Array Tissue-like P Systems

Hepzibah A. Christinal[1], Daniel Díaz-Pernil[1],
Miguel A. Gutiérrez-Naranjo[2], Mario J. Pérez-Jiménez[2]

[1] Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics I
{hepzi,sbdani}@us.es

[2] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
{magutier,marper}@us.es

University of Sevilla, Avda. Reina Mercedes s/n, 41012, Sevilla, Spain

Summary. Array grammars have been studied in the framework of Membrane Computing by using rewriting rules from transition P systems. In this paper we present a new approach to dealing with array grammars by using tissue-like P systems and present an application to the segmentation of images in two dimensional computer graphics.

1 Introduction

Array grammars 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. They have been widely studied and have a large tradition in the literature (see, e.g. [2, 6, 16, 22]).

Recently, Membrane Computing has also approximated to array grammars by setting bridges between both areas (see, e.g. [1, 14, 20]). The basic idea in such approaches is considering an array (i.e., a finite set of objects placed in points of the plane with integer coordinates) as a P system object and using rewriting rules of the type used in transition P systems [13] for replacing it. The type of rule used is $x \to y(tar)$ where $x \to y$ is a context-free rule and $tar \in here, out, in$ is the target which indicates the membrane where the generated object will be placed. Such rewriting rules capture the idea of *array production* $p : \mathcal{A} \to \mathcal{B}$ with $\mathcal{A}$ and $\mathcal{B}$ arrays.

In this paper we present a new approach for linking Membrane Computing to array grammars. Instead of using transition P systems to handle the arrays we propose to use tissue-like P systems. This approach allows us to use the power of symport-antiport rules for designing Membrane Computing algorithms which deal with array objects. In such P system model, the rules are of type $(i, u/v, j)$ with the following interpretation: If the multiset u occurs in a membrane with label i and the multiset v occurs in a membrane with label j, both multiset can be interchanged. We consider an extension of this type of rules. We will consider that two arrays A and B can appear (respectively) in the multisets u and v. The semantics of such rule will be explained below, but the intuition is that the arrays in the membranes i and j will be partially modified.

As a case study, we present an application of array tissue-like P systems to the *Segmentation Problem* in computer vision.

Segmentation in computer vision (see [8]), refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze for an human. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

In the literature, there exists different techniques to segment an image. Some of them are clustering methods [23], histogram-based methods [21], Watershed transformation methods [25] or graph partitioning methods [24]. Some of the practical applications of image segmentation are medical imaging [23], the study of anatomical structure, locate objects in satellite images (roads, forests, etc.) [19] or face recognition [7] among others.

The paper is organized as follows: First we briefly recall some basic definitions related to graphs and multisets and introduce our definition of pixel and array. Next, we introduce a new P system model called *Array tissue-like P systems* on the basis of *tissue P systems*. In Section 4, this P system model is used to find a solution to the segmentation problem in Digital Image.

2 Definitions

An *alphabet*, Σ, is a non-empty set, whose elements are called *symbols*. An ordered sequence of symbols is a *string*. The number of symbols in a string u is the *length* of the string, and it is denoted by $|u|$. As usual, the empty string (of length 0) is denoted by λ. The set of strings of length n built with symbols from the alphabet Σ is denoted by Σ^n and $\Sigma^* = \cup_{n\geq 0}\Sigma^n$. A *language* over Σ is a subset of Σ^*. A *multiset* over a set A is a pair (A, f) where $f:\ A \rightarrow \mathbb{N}$ is a mapping. If $m = (A, f)$ is a multiset then its support is defined as $supp(m) = \{x \in A \mid f(x) > 0\}$ and its *size* is defined as $\sum_{x\in A} f(x)$. A multiset is empty (resp. finite) if its support is the empty set (resp. finite). If $m = (A, f)$ is a finite multiset over A, then it is denoted by $m = a_1^{f(a_1)} a_2^{f(a_2)} \cdots a_k^{f(a_k)}$, where $supp(m) = \{a_1, \ldots, a_k\}$, and for each element a_i, $f(a_i)$ is called the multiplicity of a_i. An *undirected graph* G is a pair $G = (V, E)$ where V is the set of vertices and E is the set of edges, each one of which is an (unordered) pair of (different) vertices. If $\{u, v\} \in E$, we say that u is *adjacent* to v (and also v is *adjacent* to u). The *degree* of $v \in V$ is the number of adjacent vertices to v. In what follows we assume that the reader is already familiar with the basic notions and the terminology underlying P systems[3].

Next, we give a formalization of the arrays considered in this paper.

Definition 1. *Given a finite set V, called an* alphabet of colors*, a* pixel *on V is a pair $\langle \mathbf{x}, v\rangle$ such that $\mathbf{x} \in \mathbb{Z}^2$ and $v \in V$. An* array *on V, A, is a finite set of pixels such that if $\langle \mathbf{x}_1, v_1\rangle, \langle \mathbf{x}_2, v_2\rangle \in A$ and $v_1 \neq v_2$ then $\mathbf{x}_1 \neq \mathbf{x}_2$. Finally, the* support *of the array A is the set $supp(A) = \{\mathbf{x} \in \mathbb{Z}^2 \mid \exists v \in V$ such that $\langle \mathbf{x}, v\rangle \in A\}$.*

Given an array A and $\mathbf{z} \in \mathbb{Z}^2$, we will denote by $A + \mathbf{z}$ the set

$$A + \mathbf{z} = \{\langle \mathbf{x} + \mathbf{z}, v\rangle \mid \langle \mathbf{x}, v\rangle \in A\}$$

Example 1. Let $V = \{R, G, B\}$ be the alphabet of colors and A the array on V $A = \{\langle(3,2), R\rangle, \langle(3,3), G\rangle, \langle(5,5), G\rangle\}$. Let us consider $\mathbf{z} = (-2,1) \in \mathbb{Z}^2$. The array $A + \mathbf{z}$ is $\{\langle(1,3), R\rangle, \langle(1,4), G\rangle, \langle(3,6), G\rangle\}$.

If there are no confusion about the alphabet of colors, we will omit it and we talk about *pixels*. As usual, we will denote by V^{*2} the set of all two dimensional arrays over V.

3 Array Tissue-like P Systems

In the initial definition of the cell-like model of P systems [12], membranes are hierarchically arranged in a tree-like structure. Its biological inspiration comes from the morphology of cells, where small vesicles are surrounded by larger ones. This biological structure can be abstracted into a tree-like graph, where the root represents the skin of the cell (i.e. the outermost membrane) and the leaves represent membranes that do not contain any other membrane (elementary membranes). Besides, two nodes in the graph are connected if they represent two membranes such that one of them contains the other one.

In *tissue P systems*, the tree-like membrane structure is replaced by a general graph. This model has two biological inspirations (see [9, 10]): intercellular communication and cooperation between neurons. The common mathematical model of these two mechanisms is a net of processors dealing with symbols and communicating these symbols along channels specified in advance. The communication among cells is based on symport/antiport rules, which were introduced as communication rules for P systems in [11]. In symport rules, objects cooperate to traverse a membrane together in the same direction, whereas in the case of antiport rules, objects residing at both sides of the membrane cross it simultaneously but in opposite directions. 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

[3] We refer to [13] for basic information in this ares, to [15] for a comprehensive presentation and the web site [26] for the up-to-date information.

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. $w_1, \dots, w_q$ 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 $i, j \in \{0, 1, 2, \dots, q\}, i \neq j,\ u, v \in \Gamma^*$,
6. $i_\Pi \in \{0, 1, 2, \dots, q\}$,
7. $o_\Pi \in \{0, 1, 2, \dots, q\}$.

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 $1, 2, \dots, q$. 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 $w_1, \dots, w_q$ 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.

In order 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 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.

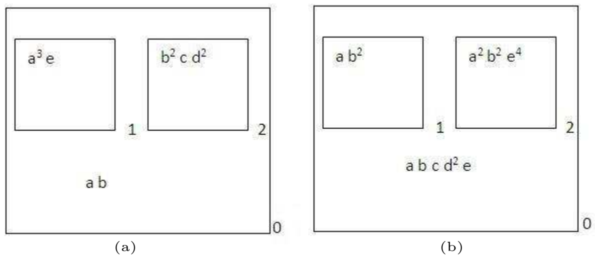

Fig. 1. (a) Initial Configuration of system Π' (b) Following Configuration of Π'

Next, we introduce a modification of this model in order to deal with arrays. An *array tissue-like P system* of degree $q \geq 1$ with input is a tuple of the form

$$\Pi = (\Gamma, V, \mathcal{E}, w_0, w_1, \dots, w_q, A_1, \dots, A_q, \mathcal{R}, i_\Pi, o_\Pi),$$

where

1. Γ is a finite *alphabet*, whose symbols will be called *objects*,
2. V is the *alphabet of colors* verifying $V \cap \Gamma = \emptyset$.
3. $\mathcal{E}$ is a finite subset of arrays on V.
4. $w_0, w_1, \dots, w_q$ are strings over Γ representing the multisets of objects associated with the cells at the initial configuration,
5. $A_1, \dots, A_n$ are arrays on V, placed on the corresponding cells at the initial configuration.
6. $\mathcal{R}$ is a finite set of communication rules of the following form: $(i, u_i W_i / u_j W_j, j)$, for $i, j \in \{0, 1, 2, \dots, q\}, i \neq j$, $u_i, u_j \in \Gamma^*$ and W_i, W_j two arrays on V.

7. $i_\Pi \in \{0, 1, 2, \ldots, q\}$ is the input cell.
8. $o_\Pi \in \{0, 1, 2, \ldots, q\}$ is the output cell.

In a similar way to tissue-like P systems, an *array 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 w_0 is the set of objects placed in the environment, each one of them available in an arbitrary large amount of copies.

For each $i \in \{1, \ldots, q\}$, each A_i is an array placed in the cell i in the initial configuration and $\mathcal{E}$ is the set of arrays placed in the environment, each one of them available in an arbitrary large amount of copies. The empty array $\emptyset$ always belongs to $\mathcal{E}$. For all the non-empty copies, we will consider that the leftmost pixel of the bottom row in the array corresponds to the coordinates $(0, 0)$.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered), regardless if the environment is involved or not. 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.

The main difference with respect tissue-like P systems is related to the application of the rules.

Definition 2. *Let us consider two index i, j such that $i \neq 0 \neq j$ and two non-empty arrays W_i and W_j. The communication rule $(i, u_i W_i / u_j W_j, j)$ is* applicable *over two cells labeled by i and j if the following conditions are verified:*

- *u_i is contained in cell i and u_j is contained in cell j*
- *There exist two arrays, A_i in the cell i and A_j in the cell j and two pairs $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{Z}^2$ such that*
  (a) $W_i + \mathbf{z}_1 \subseteq A_i$
  (b) $W_j + \mathbf{z}_2 \subseteq A_j$
  (c) $supp(W_i) \cap supp(W_j) \neq \emptyset$
  (d) $supp(A_i - (W_i + \mathbf{z}_1)) \cap supp(W_j + \mathbf{z}_1) = \emptyset$
  (e) $supp(A_j - (W_j + \mathbf{z}_2)) \cap supp(W_i + \mathbf{z}_2) = \emptyset$

The application of this rule means that the objects of the multisets represented by u_i and u_j are interchanged between the two cells. The arrays, A_i in the cell i and A_j in the cell j are substituted by A'_i and A'_j respectively, where

$$A'_i = (A_i - (W_i + \mathbf{z}_1)) \cup (W_j + \mathbf{z}_1) \qquad A'_j = (A_j - (W_j + \mathbf{z}_2)) \cup (W_i + \mathbf{z}_2)$$

Note that if either A_i or A_j is the empty array, then the rule is not applicable.

Example 2. Let us suppose that we have two cells with labels 1 and 2 with the following objects and arrays, $[\, z_2c_3A_1\,]_1$ and $[\, d_3k_3bA_2\,]_2$, with z_2, c_3, d_3, k_3, b objects and A_1, A_2 arrays over $\{R, B, G\}$

$$\begin{aligned} A_1 &= \{\langle(1,1),G\rangle, \langle(1,2),G\rangle, \langle(2,2),R\rangle, \langle(2,3),B\rangle\} \\ A_2 &= \{\langle(5,5),G\rangle, \langle(6,5),G\rangle, \langle(6,6),G\rangle\} \end{aligned}$$

Let us consider the rule $r_1 \equiv (1,\, z_2W_1 \,/\, d_3k_3W_2,\, 2)$ where W_1 and W_2 are the arrays

$$\begin{aligned} W_1 &= \{\langle(7,0),G\rangle, \langle(7,1),G\rangle, \langle(8,1),R\rangle\} \\ W_2 &= \{\langle(7,1),G\rangle, \langle(8,1),G\rangle\} \end{aligned}$$

We will check that r_1 is applicable to the cells 1 and 2

- z_2 is contained in the cell 1 and d_3k_3 is contained in the cell 2.
- Let us consider $\mathbf{z_1} = (-6,1) \in \mathbb{Z}^2$ and $\mathbf{z_2} = (-2,4) \in \mathbb{Z}^2$
  (a) $W_1 + \mathbf{z}_1 = \{\langle(1,1),G\rangle, \langle(1,2),G\rangle, \langle(2,2),R\rangle\} \subseteq A_1$
  (b) $W_2 + \mathbf{z}_2 = \{\langle(5,5),G\rangle, \langle(6,5),G\rangle\} \subseteq A_2$
  (c) $supp(W_i) \cap supp(W_j) = \{((7,0),(7,1),(8,1)\} \cap \{(7,1),(8,1)\} \neq \emptyset$
  (d) $A_1 - (W_1 + \mathbf{z_1}) = \{\langle(2,3),B\rangle\}$ and $W_2 + \mathbf{z}_1 = \{\langle(1,2),G\rangle, \langle(2,2),G\rangle\}$. By considering their supports we have $supp(A_1 - (W_1 + \mathbf{z_1})) = \{(2,3)\}$ and $supp(W_2 + \mathbf{z}_1) = \{(1,2),(2,2)\}$, then

  $$supp(A_1 - (W_1 + \mathbf{z_1})) \cap supp(W_2 + \mathbf{z}_1) = \emptyset$$

  (e) $A_2 - (W_2 + \mathbf{z_2}) = \{\langle(6,6),G\rangle\}$ and $W_1 + \mathbf{z}_2 = \{\langle(5,4),G\rangle, \langle(5,5),G\rangle, \langle(6,5),R\rangle\}$. By considering their supports we have $supp(A_2 - (W_2 + \mathbf{z_2})) = \{(6,6)\}$ and $supp(W_1 + \mathbf{z}_2) = \{(6,4),(5,5),(6,5)\}$, then

  $$supp(A_2 - (W_2 + \mathbf{z_2})) \cap supp(W_1 + \mathbf{z}_2) = \emptyset$$

The rule r_1 is applicable to the cells 1 and 2, and the result of applying the rule is $[\, d_3k_3c_3A_1'\,]_1$ and $[\, z_2bA_2'\,]_2$ where

$$\begin{aligned} A_1' &= (A_1 \,-\, (W_1 + \mathbf{z}_1)) \cup (W_2 + \mathbf{z}_1) \\ &= \{\langle(2,3),B\rangle, \langle(1,2),G\rangle, \langle(2,2),G\rangle\} \\ A_2' &= (A_2 \,-\, (W_2 + \mathbf{z}_2)) \cup (W_1 + \mathbf{z}_2) \\ &= \{\langle(6,6),G\rangle, \langle(5,4),G\rangle, \langle(5,5),G\rangle, \langle(6,5),R\rangle\} \end{aligned}$$

Next, we define the applicability of a rule if one of the regions involved is the environment and the arrays are not empty.

Definition 3. *Let us consider an index $i \neq 0$ and two non-empty arrays W_i and W_0. The communication rule $(i, u_iW_i/u_0W_0, 0)$ is* applicable *over two cells labeled by i and 0 if the following conditions are verified:*

- *u_i is contained in cell i and u_0 is contained in cell 0*
- *There exist an array A_i in the cell i and two pairs $\mathbf{z}_i, \mathbf{z}_0 \in \mathbb{Z}^2$ such that*

(a) $W_i + \mathbf{z}_i \subseteq A_i$
(b) $supp(W_i + \mathbf{z}_i) \cap supp(W_0 + \mathbf{z}_0) \neq \emptyset$
(c) $supp(A_i \;-\; (W_i + \mathbf{z}_i)) \cap supp(W_0 + \mathbf{z}_0) = \emptyset$

The application of this rule means that the objects of the multisets represented by u_i is removed from the cell i and substituted by the multiset represented by u_0. The arrays, A_i in the cell i is substituted by A'_i where

$$A'_i = (A_i \;-\; (W_i + \mathbf{z}_i)) \cup (W_0 + \mathbf{z}_0)$$

Example 3. Let us suppose the cell 1 with the following objects and arrays, $[\, z_2^3 c_3 A_1 \,]_1$ and A_1 the array over $\{R, B, G\}$

$$\begin{aligned} A_1 = \{\ & \langle(5,2),R\rangle, \langle(6,2),B\rangle, \langle(7,2),G\rangle, \langle(8,2),B\rangle\} \\ & \langle(9,2),R\rangle, \langle(6,1),B\rangle, \langle(8,1),B\rangle\} \end{aligned}$$

Let us consider the rule $r_1 \equiv (1,\, z_2 W_1 \,/\, d_2 W_0,\, 0)$ where W_1 and W_0 are the arrays

$$\begin{aligned} W_i &= \{\langle(3,3),B\rangle, \langle(3,4),B\rangle\} \\ W_0 &= \{\langle(0,0),R\rangle\} \end{aligned}$$

Let us suppose that d_2 belongs to w_0 and W_0 belongs to $\mathcal{E}$. In order to prove that r_1 is applicable, first we check that z_2 is contained in the cell 1 and, according to the previous claim, d_2 is contained in the environment.

We have several possibilities to choose the pair $\mathbf{z}_i, \mathbf{z}_0$. The different choices show the no determinism of the system. We also apply the rule with maximal parallelism. In this case we take the following option: the pair $\mathbf{z}_i, \mathbf{z}_0$ with $\mathbf{z}_i = (3,-2)$ and $\mathbf{z}_0 = (6,1)$ for the first application of the rule and the pair $\mathbf{z}_i^*, \mathbf{z}_0^*$ with $\mathbf{z}_i^* = (5,-2)$ and $\mathbf{z}_0^* = (8,2)$ for the second application.

(a) $W_1 + \mathbf{z}_i = \{\langle(6,1),B\rangle, \langle(6,2),B\rangle\} \subseteq A_1$
(a) $W_1 + \mathbf{z}_i^* = \{\langle(8,1),B\rangle, \langle(8,2),B\rangle\} \subseteq A_1$
(b) $supp(W_i + \mathbf{z}_i) \cap supp(W_0 + \mathbf{z}_0) = \{(6,1),(6,2)\} \cap \{(6,1)\} \neq \emptyset$
(b) $supp(W_i + \mathbf{z}_i^*) \cap supp(W_0 + \mathbf{z}_0^*) = \{(8,1),(8,2)\} \cap \{(8,1)\} \neq \emptyset$
(c) $supp(A_i \;-\; (W_i + \mathbf{z}_i)) \cap supp(W_0 + \mathbf{z}_0) = \{(5,2),(7,2),(8,2),(9,2),(8,1)\} \cap \{(6,1)\} = \emptyset$
(c) $supp(A_i \;-\; (W_i + \mathbf{z}_i)) \cap supp(W_0 + \mathbf{z}_0) = \{(5,2),(6,2)(7,2),(9,2),(861)\} \cap \{(8,2)\} = \emptyset$

The rule r_1 is applicable and the result of applying the rule twice is $[d_2^2 z_2 c_3 A'_1\,]_1$ where

$$\begin{aligned} A'_1 &= (A_1 \;-\; (W_1 + \mathbf{z}_1) - \;(W_1 + \mathbf{z}_1)^*) \cup (A_0 + \mathbf{z}_0) \cup (A_0 + \mathbf{z}_0^*) \\ &= \{\langle(5,2),R\rangle, \langle(7,2),G\rangle, \langle(9,2),R\rangle, \langle(6,1),R\rangle, \langle(8,2),R\rangle\} \end{aligned}$$

Finally, let us consider the case in which one of the regions involved in the rule is the environment and the array considered in the environment is the empty array.

Definition 4. *The communication rule* $(i, u_i W_i / u_0, 0)$ *is* applicable *over two cells labeled by* i *and* 0 *if the following conditions are verified:*

- u_i *is contained in cell* i *and* u_0 *is contained in cell* 0
- *There exist an array* A_i *in the cell* i *and a pair* $\mathbf{z}_i \in \mathbb{Z}^2$ *such that* $W_i + \mathbf{z}_i \subseteq A_i$

The application of this rule means that the objects of the multisets represented by u_i *is removed from the cell* i *and substituted by the multiset represented by* u_0*. The array* A_i *in the cell* i *is substituted by* A'_i *where* $A'_i = (A_i - (W_i + \mathbf{z}_i))$

Example 4. Let us suppose the cell 1 with the following objects and arrays, $[\, z_2^3 c_3 A_1 \,]_1$ and A_1 the array over $\{R, B, G\}$

$$A_1 = \{ \langle (5,2), R \rangle, \langle (6,2), B \rangle, \langle (7,2), G \rangle, \langle (8,2), B \rangle \} \\ \langle (9,2), R \rangle, \langle (6,1), B \rangle, \langle (8,1), B \rangle \}$$

Let us consider the rule $r_1 \equiv (1, W_1 \,/\, d, 0)$ where W_1 is the array $W_i = \{\langle (3,3), B \rangle\}$. Let us suppose that d belongs to w_0. In this case, we have four possibilities to choose $\mathbf{z}_i$. They are $(3,-2), (3-1), (5,-2), (5,-1)$. It is trivial to check that the rule is applicable. It will be applied with maximal parallelism, so the rule will be applied four times and the result of applying the rule twice is $[d^4 z_2^3 c_3 A'_1 \,]_1$ where

$$A'_1 = \{\langle (5,2), R \rangle, \langle (7,2), G \rangle, \langle (9,2), R \rangle\}$$

4 Using Array Tissue-like P Systems in Digital Image

In digital image terminology, given a finite *alphabet of colors* V and a *blank* object $\#$ such that $\# \notin V$, a two-dimensional (2D) digital image is a pair (S, A_S), where $S \subset \mathbb{N}^2$ and $A_S : S \rightarrow V \cup \{\#\}$ is an array on S. The size of V, $|V|$, is the number of its elements. Moreover, we can introduce an order of colors in an image. We define the *ordered alphabet associate to an image* like a pair $(V, <_V)$, where $<_V$ is an order in the set V.

The definition of *pixel* is associated with arrays, i.e., with equivalence classes of arrays. In this way, it makes sense that we study the adjoining relation of two pixels of generic positions (i, j) and (i', j') by exploring the relation among these generic coordinates. For the sake of simplicity, we write the pixel $< (i,j), a >$ as a_{ij}. There exists two natural way of defining adjacent pixels: 4-adjacency and 8-adjacency [17, 18].

In the first case, given a pixel K_{ij}, the list of adjacent pixels to this is $\{K_{ij-1}, K_{ij+1}, K_{i-1j}, K_{i+1j}\}$ i.e.; the adjacent pixels to any pixel K_{ij} are just north, south, west, east of this (no in the diagonal respect to considered pixel). In the second we consider the pixel K_{ij} (where $K = B \vee K = W$), the list of adjacent pixels to this is $\{K_{i-1j-1}, K_{i-1j}, K_{i-1j+1}, K_{ij-1}, K_{ij+1}, K_{i+1j-1}, K_{i+1j}, K_{i+1j+1}\}$ i.e.; the adjacent pixels to a any pixel K_{ij} are just up, down, right and left of this and, moreover, we consider the diagonal objects.

We will consider to work in this paper with 4-adjacency (for 2D images), because from a membrane computing point of view is more complex to design systems using this adjacency.

In this paper, we want to segment a 2D digital image. For this, we obtain the boundary of the different regions dividing the image. If we want to draw (or highlight) the boundaries we can follow two paths: *edge-based segmentation* and *region-based segmentation*. In the first option, we want to draw border line of the regions. In the second, we want to eliminate (or draw in white) and keep the resting pixels of the regions.

4.1 Segmenting 2D images

In this paper, we have decided to segment 2D digital images using array tissue-like P systems based in the first method: *edge-based segmentation*

Edge-based segmentation

We must find the border points of the regions (with different color) present in an image. So, we look for the pixels a_{ij} with some adjacent pixel of different color. We consider an input 2D digital image, and the color alphabet of the image ordered. So, for each image with $n \times m$ pixels ($n, m \in \mathbb{N}$) we define an array tissue-like P system whose input is given by an array codifying the input image. For the answer stage we use a counter $\bar{z}_i$, whose number of copies initially is $\lceil r^{1/2^7} \rceil$, where $r = max(n,m)$ because segmentation takes place in a constant number of steps.The output of the system is given by the objects appear in the output cell when it stops.

So, we can define an array tissue-like P systems to do the edge-based segmentation to a 2D image. For each $n, m \in \mathbb{N}$ we consider the tissue-like P system with input of degree 2

$$\Pi = (\Gamma, \Sigma, V, \mathcal{E}, w_0, w_1, w_2, A_1, A_2, \mathcal{R}, i_\Pi, o_\Pi),$$

defined as follows

(a) $\Gamma = \Sigma \cup \{a'_{ij} : a \in \mathcal{C}_S,\ 1 \leq i \leq n,\ 1 \leq j \leq m\} \cup \{\bar{z}_1\}$,
(b) $\Sigma = \{a_{ij} : a \in \mathcal{C}_S,\ 1 \leq i \leq n,\ 1 \leq j \leq m\}$,
(c) $V = \mathbb{N}$
(d) $\mathcal{E} = \{\, a'\,b\,, \,b\,a'\,, \begin{matrix} a' \\ b \end{matrix}, \begin{matrix} b \\ a' \end{matrix}, \begin{matrix} a' & b \\ a' & a' \end{matrix}, \begin{matrix} a' & a' \\ a' & b \end{matrix}, \begin{matrix} a' & a' \\ b & a' \end{matrix}, \begin{matrix} a' & a' \\ b & a' \end{matrix} \; : \; a, b \in V \wedge a < b\}$,
(e) $w_0 = \{\bar{z}_i : 2 \leq i \leq 9\}$,
$w_1 = \bar{z}_1^{\lceil r^{1/2^7} \rceil}$, where $r = max(n,m)$,
$w_2 = \bar{z}_1^{\lceil r^{1/2^7} \rceil}$,
(f) $A_1, A_2 = \emptyset$

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

1. $$(j, \bar{z}_i/\bar{z}_{i+1}^2, 0), \text{ for } i = 1, \ldots, 8,\ j = 1, 2$$

In this rule, we are working with a counter that it is used in the output of the systems.

2.

$$(1,\ a\ b\ /\ a'\ b\,, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1,\ b\ a\ /\ b\ a'\,, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1, \begin{matrix} a \\ b \end{matrix} / \begin{matrix} a' \\ b \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1, \begin{matrix} b \\ a \end{matrix} / \begin{matrix} b \\ a' \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

These rules are used when image has two adjacent pixels with different associated colors (border pixels). Then, the pixel with less associated color is marked (edge pixel).

3.

$$(1, \begin{matrix} a' & b \\ a & a' \end{matrix} / \begin{matrix} a' & b \\ a' & a' \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1, \begin{matrix} a' & a \\ b & a' \end{matrix} / \begin{matrix} a' & a' \\ b & a' \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1, \begin{matrix} a & a' \\ a' & b \end{matrix} / \begin{matrix} a' & a' \\ a' & b \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

$$(1, \begin{matrix} b & a' \\ a' & a \end{matrix} / \begin{matrix} b & a' \\ a' & a' \end{matrix}, 0), \text{ for } a, b \in \mathcal{C},\ \ a < b.$$

The rules mark (write in capital letters) the pixels which are adjacent to two pixels with same color and which were marked before. But, with the condition that the marked objects are adjacent to other pixel with a different color.

4.

$$(1, \bar{z}_9 a'_{ij}/\bar{z}_9, 2), \text{ for } a \in \mathcal{C}, 1 \leq i \leq n,\ 1 \leq j \leq m$$

With these rules system sends the edge pixels to the output cell.

(h) $i_\Pi = 1$

(i) $o_\Pi = 2$.

An overview of the Computation: A 2D image is codified by the input array that appear in the input cell and with them the system begins to work. Rules of type 1 initiate the counter $\bar{z}$. In a parallel manner, rules of type 2 identify the border pixels and mark the edge pixels. These rules need 4 steps to mark all the border pixels. From the second step, the rules of type 3 can be used with the first rules at the same time. So, in other 4 steps we can mark the rest of the (edge) pixels adjacent to two edge pixels and other border pixel with a different color to the other three pixels. System can apply the types of rules 2 and 3 simultaneously in some configurations, but it always applies the same number of these two types of rules because this number is given by edge pixels (we consider 4-adjacency). Finally, the fourth type of rules are applied in the following step on the system finish to mark all edge pixels in the cell 1. So with one step more we will have all the edge pixels in the output cells. Thus we need only 9 steps to obtain an edge-based segmentation for an $n \times m$ digital image.

Examples

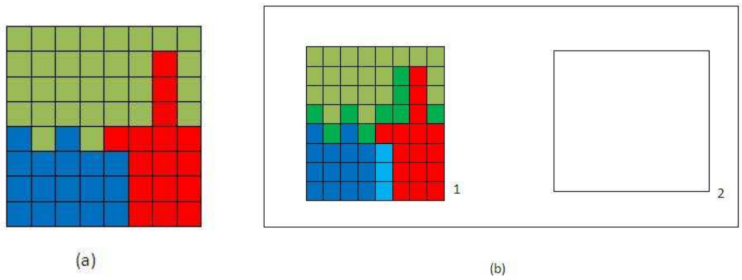

Fig. 2. (a) Input Image (b) Initial configuration

We show here some examples to see how our system does the edge-based segmentation of 2D images. We work with the images given by Figure 2 (a) and Figure 4 (a).

Initially, we consider an 8×8 image given by Figure 2 (a). A codifying of the initial configuration to segment this image is shown in Figure 2 (b). The order of the colors is the following: green, blue and red. Remember, we apply the rules in a maximally parallel manner where the pixels used by the rules are shown with different colors in the Figure 3 (a).

After nine steps, the output configuration is obtained and shown in Figure 3 (b).

Next, we segment the image of size 12×14 given by Figure 4 (a). In this example, we take the colors in the following order: Red, green, brown, orange, black, blue and light blue. The output image is shown in Figure 4(b).

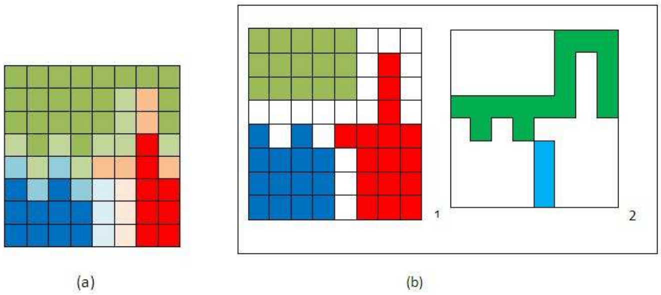

Fig. 3. (a) Process (b) Output Configuration

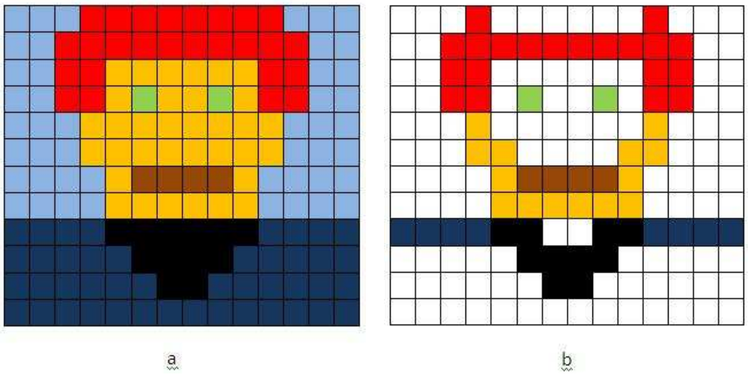

Fig. 4. (a) Input Image (b) Output Configuration

5 Final Remarks

This paper can be seen as a first attempt of formalizing the bridges between Membrane Computing and Algebraic Topology presented recently by Cristinal *et al.* [3, 4, 5].

The starting point is that problems from Digital Images, treated by techniques of Algebraic Topology, can be suitable for Membrane Computing techniques. The basis is that such problems can be treated locally by a set of processors, the information can be expressed as (multi)sets of pixels and other auxiliary objects, and the transformations can be processed by re-writing-type rules.

Many research lines are open. From the Membrane Computing point of view, we wonder whether tissue-like models is the most suitable or not, or in which way this formalization can be improved. From the Algebraic Topology point of view, the question is to find new representations and new problems which can be expressed and dealt with Membrane Computing techniques.

Acknowledgement

The first author acknowledge the support of the project "Computational Topology and Applied Mathematics" PAICYT research project FQM-296. The second,third and four author acknowledge the support of the projects TIN2008-04487-E and TIN-2009-13192 of the Ministerio de Ciencia e Innovación of Spain and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200.

References

1. Ceterchi, R., Mutyam, M., Păun, Gh., Subramanian, K.G. Array-rewriting P Systems. *Natural Computing* **2**, (2003) 229-249.
2. Cook, C.R., Wang, P.S.-P. A Chomsky Hierarchy of Isotonic Array Grammars and Languages. *Computer Graphics and Image Processing* **8**, (1978) 144-152.
3. Christinal, H.A., Díaz-Pernil, D., Real, P. Segmentation in 2D and 3D Image Using Tissue-Like P System. *Lecture Notes in Computer Science*, **5856**, (2009) 169-176.
4. Christinal, H.A., Díaz-Pernil, D., Real, P. Using Membrane Computing for Obtaining Homology Groups of Binary 2D Digital Images. *Lecture Notes in Computer Science*, **5852**, (2009) 388-401.
5. Christinal, H.A., Díaz-Pernil, D., Real, P. P Systems and Computational Algebraic Topology. Accepted paper at *Journal of Mathematical and Computer Modelling.*
6. Freund, R. Array Grammars. Technical Rep. 15/00, Research Group on Mathematical Linguistics, Rovira i Virgili University, Tarragona, (2000).
7. Kim, S.-H., Kim, H.-G., Tchah, K.-H. Object oriented face detection using colour transformation and range segmentation, *Electronics Letters, IEEE*, **34**, (1998), 979-980.
8. Linda G. Shapiro and George C. Stockman .*Computer Vision*, (2001).
9. Martín–Vide, C., Pazos, J., Păun, Gh. and Rodríguez Patón, A. A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. *Lecture Notes in Computer Science*, **2387**, (2002), 290–299.
10. Martín–Vide, C., Pazos, J., Păun, Gh. and Rodríguez Patón, A. Tissue P systems. *Theoretical Computer Science*, **296**(2), (2003), 295–326.
11. Păun, A. and Păun, Gh. The power of communication: P systems with symport/antiport. *New Generation Computing*, **20**, 3, (2002), 295-305.
12. Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1, (2000), 108–143.
13. Păun, Gh. Membrane Computing. An Introduction. *Springer–Verlag, Berlin*, (2002).
14. Păun, Gh. Grammar Systems vs. Membrane Computing: A Preliminary Approach. Grammar Systems Week 2004, July 5-9, Budapest, Hungary, (2004). `http://psystems.disco.unimib.it/download/grsp0.ps`
15. Gh. Păun, G. Rozenberg, A. Salomaa, eds: Handbook of Membrane Computing. Oxford University Press, Cambridge, 2009.
16. Rosenfeld, A. Picture Languages. Academic Press, Reading, MA (1979).
17. Rosenfeld, A. Digital Topology. *American Mathematical Monthly*, **86**, (1979), 621-630.

18. Rosenfeld, A. Connectivity in Digital Pictures. *Journal for Association of Computing Machinery*, **17**(1), (1970), 146-160.
19. Sharmay, O., Miocz, D., Antony, 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**, (2008), 235-240.
20. Subramanian, K.G. P Systems and Picture Languages. *Lecture Notes in Computes Science* **4664**, (2007) 99-109.
21. Tobias, O.J., Seara,R. Image Segmentation by Histogram Thresholding Using Fuzzy Sets. *IEEE Transactions on Image Processing*,**11**, 12, (2002) 1457-1465.
22. Wang, P.S.-P. Some New Results on Isotonic Array Grammars. *Information Processing Letters* **10**, (1980) 129-131.
23. Wang, D., Lu, H., Zhang, J., Liang, J.Z. A Knowledge-Based Fuzzy Clustering Method with Adaptation Penalty for Bone Segmentation of CT images. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, (2005), 6488–6491.
24. Yuan Xiaojing, Situ Ning and Zouridakis George. A narrow band graph partitioning method for skin lesion segmentation.*Elsevier Science Pattern Recognition*, **42**, 6, (2009), 1017-1028.
25. Yazid. H and Arof. H. Image Segmentation using Watershed Transformation for Facial Expression Recognition. *IFMBE Proceedings, 4th Kuala Lumpur International Conference on Biomedical Engineering*, **21**, (2008), 575-578.
26. P systems web page `http://ppage.psystems.eu/`