Region-based segmentation of 2D and 3D images with tissue-like P systems

Hepzibah A. Christinal\textsuperscript{a,b}, Daniel Díaz-Pernil\textsuperscript{b,*}, Pedro Real\textsuperscript{b}

\textsuperscript{a}Karunya University, Coimbatore 641 114, Tamilnadu, India  
\textsuperscript{b}Research Group on Computational Topology and Applied Mathematics, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

\begin{abstract}
Membrane Computing is a biologically inspired computational model. Its devices are called P systems and they perform computations by applying a finite set of rules in a synchronous, maximally parallel way. In this paper, we develop a variant of P-system, called tissue-like P system in order to design in this computational setting, a region-based segmentation algorithm of 2D pixel-based and 3D voxel-based digital images. Concretely, we use 4-adjacency neighborhood relation between pixels in 2D and 6-adjacency neighborhood relation between voxel in 3D for segmenting digital images in a constant number of steps. Finally, specific software is used to check the validity of these systems with some simple examples.
\end{abstract}

1. Introduction

Natural Computing studies new computational paradigms inspired on 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, such as Genetic Algorithms (Holland, 1992), Neural Networks (McCulloch and Pitts, 1988), DNA-based molecular computing (Adleman, 1994).

Membrane Computing is a theoretical model of computation inspired by the structure and functioning of cells as living organisms. In Natural Computing, a P system is called cell-like P system (Paun, 2000). Roughly speaking, a P system consists of a membrane structure, in whose compartments one places multisets of objects which evolve according to given rules. In the most common model, the rules are applied in a synchronous non-deterministic maximally parallel way, but some other semantics are being explored.

According to their architecture, these models can be split into two sets: cell-like P systems (Paun, 2000) and tissue-like P systems (Mira and Alvarez, 2007; Díaz-Pernil et al., 2008, 2009). In the first 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.

Segmentation in computer vision (Stockman and Shapiro, 2001) refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image in such a way those pixels with the same label share certain visual characteristics. There exist different techniques to segment an image. Some of them are Clustering methods (Li et al., 2008), Histogram-based methods (Tobias and Seara, 2002), Watershed transformation methods (Yuan et al., 2009). Some of the practical applications of image segmentation are medical Imaging (Campadelli et al., 2009), study of anatomical structures, location of objects in satellite images (roads, forests, etc.) (Gamanya et al., 2007), and face recognition (Zhao et al., 2003).

Previous work putting into relation Natural Computing with Digital Imagery are (Ceterchi et al., 2003a,b), among others. Here, we develop massively parallel algorithms for segmentation of digital images using a recent scheme in Natural Computing: tissue-like P systems.

The paper is structured as follows. In Section 2, we introduce the definition of basic tissue-like P systems and show an example to understand how these systems work. In Section 3, we design a family of systems for region-based segmentation in 2D image \((n \times m)\). Afterwards, we check our model using a program called Tissue Simulator with very easy images. At the end of this section, we introduce a family of tissue-like P systems to obtain a region-based segmentation of 3D images. Finally, some conclusions and future work are given.
2. Description of a model of membranes

We begin this section by briefly recalling some of the concepts used later on in the paper.

An alphabet, \( \Sigma \), 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 (with length 0) will be denoted by \( \lambda \). The set of strings of length \( n \) built with symbols from alphabet \( \Sigma \) is denoted by \( \Sigma^n \) and \( \Sigma^* = \bigcup_{i \geq 0} \Sigma^i \). A language over \( \Sigma \) is a subset of \( \Sigma^* \).

A multiset over a set \( A \) is a pair \((A, f)\) where \( f : A \rightarrow \mathbb{N} \) is a mapping. The set of all multisets on \( A \) will be denoted by \( \Lambda\lambda(A) \). If \( m = (A, f) \) is a multiset then its support is defined as \( \text{supp}(m) = \{ x \in A | f(x) > 0 \} \) and its size is defined as \( \sum_{x \in \text{supp}(m)} f(x) \). 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 will be denoted as \( m = a_1^{f_1}a_2^{f_2} \ldots a_k^{f_k} \), where \( \text{supp}(m) = \{ a_1, \ldots, a_k \} \), and for each element \( a_i, f(a_i) \) is called the multiplicity of \( a_i \). If \( f(a_i) = 1 \), we will write \( a_i \) instead of \( a_i^1 \). In what follows, we assume the reader is already familiar with the basic notions and the terminology underlying \( \mathbb{P} \) systems.\(^1\)

Martin-Vide et al. introduced in (Martín-Vide et al., 2003) a new variant of \( \mathbb{P} \) systems where the cells are ordered in tissues. It has two biological inspirations: intercellular communication and cooperation between neurons. In this paper, we work with a new tissue-like model presented in (Mira and Álvarez, 2007; Díaz-Permil et al., 2009) closer to the cell-like systems (classical \( \mathbb{P} \) systems).

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.

Formally, a tissue-like \( \mathbb{P} \) system of degree \( q \geq 1 \) is a tuple

\[ \Pi = (\Gamma, \Sigma, \varepsilon, w_1, w_2, w_3, \mathcal{R}, \mathcal{I}_I, \mathcal{O}_I), \]

where

1. \( \Gamma \) is a finite alphabet, whose symbols will be called objects,
2. \( \Sigma \subseteq \Gamma \) is the input alphabet,
3. \( \varepsilon \subseteq \Gamma \) (the objects in the environment),
4. \( w_1, \ldots, w_q \) are strings over \( \Gamma \) 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. \( \mathcal{I}_I \subseteq \{0, 1, 2, \ldots, q\} \),
7. \( \mathcal{O}_I \subseteq \{0, 1, 2, \ldots, q\} \).

A tissue-like \( \mathbb{P} \) system of degree \( q \geq 1 \) can be seen as a set of \( q \) cells (each one consisting of an elementary membrane) labelled 1, 2, \ldots, \( q \). We will use 0 to refer to the label of the environment, \( \mathcal{I}_I \) denoting the input region and \( \mathcal{O}_I \) denoting the output region (which can be the region inside a cell or the environment).

The strings \( w_1, \ldots, w_q \) describe the multisets of objects placed in the \( q \) cells of the system. We interpret \( \varepsilon \subseteq \Gamma \) as the set of objects placed in the environment, each of them available in an arbitrary large amount of copies.

The communication rule \((i, u | v_j)\) can be applied over two cells labelled \( i \) 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. Therefore, some objects not belonging to \( \varepsilon \) can go to the environment. So, in a

---

\(^1\) We refer to (Plân, 2002) for basic information in this area, to (Plân et al., 2010) for a comprehensive presentation and the web site [The \( \mathbb{P} \) Systems Webpage, 2008] for the up-to-date information.
cannot be applied by the system in this computation step, because there is not an object \( f \) in cell 1. The question is the application of the rules \( r_2 \) and \( r_4 \). System has two possibilities: it can apply rules \( r_2 \) and \( r_4 \) twice each one or, \( r_4 \) could works four times (\( r_2 \) no one). System chooses one or another possibility in a non deterministic manner. In this example we have chosen by the system the first one. So, two copies of \( e \) go to cell 1 and two other copies go to the environment. We must remember that system can apply rule \( r_2 \) twice because \( f \) is an element of the alphabet (of the environment) \( E \). Moreover, we do not show the objects of \( E \) in the environment in Fig. 1, but system can take them of the environment when it needs them.

Moreover, we do not show the objects of \( E \) in Fig. 2. But we need to define more types of rules with respect to the number of objects in the set \( E \). System does not stop in this computation step because it can apply rules \( r_1 \) four times and \( r_2 \) once. Next, system applies four times rule \( r_3 \) again and then, system halts.

3. Segmenting digital images in a constant time

A 2-D digital image \( I \) with size \( n \times m \) \((n, m \in \mathbb{N})\) is a rectangular net of objects \((i, j)\) called pixels (voxels in 3-D images), with \( 1 \leq i \leq n \) and \( 1 \leq j \leq m \). Let the alphabet of colors of \( I (C \subseteq \mathbb{N}) \) be the ordered set of all colors in \( I \). We define the size of \( C, |C| \), as the number of colors of this alphabet. Moreover, we will assume that each pixel of \( I \) is associated to a color of \( C \). So, we encode the pixel \((i, j)\) with associated color \( a \in C \) as the object \( a_{i,j} \). Therefore, we codify image \( I \) as the set \( \{a_{i,j} : a \in C \land 1 \leq i \leq n \land 1 \leq j \leq m\} \).

Given two pixels \( p_1 = (i_1,j_1) \) and \( p_2 = (k_1,l_1) \), we say they are adjacent when the distance \( d(p_1,p_2) = \sqrt{(i_1-k_1)^2 + (j_1-l_1)^2} \) is 1. Then, we consider a 4-adjacency (Rosenfeld, 1979) neighborhood relation between pixels. In an analogous way, we can design a segmentation algorithm for the 8-adjacency relation, but with first adjacency we need to define more types of rules with respect to the second one, i.e., we consider a more complex and more interesting problem from a membrane computing point of view.

Following this way, we can divide an image in regions, where each region is a set of two to two adjacent pixels and everyone has the same associated color. We define a boundary of a region as the set of pixels of this region with the property to be adjacent to other pixel with a different associated color.

In this section, we segment pixel-based digital images using a region-based segmentation algorithm. This type of segmentation finds regions with different color from each other present in an image. We define two family of tissue-like \( P \) systems to realize this type of segmentation. \( \Pi_1 \) segments 2D digital images and \( \Pi_2 \) is the adaptation of the first system to segment 3D voxel-based digital images.

Region-based segmentation of \( n-D \) digital images (\( n-D-ES \)) problem: Given a \( n-D \) (with \( n = 2 \vee 3 \)) digital image with pixels (or voxels) of (possibly) different colors, eliminate the boundaries between regions in that image.

3.1. Region-based segmentation of a 2D digital image

Given a digital image with \( n \times m \) pixels \((n, m \in \mathbb{N})\) we define a tissue-like \( P \) system whose input is given by the set \( \{a_i : a \in C \land 1 \leq i \leq n \land 1 \leq j \leq m\} \).

Next, we shall give some outlines of how to prove that the 2D-ES problem can be solved in a constant amount of time with respect to the input data using a family of basic \( P \) systems.

First, the system marks the boundary pixels. Next, it marks the necessary pixels to connect all the boundary pixels of the same color. The system needs 8 computation steps for this and uses a counter, \( z_i \), to send the marked objects to the cell 2. But as the system only uses 8 steps to do the segmentation we need initially \( \lceil \frac{r_1}{2} \rceil \) copies of object \( z_1 \), with \( r_1 = \text{max}(n,m) \), to generate enough copies of \( z_1 \) for the system output. It is given by the objects that appear in the cell 2 when it stops. So, the system is ready to send the objects codifying the complete image to the cell 2 in the last step of computation.

3.1.1. A family of tissue-like \( P \) systems

So, we define a family of tissue-like \( P \) systems to do the region-based segmentation of a 2D image. For each \( n, m \in \mathbb{N} \) we consider the tissue-like \( P \) system of degree 2

\[
\Pi_1(n, m) = (F, \Sigma, \varepsilon, \omega_1, \omega_2, R, I_0, o_1),
\]

which is defined by the following conditions:

(a) The working alphabet is \( F = \Sigma \cup \varepsilon \),
(b) the input alphabet is \( \Sigma = \{a_i : a \in C, 1 \leq i \leq n, 1 \leq j \leq m\} \),
(c) the environment alphabet is \( \varepsilon = \{a_i : 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{z_i : 1 \leq i \leq 9\} \),
(d) the multisets of cells 1 and 2 are \( \omega_1 = z_1^{(r_1/2^{|\varepsilon|})} \), \( \omega_2 = \emptyset \), respectively,
(e) \( R \) is the following set of communication rules:

1. \((1, z_i/z_{i+1}^j, 0)\), for \( i = 1, \ldots, 8 \).

These rules are used to update the counter \( z_i \), duplicating the number of copies in each step.

2. \((1, a_i b_{j} / a_i b_{j}, 0)\), for \( a, b \in C, a < b \), \( 1 \leq i, k \leq n \) and \( 1 \leq j, l \leq m \) (see Fig. 2, center image).

These rules are used when the image has two adjacent pixels of different colors.
pixels with different associated colors (border pixels). For example, in Fig. 2 we can find pair of adjacent pixels of the form:

\[
\begin{array}{ccc}
1 & 3 & 1 \\
1 & 3 & 1 \\
\end{array}
\]

Then, the pixel with smallest associated color is marked and system brings an object representing this marked pixel (edge pixel) from the environment. For example, in the center of Fig. 2 we use bold fonts to highlight the edge pixels.

adding a new edge pixel. We use bold font to the edge pixels, i.e. objects of the form \(a\) (in this case we use \(1\) for \(1\)).

4. \((1, \overline{a_{ij}/\overline{a_{lk}}}, 2), \) for \(1 \leq i \leq n, 1 \leq j \leq m.\) These rules send the marked pixels to the environment (see Fig. 3):

\((f)\) the input cell is \(i_{jk} = 1,\)

\((g)\) the output cell is \(o_{jk} = 0.\)

3.1.2. An overview of the computation

When the input objects \(a_{ij}\) encoding the colored pixels from a 2D digital image and the counter \(z\) appear in the input cell, the system begins to work. Rules of type 1 identify the border pixels and bring the edge pixels from the environment in a parallel manner. At the same time, rules of type 3 duplicate the counter \(z\) during eight steps to obtain the needed copies of the object \(z_{lp}.\) The rules of type 1 need four steps to mark all the border pixels similar to the edge-based segmentation. From the second step the rules of type 2 can be used in parallel with the first rules. So, in another four steps, we can bring from the environment the edge pixels adjacent to two border pixels. The system can apply the first two types of rules simultaneously in some configurations. The system always applies the same number of these two types of rules because this number is given by the edge pixels (we consider 4-adjacency).

Finally, the fourth type of rules are applied in the following step, using the objects \(z_{lp}\) the edge pixels are sent to the environment leaving behind different color regions. We need only 9 steps to obtain a region-based segmentation for a \(n \times m\) image.

3.1.3. Complexity and resources needed

Taking into account that the size of the input data is \(O(n \cdot m)\) and the number of colors of the image \((|C|)\) is \(h\), the amount of resources needed to define the systems of our family and the complexity of our problem are determined in the following table:

<table>
<thead>
<tr>
<th>2D-ES Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
</tr>
<tr>
<td>Number of steps of a computation</td>
</tr>
<tr>
<td>Resources needed</td>
</tr>
<tr>
<td>Size of the alphabet</td>
</tr>
<tr>
<td>Initial number of cells</td>
</tr>
<tr>
<td>Initial number of objects</td>
</tr>
<tr>
<td>Number of rules</td>
</tr>
<tr>
<td>Upper bound for the length of the rules</td>
</tr>
</tbody>
</table>

3.1.4. Checking this family of systems with Tissue Simulator software

We have used a program called Tissue Simulator (see Fig. 4) introduced by Borrego et al. (2007). We have simulated our family of systems to segment 2D digital images with this software. Finally, we have introduced as instances of our system the examples appearing in Fig. 5 and, in a constant number of steps, we have obtained an encoding of a region-segmentation (showed in Fig. 6) for each one of the examples introduced before.

3.2. Segmenting 3D digital images

The next step consists of extending our models to work with 3D images. Now, the input data are voxels \((i,j,k) \in \mathbb{N}^3\) encoded by the objects \(a_{ijk}\) with \(a \in C.\) We consider the 26-adjacency relation for determining the neighbors of one voxel.

3.2.1. A family of tissue-like P systems

Given a digital image with \(n \times m \times l\) voxels \((n, m, l \in \mathbb{N})\) we define a tissue-like P system whose input is given by the voxels of the image encoded by the objects \(a_{ijk}\), where \(1 \leq i \leq n, 1 \leq j \leq m\) and \(1 \leq k \leq l.\)

Next, we shall give an outline of how to prove that the 3D-ES problem can be solved in a constant amount of time with respect to the input data using a family of basic P systems.

A system of this family works in the following way: First, the system marks the boundary pixels. Next, the system uses a counter \((z_1, \) whose number of initial copies in the system is \(r_1 = \left\lceil \frac{1}{h^2} \right\rceil,\) where \(r_2 = \text{max}(n, m, l)\)) to send the marked objects to the environment.
The output is given by the objects appearing in the output cell when the system stops.

\[ \Pi_2(n, m, l) = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, R, i_0, o_0) \]

defined as follows:

(a) The working alphabet is \( \Gamma = \Sigma \cup \mathcal{E} \).
(b) The input alphabet is \( \Sigma = \{ \alpha_{ijk} : \alpha \in \mathcal{C}, 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq l \} \).
(c) The environment alphabet is \( \mathcal{E} = \{ z_i, z_j, \alpha_{ijk} \in \mathcal{C}, 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq l \} \).
(d) The multisets of cells 1 and 2 are \( w_1 = z_i^{j=1,2} \), \( w_2 = \emptyset \), respectively.
(e) \( R \) is the following set of communication rules:

1. \( (1, z_i^{j=1}, 0) \), for \( i = 1, \ldots, 26 \). These rules are used to update the counter \( z_i \) duplicating the number of copies in each step.
2. \( (1, a_{ij1}, \alpha_{ijl}, b_{ij2}, \alpha_{ijl}, b_{ij2}, c_{ijl}, 0) \), for \( 1 \leq i, i_2 \leq n, 1 \leq j, j_2 \leq m, 1 \leq k, k_2 \leq l \), with \( (i_1, j_1, k_1) \) and \( (i_2, j_2, k_2) \) adjacent voxels, and finally \( a, b \in \mathcal{C}, a < b \). These rules are used when the image has two adjacent voxels with different associated colors (border voxels). Then, the voxel with the smallest associated color is marked and the system brings from the environment an object representing this marked voxel (edge voxel).
3. \( (1, z_i^{j=1}, 0) \), for \( 1 \leq i \leq n, 1 \leq j \leq m \) and \( 1 \leq k \leq l \). These rules send the marked pixels to the environment.

Fig. 4. Two images of Tissue Simulator.
3.2.2. An overview of the computation

This computation would be very similar if we consider an 8-adjacency in 2D. Rules of type 1 identify the border voxels and bring the edge voxels from the environment. These rules need as much as 26 steps for this. Finally, the second type of rules are applied in the following step sending the edge voxels to the environment. So we need again a constant number of steps in this 3D region-segmentation massively parallel algorithm.

3.2.3. Complexity and resources needed

Taking into account the size of the input data is $O(n \cdot m \cdot l)$ and the number of colors of the image $|g|$ is $h$, 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:

<table>
<thead>
<tr>
<th>3D-RS problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
</tr>
<tr>
<td>Resources needed</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

4. Conclusions and future work

In this paper, tissue-like P-systems (Díaz-Pernil et al., 2008) are used for theoretically solving a region-based segmentation prob-
lem for digital images. This work opens new promising research lines: (a) developing efficient sequential software mimicking these Natural Computing techniques; (b) developing efficient parallel software working with a cluster or a graphic card (GPUs); (c) and also designing topology-based processes (topological noise removal, topological simplification, snakes and active contours, topological filling, image compression…) in Digital Imagery using Membrane Computing. Segmentation can be seen as a topology-based process in Digital Imagery, attending to the topological characteristics (connected components and holes, mainly) of each object defined by one unique label. To have at hand representation models for digital images trying to capture full topological and geometrical information of them (González-Díaz and Real, 2005; Gonzalez-Díaz et al., 2009a,b) could be of help for getting this last goal.

Acknowledgement

The first author acknowledges the support of the project “Computational Topology and Applied Mathematics” PAICYT research project FQM-296. The second author acknowledge the support of the project TIN2009-13192 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence P08-TIC-04200 of the Junta de Andalucía. The third author acknowledge the support of the project Project of Excellence P08-TIC-04200 of the Junta de Andalucía.

References


