru R, Bakir ME, Konur S, Stannett M, Ipate F. Modelling and validating an engineering application in kernel P systems. In: Gheorghe M, Rozenberg G, Salomaa A, Zandron C, editors. Membrane computing—18th international conference, CMC 2017, Bradford, UK, July 25–28, 2017, revised selected papers. Lecture notes in computer science, vol. 10725. Springer; 2017. p. 183–95.
100. Li X, Zhang T, Qu Z. Image segmentation using fuzzy clustering with spatial constraints based on Markov random field via bayesian theory. IEICE Trans Fundam Electron Commun Comput Sci. 2008;91–A(3):723–9.
101. Lindenmayer A. Mathematical models for cellular interaction in development: parts I and II. J Theor Biol. 1968;18:280–315.
102. Lindenmayer A. Developmental systems without cellular interactions, their languages and grammars. J Theor Biol. 1971;30(3):455–84.
103. Litjens GJS, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60–88.
104. Liu D, Jiang Z, Feng H. A novel fuzzy classification entropy approach to image thresholding. Pattern Recognit Lett. 2006;27(16):1968–75.
105. Lü HE, Wang PSP. A comment on a fast parallel algorithm for thinning digital patterns. Commun ACM. 1986;29(3):239–42.
106. Mandelbrot BB. The fractal geometry of nature. New York: W. H. Freedman and Co.; 1983.
107. Martín-Vide C, Mauri G, Păun G, Rozenberg G, Salomaa A, editors. Membrane computing, international workshop, WMC 2003, Tarragona, Spain, July 17–22, 2003, revised papers. Lecture notes in computer science, vol. 2933. Berlin: Springer; 2004.
108. Molina-Abril H, Real P. Advanced homology computation of digital volumes via cell complexes. In: da Vitoria Lobo N, Kasparis T, Roli F, Kwok JTY, Georgiopoulos M, Anagnostopoulos GC, Loog M, editors. SSPR/SPR. Lecture notes in computer science, vol. 5342. Berlin: Springer; 2008. p. 361–71.
109. Moulik S, Boonn WW. The role of GPU computing in medical image analysis and visualization. In: Boonn WW, Liu BJ, editors. Medical imaging 2011: advanced PACS-based imaging informatics and therapeutic applications, vol. 7967. Proceedings of the SPIE; 2011. p. 79670L.
110. Mutyam M, Krithivasan K. P systems with membrane creation: universality and efficiency. In: Margenstern M, Rogozhin Y, editors. MCU, lecture notes in computer science, vol. 2055. Springer; 2001. p. 276–87.
111. Nicolescu, R.: Parallel thinning with complex objects and actors. In: Gheorghe M, Rozenberg G, Salomaa A, Sosík P, Zandron C, editors. Membrane computing—15th international conference, CMC 2014, Prague, Czech Republic, August 20–22, 2014, revised selected papers, lecture notes in computer science, vol. 8961. Springer; 2014. p. 330–54.
112. Ogawa K, Ito Y, Nakano K. Efficient canny edge detection using a GPU. In: Proceedings of the 2010 first international conference on networking and computing, ICNC '10. IEEE Computer Society, Washington, DC, USA; 2010. p. 279–80.
113. Osareh A, Mirmehdi M, Thomas B, Markham R. Automated identification of diabetic retinal exudates in digital colour images. Br J Ophthalmol. 2003;87(10):1220–3.

114. Owens JD, Houston M, Luebke D, Green S, Stone JE, Phillips JC. GPU computing. Proc IEEE. 2008;96(5):879–99.
115. Pal NR, Pal SK. A review on image segmentation techniques. Pattern Recognit. 1993;26(9):1277–94.
116. Parker J. Algorithms for image processing and computer vision. New York: Wiley; 2010.
117. Peña-Cantillana F, Díaz-Pernil D, Berciano A, Gutiérrez-Naranjo MA. A parallel implementation of the thresholding problem by using tissue-like P systems. In: Real P, Díaz-Pernil D, Molina-Abril H, Berciano A, Kropatsch WG, editors. Computer analysis of images and patterns—14th international conference, CAIP 2011, Seville, Spain, August 29–31, 2011, proceedings, part II. Lecture notes in computer science, vol. 6855. Springer; 2011. p. 277–84.
118. Peña-Cantillana F, Díaz-Pernil D, Christinal HA, Gutiérrez-Naranjo MA. Implementation on CUDA of the smoothing problem with tissue-like P systems. Int J Nat Comput Res. 2011;2(3):25–34.
119. Peng H, Shao J, Li B, Wang J, Pérez-Jiménez MJ, Jiang Y, Yang Y. Image thresholding with cell-like P systems. In: García-Quismondo et al. [63]; 2012. p. 75–88.
120. Peng H, Wang J, Ming J, Shi P, Pérez-Jiménez MJ, Yu W, Tao C. Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems. IEEE Trans Smart Grid. 2018;9(5):4777–84.
121. Peng H, Wang J, Pérez-Jiménez MJ. Optimal multi-level thresholding with membrane computing. Digit Signal Process. 2015;37:53–64.
122. Peng H, Wang J, Pérez-Jiménez MJ, Shi P. A novel image thresholding method based on membrane computing and fuzzy entropy. J Intell Fuzzy Syst. 2013;24(2):229–37.
123. Peng H, Yang Y, Zhang J, Huang X, Wang J. A region-based color image segmentation method based on P systems. Rom J Inf Sci Technol. 2014;17(1):63–75.
124. Pérez-Jiménez M.J, Riscos-Núñez A, Romero-Jiménez A, Woods D. Complexity—membrane division, membrane creation. In: Păun et al. [129]; 2010. p. 302–36.
125. Păun G. Membrane computing: an introduction. Berlin: Springer; 2002.
126. Păun G. Computing with membranes. Tech. Rep. 208, Turku Centre for Computer Science, Turku, Finland; 1998.
127. Păun G. Computing with membranes. J Comput Syst Sci. 2000;61(1):108–43 (See also [126]).
128. Păun G. Grammar systems vs. membrane computing: a preliminary approach. In: Pre-proceedings of the workshop on grammar systems, MTA SZTAKI Budapest; 2004. p. 225–45.
129. Păun G, Rozenberg G, Salomaa A, editors. The Oxford handbook of membrane computing. Oxford: Oxford University Press; 2010.
130. Rawat W, Wang Z. Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. 2017;29(9):2352–449.
131. Real P. Homological perturbation theory and associativity. Homol Homotopy Appl. 2000;2(5):51–88.
132. Real P, Molina-Abril H. Cell at-models for digital volumes. In: Torsello A, Escolano F, Brun L, editors. GbRPR, vol. 5534., Lecture notes in computer scienceBerlin: Springer; 2009. p. 314–23.
133. Real P, Molina-Abril H, Kropatsch WG. Homological tree-based strategies for image analysis. In: Jiang X, Petkov N, editors. CAIP, vol. 5702., Lecture notes in computer scienceBerlin: Springer; 2009. p. 326–33.
134. Reina-Molina R, Carnero Iglesias J, Díaz-Pernil D. Image segmentation using tissue-like P systems with multiple auxiliary cells. Image A Appl Math Image Eng. 2011;2(4):25–8.
135. Reina-Molina R, Díaz-Pernil D. Bioinspired parallel 2D or 3D skeletonization. Image A Appl Math Image Eng. 2013;3(6):41–4.
136. Reina-Molina R, Díaz-Pernil D, Gutiérrez-Naranjo MA. Cell complexes and membrane computing for thinning 2D and 3D images. In: García-Quismondo et al. [63]. p. 167–86.
137. Reina-Molina R, Díaz-Pernil D, Real P, Berciano A. Membrane parallelism for discrete Morse theory applied to digital images. Appl Algebra Eng Commun Comput. 2015;26(1–2):49–71.
138. Reina-Molina R, Díaz-Pernil D, Real P, Berciano A. Effective homology of k-d digital objects (partially) calculated in parallel. Pattern Recognit Lett. 2016;83:59–66.
139. Rivero-Gil E, Gutiérrez-Naranjo MA, Pérez-Jiménez MJ. Graphics and P systems: experiments with JPLANT. In: Díaz-Pernil D, Graciani C, Gutiérrez-Naranjo MA, Păun G, Pérez-Hurtado I, Riscos-Núñez A, editors. Sixth brainstorming week on membrane computing. Sevilla: Fénix Editora; 2008. p. 241–53.
140. Rivero-Gil E, Gutiérrez-Naranjo MA, Romero Jiménez Á, Riscos-Núñez A. A software tool for generating graphics by means of P systems. Nat Comput. 2011;10(2):879–90.
141. Romero A, Rubio J, Sergeraert F. Effective homology of filtered digital images. Pattern Recognit Lett. 2016;83:23–31.
142. Romero-Jiménez Á, Gutiérrez-Naranjo MA, Pérez-Jiménez MJ. Graphical modeling of higher plants using P systems. In: Hoogeboom HJ, Păun G, Rozenberg G, Salomaa A, editors. Workshop on membrane computing. Lecture notes in computer science, vol. 4361. Berlin: Springer; 2006. p. 496–506.
143. Romero-Jiménez A, Gutiérrez-Naranjo MA, Pérez-Jiménez MJ. The growth of branching structures with P systems. In: Graciani et al. [76]. p. 253–65.
144. Rosenfeld A. Picture languages. Reading: Academic Press; 1979.
145. Rosin PL. Training cellular automata for image processing. IEEE Trans Image Process. 2006;15(7):2076–87.
146. Rozenberg G, Salomaa A. The mathematical theory of L systems. Pure and applied mathematics. New York: Elsevier Science; 1980.
147. Saeed K, Tabedzki M, Rybnik M, Adamski M. K3M: a universal algorithm for image skeletonization and a review of thinning techniques. Appl Math Comput Sci. 2010;20(2):317–35.
148. Sahoo P, Soltani S, Wong A. A survey of thresholding techniques. Comput Vis Graph Image Process. 1988;41(2):233–60.
149. Sanduja V, Patial R. Sobel edge detection using parallel architecture based on FPGA. Int J Appl Inf Syst. 2012;3(4):20–4.
150. Sekhar S, Al-Nuaimy W, Nandi A.K. Automated localisation of retinal optic disk using Hough transform. In: IEEE 5th international symposium on biomedical imaging: from nano to macro. ISBI; 2008, pp 1577–80.
151. Selvapeter PJ, Hordijk W. Cellular automata for image noise filtering. In: IEEE world congress on nature and biologically inspired computing. NaBIC 2009IEEE; 2009. p. 193–7.
152. Sergeraert F. The computability problem in algebraic topology. Adv Math. 1994;104:1–29.
153. Shapiro LG, Stockman GC. Computer vision. Upper Saddle River: Prentice Hall PTR; 2001.
154. Sharma O, Mioc D, Anton F. Polygon feature extraction from satellite imagery based on color image segmentation and medial axis. The international archives of the photogrammetry, remote sensing and spatial information sciences, XXXVII, Part B3a, Commission III; 2008. p. 235–40.
155. Sheeba F, Thamburaj R, Nagar AK, Mammen JJ. Segmentation of peripheral blood smear images using tissue-like P systems. In: 2011 sixth international conference on bio-inspired computing: theories and applications (BIC-TA); 2011. p. 257–61.
156. Smith SJ, Bourgoin MO, Sims K, Voorhees HL. Handwritten character classification using nearest neighbor in large databases. IEEE Trans Pattern Anal Mach Intell. 1994;16(9):915–9.

157. Sobel IE. Camera models and machine perception. Ph.D. thesis, Dept. of Computer Sciences, Stanford, CA, USA. AAI7102831; 1970.
158. Staal J, Abràmoff MD, Niemeijer M, Viergever MA, van Ginneken B. Ridge-based vessel segmentation in color images of the retina. IEEE Trans Med Imaging. 2004;23(4):501–9.
159. Subramanian K, Saravanan R, Geethalakshmi M, Chandra PH, Margenstern M. P systems with array objects and array rewriting rules. In: Pan L, Păun G, editors. Proceedings of bio-inspired computing—theory and applications conference, BIC-TA 2006, Wuhan, China, September 2006, membrane computing section; 2006. p. 160–67.
160. Subramanian KG. P systems and picture languages. In: Durand-Lose JO, Margenstern M, editors. MCU, vol. 4664., Lecture notes in computer scienceBerlin: Springer; 2007. p. 99–109.
161. Subramanian KG, Ali RM, Nagar AK, Margenstern M. Array P systems and t-communication. Fundamenta Informaticae. 2009;91(1):145–59.
162. Subramanian KG, Isawasan P, Venkat I, Pan L, Nagar A. Array P systems with permitting features. J Comput Sci. 2014;5(2):243–50.
163. Subramanian KG, Pan L, Lee SK, Nagar AK. A P system model with pure context-free rules for picture array generation. Math Comput Model. 2010;52(11–12):1901–9.
164. Subramanian KG, Sriram S, Song B, Pan L. An overview of 2d picture array generating models based on membrane computing. In: Adamatzky A, editor. Reversibility and Universality, Essays Presented to Kenichi Morita on the Occasion of his 70th Birthday. Emergence, complexity and computation, vol. 30. Springer; 2018. p. 333–56.
165. Suzuki S, Abe K. Binary picture thinning by an iterative parallel two-subcycle operation. Pattern Recognit. 1987;20(3):297–307.
166. Snchez-Karhunen E, Valencia-Cabrera L. Membrane computing applications in computational economics. In: Graciani C, Păun G, Riscos-Núñez A, Valencia-Cabrera L, editors. Fifteenth brainstorming week on membrane computing. Sevilla: Fénix Editora; 2017. p. 189–214.
167. Tarabalka Y, Chanussot J, Benediktsson JA. Segmentation and classification of hyperspectral images using Watershed transformation. Pattern Recognit. 2010;43(7):2367–79.
168. Tobias OJ, Seara R. Image segmentation by histogram thresholding using fuzzy sets. IEEE Trans Image Process. 2002;11(12):1457–65.
169. Wang D, Lu H, Zhang J, Liang JZ. 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, vol. 6. 2005. p. 6488–91.
170. Wang H, Peng H, Shao J, Wang T. A thresholding method based on P systems for image segmentation. ICIC Express Lett. 2012;6(1):221–7.
171. Wang PSP. Some new results on isotonic array grammars. Inf Process Lett. 1980;10(3):129–31.
172. Yahya RI, Hasan S, George LE, Alsalibi B. Membrane computing for 2D image segmentation. Int J Adv Soft Comput Appl. 2015;7(1):35–50.
173. Yahya RI, Shamsuddin SM, Hasan S, Yahya SI. Tissue-like P system for segmentation of 2D hexagonal images. ARO Sci J Koya Univ. 2016;IV(1):35–42.
174. Yahya RI, Shamsuddin SM, Yahya SI, Hasan S, Al-Salibi B, Al-Khafaji G. Image segmentation using membrane computing: A literature survey. In: Gong M, Pan L, Song T, Zhang G, editors. Bio-inspired computing: theories and applications—11th international conference, BIC-TA 2016, Xi'an, China, October 28–30, 2016, revised selected papers, part I, communications in computer and information science. Springer; 2016. p. 314–35.
175. Yahya RI, Shamsuddin SM, Yahya SI, Hasan S, Alsalibi B. Automatic 2d image segmentation using tissue-like P system. Int J Adv Soft Comput Appl. 2018;10(1):36–54.
176. Yahya SI, Yahya RI III, Al-Salibi B, Al-Khafaji GK, Shamsuddin SM. Three-dimensional image segmentation using tissue-like P system. ARO Sci J Koya Univ. 2017;2:67–74.
177. Yang X. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N, editors. Nature inspired cooperative strategies for optimization, NICSO 2010, May 12–14, 2010, Granada, Spain, studies in computational intelligence, vol. 4. Springer; 2010. p. 65–74.
178. Yang Y, Peng H, Jiang Y, Huang X, Zhang J. A region-based image segmentation method under P systems. J Inf Comput Sci. 2013;10(10):2943–50.
179. Yazid H, Arof H. Image segmentation using watershed transformation for facial expression recognition. In: IFMBE proceedings, 4th Kuala Lumpur international conference on biomedical engineering; 2008. p. 575–78.
180. Ye QZ, Danielsson PE. Inspection of printed circuit boards by connectivity preserving shrinking. IEEE Trans Pattern Anal Mach Intell. 1988;10(5):737–42.
181. Yuan X, Situ N, Zouridakis G. A narrow band graph partitioning method for skin lesion segmentation. Pattern Recognit. 2009;42(6):1017–28.
182. Zhang G, Gheorghe M, Pan L, Pérez-Jiménez MJ. Evolutionary membrane computing: a comprehensive survey and new results. Inf Sci. 2014;279:528–51.
183. Zhang GX, Gheorghe M, Li Y. A membrane algorithm with quantum-inspired subalgorithms and its application to image processing. Nat Comput. 2012;11(4):701–17.
184. Zhang TY, Suen CY. A fast parallel algorithm for thinning digital patterns. Commun ACM. 1984;27(3):236–9.
185. Zhang Z, Peng H. Object segmentation with membrane computing. J Inf Comput Sci. 2012;9(17):5417–24.
186. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition: a literature survey. ACM Comput Surv. 2003;35(4):399–458.
187. Zhou Y, Chellappa R. Artificial neural networks for computer vision. Research notes in neural computing. Berlin: Springer; 1992.
188. Zitová B, Flusser J. Image registration methods: a survey. Image Vis Comput. 2003;21(11):977–1000.
189. NVIDIA Corporation. NVIDIA CUDA Programming Guide. http://www.nvidia.com/ (2012).

Dr. Daniel Díaz-Pernil obtained his Ph.D at the University of Sevile. He has been member of the topological pattern analysis and recognition resarch group. Author of more than 20 papers in prestigious international journal and reviewer of several international journal. His most important researching lines are related with membrane computing, digital image, pattern recognition and algebraic topology.

Miguel A. Gutiérrez-Naranjo obtained his Ph.D. in Mathematics at the University of Seville. Currently, he is associate professor in the Department of Computer Science and Artificial Intelligence at the University of Seville, Spain. His research interest includes topics related to Artificial Intelligence and Natural Computing, both from a theoretical and practical point of view. He has co-authored more than 40 scientific papers in these areas.

Hong Peng received the BSc degree and the ME degree in Mathematics from Sichuan Normal University, Chengdu, China in 1987 and 1990, and the PhD 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 School of Mathematics and Computer Engineering, Xihua University, China since 2005. His research interests include membrane computing, machine learning and image processing. He has published over 100 scientific papers in international journals and conferences.