Cell Complexes and Membrane Computing for Thinning 2D and 3D Images

Raúl Reina-Molina[1], Daniel Díaz-Pernil[1], Miguel A. Gutiérrez-Naranjo[2]

[1]Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics
University of Sevilla
raureimol@alum.us.es, sbdani@us.es

[2]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
magutier@us.es

Summary. In this paper, we show a new example of bridging Algebraic Topology, Membrane Computing and Digital Images. In [24], a new algorithm for thinning multidimensional black and white digital images by using cell complexes was presented. Such cell complexes allow a discrete partition of the space and the algorithm preserves topological and geometrical properties of the image. In this paper, we present a parallel adaptation of such algorithm to P systems, by introducing some concepts of Algebraic Topology in the Membrane Computing framework. The chosen model for the implementation is tissue-like P systems with promoters, inhibitors and priorities.

1 Introduction

Computer vision [36] is one of the challenges for Computer Science in the next years. From a biological point of view, vision is an extremely complex process involving the transformation of the light energy into a signal which leaves the eye by way of the optic nerve and arrives to the brain, where is interpreted. From the computational side, a 2D digital image can be roughly defined as a function from a two dimensional surface which maps each point from the surface onto a set of attributes as bright or color. Analogously, a 3D image maps a region of a tridimensional space onto a set of attributes. The different treatments of such mappings provide a big amount of current applications in computer vision as biometrics [1], surveillance [11] or medical imaging [2].

Many problems in the processing of 2D or 3D digital images have features which make it suitable for techniques inspired by nature. The subset of the integer plane or space taken to be the *support* of the image and the set of possible features

associated to each 2D or 3D point can be considered finite and hence, the transformation of an image into another can be made in a *discrete* way. Other of such features is that the treatment of the image can be parallelized and locally modified. Regardless how large is the picture, the process can be performed in parallel in different local areas of it. Another interesting feature is that the information of the image can also be easily encoded in the data structures used in Natural Computing.

In the literature, we can find many examples of the use of Natural Computing techniques for dealing with such problems. One of the classic examples is the use of cellular automata [33, 35]. Other efforts are related to artificial neural networks as in [18, 38].

In Membrane Computing, there is a large tradition in the study of dealing information structured as two dimensional objects (see, e.g., [5, 6, 12, 23]). The main motivation for these studies is to bring together Membrane Computing and Picture Grammars. From a technical point of view, arrays are two-dimensional objects placed inside the membranes as strings are one-dimensional objects in the model of P systems with string objects [19, 31].

Recently, a new research line has been open by applying well-known membrane computing techniques for solving problems from digital imagery. For example, the *segmentation* problem, [8, 10, 13, 14], *thresholding* [7] or *smoothing* [29]. Special attention deserves Gimel'farb *et al.* [20], where the *symmetric dynamic programming stereo* (SDPS) algorithm [21] for stereo matching was implemented by using simple P modules with duplex channels.

In this paper, we focus on the problem of skeletonizing a 2D or 3D image. Skeletonization is one of the approaches for representing a shape with a small amount of information by converting the initial image into a more compact representation and keeping the meaning features. The conversion should remove redundant information, but it should also keep the basic structure. There are many different definitions of the skeleton of a black and white image and many skeletonizing algorithms[1], but in general, the image B is a skeleton of the image A, if it has fewer black pixels than A, preserves its topological properties and, in some sense, keeps its *meaning*. The most important features concerning a shape are its *topology* (represented by connected components, holes, etc.) and its *geometry* (elongated parts, ramifications, etc.), thus these terms have to be preserved. When the skeletonizing process is made by the iterative removal of non-significant elements of the image, the process is known as *thinning*.

In this paper, we present an implementation of the Liu's algorithm [24] for thinning images based on Membrane Computing techniques. The basic notion of this algorithm is the *cell complex*. It can be seen as a mathematical abstraction of a space unit. This space unit is built in some n dimensional space and embedded in a space of higher dimension, as a 2-dimensional square can be embedded in a 3D space. All these concepts will be formalized below.

[1] A detailed description of skeletonizing algorithms is out of the scope of this paper. For a survey in this topic, see e.g., [34].

In Liu's work [24], a cell complex is processed in order to obtain another complex with the same topology, and the same geometry. We will start from a black and white 2D or 3D digital image by building a cell complex from it. This complex will be, then, processed by consecutive parallel removal of certain cells. The removal process does not change the topology nor the geometry of the starting cell complex. At the end of this process, the set of non-removed cells will make the skeleton.

For implementing these ideas in the Membrane Computing framework, we present a family of tissue-like P systems endowed with priorities among rules, promoters and inhibitors. This paper follows the research line open with [9], but, to the best of our knowledge, this is the first work which put together Membrane Computing, Cells Complexes and thinning processes.

The paper is organized as follows. In the first section, all technical requirements of Algebraic Topology are reviewed. Next, the basics for understanding the proposed algorithm are introduced, followed by the presentation of the Membrane Computing framework and the bioinspired 2D and 3D black and white image thinning algorithm. Next, an overview of the computation is presented, finishing with conclusions and future work.

2 Cubical Complexes

As pointed above, cubical complexes are mathematical abstractions to handle structured portions of a n dimensional space. On such abstractions, we can define several operators as the *border* one, which associates, for example, a 3D cell (cube) with six 2D cells (squares), or properties to define *free cells* or *isolated cells*.

We follow T. Kaczyński, K. Mischaikow and M. Mrozek [22] in the description of a kind of combinatorial structure on a topological space.

Definition 1. *[22] An* elementary interval *is a closed interval $\overline{I} \subset \mathbb{R}$ of the form $\overline{I} = [l, l+1]$ or $\overline{I} = [l, l]$ for some $l \in \mathbb{Z}$. The former are called* nondegenerated*, while the latter are called* degenerated*. The interval $[l, l]$ that contains only one point will be denoted by $[l]$.*

Degenerated elementary intervals are simply points with 0 dimensions. Nondegenerated elementary intervals are segments (objects with one dimension). Next, we generalize this notion to any dimension.

Definition 2. *An elementary cube $\overline{\sigma}$ is a finite product of elementary intervals:*

$$\overline{\sigma} = \overline{I}_1 \times \overline{I}_2 \times \cdots \times \overline{I}_d \subset \mathbb{R}^d$$

where each $\overline{I}_j$ is an elementary interval, $j \in \{1, \ldots, d\}$. The set of all elementary cubes in $\mathbb{R}^d$ is denoted by $\mathcal{K}^d$. The set of all elementary cubes is

$$\mathcal{K} = \bigcup_{d=1}^{\infty} \mathcal{K}^d$$

For example $\{(0,0,0)\}$, $\{(x,0,0)\,|\,0 \leq x \leq 1\}$, $\{(x,y,0)\,|\,0 \leq x,y \leq 1\}$ and $\{(x,y,z)\,|\,0 \leq x,y,z \leq 1\}$ are elementary cubes. Given an elementary cube $\overline{\sigma} = \overline{I}_1 \times \overline{I}_2 \times \cdots \times \overline{I}_d$ in $\mathbb{R}^d$, its *embedding number* d is denoted by $\operatorname{emb} \overline{\sigma}$. The dimension of $\overline{\sigma}$ is defined to be the number of nondegenerated intervals in its definition and is denoted by $\dim \overline{\sigma}$. In this way, for the elementary cube $Q \equiv \{(x,y,0)\,|\,0 \leq x,y \leq 1\}$, $\operatorname{emb} Q$ is 3 and $\dim Q$ is 2.

The set of all elementary cubes with dimension p is denoted by $\mathcal{K}_p$. The set of all elementary cubes in $\mathbb{R}^d$ with dimension p is denoted by $\mathcal{K}_p^d$.

The following definition gives sense to the decomposition of elementary cubes into lower-dimensional objects.

Definition 3. *Let $\overline{\delta}$ and $\overline{\sigma}$ be two elementary cubes of any dimension. If $\overline{\delta} \subset \overline{\sigma}$, then $\overline{\delta}$ is a* face *of $\overline{\sigma}$. If $\overline{\delta}$ is a face of $\overline{\sigma}$ and $\overline{\delta} \neq \overline{\sigma}$, then $\overline{\delta}$ is a* proper face *of $\overline{\sigma}$. $\overline{\delta}$ is a* primary face *of $\overline{\sigma}$ if it is a face of $\overline{\sigma}$ and $\dim \overline{\delta} = \dim \overline{\sigma} - 1$. Given an elementary cube $\overline{\sigma} \in \mathcal{K}_p^d$, the set of all primary faces of $\overline{\sigma}$ is called the* border *of $\overline{\sigma}$ and it is denoted by $\partial\,\overline{\sigma}$.*

For example, let us consider the elementary cubes $\overline{\sigma}_1 = \{(x,0,0)\,|\,0 \leq x \leq 1\}$, $\overline{\sigma}_2 = \{(x,y,0)\,|\,0 \leq x,y \leq 1\}$ and $\overline{\sigma}_3 = \{(x,y,z)\,|\,0 \leq x,y,z \leq 1\}$. Notice that $\overline{\sigma}_1 \subseteq \overline{\sigma}_2 \subseteq \overline{\sigma}_3$ holds, and hence $\overline{\sigma}_1$, $\overline{\sigma}_2$ and $\overline{\sigma}_3$ are faces of $\overline{\sigma}_3$; $\overline{\sigma}_1$ and $\overline{\sigma}_2$ are proper faces of $\overline{\sigma}_3$; $\overline{\sigma}_1$ is a primary face of $\overline{\sigma}_2$ and $\overline{\sigma}_2$ is a primary face of $\overline{\sigma}_3$. We also have that $\partial\,\overline{\sigma}_2 = \{\overline{\sigma}_1, \overline{\sigma}_1', \overline{\sigma}_1'', \overline{\sigma}_1'''\}$ with $\overline{\sigma}_1' = \{(x,1,0)\,|\,0 \leq x \leq 1\}$, $\overline{\sigma}_1'' = \{(0,x,0)\,|\,0 \leq x \leq 1\}$, $\overline{\sigma}_1''' = \{(1,x,0)\,|\,0 \leq x \leq 1\}$.

Definition 4. *Let $\overline{I}$ be an elementary interval. The associated* elementary cell *is*

$$I = \begin{cases} (l, l+1) & \text{if } I = [l, l+1], \\ [l] & \text{if } I = [l]. \end{cases}$$

Let $\overline{\sigma} = \overline{I}_1 \times \overline{I}_2 \times \cdots \times \overline{I}_d \subset \mathbb{R}^d$ be an elementary cube, the associated elementary cell *is*

$$\sigma = I_1 \times I_1 \times \cdots \times I_d$$

The *dimension* of an *elementary cell* σ is defined as $\dim \overline{\sigma}$, i.e., the dimension of the associated elementary cube. The border for an elementary cell σ can also be defined as the set $\partial\,\sigma = \{\delta : \overline{\delta} \in \partial\,\overline{\sigma}\}$.

Definition 5. *A* cubical complex *is a set of elementary cells such that, given an elementary cell σ in the complex, all of its principal faces (the cells in $\partial\,\sigma$) are in the complex.*

For the sake of simplicity, hereafter we will say *cells* instead of elementary cells, bearing in mind that we refer to such kind of objects.

For example, Figure 1 (left) shows the cubical complex

$$K = \{ABCD, AC, CD, BD, AB, BE, A, B, C, D, E\}$$

This cubical complex has 1 cell of dimension 2 ($ABCD$), 5 cells of dimension 1 (AC, CD, BD, AB, BE) and 5 cells of dimension 0 (A, B, C, D, E).

When a cell is not a proper face of any cell in a given cell complex, it will be called *isolated cell*. A cell that is a proper face of exactly one cell in the complex is called *free face*. The following proposition links the concepts of free faces, proper faces and dimension. The proof can be found in [22].

Proposition 1. *Let δ be a free face in a cell complex and assume δ is a proper face of σ. Then σ is an isolated cell and* $\dim \delta = \dim \sigma - 1$.

As we are interested in obtaining a simpler representation for a cell complex whilst the topology is preserved. In the following definition, a way to *reduce* the number of cells in a cell complex is presented. This process reduces the number of cells by two and it does not change the topology of the cell complex.

For example, let us consider the cell complex of Figure 1 (left). The cells $ABCD$ and BE are isolated. The cells AC, CD, BD AB and E rare free faces, but A, B, C and D are not free faces, since they are proper faces of more than 1 cell complex.

Definition 6. *Let K be a cubical complex and δ a free cell in K. Let σ be the only cell in K such that δ is a proper face of σ. Let $K' = K \setminus \{\delta, \sigma\}$. K' is obtained from K via a process called* elementary collapse *of σ by δ.*

Let us consider again the cell complex K of Figure 1 (left). The cell E is a free face of BE and, hence, we can consider the elementary collapse of BE by E. The effect of such elementary collapse is the removal of E and BE from the cell complex K. Analogously, AC is a free face of $ABCD$. The elementary collapse of $ABCD$ by AC is the removal of both cells ($ABCD$ and AC) from K. Figure 1 (right) shows the final cubical complex obtained after both collapses.

Definition 7. *Let K be a cubical complex. A pair of cells $\langle \delta, \sigma \rangle$ is said to be a* simple pair *if following conditions hold:*

- *δ is a free cell in K.*
- *σ is the only cell such that $\delta \in \partial\, \sigma$.*

The cell σ is called the facet *of the simple pair.*

As shown in related literature [22, 37], simple pairs removal does not change the topology of the given cell complex.

3 Cell Complex Thinning

Skeletonization is usually considered as a pre-process in pattern recognition algorithms, but its study is also interesting by itself for the analysis of line-based

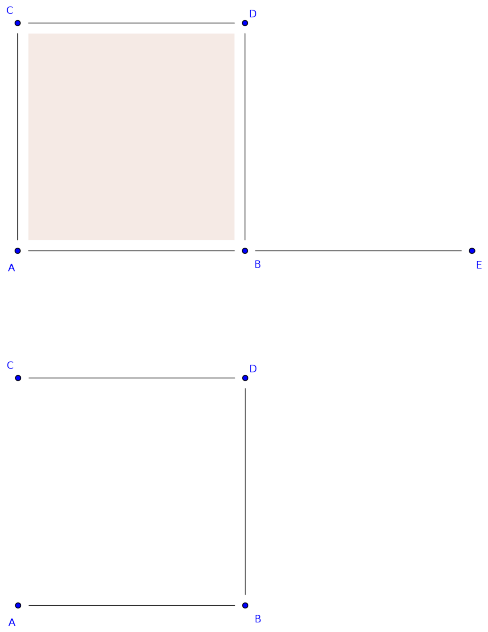


Fig. 1. Elementary collapse example: E collapses onto BE and AC collapses onto $ABDC$ in the image at the left, producing the image at the right.

images as texts, line drawings, human fingerprints or cartography. Skeletonization is a common transformation in Image Analysis. The concept of skeleton was introduced by Blum in [3], under the name of medial axis transform.

Let K be a cubical[2] cell complex and let ∂ be its border operator. As seen in the previous section, if only simple pairs of cells are removed, the topology is kept. For geometry preservation it is necessary to require some additional properties to those cells to be removed.

The basic idea of the algorithm is to define an iterative process where *outer* cells are removed. Here, the idea of *outer* cells makes reference to simple pairs, since in a simple pair $\langle \delta, \sigma \rangle$ the cell δ is a "terminal" cell as it does not lie in the border of any other one rather than σ.

In the process of iterative thinning, given a cell σ, we will denote the later iteration when σ is the facet of a simple pair by $R(\sigma)$. The earlier iteration when σ becomes isolated will be denoted by $I(\sigma)$. Liu *et al.* describe in [25] the relation between $I(\sigma)$ and $R(\sigma)$, and the maximum *isotropic* elongation in $p+1$ and p

[2] In the original work by Liu, [24], the thinning algorithm is designed for cell complexes of any kind, however we restrict to cubical complexes.

directions, respectively, since $\dim \sigma = p$. Thus, if σ is a p-cell in a cell complex, $I(\sigma)$ measures the shortest discrete distance from σ to the object boundary. This gives an idea of the size of the maximum disk centered at σ and inscribed in the object. On the other hand, $R(\sigma)$ measures the longest distance from σ to the object boundary going along the skeleton $(p-1)$-cells.

From the observation of the behaviour of previous measures, Liu defined two *difference measures*. The absolute one, $R(\sigma) - I(\sigma)$, is called *absolute medial persistence* and is denoted by MP_{abs}. On the other hand, *relative medial persistence* is defined as $1 - \frac{I(\sigma)}{R(\sigma)}$ and denoted by MP_{rel}. Both of them measure the duration in which a cell remains isolated during thinning process.

The cell complex thinning algorithm is shown in algorithm 1. It starts by initializing the isolated cells. Next, the thinning iterations start. In each iteration, all simple pairs are selected, all the pairs where the facet cell has one of the medial persistence measures less than given thresholds are chosen. Finally, the cells in selected simple pairs are removed from the cell complex. Otherwise, the cells are removed and the thinning iterations stop, else, the iteration counter increases and the thinning iterations continue. When the algorithm halts, a cell complex representing the skeleton for the initial one is obtained.

Algorithm 1 Cell complex thinning algorithm

Require: K cell complex, $\varepsilon_a, \varepsilon_r > 0$
  for all $\sigma \in K$ isolated **do**
    $I(\sigma) \leftarrow 0$
  end for
  iter $\leftarrow 1$
  repeat
    Let $S = \{\langle \delta, \sigma \rangle : \langle \delta, \sigma \rangle$ is a simple pair$\}$
    for all $\sigma \in \pi_2(S)$ **do**
      $R(\sigma) \leftarrow$ iter
    end for
    Let $S' = \{\langle \delta, \sigma \rangle \in S : MP_{abs}(\sigma) < \varepsilon_a \wedge MP_{rel}(\sigma) < \varepsilon_r\}$
    $K = K \setminus \{\sigma, \delta : \langle \delta, \sigma \rangle \in S'\}$
    for all $\sigma \in K$ new isolated cell **do**
      $I(\sigma) \leftarrow$ iter
    end for
    iter $\leftarrow$ iter $+ 1$
  until $S' = \emptyset$

Here $\pi_2(\langle \delta, \sigma \rangle) = \sigma$ is the second projection for the pair $\langle \delta, \sigma \rangle$.

4 Formal Framework

The chosen P system model for a Membrane Computing implementation of the algorithm is the *tissue-like P systems model* endowed with some extra ingredients.

As it is well-known, the biological inspirations of this model are intercellular communication and cooperation between neurons [26, 27]. The communication among cells is based on symport/antiport rules[3]. Tissue-like P systems have been widely used to solve computational problems in other areas (see e.g. [15, 16]), but recently, they have been also used in the study of digital images (e.g., [4, 8, 10, 17, 28, 29]). In this paper, we use a variant of tissue-like P systems where the application of the rules are regulated by *promoters* and *inhibitors*. These promoters have a clear biological inspiration. The rule is applied if the reactants are present, but it is also necessary the presence of all the promoters and none of the inhibitors in the corresponding cell. The promoters are not *consumed* nor *produced* by the application of the rule, but if they are not in the cell, the rule cannot be applied. In one step, each reactant in a membrane can only be used for one rule, but if several rules need the presence of the same promoter, then the presence of one unique copy of the promoter suffices for the application of the rules. In the general case, if there are several possibilities, the rule is non-deterministically chosen, but sometimes we will consider a priority relation between rules, so we need the concept of *priority* in our P systems. Next, we recall the formal definition of these P systems.

Definition 8. *A* tissue-like P system with promoters, inhibitors and priorities *of degree* $q \geq 1$ *is a tuple of the form*

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

where q *is the number of cells (or membranes) of the P system and*

1. Γ *is a finite alphabet, whose symbols will be called objects. These objects can be placed in the cells or in the surrounding space (called the environment).*
2. $\Sigma \subseteq \Gamma$ *is the input alphabet. The input of the computation performed by the P system is encoded by using this alphabet.*
3. $\mathcal{E} \subseteq \Gamma$ *is a finite alphabet representing the set of the objects in the environment. Following a biological inspiration, the objects in the environment are available in an arbitrary large amount of copies;*
4. $w_1, \ldots, w_q$ *are strings over* Γ *representing the multisets of objects placed inside the cells at the starting of the computation;*
5. $\mathcal{R}$ *is a finite set of rules of the following form:*

$$(pro\, \neg inh \,|\, i, u/v, j), \quad \text{for } 0 \leq i \neq j \leq q,\ pro, inh, u, v \in \Gamma^*$$

6. Pri *is a finite set of relations* $R_i > R_j$*, where* R_i *and* R_j *are rules from* $\mathcal{R}$*. It means that if* R_i *and* R_j *can be applied, then the application of* R_i *has priority on the application of* R_j*.*
7. $i_{in} \in \{1, 2, \ldots, q\}$ *denotes the input cell, i.e., the cell where the input of the computation will be placed.*
8. $i_{out} \in \{1, 2, \ldots, q\}$ *denotes the output cell, i.e., the cell where the output of the computation will be placed.*

[3] Introduced in Membrane Computing in [30].

Informally, a tissue-like P system with promoters, inhibitors and priorities of degree $q \geq 1$ can be seen as a set of q cells labeled by $1, 2, \ldots, q$. The cells are the nodes of a virtual graph, where the edges connecting the cells are determined by the communication rules of the P system, i.e., as usual in tissue-like P systems, the edges linking cells are not provided explicitly: If a rule $(pro\, \neg inh \,|\, i, u/v, j)$ is given, then cells i and j are considered linked. The application of a rule $(pro\, \neg inh \,|\, i, u/v, j)$ consists of trading the multiset u (initially in the cell i) against the multiset v (initially in j). After the application of the rule, the multiset u disappears from the cell i and it appears in the cell j. Analogously, the multiset v disappears from the cell j and it appears in the cell i. The trade can also be between one cell and the environment, labeled by 0. The rule is applied if in the cell with label i the objects of pro are present in the cell i (*promoters*), while any of the objects in inh do not appear in the cell (*inhibitors*). The promoters or the inhibitors are not modified by the application of the rule. If the promoters and inhibitors are empty, we will write $(i, u/v, j)$ instead of $(\emptyset\, \neg\emptyset |\, i, u/v, j)$. Finally, we write $(pro\,|i, u/v, j)$ or $(\neg inh\,|i, u/v, j)$ when only promoters or inhibitors appear, respectively.

As usual, we also consider that some objects not belonging to $\mathcal{E}$ can arrive to the environment during a computation. So, in a configuration (not initial) we could find two types of objects in the environment: Firstly, those which belong to the environment and appear in an arbitrary large number of copies. Secondly, those which not belong to the environment but are been sent to the environment by the application of a rule.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). A *configuration* is an instantaneous description of the P system and it is represented as a tuple $(w_0, w_1, \ldots, w_q)$, where 'W_0 is the multiset of objects from $\Gamma - \mathcal{E}$ placed in the environment (initially, $w_0 = \emptyset$). 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 configuration is *halting* when no rules can be applied to it. A *computation* is a sequence of computation steps such that either it is infinite or it is finite and the last step yields a halting configuration (i.e., no rules can be applied to it). Then, a computation halts when the P system reaches a halting configuration. The output of a computation is collected from its halting configuration by reading the objects contained in the output cell.

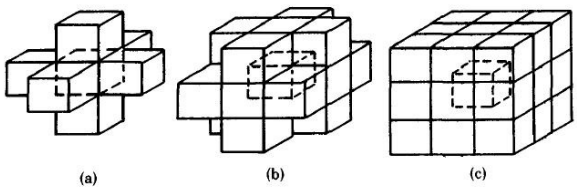
4.1 Image Algebra

Next, we recall some basic definitions from Image Algebra used in thi paper[4].

For a point set $X \subset \mathbb{Z}^2$, a *neighborhood function* is a function $N : X \to 2^{\mathbb{Z}^2}$. For each point $x \in X$, $N(x) \subset \mathbb{Z}^2$. The set $N(x)$ is called a *neighborhood* for x. There are two neighborhood function on subsets of $\mathbb{Z}^2$ which are of particular importance in image processing, the *von Neumann* neighborhood and the *Moore*

[4] A detailed introduction can be found in [32].

neighborhood. The first one, $N : X \to 2^{\mathbf{Z}^2}$, is defined by $N(x) = \{y : y = (x_1 \pm j, x_2)$ *or* $y = (x_1, x_2 \pm k),\ j, k \in \{0, 1\}\}$, where $x = (x_1, x_2) \in X \subset \mathbf{Z}^2$. While the Moore neighborhood $M : X \to 2^{\mathbf{Z}^2}$ is defined by $M(x) = \{y : y = (x_1 \pm j, x_2 \pm k),\ j, k \in \{0, 1\}\}$, where $x = (x_1, x_2) \in X \subset \mathbf{Z}^2$. The von Neumann and Moore neighborhood are also called the *four neighborhood* (4-adjacency) and *eight neighborhood* (8-adjacency), respectively.



(a)6-Neighbors (b) 18-Neighbors (c) 26-Neighbors

Fig. 2. Neighbors of a voxel in a cube

In $\mathbb{Z}^3$ two voxels are said to be 26-adjacent if they are distinct and each coordinate of one differs from the corresponding coordinate of the other by at most 1. Two voxels are 18-adjacent if they are 26-adjacent and differ in at most two of their coordinates; and two voxels are 6-adjacent if they are 26-adjacent and differ in at most one coordinate. That is to say each voxel has three kinds of *neighbors*: 6-neighbors which are also called face neighbors, 18-neighbors which are face and edge neighbors and 26-neighbors which are face, edge, and vertex neighbors, as they are shown in Figure 2. For $n = 4; 8; 6; 18$ or 26 an *n-neighbor* of a voxel p is a point that is n-adjacent to p.

The point sets with the usual operations has an algebra structure (see [32]).

A *Z-valued image* on X is any element of Z^X. Given a Z-valued image $I : X \to Z$, we will refer to Z as the set of possible range values of I, and to X as the spatial domain of I. The graph of an image is also referred to as the *data structure representation* of the image. Given the data structure representation $I = \{(x, I(x)) : x \in X\}$, then an element $(x, I(x))$ is called a *picture element* or *resel*[5]. The first coordinate x of a resel is called the *resel location* or *image point*, and the second coordinate $I(x)$ is called the *resel value* of I at location x.

For example, X could be a subset of $\mathbb{Z}^2$ where $x = (i, j)$ denotes spatial location, and Z could be a subset of $\mathbb{N}$, $\mathbb{N}^3$, etc. We call to the image set of the function I with domain X the *set of colors* or *alphabet of colors* and the image point of each resel is called *associated color*.

[5] The elements of a two-dimensional image are usually called *pixels*; the elements of a three dimensional image are usually called *voxels*, and the elements of a four-dimensional image are usually called *doxels* (*resel* in general).

5 Description of the Algorithm

In previous sections two kinds of objects has been reviewed. On one side, cell complexes achieves an useful link between continuous spaces and discrete structures where combinatorial algorithms may be developed using well-established properties and results by continuous topology. On the other hand, it has been settled a theoretical framework for working with images, considering them as a function from a topological discrete space to a set of "colors".

Our main goal is, starting from a k-dimensional binary image, build another image which represents a skeleton for the original one. In this process we will get a cell complex from the original image, skeletonize it and build back an image from the last skeleton. In this process no topological or shape information will be lost.

The set of points for our source images will be the set $[0,n)^k = \{0,1,\ldots,n-1\}^k \subset \mathbb{Z}^k$ equipped with a *cubic* neighbourhood function, described as follows: Two resels $\mathbf{i} = (i_1,\ldots,i_p,\ldots,i_k)$ and $\mathbf{j} = (j_1,\ldots,j_p,\ldots,j_k)$ are to be said $2k$-adjacent if $i_l = j_l$ for $l \neq p$ and $|i_p - j_p| = 1$. More formally, the neighbourhood function is given by

$$N(i_1,\ldots,i_k) = \left\{ (j_1,\ldots,j_k) \in [0,n)^k : j_l = \begin{cases} i_l & \text{if } l \neq p \\ i_l \pm 1 & \text{if } l = p \end{cases} ; 1 \leq p \leq k \right\}$$

This neighborhood function, when restricted to $k = 2$, gives the 4-adjacency, and 8-adjacency when $k = 3$.

Let $I : [0,n)^k \to \{0,1\}$ be a k-D binary image of size n^k, where the set of points in the object (or black points) is $I^{-1}(1)$. Let $K = K(I)$ be the cubic cell complex built from I. In K, the 0-cells represent points in the object, the 1-cells represent pairs of $2k$-adjacent points, the 2-cells represent unit squares where its edges are pairs of $2k$-adjacent points, and so on. In general, each p-cell is a p-dimensional unit hypercube whose edges are pairs of $2k$-adjacent points.

The cubical complex K built from an image I can be encoded using tuples in $[0,2n-1)^k$. The 0-cell $(i_1,\ldots,i_p)$ is encoded using the tuple $(2i_1,\ldots,2i_p)$. Higher dimension cells are encoded using tuples in $[0,2n-1)^k$ with many odd coordinates as the dimension of the cell. The way a p-cell is encoded using only one tuple is based in the idea of barycenter. Exactly, let σ be a p-cell with vertices given by $\mathbf{i}_1,\ldots,\mathbf{i}_{2p}$, and let us suppose that the vertices are sorted by lexicographic order. The set $\{\mathbf{v}_j = \mathbf{i}_j - \mathbf{i}_1 : 2 \leq j \leq 2p\}$ can be thought as a set of vectors in $\mathbb{R}^k$. From this set, we can extract a basis formed by vectors from the canonical one. Let $\{\mathbf{u}_1,\ldots,\mathbf{u}_p\}$ be that basis. In such situation, the cell σ is encoded by the tuple

$$2\mathbf{i}_1 + \sum_{j=1}^{p} \mathbf{u}_p$$

As the vectors $\mathbf{u}_j$ have all the coordinates 0, except one of them with value 1, and all of them are linearly independent, the dimension of cell σ is the number of odd coordinates in its encoded tuple, as we have said before.

The operator $\partial : [0, 2n-1)^k \to 2^{[0,2n-1)^k}$ given by

$$\partial(i_1, \ldots, i_k) = \\ = \left\{ (j_1, \ldots, j_k) \in [0, 2n-1)^k : j_l = \begin{cases} i_l \pm 1 \text{ if } i_l \equiv 1 (mod\ 2) \wedge l = p \\ i_l \qquad \text{in other case} \end{cases} ; 1 \leq p \leq k \right\}$$

gives all the possible cells in the border of the one represented by $(i_1, \ldots, i_k)$. When we would like to find the border cells for one in a complex K, we may use the *restricted border operator* given by

$$\partial|_K \, \mathbf{i} = \partial \, \mathbf{i} \cap K$$

In the definition of the rules for the family of tissue-like P systems which solves the proposed skeletonization problem, the use of the *inverse border operator* will be useful. It is defined as follows.

$$\partial^{-1}(i_1, \ldots, i_k) = \\ = \left\{ (j_1, \ldots, j_k) \in [0, 2n)^k : j_l = \begin{cases} i_l \pm 1 \text{ if } i_l \equiv 0 (mod\ 2) \wedge l = p \\ i_l \qquad \text{in other case} \end{cases} ; 1 \leq p \leq k \right\}$$

There is no difficult in observing that, for any $\mathbf{j} \in \partial \mathbf{i}$ is $\mathbf{i} \in \partial^{-1} \mathbf{j}$. So the use of the name "inverse border operator" is plenty justified.

The tissue-like P systems presented in this paper have six membranes. The first membrane is used as input and for marking the isolated cells before starting the thinning iterations. The second membrane is used to mark simple pairs. The third membrane selects the cells to be removed. The fourth membrane marks new isolated cells and update the counter I. The fifth membrane updates counter R and the sixth one is used as output membrane. Next, the P system is formally described.

Let I be a k-D binary image of size n^k, let K be the cubical cell built from I, let $\varepsilon_{abs} \in \{1, 2, \ldots, n\}$ and $\varepsilon_{rel} \in \{\tau_1, \ldots, \tau_m\} \subset (0, 1) \cap \mathbb{Q}$, where $\tau_j < \tau_{j+1}$ for $1 \leq j < m$. For every tuple $\langle n, \varepsilon_{abs}, \varepsilon_{rel} \rangle$ we will define a tissue-like P system with promoters, inhibitors, priorities and input, denoted by $\Pi(n, \varepsilon_{abs}, \varepsilon_{rel})$ and defined as follows:

$$\Pi(n, \varepsilon_{abs}, \varepsilon_{rel}) = (\Gamma, \Sigma, \mathcal{E}, w_1, \ldots, w_6, \mathcal{R}, Pri, i_{in}, i_o)$$

where:

- $\Gamma = \{\mathbf{i} : \mathbf{i} \in [0, 2n-1)^k\} \cup \{(I, \mathbf{i}) : \mathbf{i} \in [0, 2n-1)^k\} \cup$
  $\{(R, \mathbf{i}, d) : \mathbf{i} \in [0, 2n-1)^k, 1 \leq d \leq n\} \cup \{(I, \mathbf{i}, D) : \mathbf{i} \in [0, 2n-1)^k, 0 \leq D \leq n\} \cup$
  $\{S_{\mathbf{i}} : \mathbf{i} \in [0, 2n-1)^k\} \cup \{U_{\mathbf{i}} : \mathbf{i} \in [0, 2n-1)^k\} \cup \{R\}$
- $\Sigma = \{\mathbf{i} \in [0, 2n-1)^k : \mathbf{i} \in K\}$
- $w_1 = \{(R, \mathbf{i}, 1) : \mathbf{i} \in K\} \cup \{(I, \mathbf{i}, 0) : \mathbf{i} \in K\}$
- $w_2 = \ldots = w_6 = \emptyset$
- $\mathcal{E} = \Gamma \setminus \Sigma$
- $\mathcal{R}$ is the set of rules:

- $R_1 \equiv \big(\{\mathbf{i}\}\, \neg\partial^{-1}\mathbf{i}|1, \lambda/(I, \mathbf{i}), 0\big)$
  for $\mathbf{i} \in [0, 2n-1)^k$
  These rules mark isolated cells before starting thinning iterations.
- $R_2 \equiv (1, \mathbf{i}\,(R, \mathbf{i}, 1)\,(I, \mathbf{i}, 0)/\lambda, 2)$
  for $\mathbf{i} \in [0, 2n-1)^k$
- $R_3 \equiv (1, (I, \mathbf{i})/\lambda, 2)$
  for $\mathbf{i} \in [0, 2n-1)^k$
  These rules move objects to the second membrane for starting the thinning iterations.
- $R_4 \equiv \big(\{\mathbf{i}, \mathbf{j}\}\, \neg\left(\partial^{-1}\mathbf{j} \setminus \{\mathbf{i}\} \cup \{S_\mathbf{i}, S_\mathbf{j}\}\right)|2, \lambda/S_\mathbf{i}\, S_\mathbf{j}, 0\big)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $\mathbf{j} \in \partial\mathbf{i}$.
  These rules mark simple pairs.
- $R_5 \equiv (2, \mathbf{i}\,(R, \mathbf{i}, d)\,(I, \mathbf{i}, D)/\lambda, 3)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
- $R_6 \equiv (2, (I, \mathbf{i})/\lambda, 3)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
- $R_7 \equiv (2, S_\mathbf{i}\, S_\mathbf{j}/\lambda, 3)$
  for $\mathbf{i}, \mathbf{j} \in [0, 2n-1)^k$.
  These rules move objects to the third membrane for marking cells to be removed.
- $R_8 \equiv (\{S_\mathbf{i}, S_\mathbf{j}, (R, \mathbf{i}, d), (I, \mathbf{i}, D)\}\, \neg\{R_\mathbf{i}, R_\mathbf{j}\}|3, \lambda/RR_\mathbf{i}\, R_\mathbf{j}, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$, $\mathbf{j} \in \partial\mathbf{i}$,
  $0 \leq d, D \leq n$, $d \neq 0$,
  $d - D < \varepsilon_{abs}$ and $1 - \frac{D}{d} < \varepsilon_{rel}$
  These rules will mark for removal only those simple pairs whose higher dimension cell has not enough shape signification. Shape signification is calculated using medial persistence measures from [24, 25] . A cell is significant enough if both medial persistence measures are greater than some thresholds, given by ε_{abs} and ε_{rel} for MP_{abs} and MP_{rel}, respectively.
- $R_9 \equiv (\{R_\mathbf{i}\}|3, \mathbf{i}\,(R, \mathbf{i}, d)\,(I, \mathbf{i}, D)/\lambda, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
- $R_{10} \equiv (\{R_\mathbf{i}\}|3, (I, \mathbf{i})/\lambda, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
- $R_{11} \equiv (\{R_\mathbf{i}, R_\mathbf{j}\}|3, S\, S_\mathbf{i}\, S_\mathbf{j}/\lambda, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $\mathbf{j} \in \partial\mathbf{i}$
  These rules remove those simple pairs which are not significant enough.
- $R_{12} \equiv (\neg\{R_\mathbf{i}\}|3, \mathbf{i}\,(R, \mathbf{i}, d)\,(I, \mathbf{i}, D)/\lambda, 4)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
- $R_{13} \equiv (\neg\{R_\mathbf{i}\}|3, (I, \mathbf{i})/\lambda, 4)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
- $R_{14} \equiv (\neg\{R_\mathbf{i}, R_\mathbf{j}\}|3, S_\mathbf{i}\, S_\mathbf{j}/\lambda, 4)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $\mathbf{j} \in \partial\mathbf{i}$
  These rules send objects to the fourth membrane for marking new isolated cells.

- $R_{15} \equiv \big(\{\mathbf{i}, (R, \mathbf{i}, d)\}\, \neg(\partial^{-1}\mathbf{i} \cup \{(I, \mathbf{i})\}|4, (I, \mathbf{i}, D)/(I, \mathbf{i})\, (I, \mathbf{i}, d), 0\big)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
  These rules mark new isolated cells and update counter I.
- $R_{16} \equiv (4, \mathbf{i}\, (R, \mathbf{i}, d)\, (I, \mathbf{i}, D)/\lambda, 5)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
- $R_{17} \equiv (4, (I, \mathbf{i})/\lambda, 5)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
- $R_{18} \equiv (4, S\, S_{\mathbf{i}}\, S_{\mathbf{j}}/\lambda, 5)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $\mathbf{j} \in \partial\mathbf{i}$
  These rules send objects to the fifth membrane for updating counter R.
- $R_{19} \equiv (\{R\}\, \neg\{U_{\mathbf{i}}\}|5, (R, \mathbf{i}, d)/(R, \mathbf{i}, d+1)\, U_{\mathbf{i}}, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $1 \leq d \leq n$
  These rules update counter R.
- $R_{20} \equiv (\{U_{\mathbf{i}}\}|5, \mathbf{i}\, (R, \mathbf{i}, d)\, (I, \mathbf{i}, D)/\lambda, 2)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $0 \leq d, D \leq n$.
- $R_{21} \equiv (\{U_{\mathbf{i}}\}|5, (I, \mathbf{i})/\lambda, 2)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
  These rules move objects back to the second membrane for starting the next thinning iteration.
- $R_{22} \equiv (\{U_{\mathbf{i}}, U_{\mathbf{j}}\}|5, R\, S_{\mathbf{i}}\, S_{\mathbf{j}}/\lambda, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$ and $\mathbf{j} \in \partial\mathbf{i}$.
- $R_{23} \equiv (5, U_{\mathbf{i}}/\lambda, 0)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
  These rules remove unnecessary objects.
- $R_{24} \equiv (\neg\{R\}|5, \mathbf{i}/\lambda, 6)$
  for $\mathbf{i} \in [0, 2n-1)^k$.
  These rules send the skeletonized cell complex to the output membrane, when no cell has been removed in prior steps.

- $Pri = \{R_1 > R_p : p = 2, 3\} \cup \{R_4 > R_p : 5 \leq p \leq 7\} \cup$
  $\{R_{15} > R_p : 16 \leq p \leq 18\} \cup \{R_p > R_{23} : 19 \leq p \leq 22\} \cup$
  $\{R_8 > R_p : 12 \leq p \leq 14\}$
- $i_{in} = 1$ is the input cell.
- $i_{out} = 6$ is the output cell.

6 Overview of the Computation

Let $K \subset [0, 2n-1)^k$ be a cubical cell complex encoded as described above. Next, we will describe the behaviour of the P systems in the family Π when the input is set to K with thresholds set to ε_{abs} and ε_{rel} respectively. From now, $\mathcal{C}_c$ will denote the c-th configuration for the P system.

In order to make this overview more understandable, the process will be illustrated by the thinning process of image shown in figure 3.

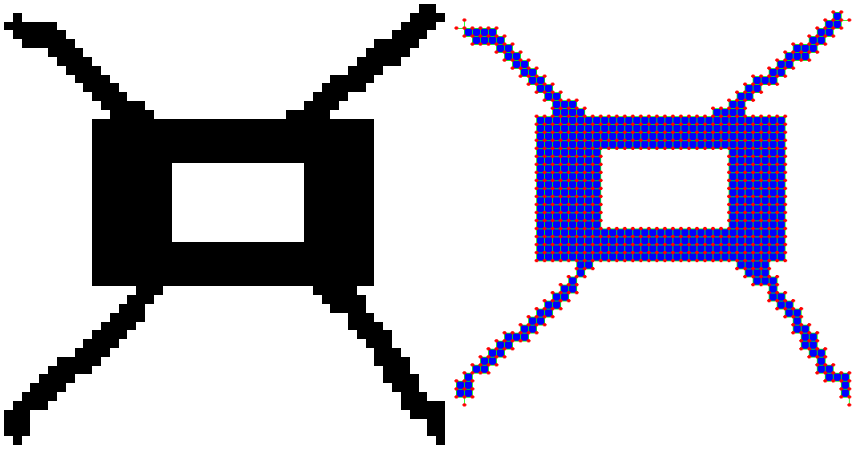


Fig. 3. Example image to show the thinning process, on the left. On the right is the cell complex representation for the image. Blue squares represent 2-cells, green lines represent 1-cells and red dots represent 0-cells.

In the initial state $\mathcal{C}_0$, the first membrane stores one object $\mathbf{i}$ for each cell in K. The initial values for counters R and I, given by objects $(R, \mathbf{i}, 1)$ and $(I, \mathbf{i}, 0)$, are also stored in the first membrane. In this situation, only rules R_1, R_2 or R_3 can be applied. For priority reasons, the rules R_1 are the only one that can be selected. After apply the selected rules from R_1, in $\mathcal{C}_1$, the first membrane contains objects $\mathbf{i}$ (for cells in K), counters R and I, and *isolation marks* $(I, \mathbf{i})$ for each isolated cell $\mathbf{i}$.

In the configuration $\mathcal{C}_1$, only the rules R_2 and R_3 can be selected, moving the cell objects $\mathbf{i}$, along with the isolation marks $(I, \mathbf{i})$ and counters $(R, \mathbf{i}, 1)$ and $(I, \mathbf{i}, 0)$, to the second membrane. The application of these rules gives as result the next configuration, $\mathcal{C}_2$. In this situation, only the rules establishing communications with the second membrane can be selected. Hence, the P system must select rules from $\{R_4, R_5, R_6, R_7\}$. However, for priority reasons, only the rules R_4 can be selected and applied, arising to the next configuration, where simple pairs $\langle j, i \rangle$ are marked by the presence of objects $S_{\mathbf{j}}$ and $S_{\mathbf{i}}$ in the second membrane.

In the current configuration, $\mathcal{C}_3$, only rules R_5, R_6 and R_7 can be selected. The application of them gives as result the configuration $\mathcal{C}_4$, where objects have been moved from the second to the third membrane. In the third membrane the simple pairs are going to be examined in order to detect those to be marked for removal, when they were not significant enough. In this situation, only rules R_8 can be selected and their application arises to the next configuration, $\mathcal{C}_5$, where those simple pairs $\langle \mathbf{j}, \mathbf{i} \rangle$ that can be removed are marked by $R_{\mathbf{j}}$ and $R_{\mathbf{i}}$.

In the previous configuration, only rules R_p, for $9 \leq p \leq 14$, can be selected. The application of these rules makes the P system evolve to the configuration $\mathcal{C}_6$,

where selected simple pairs have been removed, along with the auxiliary objects, and the remaining objects have been moved from the third to the fourth membrane.

In the configuration $\mathcal{C}_6$, for priority reasons again, only the rules R_{15} can be selected, and their application marks the new isolated cells and updates the counter $(I, \mathbf{i}, D)$. Now, all available objects are updated in the fourth membrane, in the configuration $\mathcal{C}_7$. Then, only rules R_{16}, R_{17} and R_{18} can be applied, resulting in the configuration $\mathcal{C}_8$ where all the objects in the fourth membrane are moved to the fifth one.

If no simple pairs have been marked for removal in configuration $\mathcal{C}_9$, there is no marker R in the fifth membrane. In this situation, the only rules that can be applied are those in R_{24}. The application of these rules leaves the P system in the configuration $\mathcal{C}_{10}$ which also is a halting configuration.

Let us Ssppose there have been some simple pairs marked for removal in configuration $\mathcal{C}_8$, which ensures the presence of marker R in the P system. Then, the application of rules in R_{19} updates the counter R, leaving the P system in the configuration C_9. In this situation, only rules in R_{20}, R_{21} and R_{22} can be applied. The former move objects to the second membrane, where the thinning iterations restart, the latter removes auxiliary objects from the fifth membrane. In this situation, the P system is in the configuration $\mathcal{C}_{10}$. The result for the example image is shown in Figure 4 (Right).

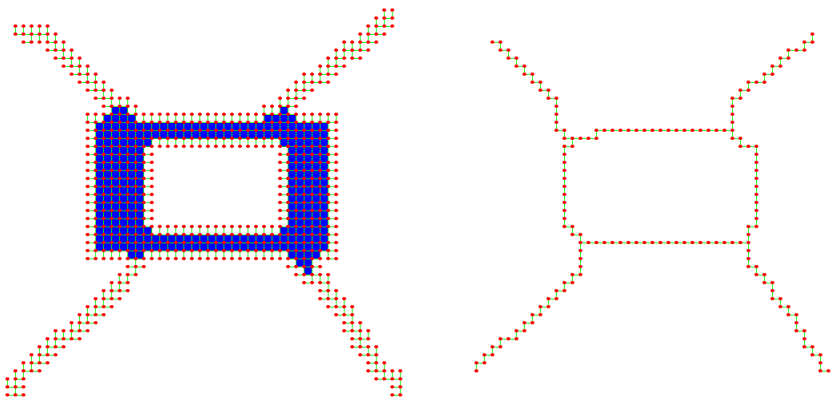


Fig. 4. (Left) Cell complex representation for cells in membrane 2 after the first thinning iteration. (Right) The thinned cell complex.

In previous situation, only the rules in R_4 and R_{23} can be applied. The former marks simple pairs in the second membrane, while the latter remove auxiliary remaining objects in the fifth membrane, leaving the P system in the configuration $\mathcal{C}_{11}$. From this point, the P system will evolve as above until it reaches the configuration $\mathcal{C}_{16}$ whether the halting condition may be reached in next configuration

$\mathcal{C}_{17}$, or not, depending on the presence of marker R. In the first case, the P system will start a new thinning iteration. In the second situation, the P system sends out the skeleton to the output membrane.

In any case, the P system will reach the halting configuration in $7t + 3$ steps, where t stands for the thinning iterations performed. If we start from a k-D binary image of size n^k where all the resels are black, and we do not pay attention to the shape significance, we perform a full thinning in a number of thinning iterations which, in addition, is the maximum. We have found that, in situation above, the greater number of thinning iterations is given by $k(n + 1)$. Hence, we can ensure that the P system halts in, at most, $7k(n + 1) + 3$ computation steps.

In Figure 4 (Right), the resulting image, representing the cell complex in the sixth membrane when the halting condition is reached, is shown.

The required computational resources for the family of tissue-like P systems defined in this paper is given in the table 1.

k-D binary image thinning problem	
Complexity	
Number of steps of computation	$\leq 7k(n+1)+3$
Resources needed	
Size of the alphabet	$O(n^{k+1})$
Initial number of cells	6
Initial number of objects	$3\|K\|$
Number of rules	$O(n^{k+2})$
Upper bound for the length of the rules	3

Table 1. Complexity aspects, where the size of the input data is $O(n^k)$, $|K|$ is the number of cells in the input cell complex K.

7 Conclusions and Future Work

In this paper, we bring together Membrane Computing and Cell Complexes. Both disciplines deal with compartments of the Euclidean space on their foundations, but their inspiration and motivation are quite different. The former is a computation model inspired in the functioning of living cells and tissues and the latter is born as a tool for handle concepts of Algebraic Topology.

In this paper, we use Membrane Computing techniques to implement a cell complex based algorithm for thinning images and show a new proof that the Membrane Computing framework is flexible enough to adapt to unexpected situations. In this way, this is a pioneer work that open a new research line that can be followed at different levels.

Firstly, we can study if other P system models (cell-like P systems, SN P systems, a most restrictive model of tissue-like P systems, ...) are *better* than

the one used in this paper to implement the Liu's algorithm in the Membrane Computing framework. *Better* should be considered here in a broad sense, since it can mean with a lower amount of resources, with less *ingredients* in the P system model o more efficient in some sense.

Another line to follow is to study if other problems in Algebraic Topology already studied with Cells Complexes can be considered in the framework of Membrane Computing. This research line can open a flow of inquiries and solutions in both directions enriching both disciplines with new points of view.

Finally, a more general question is the study of links on the foundations of Membrane Computing and Cell Complexes. As pointed out above, both disciplines shares a compartmental view of the Euclidean space and this can be a starting point for a deeper study of their common properties.

Acknowledgements

DDP and MAGN 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. Adeoye, O.S.: A survey of emerging biometric technologies. International Journal of Computer Applications 9(10), 1–5 (November 2010)
2. Ayache, N.: Medical image analysis and simulation. In: Shyamasundar, R.K., Ueda, K. (eds.) ASIAN. Lecture Notes in Computer Science, vol. 1345, pp. 4–17. Springer (1997)
3. Blum, H.: An associative machine for dealing with the visual field and some of its biological implications. In: Bernard, E.E., Kare, M.R. (eds.) Biological Prototypes and Synthetic Systems. vol. 1, pp. 244–260. Plenum Press, New York (1962)
4. Carnero, J., Díaz-Pernil, D., Gutiérrez-Naranjo, M.A.: Designing tissue-like P systems for image segmentation on parallel architectures. In: del Amor, M.A.M., Păun, Gh., de Mendoza, I.P.H., Romero-Campero, F.J., Cabrera, L.V. (eds.) Ninth Brainstorming Week on Membrane Computing. pp. 43–62. Fénix Editora, Sevilla, Spain (2011)
5. Ceterchi, R., Gramatovici, R., Jonoska, N., Subramanian, K.G.: Tissue-like P systems with active membranes for picture generation. Fundamenta Informaticae 56(4), 311–328 (2003)
6. Ceterchi, R., Mutyam, M., Păun, Gh., Subramanian, K.G.: Array-rewriting P systems. Natural Computing 2(3), 229–249 (2003)
7. Christinal, H.A., Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Thresholding of 2D images with cell-like P systems. Romanian Journal of Information Science and Technology 13(2), 131–140 (2010)

8. Christinal, H.A., Díaz-Pernil, D., Real, P.: Segmentation in 2D and 3D image using tissue-like P system. In: Bayro-Corrochano, E., Eklundh, J.O. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Lecture Notes in Computer Science, vol. 5856, pp. 169–176. Springer (2009)
9. Christinal, H.A., Díaz-Pernil, D., Real, P.: P systems and computational algebraic topology. Mathematical and Computer Modelling 52(11-12), 1982 – 1996 (2010)
10. Christinal, H.A., Díaz-Pernil, D., Real, P.: Region-based segmentation of 2D and 3D images with tissue-like P systems. Pattern Recognition Letters 32(16), 2206 – 2212 (2011)
11. Collins, R., Lipton, A., Kanade, T.: Introduction to the special section on video surveillance. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8), 745 –746 (2000)
12. Dersanambika, K.S., Krithivasan, K.: Contextual array P systems. International Journal of Computer Mathematics 81(8), 955–969 (2004)
13. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Molina-Abril, H., Real, P.: A bio-inspired software for segmenting digital images. In: Nagar, A.K., Thamburaj, R., Li, K., Tang, Z., Li, R. (eds.) Proceedings of the 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications BIC-TA. vol. 2, pp. 1377 – 1381. IEEE Computer Society, Beijing, China (2010)
14. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Molina-Abril, H., Real, P.: Designing a new software tool for digital imagery based on P systems. Natural Computing pp. 1–6 (2011)
15. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: A linear-time tissue P system based solution for the 3-coloring problem. Electronic Notes in Theoretical Computer Science 171(2), 81–93 (2007)
16. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Solving subset sum in linear time by using tissue P systems with cell division. In: Mira, J., Álvarez, J.R. (eds.) IWINAC (1). Lecture Notes in Computer Science, vol. 4527, pp. 170–179. Springer (2007)
17. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Real, P., Sánchez-Canales, V.: Computing homology groups in binary 2D imagery by tissue-like P systems. Romanian Journal of Information Science and Technology 13(2), 141–152 (2010)
18. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networks - a review. Pattern Recognition 35(10), 2279–2301 (2002)
19. Ferretti, C., Mauri, G., Zandron, C.: P systems with string objects. In: Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) The Oxford Handbook of Membrane Computing, pp. 168 – 197. Oxford University Press, Oxford, England (2010)
20. Gimel'farb, G., Nicolescu, R., Ragavan, S.: P systems in stereo matching. In: Real, P., Díaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 6855, pp. 285–292. Springer (2011)
21. Gimel'farb, G.L.: Probabilistic regularisation and symmetry in binocular dynamic programming stereo. Pattern Recognition Letters 23(4), 431–442 (2002)
22. Kaczyński, T., Mischaikow, K., Mrozek, M.: Computational homology. Applied mathematical sciences, Springer (2004)
23. Krishna, S.N., Rama, R., Krithivasan, K.: P systems with picture objects. Acta Cybernetica 15(1), 53–74 (2001)
24. Liu, L.: 3D thinning on cell complexes for computing curve and surface skeletons. Washington University (2009)

25. Liu, L., Chambers, E.W., Letscher, D., Ju, T.: A simple and robust thinning algorithm on cell complexes. Computer Graphics Forum 29(7), 2253–2260 (2010)
26. Martín-Vide, C., Pazos, J., Păun, Gh., Rodríguez-Patón, A.: A new class of symbolic abstract neural nets: Tissue P systems. In: Ibarra, O.H., Zhang, L. (eds.) COCOON. Lecture Notes in Computer Science, vol. 2387, pp. 290–299. Springer (2002)
27. Martín-Vide, C., Păun, Gh., Pazos, J., Rodríguez-Patón, A.: Tissue P systems. Theoretical Computer Science 296(2), 295–326 (2003)
28. Peña-Cantillana, F., Díaz-Pernil, D., Berciano, A., Gutiérrez-Naranjo, M.A.: A parallel implementation of the thresholding problem by using tissue-like P systems. In: Real, P., Díaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W.G. (eds.) CAIP (2). Lecture Notes in Computer Science, vol. 6855, pp. 277–284. Springer (2011)
29. Peña-Cantillana, F., Díaz-Pernil, D., Christinal, H.A., Gutiérrez-Naranjo, M.A.: Implementation on CUDA of the smoothing problem with tissue-like P systems. International Journal of Natural Computing Research 2(3), 25–34 (2011)
30. Păun, A., Păun, Gh.: The power of communication: P systems with symport/antiport. New Generation Computing 20(3), 295–306 (2002)
31. Păun, Gh.: Computing with membranes. Journal of Computer and System Sciences 61(1), 108–143 (2000)
32. Ritter, G.X., Wilson, J.N., Davidson, J.L.: Image algebra: An overview. Computer Vision, Graphics, and Image Processing 49(3), 297–331 (1990)
33. Rosin, P.L.: Training cellular automata for image processing. IEEE Transactions on Image Processing 15(7), 2076–2087 (2006)
34. Saeed, K., Tabedzki, M., Rybnik, M., Adamski, M.: K3M: A universal algorithm for image skeletonization and a review of thinning techniques. Applied Mathematics and Computer Science 20(2), 317–335 (2010)
35. Selvapeter, P.J., Hordijk, W.: Cellular automata for image noise filtering. In: NaBIC. pp. 193–197. IEEE (2009)
36. Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA (2001)
37. Zhou, Q.Y., Ju, T., Hu, S.M.: Topology repair of solid models using skeletons. IEEE Transactions on Visualization and Computer Graphics 13(4), 675–685 (2007)
38. Zhou, Y., Chellappa, R.: Artificial neural networks for computer vision. Research notes in neural computing, Springer-Verlag (1992)