Using Membrane Computing for Obtaining Homology Groups of Binary 2D Digital Images

Hepzibah A. Christinal[1,2], Daniel Díaz-Pernil[1], and Pedro Real Jurado[1]

[1] Research Group on Computational Topology and Applied Mathematics
University of Sevilla, Avda. Reina Mercedes s/n, 41012, Sevilla, Spain
[2] Karunya University, Coimbatore, Tamilnadu, India
{hepzi,sbdani,real}@us.es

Abstract. Membrane Computing is a new paradigm inspired from cellular communication. Until now, P systems have been used in research areas like modeling chemical process, several ecosystems, etc. In this paper, we apply P systems to Computational Topology within the context of the Digital Image. We work with a variant of P systems called *tissue-like P systems* to calculate in a general maximally parallel manner the homology groups of 2D images. In fact, homology computation for binary pixel-based 2D digital images can be reduced to connected component labeling of white and black regions. Finally, we use a software called *Tissue Simulator* to show with some examples how these systems work.

Keywords: computational topology, homology groups, membrane computing, P systems.

1 Introduction

Natural Computing studies new computational paradigms inspired from Nature. It abstracts the way in which Nature "computes", conceiving new computing models. There are several fields in Natural Computing that are now well established. To mention a few of these, Genetic Algorithms introduced by J. Holland[22] which is inspired by natural evolution and selection in order to find an optimal solution in a large set of feasible candidate solutions; Neural Networks introduced by W.S. McCulloch and W. Pitts[24] which is based on the interconnections of neurons in the brain; or DNA-based molecular computing, that was initiated when L. Adleman[1] published a solution to an instance of the Hamiltonian path problem by manipulating DNA strands in a lab.

Membrane Computing[1] is a theoretical model of computation inspired by the structure and functioning of cells like living organisms able to process and generate information. The computational devices in Membrane Computing are called *P systems*. Roughly speaking, a P system consists of a membrane structure,

[1] A layman-oriented introduction can be found in [32], a comprehensive presentation can be found in [30] and further updated bibliography in [40]. A presentation of applications can be found in [5].

P. Wiederhold and R.P. Barneva (Eds.): IWCIA 2009, LNCS 5852, pp. 383–396, 2009.

in the compartments of which one places multisets 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. Ever since the seminal paper [29]was introduced, different models of P systems have been studied. According to their architecture, these models can be split into two sets: cell-like P systems and tissue-like P systems[33,7]. In cell-like P systems, membranes are hierarchically arranged in a tree-like structure. The inspiration for such architecture is the set of vesicles inside the cell. All of them perform their biological processes in parallel and life is the consequence of the harmonious conjunction of such processes.

This paper is devoted to the second approach: tissue-like P systems. According to the architecture, the main difference with cell-like P systems is that the structure of membranes is defined by a general graph instead of a tree-like graph. These models were first presented by Martín–Vide *et al.* in [25] and it has two biological inspirations (see [26]): 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 communication among cells is based on symport/antiport rules. This way of communication for P systems was introduced in [31] on the basis of the communication between cells. Symport rules move objects across a membrane together in one direction, whereas antiport rules move objects across a membrane in opposite directions.

On the other hand, homology groups related to the "different" n-dimensional holes (connected component, tunnels, cavities,...)are invariants from Algebraic Topology which are frequently used in Digital Image Analysis and Structural Pattern Recognition. In some sort, they reflect the topological nature of the object in terms of the number and characteristics of its holes. In a binary 2D image, the computation of homology groups can be reduced to a process of black and white connected components labeling. The different black connected components are the generators of the 0-dimensional homology group which is the "black" part of the image. On the other hand, the closed "black" curves surrounding the different white connected components of the image are the generators of its 1-dimensional homology group.

J. Chao and J. Nakayama connected Natural Computing and Algebraic Topology using Neural Networks[4] (extended Kohonen mapping). Moreover, the idea to relate P systems and image processing already appeared in [3,6]. Here, we use for the first time, the power and efficiency of a variant of P systems called tissue-like P systems[8,9] to calculate the homology groups to binary pixel-based 2D images. The parallelism is massive in this model (see [20,23]), so the time used to obtain the homology groups does not depend on the number of black and white connected components, but only on the thickness of them.

The paper is structured as follows: in the next section we present the definition of basic P systems with input. In Section 3, we design two systems for calculating H_0 and H_1 for any binary pixel-based 2D digital image (having $n \times n$ pixels) and we show how both systems calculate the homology groups to two specific

8×8 images in the following section. In final part of the paper, we present some conclusions and future work.

2 Description of a Model of Membranes

In the first definition of tissue P systems in [25,26] the membrane structure did not change along the computation. Based on the cell-like model of P systems with active membranes, Gh. Păun et al. presented in [33] a new model of tissue-like P systems *with cell division*. The biological inspiration is clear: alive tissues are not *static* network of cells, since cells are duplicated via mitosis in a natural way. Díaz-Pernil presented in [7] a formalization of *Tissue-like P systems* (without cellular division), and these are the systems that we use in this paper.

The main features of this model, from the computational point of view, are that cells do not have polarizations (the contrary holds in the cell-like model of P systems, see [30]) and the membrane structure is a general graph, not a tree (i.e., not a cell-like model).

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

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

where

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

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) labeled by $1, 2, \ldots, q$. We will use 0 to refer to the label of the environment, i_Π and o_Π denote the input region and the output region (which can be the region inside a cell or the environment)respectively.

The strings $w_1, \ldots, w_q$ describe the multisets 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 labeled 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 multisets 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.

Now, to understand how we can obtain a computation of one of these P systems we present an example of them:

Consider us the following tissue-like P system

$$\Pi' = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$$

where

1. $\Gamma = \{a, b, c, d, e\}$,
2. $\Sigma = \emptyset$,
3. $\mathcal{E} = \{a, b, e\}$,
4. $w_1 = a^3\, e$, $w_2 = b^2\, c\, d$,
5. $\mathcal{R}$ is the following set of communication rules
   (a) $(1, a/b, 2)$,
   (b) $(2, c/b^2, 0)$,
   (c) $(2, d/e^2, 0)$,
   (d) $(1, e/\lambda, 0)$,
6. $i_\Pi = 1$,
7. $o_\Pi = 0$

We can observe the initial configuration of this system in the Figure 1 (a). We have four rules to apply. First rule is $(1, a/b, 2)$. The rule can be applied whenever an object 'a' is founded in cell 1 and one copy of 'b' appear in cell 2. This rule sends 'a' to cell 2 and 'b' from cell 2 to cell 1. Rule 2 is $(2, c/b^2, 0)$ and implies that when symbol 'c' present in cell 2 then this rule takes two copies of 'b' from environment and sends 'c' to the environment (i.e. cell 0). Rule 3 is similar to rule 2. Rule 4, $(1, e/\lambda, 0)$, sends the object 'e' to the environment. So, as we have 3 copies of 'a' and 1 copy of 'e' in cell 1 and 2 copies of 'b', one copy of 'c' and two copies of 'd' appear in cell 2. Then, all the rules can be applied in a parallel manner. Figure 1(b) show the next configuration of the system after applying the rules. If reader observes the initial elements in the environment of a tissue-like P systems (in this

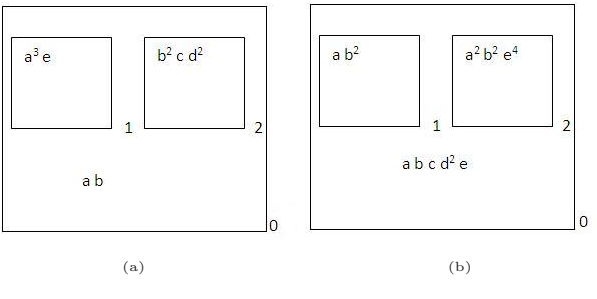

Fig. 1. (a) Initial Configuration of system Π' (b) Following Configuration of Π'

case a, b), one can observe the number of the copies of these elements always appear as one, because we have an arbitrary large amount of copies of them. The only objects changing its number of copies in the environment during a computation are the elements were not appear there initially. In this example, d has two copies because it is not an initial element of the environment.

3 Calculating Homology Groups

In the following, we will try to calculate homology groups, H_0 and H_1, to a digital image of two dimensions. The image is given by multiple pixels forming a $\mathbb{N}^2$ mesh. We suppose each pixel has associated with one of the two possible colors, black and white. Then, the black or white pixel in the position (i, j) is codified by the object B_{ij} or W_{ij}.

H_0 is given by the number of connected components formed by the black pixels and H_1 is given by the number of the holes created by the black pixels; i.e., the number of connected components of white pixels surrounded by black pixels. So, we consider 4-adjacency to see which are the neighboring pixels (if we consider 8-adjacency the systems will be very similar to the systems appear in this paper).

3.1 A Family of Tissue-Like P Systems to Obtain H_0

At this point, we want to know the number of connected components formed by the black pixels. We define a family of tissue-like P systems and for all digital images with n^2 pixels ($n \in \mathbb{N}$) we take a tissue-like P system whose input is given by two types of elements: B_{ij} codifying a black pixel, W_{ij} codifying a white pixel of the input image. The output is given by the number of objects C that appear in the output cell when the system stops (the number of connected components).

Below, we describe the rules of the family of systems in a schematic manner. For each type of rules we show a representative rule. For example, we describe the rules of type 1 as follows:

$$(1, \begin{matrix} & W & K \\ W & B & B \\ & W & B \end{matrix} \quad / \quad \begin{matrix} & W & K \\ W & W & B \\ & W & B \end{matrix}, 0)$$

where K could be B or W.

We describe the rules of each type depending on the position of the black pixels (up, down, left and right) respect to white pixels. For example, with the above schema we represent 8 subtypes of rules of type 1: 4 for each possible position and we must consider the possible values of K.

So, we can define a family of tissue-like P systems to calculate H_0 to any 2D image. For each $n \in \mathbb{N}$ we will consider the tissue-like P system of the family with input of degree 2:

$$\Pi_0(n) = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi),$$

defined as follows

a) $\Gamma = \Sigma \cup \{G_{ij} : 1 \leq i,j \leq n\} \cup \{C\}$,
b) $\Sigma = \{B_{ij}, W_{ij} : 1 \leq i,j \leq n\}$,
c) $\mathcal{E} = \Sigma \cup \{C\}$,
d) $w_1 = \{W_{ij} : (i = 0 \wedge 1 \leq j \leq n) \vee$
$(i = n+1 \wedge 1 \leq j \leq n) \vee$
$(j = 0 \wedge 1 \leq i \leq n) \vee$
$(j = n+1 \wedge 1 \leq i \leq n)\}$,
$w_2 = \emptyset$,

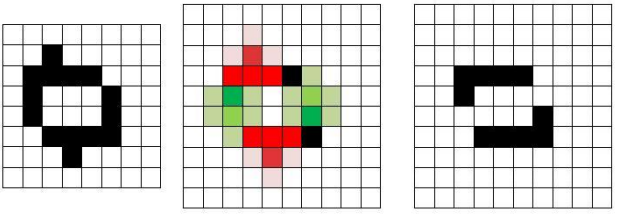

Fig. 2. Cutting branches of two black connected components

e) R is the following set of communication rules:

1. $(1, \begin{matrix} & W & K & \\ W & B & B \\ & W & B & \end{matrix} / \begin{matrix} & W & K & \\ W & W & B \\ & W & B & \end{matrix}, 0)$, where $K = B$ or $K = W$.

2. $(1, \begin{matrix} & W & W & \\ W & B & B \\ & W & W & \end{matrix} / \begin{matrix} & W & W & \\ W & W & B \\ & W & W & \end{matrix}, 0)$

The above two types of rules are used to eliminate single points, i.e. branches of black connected components, as seen in Figure 2 where the necessary pixels to apply two rules of type 1 are colored in red, and then colored in green when pixels are used to apply rules of type 2.

3. $(1, \begin{matrix} & W & W & \\ W & B & B \\ W & B & B & \end{matrix} / \begin{matrix} & W & W & \\ W & W & B \\ W & B & B & \end{matrix}, 0)$

4. $(1, \begin{matrix} & & W & & \\ W & B & B & B \\ & B & W & W & \\ & & W & & \end{matrix} / \begin{matrix} & & W & & \\ W & W & B & B \\ & B & B & W & \\ & & W & & \end{matrix}, 0)$

5. $(1, \begin{matrix} & W & W & W & \\ B & B & B & W \\ & B & W & B & \\ & & W & & \end{matrix} / \begin{matrix} & W & W & W & \\ B & W & W & W \\ & B & B & B & \\ & & W & & \end{matrix}, 0)$

6. $(1, \begin{matrix} & B & B & B & \\ B & W & B \\ & B & B & B & \end{matrix} / \begin{matrix} & B & B & B & \\ B & B & B \\ & B & B & B & \end{matrix}, 0)$

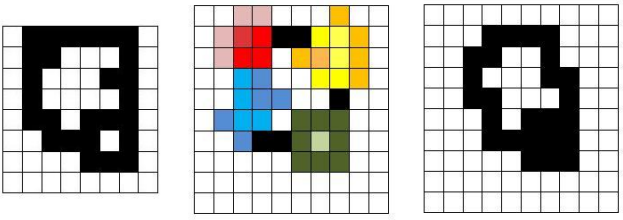

Fig. 3. Reducing black connected components with rules of type 2 to 6

These of rules are used to reduce the dimensions of the black connected components with white pixels inside them, as seen in Figure 3 where we have colored pixels used by rules of types 3 to 6 with different colors: red, blue, yellow and green, respectively.

7. $(1, \begin{matrix} & W & \\ W & B & W \\ & W & \end{matrix} / \begin{matrix} & W & \\ W & G & W \\ & W & \end{matrix}, 0)$

The 7-th type of rule is used when the system reduces the black connected components to only one pixel. So, these rules change the color of the pixel to green (codify with the object G_{ij}).

8. $(1, G_{ij}/W_{ij}\, C, 0)$, for $1 \leq i, j \leq n$

The 8-th type of rules brings an object C and W_{ij} to membrane 1 and sends G_{ij} to the environment.

9. $(1, C/\lambda, 2)$

The 9-th type of rule sends one copy of the object C to the output cell. Then, so much copies of C as connected components of black pixel arrive to cell 2.

f) $i_\Pi = 1$,

g) $o_\Pi = 2$.

Overview of a Computation: Given an image as input data whose size is $n \times n$, there exists a system of this family working in a parallel manner: First, it eliminates the branches of the black connected components that appear in the image in 4 steps. For this, the system uses rules of type 1 and 2. Secondly, the system reduces the size of the black connected components from four directions-up, down, left and right. The system takes the rules of types 3 to 6 to realize this task, and needs a logarithmic number of steps proportionate to the size of the biggest black connected component and reduce each component to only one black pixel. In this manner, we have obtain the complexity of the problem to obtain homology group H_0 of binary 2D digital image using tissue-like P systems.

Complexity and Necessary Resources: Taking account the size of the input data is $O(n^2)$, the amount of necessary resources for defining the systems of our family and the complexity of our problem can be observed in the following table:

H_0 Problem	
Complexity	
Number of steps of a computation	$O(n)$
Necessary Resources	
Size of the alphabet	$3n^2+1$
Initial number of cells	2
Initial number of objects	$4n-3$
Number of rules	$O(n^2)$
Upper bound for the length of the rules	22

3.2 A Family of Tissue-Like P System to Obtain H_1

Now, we want to know the number of white connected components surrounded by one or more black connected components. So, we define a family of tissue-like P systems and for each digital image with n^2 pixels ($n \in \mathbb{N}$), we take a specific tissue-like P system of the family for all the images with size $n \times n$.

For each $n \in \mathbb{N}$ we will consider the tissue-like P system

$$\Pi_1(n) = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi),$$

defined as follows

a) $\Gamma = \Sigma \cup \{b_{ij}, g_{ij}, w_{ij} : 1 \leq i,j \leq n\} \cup \{C\}$,
b) $\Sigma = \{B_{ij}, W_{ij} : 1 \leq i,j \leq n\} \cup \{P_{ij} : (i = 0 \wedge 1 \leq j \leq n) \vee (i = n+1 \wedge 1 \leq j \leq n) \vee (j = 0 \wedge 1 \leq i \leq n) \vee (j = n+1 \wedge 1 \leq i \leq n)\}$,
c) $\mathcal{E} = \Gamma$,
d) $w_1 = \{a_1\} \cup \{P_{ij} : (i = 0 \wedge 1 \leq j \leq n) \vee (i = n+1 \wedge 1 \leq j \leq n) \vee (j = 0 \wedge 1 \leq i \leq n) \vee (j = n+1 \wedge 1 \leq i \leq n)\}$,
$w_2 = \emptyset$,
e) R is the following set of communication rules:

1. $(1, a_i/a_{i+1}^2, 0)$, for $i = 1 \ldots n/2$
It is a counter used to decide when the objects codifying pixels are sent to cell 2.

2. $(1, P\,W\,/\,P\,P, 0)$
The system eliminates all the white pixels (pass to be colored in pink) that are not inside black connected component.

3. $(1, a_{2^{\lceil \lg n \rceil}} K_{ij}/k_{ij}, 0)$, for $1 \leq i, j \leq n$ and $K = B \vee W$.
When the objects $a_{2^{\lceil \lg n \rceil}}$ appear in the cell 1 system sends all the objects codifying the black or white pixels to the cell 2.
The rest of the rules are the same of the system Π_0, but exchanging the white pixels by black pixels and in the other way:

4. $(1, \begin{matrix} & w\ k & \\ w & b\ b & \\ & w\ b & \end{matrix} / \begin{matrix} & w\ k & \\ w\ w & b & , 0) \\ & w\ b & \end{matrix}$, where $k = b$ or $k = w$.

5. $(1, \begin{matrix} & w\ w & \\ w & b\ b & \\ & w\ w & \end{matrix} / \begin{matrix} & w\ w & \\ w\ w & b & , 0) \\ & w\ w & \end{matrix}$

6. $(1, \begin{matrix} & w\ w & \\ w & b\ b & \\ w & b\ b & \end{matrix} / \begin{matrix} & w\ w & \\ w\ w & b & , 0) \\ w & b\ b & \end{matrix}$

7. $(1, \begin{matrix} & w & \\ w & b\ b\ b & \\ & b\ w\ w & \\ & w & \end{matrix} / \begin{matrix} & w & \\ w\ w & b\ b & , 0) \\ & b\ b\ w & \\ & w & \end{matrix}$

8. $(1, \begin{matrix} & w\ w\ w & \\ b & b\ b\ w & \\ & b\ w\ b & \\ & w & \end{matrix} / \begin{matrix} & w\ w\ w & \\ b & w\ w\ w & , 0) \\ & b\ b\ b & \\ & w & \end{matrix}$

9. $(1, \begin{matrix} b\ b\ b \\ b\ w\ b \\ b\ b\ b \end{matrix} / \begin{matrix} b\ b\ b \\ b\ b\ b\ , 0) \\ b\ b\ b \end{matrix}$

10. $(1, \begin{matrix} & w & \\ w & b & w \\ & w & \end{matrix} / \begin{matrix} & w & \\ w & g & w\ , 0) \\ & w & \end{matrix}$

11. $(1, g_{ij}/w_{ij}\, C, 0)$, for $1 \leq i, j \leq n$
12. $(1, C/\lambda, 2)$

f) $i_\Pi = 1$,
g) $o_\Pi = 2$.

Overview of a Computation: Using a tissue-like P system, to compute H_1 of a digital image is similar to compute H_0. There exists a system of this family working in a parallel manner: First, it takes the white pixels not contained in black connected components and transforms these pixels in pink (type of rules 2). Using the counter a_i, white and black pixels are transformed in other objects (small letters) in $n/2+1$ steps (types of rules 1 and 3). In this form, we can apply the rest of rules (those similar to Π_0). So system eliminates the branches of the white connected components that appear in the image in 4 steps. For this, the system uses types of rules 4 and 5. Then, the system reduces the size of the white

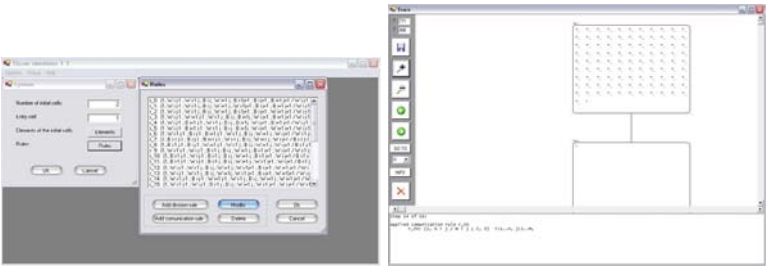

Fig. 4. Two images about Tissue Simulator

connected components from four directions: up, down, left and right. The system takes the rules of types 6 to 9 to realize this task, and needs a logarithmic number of steps proportionate to the size of the biggest white connected component and reduce each component to only one white pixel (less than $O(\mathrm{n})$). In this manner, we have obtained the complexity of the problem to obtain homology group H_1 of binary 2D digital image using tissue-like P systems.

Complexity and Necessary Resources: Taking account the size of the input data is $O(n^2)$, the amount of necessary resources to construct the tissue-like P systems of our family and the cellular complexity respect to time of our problem can be observed in the following table:

H_1 Problem	
Complexity	
Number of steps of a computation	$O(n)$
Necessary Resources	
Size of the alphabet	$5n^2 + 4n - 2$
Initial number of cells	2
Initial number of objects	$4n - 2$
Number of rules	$O(n^2)$
Upper bound for the length of the rules	22

4 Some Examples

In this section, we check the tissue-like P systems in section 3 above with some images that appear in Figure using a specific sequential software, called *Tissue Simulator* (see [39]) and developed by R. Borrego-Ropero et al. in [2]. This software was developed to help researchers to understand how these systems obtain a possible computation. Although, this program was developed in Java, it was not meant to be used in Digital Image. So, we do not work with images

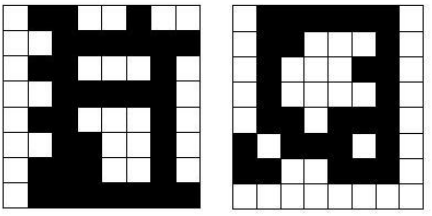

Fig. 5. Two images to check

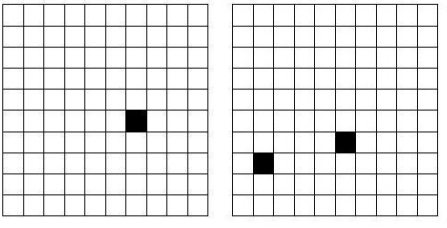

Fig. 6. Number of black connected components in the previous images

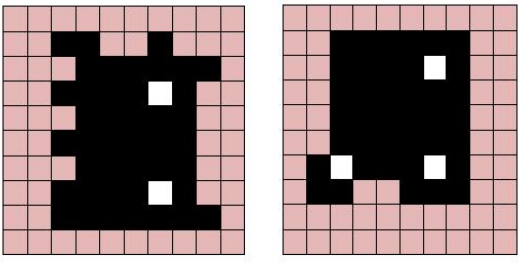

Fig. 7. Number of white holes in the previous images

directly, but we work with elements (of an alphabet) codifying the pixels of an image, and the output is given by these elements (see Figure 4).

First, we are going to obtain the different connected components for the images given by Figure 5.

After a logarithmic number of steps with respect to the input data, the Tissue Simulator stops and gives the output data that appears in the output cell (cell 2) of the system Π_1 (created to calculate the number of black connected components). This output is given using elements codifying the images which are shown in the Figure 6.

On the other hand, using a logarithmic number of steps again with respect to the input data, Tissue Simulator calculates the number of white connected

components inside black connected components. It is shown in Figure 7, the output codified by the elements that appear in the output cell for each one of images of Figure 5.

5 Conclusions and Future Work

We have shown in this paper that the homology for 2D digital objects (using 4-connectivity or 8-connectivity for neighbor pixels) can be efficiently obtained using P systems. The most important issue we want to deal with the near future, is to use P systems for getting homological and cohomological information (Reeb graphs[17], AT-models ([13,14,15,16,18]), homology gradient vector field [28,27,37,36], representative (co)cycles of (co)homology generators [19,11,12], cohomology algebra [21,11], cohomology operations [38,10], torsion numbers [21], homotopy groups [34,35] for $3D$ and $4D$ geometric objects. The complexity in time for most of the algorithms previously cited ranges from linear (for connected component labeling), passing through cubical (for homology gradient vector fields or homology groups), and $O(n^5)$ for cohomology algebra and cohomology operations, up to exponential and more in the case of homotopy groups. The predictable drastic improvements in complexity that P-systems could mean in Computational Algebraic Topology methods. This would allow in the future to handle with optimism the computation processes of these complex topological invariants.

Acknowledgement

The first author acknowledge the support of the project "Computational Topology and Applied Mathematics" PAICYT research project FQM-296. The second author acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200. The third author acknowledge the support of the project MTM2006-03722 of the Ministerio español de Educación y Ciencia and the project PO6-TIC-02268 of Excellence of Junta de Andalucía.

References

1. Adleman, L.M.: Molecular computations of solutions to combinatorial problems. Science 226, 1021–1024 (1994)
2. Borrego–Ropero, R., Díaz–Pernil, D., Pérez–Jiménez, M.J.: Tissue Simulator: A Graphical Tool for Tissue P Systems. In: Proc. International Workshop, Automata for Cellular and Molecular Computing, MTA SZTAKI, Budapest, pp. 23–34 (2007)
3. Ceterchi, R., Mutyam, M., Păun, G., Subramanian, K.G.: Array-rewriting P systems. Natural Computing 2, 229–249 (2003)
4. Chao, J., Nakayama, J.: Cubical Singular Simples Model for 3D Objects and Fast Computation of Homology Groups. In: Proc. ICPR 1996, pp. 190–194. IEEE, Los Alamitos (1996)

5. Ciobanu, G., Păun, G., Pérez–Jiménez, M.J. (eds.): Applications of Membrane Computing. Springer, Heidelberg (2005)
6. Dersanambika, K.S., Krithivasan, K., Subramanian, K.G.: P Systems Generating Hexagonal Picture Languages. In: Martín-Vide, C., Mauri, G., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2003. LNCS, vol. 2933, pp. 168–180. Springer, Heidelberg (2004)
7. Díaz–Pernil, D.: Sistemas P de Tejido: Formalización y Eficiencia Computacional. PhD Thesis, University of Seville (2008)
8. Díaz–Pernil, D., Gutiérrez, M.A., Pérez–Jiménez, M.J., Riscos–Núñez, A.: A uniform family of tissue P systems with cell division solving 3-COL in a linear time. Theoretical Computer Science 404(1–2), 76–87 (2008)
9. Díaz–Pernil, D., Pérez–Jiménez, M.J., Romero, A.: Computational efficiency of cellular division in tissue-like P systems. Romanian Journal of Information Science and Technology 11(3), 229–241 (2008)
10. González Díaz, R., Real, P.: Computation of Cohomology Operations of Finite Simplicial Complexes. Homology Homotopy and Applications 2, 83–93 (2003)
11. Gonzalez-Diaz, R., Real, P.: Towards Digital Cohomology. In: Nyström, I., Sanniti di Baja, G., Svensson, S. (eds.) DGCI 2003. LNCS, vol. 2886, pp. 92–101. Springer, Heidelberg (2003)
12. Gonzalez-Diaz, R., Real, P.: On the cohomology of 3D digital images. Discrete Applied Mathematics 147, 245–263 (2005)
13. Gonzalez-Diaz, R., Medrano, B., Real, P., Sanchez-Pelaez, J.: Algebraic Topological Analysis of Time-sequence of Digital Images. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2005. LNCS, vol. 3718, pp. 208–219. Springer, Heidelberg (2005)
14. Gonzalez-Diaz, R., Medrano, B., Real, P., Sánchez, J.: Reusing Integer Homology Information of Binary Digital Images. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 199–210. Springer, Heidelberg (2006)
15. Gonzalez-Diaz, R., Medrano, B., Real, P., Sánchez-Peláez, J.: Simplicial Perturbation Technique and Effective Homology. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2006. LNCS, vol. 4194, pp. 166–177. Springer, Heidelberg (2006)
16. Gonzalez-Diaz, R., Jiménez, M.J., Medrano, B., Real, P.: Extending AT-Models for Integer Homology Computation. In: Escolano, F., Vento, M. (eds.) GbRPR 2007. LNCS, vol. 4538, pp. 330–339. Springer, Heidelberg (2007)
17. Gonzalez-Diaz, R., Jiménez, M.J., Medrano, B., Real, P.: A Graph-with-Loop Structure for a Topological Representation of 3D Objects. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 506–513. Springer, Heidelberg (2007)
18. Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B., Molina-Abril, H., Real, R.: Integral Operators for Computing Homology Generators at Any Dimension. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 356–363. Springer, Heidelberg (2008)
19. Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B., Real, P.: Chain homotopies for object topological representations. Discrete Applied Mathematics 157, 490–499 (2009)
20. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Riscos-Núñez: On the degree of parallelism in membrane systems. Theoretical Computer Science 372, 183–195 (2007)
21. Hatcher, A.: Algebraic Topology. Cambridge Univ. Press, Cambridge (2001)
22. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
23. Loos, R., Nagy, B.: Parallelism in DNA and Membrane Computing. In: CiE, Local Proceedings, pp. 283–287 (2007)

24. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943)
25. Martín–Vide, C., Pazos, J., Păun, G., Rodríguez Patón, A.: A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. In: Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, pp. 290–299. Springer, Heidelberg (2002)
26. Martín–Vide, C., Pazos, J., Păun, G., Rodríguez Patón, A.: Tissue P systems. Theoretical Computer Science 296, 295–326 (2003)
27. Molina-Abril, H., Real, P.: Cell AT-models for digital volumes. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 314–323. Springer, Heidelberg (2009)
28. Molina-Abril, H., Real, P.: Advanced Homological information on 3D Digital volumes. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 361–371. Springer, Heidelberg (2008)
29. Păun, G.: Computing with membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000)
30. Păun, G.: Membrane Computing. An Introduction. Springer, Berlin (2002)
31. Păun, A., Păun, G.: The power of communication: P systems with symport/antiport. New Generation Computing 20(3), 295–305 (2002)
32. Păun, G., Pérez–Jiménez, M.J.: Recent computing models inspired from biology: DNA and membrane computing. Theoria 18(46), 72–84 (2003)
33. Păun, G., Pérez–Jiménez, M.J., Riscos–Núñez, A.: Tissue P System with cell division. In: Second Brainstorming Week on Membrane Computing, Sevilla, Report RGNC 01/2004, pp. 380–386 (2004)
34. Real, P.: An Algorithm Computing Homotopy Groups. Mathematics and Computers in Simulation 42(4-6), 461–465 (1996)
35. Real, P.: Homological Perturbation Theory and Associativity. Homology Homotopy and Applications, pp. 51–88 (2000)
36. Real, P.: Connectivity forests for homological analysis of digital volumes. In: Cabestany, J., et al. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 415–423. Springer, Heidelberg (2009)
37. Real, P., Molina-Abril, H., Kropatsch, W.: Homological tree-based strategies for image analysis. In: Computer Analysis and Image Patterns, CAIP (2009)
38. Sergeraert, F.: The computability problem in algebraic topology. Advances in Mathematics 104, 1–29 (1994)
39. The Tissue Simulator Website: `http://www.tissuesimulator.es.kz`
40. The P Systems Website: `http://ppage.psystems.eu/`