Pattern Recognition Letters 32 (2011) 2206–2212

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

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

Hepzibah A. Christinal [a,b], Daniel Díaz-Pernil [b,*], Pedro Real [b]

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

ARTICLE INFO

Article history:
Available online 12 May 2011

Keywords:
P systems
Membrane Computing
Digital image
Region-based segmentation
Digital topology

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.

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 able to process and generate information. The computational devices in Membrane Computing are called *P systems* (Păun, 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 (Păun, 2000) and tissue-like P systems (Mira and Álvarez, 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 (Tarabalka et al., 2010), Graph partitioning 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.

* Corresponding author.
E-mail addresses: hepzi@us.es (H.A. Christinal), sbdani@us.es (D. Díaz-Pernil), real@us.es (P. Real).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.05.004

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*, 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 (with length 0) will be denoted by λ. The set of strings of length n built with symbols from alphabet Σ is denoted by Σ^n and $\Sigma^* = \cup_{n\geqslant 0}\Sigma^n$. A *language* over Σ is a subset from Σ^*.

A *multiset* over a set A is a pair (A,f) where $f : A \to \mathbb{N}$ is a mapping. The set of all multisets on A will be denoted by $\mathcal{M}(A)$. If $m = (A,f)$ is a multiset then its *support* is defined as $supp(m) = \{x \in A | f(x) > 0\}$ 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 will be denoted as $m = a_1^{f(a_1)}a_2^{f(a_2)}\ldots a_k^{f(a_k)}$, where $supp(m) = \{a_1,\ldots,a_k\}$, and for each element a_i, $f(a_i)$ is called the multiplicity of a_i. 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 P systems.[1]

Martín-Vide et al. introduced in (Martín-Vide et al., 2003) a new variant of 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-Pernil et al., 2009) closer to the cell-like systems (classical 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 P system* of degree $q \geqslant 1$ is a tuple

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

where

1. Γ is a finite *alphabet*, whose symbols will be called *objects*,
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. $i_\Pi \in \{1,2,\ldots,q\}$,
7. $o_\Pi \in \{0,1,2,\ldots,q\}$.

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

The strings $w_1,\ldots,w_q$ describe the multisets of objects placed in the q cells of the system. We interpret $\mathcal{E} \subseteq \Gamma$ 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 $\mathcal{E}$ can go to the environment. So, in a configuration (not initial) we could find two types of objects in the environment: First, those which belong to the environment that appear in an arbitrary large number of copies. So, system could take many copies of them as it needs in each computation step. Second, those which not belong to the environment. For them, the environment works like a cell, i.e., if three copies of an object, for example $a \notin \mathcal{E}$, arrive to the environment during a computation step. There was not copies of a in the environment before. Then, system has only three copies of element a to work, no more.

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 so, i.e., in each step we apply a maximal set of rules.

A *configuration* is an instantaneous description of the system Π, and it is represented as a tuple $(w_0,w_1,\ldots,w_q)$. Given a configuration, we can perform a computation step and obtain a new configuration by applying the rules in a parallel manner as it is shown above. A sequence of computation steps is called a *computation*. A configuration is *halting* when no rules can be applied to it. Then, a computation halts when the system reaches a halting configuration. In the literature, the output of a computation is collected from its halting configuration by reading the objects contained in the output cell.

Let us now show a simple example:

Consider a tissue-like P system

$$\Pi' = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, w_3, \mathcal{R}, i_\Pi, o_\Pi),$$

where

1. $\Gamma = \{a,b,e,f\}$ is the working alphabet,
2. $\Sigma = \emptyset$ is the input alphabet. In this case we consider empty,
3. $\mathcal{E} = \{a,f\}$ is the environment alphabet,
4. $w_1 = ab^4e$, $w_2 = a^2bf$, $w_3 = e^4f$, are the multiset of cells 1, 2 and 3, respectively,
5. $\mathcal{R} = \{r_1 \equiv (1,b/f,3),\ r_2 \equiv (2,a/e,3),\ r_3 \equiv (1,a/b,2),\ r_4 \equiv (3,e/f^2,0),\ r_5 \equiv (1,f/\lambda,0)\}$ is the set of communication rules,
6. $i_\Pi = 1$ is the label of input cell,
7. $o_\Pi = 0$ is the label of output cell.

We can see the initial configuration of this system in first image of Fig. 1. There are five rules to use, r_1 is applied once and system exchanges an object b in cell 1 by an object f in cell 3. r_3 is applied once too and an object a in cell 1 is traded against an object b. r_5

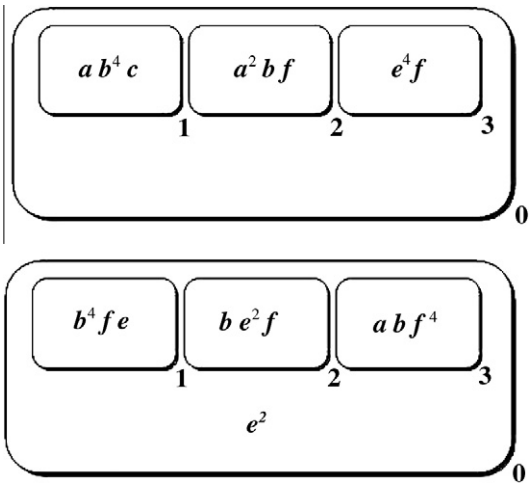

Fig. 1. Initial configuration of system Π' and the next configuration.

[1] We refer to (Păun, 2002) for basic information in this ares, to (Păun et al., 2010) for a comprehensive presentation and the web site (The P Systems Webpage, 2008) for the up-to-date information.