Int. J. Advance Soft Compu. Appl, Vol. 7, No. 1, March 2015
ISSN 2074-8523

Membrane Computing for 2D Image Segmentation

Rafaa Ismael Yahya[1] , Shafaatunnur Hasan[1], Loay Edwar George [2], and Bisan Alsalibi[3]

[1]UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research
Universiti Teknologi Malaysia, UTM Skudai, 81310 Johor, Malaysia
e-mail: rafaa_ismaeel@yahoo.com, shafaatunnur@utm.my

[2]Computer Department,
University of Baghdad, Bghdad, IRAQ
e-mail: loayedwar57@yahoo.com

[3]School of Computer Sciences
Universiti Sains Malaysia, USM, 11800 Penang, Malaysia
e-mail: besansalipi@gmail.com

Abstract

Inspired by the structure and functioning of the biological cell, membrane computing is a novel class of computational models where its devices are called P systems. In this paper tissue-like P system is proposed to improve region based segmentation using 2D image and 4-adjacency neighborhood relation between pixels. Artificial image is used and the algorithm is simulated using tissue simulator. This work proves that region based segmentation with membrane computing is done in a constant number of steps (9 steps) regardless of the size of the image Furthermore, different color relations have been explored to show the effect of color on the segmentation results.

Keywords: *Membrane computing, Region-based image segmentation, Tissue-like P systems.*

1 Introduction

Natural computing is a research field of computational paradigms inspired from nature. It is growing very fast and there are several fields that are already well established like cellular automata [1], genetic algorithms [2], neural networks [3], Amorphous computing [4], DNA-based molecular computing [5] and recently membrane computing. Membrane computing is a branch of natural computing that takes its inspiration from the cell structure and function to solve many problems.

Membrane computing explores, abstracts and formalizes new method of computation inspired by the natural membrane model [6]. In addition, cell membranes do not bound only compartments where reactions in a cell develop on the membranes, that is catalyzed by the many proteins etc., but in general the cells are organized as tissues, organs and organisms [7].

Membrane computing was first initiated by Gheorge Paun and generated a new computing model inspired from cells termed P systems. In recent years, many different models of P systems have been proposed, such as cell-like (inspired from the cell structure), tissue-like (inspired from the organization of cells in tissues) and neural-like (related to the way neurons are linked in neural nets).

The huge inherent parallelism of membrane computing has drawn great attention recently and there are a number of applications reported in several areas; biology, bio-medicine, linguistics, computer graphics, economics, approximate optimization and cryptography [8] .

Membrane Computing has features such as the encapsulation of data and the simple representation of information as well as parallelism, all of which are most appropriate when dealing with digital images. And because features in the segmentation of digital images that are suitable and easy to implement in any technique inspired by nature. One of the characteristics is that it can be parallelized and locally solved. Unconcerned with the size of the picture, it can be performed in parallel in different local areas. Another interesting characteristic is that the basic necessary information can be easily encoded by bio-inspired representation.

Segmentation in computer vision [9] means to partition image into several parts that can be easily understood and analyzed. The result of the process of segmentation is to locate the objects and boundaries in the form of lines and curves in an image. More precisely it is assigning labels to every pixel in an image which have the same characteristics. The result of this labelling of pixels is to share certain visual characteristics. There are many published papers in segmentation with membrane computing, and these include study several methods which have been designed to segment images with membrane computing. [10] proposed a tissue-like P systems to improve the standard edge-based segmentation method. [11] presented a cell-like P systems to solve the threshold problem in linear time for a number of pixels. [12] proposed a new hardware system using a tissue like P system for threshold metrics and the implementation of edge based detection for noise removal. [13] presented a new segmentation technique using a tissue like P systems with multiple auxiliary cells. [14] developed a tissue-like P systems to design a region based segmentation algorithm for 2D and 3D imaging.

To solve the threshold problem, [15] presented a membrane computing approach with novel architecture called CUDA^{TM} (Compute Unified Device Architecture). [16] used a tissue-like P systems to solve segmentation problems via hardware implementation in Field Programmable Gate Arrays (FPGAs). In medical imaging, [17] proposed membrane computing to enhance morphological segmentation methods. To segment color images via threshold segmentation, [18] presented a membrane computing designed threshold approach to segment color images. [19] presented a tissue-like P systems to improve region growing for the segmentation process. To improve threshold segmentation, [20] proposed a novel threshold approach using a cell-like P systems and [21] used a tissue-like P systems algorithm to design a gradient-based edge detection technique implemented with the CUDA^{TM} device. [22] proposed a novel segmentation using tissue-like P systems to adaptive tradition method region based color segmentation.

From the given previous work, there are still many improvements that can be made dealing with membrane computing for image segmentation. However, the difficulties of understanding the concept for new comers are always arising since no such simple methodology has been given accordingly.

Hence, in this paper, a further illustration of the membrane computing concept will be presented using a simple image of an apple that will be the input image as shown in Fig. 1, with results and segmentation shown later.

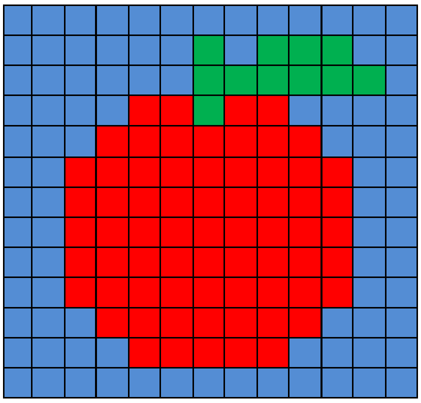


Fig. 1: Image of an apple

In this paper, region-based segmentation is used together with membrane computing. The structure of the paper is organized as follows: Section 2 provides definition the membrane computing. Section 3 describes tissue-like P systems. Section 4 gives the tissue-like P systems for 2D image segmentation, Section 5 presents the experimental results ,Section 6 concludes the paper with suggested future work and Section 7 acknowledgment.

2 Membrane Computing

Membrane computing is the theoretical model of computation inspired by the functioning and structure of living cells of organisms [12]. Membrane computing was first initiated by Gheorge Paun in 1998 where he generated a new computing model inspired from cells termed P systems. The basic elements of a P system are inspired from the cells structure and functions that make P system computation devise consisting of (1) membrane structure (2) set of evolution rules (3) multisets.

The design of membrane computing is presented in a hierarchically structured as a cell. It is divided into many compartments (according to the cell) and the external membranes look like plasma membrane in the cell containing several sub-membranes called skin. Each membrane surrounding the compartment is called a region. A membranes, which do not have a sub-membrane is called an elementary membrane. Every membrane has a label starting from number 1 to the skin membranes. The structure of the membrane can be represented like a tree inspired from the vesicles where the root of the tree is the skin membrane and the leaves are the elementary membrane. This tree structure is represented by parentheses to explain the structure of membrane as shown in Fig. 2. The motorists are the set of objects placed in the region, according to the chemical objects in the cell compartment. These objects are described by the symbolic alphabet [23].

There are many rules for handling the creation, destruction, division, merging, etc. of membranes. These rules can have promoters or inhibitors and can be regulated by a priority relation. The permeability of membranes can be controlled by the rules indefinitely either with a biological approach or with a mathematical motivation [24].

There are several features that are genuinely suitable to membrane computing and which are of interest to many applications such as distribution, discrete mathematics, algorithmicity, scalability/extensibility, transparency, massive parallelism, non-determinism and communication [25].

Many applications of membrane computing have been reported in several areas like linguistics, cryptography, bio-medicine, biology, computer graphics, economics, approximate optimization, and others. Several software products for simulating P systems and attempts to implement P systems on a dedicated hardware were also reported [26].

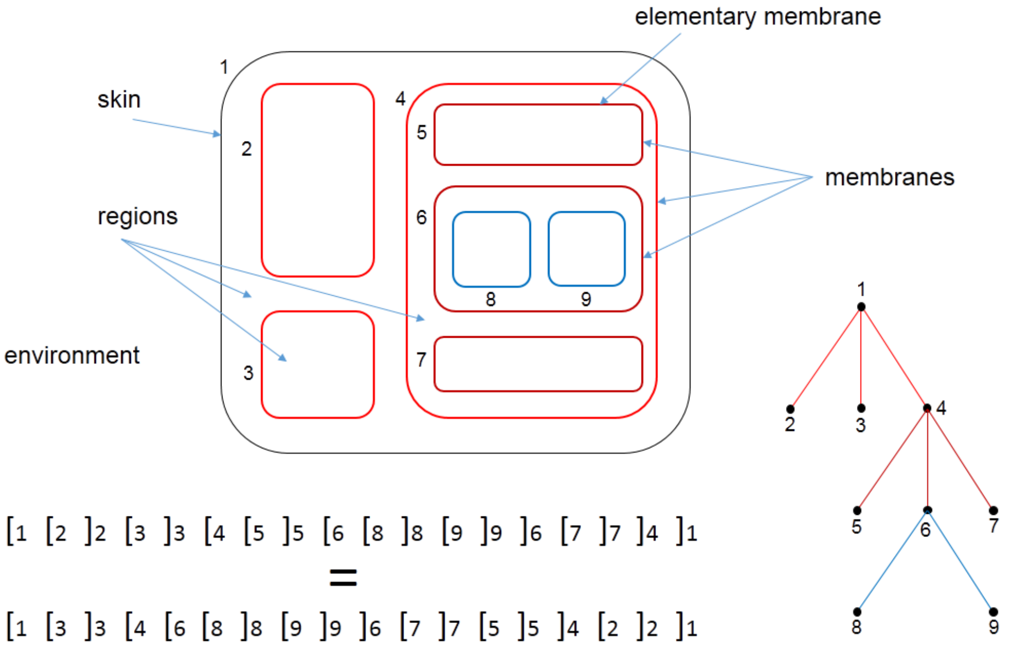


Fig. 2: A membrane structure and its associated tree [9]

3 Tissue like-P systems Description

Martin Vide [27] was the pioneer of tissue-like P system. Tissue-like P systems has two biological inspirations: intercellular communication and cooperation between neurons. These two mechanisms have a common mathematical model which is a network of processors that works with symbols and communicates these symbols by specific channel. The main feature of tissue-like P systems is that the cell does not have polarization and the structure of the graph is general.

The form of tissue-like P systems model with input of degree q ≥ 1 is a tuple

$$\Pi = (\Gamma, \Sigma, \varepsilon, w_1, \dots, w_q, R, i_{\Pi,} o_{\Pi})$$

Where

a) Γ is a finite alphabet, whose symbols will be called objects;

b) $\Sigma(\subset \Gamma)$ is the input alphabet;

c) $\varepsilon \subseteq \Gamma$ (the objects in the environment);

d) $w_1, . . ., w_q$ are strings over Γ representing the multi-sets of objects associated with the cells at the initial configuration;

e) R is a finite set of rule of the form.

1. Communication rule: $(i, u/v, j)$, for $i, j \in \{0,1,2, \dots, q\}, i \neq j, u, v \in \Gamma$;
2. Division rules: $[a]_i \rightarrow [b]_i\,[c]_i$, where $i \in \{1,2, \dots, q\}$ and $a, b, c \in \Gamma$;

f) $i_\Pi \in \{1,2, \dots ,q\}$ is the input cell;

g) $o_\Pi \in \{0,1,2, \dots ,q\}$ is the output cells

We can see a tissue- like P systems of degree q ≥ 1 as a set of q cells (each cell consists of an elementary membrane) the sets labeled by 1, 2...q. We refer 0 to the label of the environment, the input of the region denoted by i_Π and the output of the region cell (inside the region or the environment) is denoted by i_0 . The communication rules can determine a virtual graph, where the cells can be as nodes and when the cell communicates directly it is possible to indicate edges. The graph will be dynamic, because from the application of division rules new nodes can emerge and are produced by the application of division rules.

String $w_i, \dots, w_q$ is the initial multi-sets of objects located in q cells of the system.

We explicate that $\mathcal{E} \subseteq \Gamma$ is the set of objects located in the environment, each object having an arbitrarily large number of copies.

The division rule $[a]_i \rightarrow [b]_i\,[c]_i$ is used to divide the cell i which contain object *a* to two new cells and all objects in the original cell i are replicated and copied with each of the new cells resulted from the division rule, with the exception of the object *a* which is replaced by object b in the first cell and c in the other cell. In membrane computing framework, all rules are used maximally in parallel. In the first step, one rule takes one object from the cell, and where there are several possibilities the rule is non-deterministically chosen, and in each step a set of rules will apply. By applying the rules, there is only one restriction which is that when a cell is divided in that step the division rule is the only rule applied and the object inside the cell cannot be communicated within that step.

The cells resulting from the division rules have the same label of the original cell and if a cell is divided, the interaction is blocked with other cell or the environment during the mitosis process. It means that while a cell is dividing it closes the communication channels with other cells and with the environment.

The configuration provides a quick description of the P systems. From the configuration, we can implement a computation step and get a new configuration by applying the rules in parallel as shown above. A computation is a sequence of steps and the result is either finite or infinite and at the end of computation it yields a halting configuration (i.e., no rules can be applied to it). Then, a computation halts when the system reaches a halting configuration [21].

However, in our study, the findings are based on the region segmentation for the simple image, i.e., an apple using region segmentation of 2D image of Christinal method [14], and will be discussed in the next section.

4 Tissue-like P systems for 2D Image Segmentation

Segmentation is used on a region based image. Regions in an image are a group of pixel connected together according to similarity of properties of these pixels. Region segmentation segments the image into parts of the region (which are the pixels being adjacent and having the associated color) by using predefined criteria. In tissue like P systems the images are segmented according to the different colors between the pixels.

In this paper, the previous work of Hepzipah A. Chrestinal et al. is dopted [14] to implement a simple apple image for better illustration of how the membrane works. In previous work they used 8×8 with very simple image. In this work image size is 13×13 with more details to illustrate and prove that in membrane computing the algorithm requires only 9 steps to obtain region based segmentation regardless of the size of the image. The region image based segmentation method is applied by tissue like P system to segment the image.

5 Experimental results

The image shown in Fig. 1 is used in this paper, this is an artificial image (apple), contains only three colors. Hence, the number of communication rules used in the segmentation will depend on these three colors. In case the image has more than three colors the number of the communication rules will increase. The program used to check the validity of these systems is tissue simulator, as proposed by Borrego [28]. The artificial image (see the Fig. 3) is encoded into the input objects a_{ij} that every pixel have the position (matrix) and colored pixels from a 2D digital image as shown in Fig. 4. The family of the system is simulated to segment 2D images with this simulation. The rules are given as follows.

For each n, m ∈ N, we consider the tissue-like *P-systems* of degree 2 as:

$$\Pi(n,m) = (\Gamma, \Sigma, \varepsilon, w_1, w_2, R, i_{\pi}, 0_{\pi}),$$

a) The working alphabet is $\Gamma = \Sigma \cup \varepsilon$,

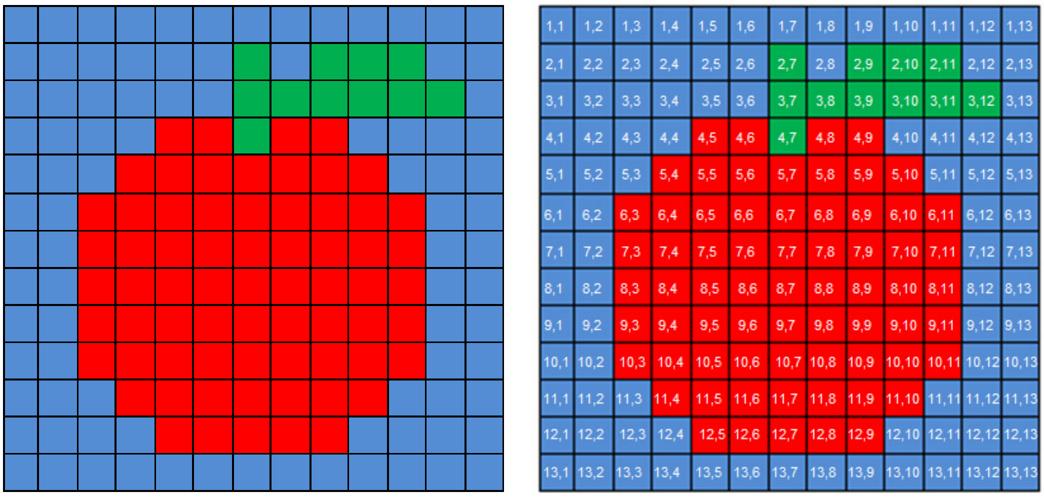
b) The input alphabet is

$$\Sigma = \{a_{ij}, b_{ij}, . a, b \in c, 1 \leq i \leq n, 1 \leq j \leq m, a < b\},$$

c) The environment alphabet is

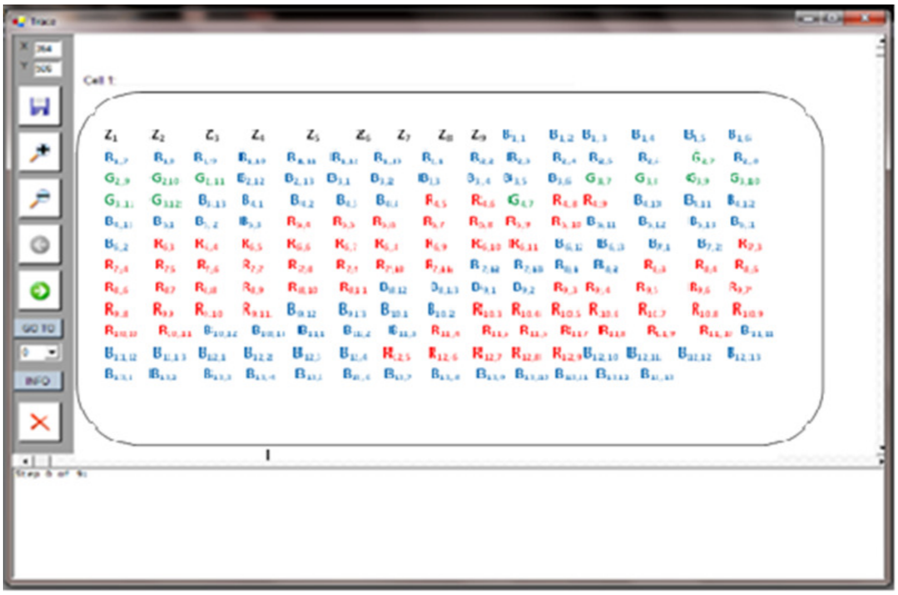
$$\varepsilon = \{X_{ij}, 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{z_i : 1 \leq i \leq 9\};$$

The multisets of cells 1 and 2 are w1= $z_1^{\lceil r_1^{(1/2^7)} \rceil}$, $w_2 = \theta$, respectively.



1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11	2,12	2,13
3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	3,13
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13
6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9	6,10	6,11	6,12	6,13
7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9	7,10	7,11	7,12	7,13
8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9	8,10	8,11	8,12	8,13
9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9	9,10	9,11	9,12	9,13
10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9	10,10	10,11	10,12	10,13
11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9	11,10	11,11	11,12	11,13
12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9	12,10	12,11	12,12	12,13
13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9	13,10	13,11	13,12	13,13

Fig. 3: Input image



Cell 1:

z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8 z_9 $B_{1,1}$ $B_{1,2}$ $B_{1,3}$ $B_{1,4}$ $B_{1,5}$ $B_{1,6}$
$B_{1,7}$ $B_{1,8}$ $B_{1,9}$ $B_{1,10}$ $B_{1,11}$ $B_{1,12}$ $B_{1,13}$ $B_{2,1}$ $B_{2,2}$ $B_{2,3}$ $B_{2,4}$ $B_{2,5}$ $B_{2,6}$ $G_{2,7}$ $B_{2,8}$
$G_{2,9}$ $G_{2,10}$ $G_{2,11}$ $B_{2,12}$ $B_{2,13}$ $B_{3,1}$ $B_{3,2}$ $B_{3,3}$ $B_{3,4}$ $B_{3,5}$ $B_{3,6}$ $G_{3,7}$ $G_{3,8}$ $G_{3,9}$ $G_{3,10}$
$G_{3,11}$ $G_{3,12}$ $B_{3,13}$ $B_{4,1}$ $B_{4,2}$ $B_{4,3}$ $B_{4,4}$ $R_{4,5}$ $R_{4,6}$ $G_{4,7}$ $R_{4,8}$ $R_{4,9}$ $B_{4,10}$ $B_{4,11}$ $B_{4,12}$
$B_{4,13}$ $B_{5,1}$ $B_{5,2}$ $B_{5,3}$ $R_{5,4}$ $R_{5,5}$ $R_{5,6}$ $R_{5,7}$ $R_{5,8}$ $R_{5,9}$ $R_{5,10}$ $B_{5,11}$ $B_{5,12}$ $B_{5,13}$ $B_{6,1}$
$B_{6,2}$ $R_{6,3}$ $R_{6,4}$ $R_{6,5}$ $R_{6,6}$ $R_{6,7}$ $R_{6,8}$ $R_{6,9}$ $R_{6,10}$ $R_{6,11}$ $B_{6,12}$ $B_{6,13}$ $B_{7,1}$ $B_{7,2}$ $R_{7,3}$
$R_{7,4}$ $R_{7,5}$ $R_{7,6}$ $R_{7,7}$ $R_{7,8}$ $R_{7,9}$ $R_{7,10}$ $R_{7,11}$ $B_{7,12}$ $B_{7,13}$ $B_{8,1}$ $B_{8,2}$ $R_{8,3}$ $R_{8,4}$ $R_{8,5}$
$R_{8,6}$ $R_{8,7}$ $R_{8,8}$ $R_{8,9}$ $R_{8,10}$ $R_{8,11}$ $B_{8,12}$ $B_{8,13}$ $B_{9,1}$ $B_{9,2}$ $R_{9,3}$ $R_{9,4}$ $R_{9,5}$ $R_{9,6}$ $R_{9,7}$
$R_{9,8}$ $R_{9,9}$ $R_{9,10}$ $R_{9,11}$ $B_{9,12}$ $B_{9,13}$ $B_{10,1}$ $B_{10,2}$ $R_{10,3}$ $R_{10,4}$ $R_{10,5}$ $R_{10,6}$ $R_{10,7}$ $R_{10,8}$ $R_{10,9}$
$R_{10,10}$ $R_{10,11}$ $B_{10,12}$ $B_{10,13}$ $B_{11,1}$ $B_{11,2}$ $B_{11,3}$ $R_{11,4}$ $R_{11,5}$ $R_{11,6}$ $R_{11,7}$ $R_{11,8}$ $R_{11,9}$ $R_{11,10}$ $B_{11,11}$
$B_{11,12}$ $B_{11,13}$ $B_{12,1}$ $B_{12,2}$ $B_{12,3}$ $B_{12,4}$ $R_{12,5}$ $R_{12,6}$ $R_{12,7}$ $R_{12,8}$ $R_{12,9}$ $B_{12,10}$ $B_{12,11}$ $B_{12,12}$ $B_{12,13}$
$B_{13,1}$ $B_{13,2}$ $B_{13,3}$ $B_{13,4}$ $B_{13,5}$ $B_{13,6}$ $B_{13,7}$ $B_{13,8}$ $B_{13,9}$ $B_{13,10}$ $B_{13,11}$ $B_{13,12}$ $B_{13,13}$

Fig. 4: The input image using the tissue simulator

R is the following set of communication rules:

Type 1: These rules are used to update the counter z_i duplicating the number of copies in each step.

$$(1, z_i / z_{i+1}^2, 0) \text{ for } i=1...9;$$

Type 2: the image has two adjacent pixels with variant associated colors (border pixels) we used this rule.

$$(1, a_{ij}, b_{kl} / x_{ij}, b_{kl}, 0) \text{ for } a, b \in C, a<b, 1 \leq i, k \leq n\ 1 \leq j, 1 \leq m .$$

The pixel which is smallest associated color is marked and the system will bring an object(x) from the environment.

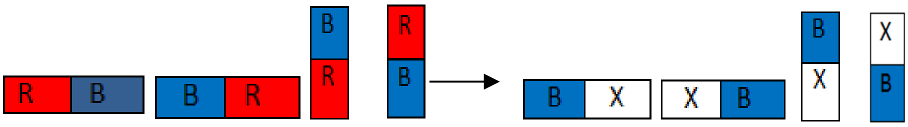
1) When the Red pixel color adjacent to pixel with Blue color (where R<B, red pixels will be marked)

$$(1, R_{ij} ; B_{i+1\,j} / X_{ij} ; B_{i+1j}, 0) \quad i:1..n \quad , j:1..m;$$

$$(1, R_{ij} ; B_{i-1\,j} / X_{ij} ; B_{i-1\,j}, 0) \; i: 1..n \quad , j: 1..m;$$

$$(1, R_{ij} ; B_{i\,j-1} / X_{ij} ; B_{i\,j-1}, 0) \quad i: 1..n, j: 1..m;$$

$$(1, R_{ij} , B_{i\,j+1} / X_{ij} , B_{i\,j+1}, 0) \; 1: i..n \quad , \quad j:1..m;$$



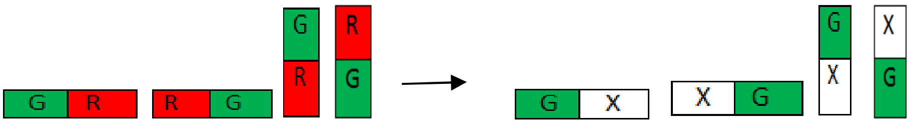
2) When the Red pixel color is adjacent to Green pixel color (where R<G, red pixels will be marked)

$$(1, R_{ij} ; G_{i+1\,j} / X_{ij} ; G_{i+1\,j}, 0) \quad i: 1..n \quad , \quad j:1..m;$$

$$(1, R_{ij} ; G_{i-1\,j} / X_{ij} ; G_{i-1\,j}, 0) \quad i: 1..n \quad , j:1..m;$$

$$(1, R_{ij} ; G_{i\,j-1} / X_{ij} ; G_{i\,j-1}, 0) \quad i: 1..n \quad , \quad j:1..m;$$

$$(1, R_{ij} ; G_{i\,j+1} / X_{ij} ; G_{i\,j+1}, 0) \quad i: 1..n \quad , \quad j:1..m;$$



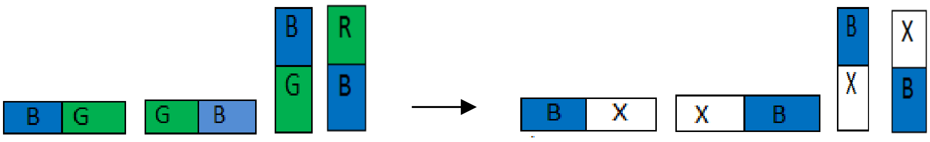
3) When the Green pixel color is adjacent to Blue pixel color (where G<B, green pixels will be marked).

$(1, G_{ij} ; B_{i+1\,j} \,/\, X_{ij} \;; B_{i+1},\ 0\,)$ i: 1..n , j:1..m;

$(1, G_{ij} ; B_{i-1\,j} \,/\, X_{ij} \;; B_{i-1\,j},\ 0\,)$ i: 1..n , j:1..m;

$(1, G_{ij} ; B_{i\,j-1} \,/\, X_{ij} \;; B_{i\,j-1},\ 0\,)$ i: 1..n , j:1..m;

$(1, G_{ij} ; B_{i\,j+1} \,/\, X_{ij} \;; B_{i\,j+1},\ 0\,)$ i: 1..n , j:1..m;



Type 3: the image has four adjacent pixels.

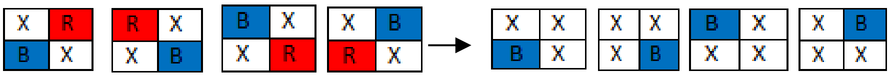
1) The Red color pixel adjacent with two marked pixels and with Blue pixel.

$(1, X_{ij} ; R_{ij+1} ; X_{i+1j+1} ; B_{i+1\,j} / X_{ij} ; X_{ij+1} ; X_{i+1j+1} ; B_{i+1\,j}, 0)$ i: 1..n-1, j :1..m-1;

$(1, X_{ij} ; R_{i-1j} ; X_{i-1j+1} ; B_{ij+1} \,/\, X_{ij} ; X_{i-1j} ; X_{i-1j+1} ; B_{ij+1}, 0)$ i:2.. n, j: 1..m-1;

$(1, X_{ij} ; R_{ij+1} ; X_{i-1j+1} ; B_{i-1j} / X_{ij} ; X_{ij+1} ; X_{i-1j+1} ; B_{i-1j}, 0)$ i: 2.. n, j: 1..m-1;

$(1, X_{ij} ; R_{i+1j} ; X_{i+1j+1} ; B_{ij+1} / X_{ij} ; X_{i+1j} ; X_{i+1j+1} ; B_{ij+1}, 0)$ i:1.. n-1, j: 1..m-1;



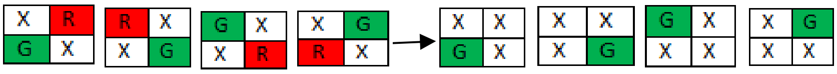
2) When the Red pixel adjacent with to marked pixel and Green pixel.

$(1 ,X_{ij} ;R_{ij+1};X_{i+1j+1} ;G_{i+1\,j}/X_{ij} ; X_{ij+1}; X_{i+1j+1}; G_{i+1\,j},0)$ i:1..n-1 , j :1..m-1;

$(1 , X_{ij}; R_{i-1j}; X_{i-1j+1}; G_{ij+1} / X_{ij}; X_{i-1j}; X_{i-1j+1}; G_{ij+1}, 0)$ i:2.. n, j :1..m-1;

$(1,X_{ij} ; R_{ij+1}; X_{i-1j+1} ; G_{i-1j}/ X_{ij}; X_{ij+1} ;X_{i-1j+1} ; G_{i-1j},0)$ i:2.. n, j :1..m-1;

$(1,X_{ij}; R_{i+1j}; X_{i+1j+1}; G_{ij+1} / X_{ij}; X_{i+1j}; X_{i+1j+1}; G_{ij+1} , 0)$ i:1.. n-1, j: 1..m-1



3) When the Green pixel adjacent with two marked pixels and Blue pixel.

$(1 ,X_{ij}; G_{ij+1}; X_{i+1j+1};B_{i+1\,j}/ X_{ij}; X_{ij+1}; X_{i+1j+1} ;B_{i+1\,j},0)$ i:1.. n-1, j: 1..m-1;

$(1, X_{ij}; G_{i-1j}; X_{i-1j+1}; B_{ij+1} / X_{ij}; X_{i-1j}; X_{i-1j+1}; B_{ij+1}, 0)$ i:2.. n, j: 1..m-1;

$(1, X_{ij} ; G_{ij+1}; X_{i-1j+1} ; B_{i-1j} / X_{ij}; X_{ij+1} ;X_{i-1j+1} ; B_{i-1j} , 0)$ i:2.. n, j:1..m-1;

$(1 ,X_{ij}; G_{i+1j};X_{i+1j+1}; B_{ij+1} / X_{ij}; X_{i+1j}; X_{i+1j+1}; B_{ij+1}, 0)$ i:1..n-1, j: 1..m-1;

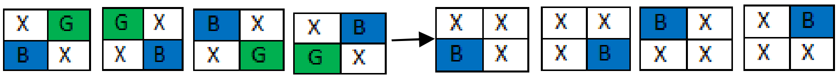


Fig. 5: shows the output using tissue simulator

Type 4: This rule sends the marked pixels to the cell2.

$$(1,Z_9X_{ij} \; / \quad , 2) \quad \text{i: 1... n, j: 1...m ;}$$

The system begins to work when the input objects a_{ij} encode the colored pixels from a 2D digital image, as shown in Fig. 3 and the counter Z_i appears in the input cell. The rule of type 1 is used to update the counter Z_i in each step (we need initially $\lceil r_1^{1/2^7} \rceil$ copies of object Z_1 with r1 = max (n, m), to generate enough copies of z_1 for the system output, It is given by the objects that appear in cell 2 when it stops). The rules of type 2 are used in a a parallel and non deterministic manner manner to identify the border pixels and bring the edge pixels from the environment. The rules of type 3 need four steps to mark all the border pixels.

Similar to edge based segmentation, in another four steps, we can bring from the environment the edge pixels adjacent to two border pixels. The system can first apply the type 2 and type 3 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). The system uses only 8 steps to do the segmentation and one step uses a counter, Z_i, to send the marked objects to cell 2. The system is ready to send the objects codifying the complete image to cell 2 in the last step of computation. We need only 9 steps to obtain a region-based segmentation of an $(n \times m)$ image (see the Fig. 5). The last configuration of the system which contains the resulting marked pixels in the output cell 2 as can bee seen in Figure 5. The illustration of the configuration steps is shown in Figure 6. Notably, only four steps have been used to mark all the pixels and the remaining steps have been used to increase the counter. Finally, in step 9, the communication rules are applied to send the resulting marked pixels into the output cell 2.

The size of the input data is $O(n\,.m\,)$ and the number of colors of the image $(|C|)$ is h, so the complexity of the problem depends on the is a number of rules which is $O(n\,.m.\,h^2\,)$. The upper bound for the length of the rule is 8. The computation steps are constant at 9 steps.

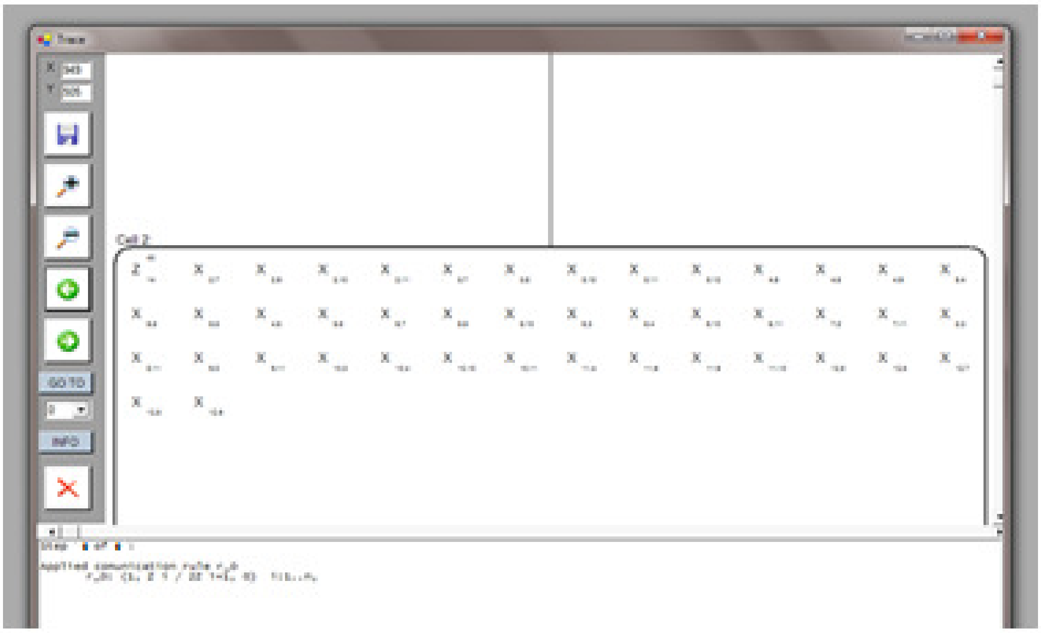
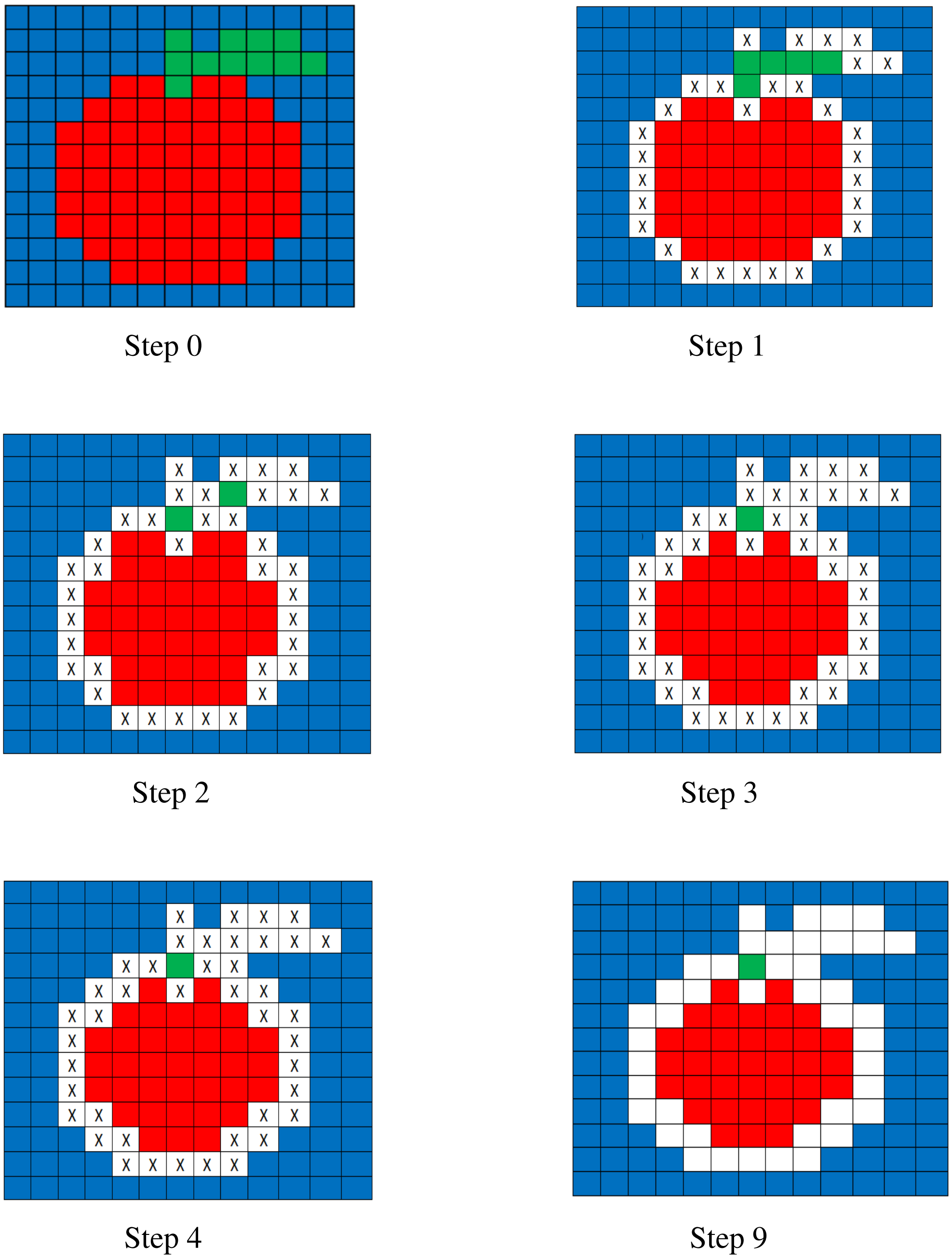


Fig. 5: The output of tissue simulator when (R<G<B)



Step 0

Step 1

Step 2

Step 3

Step 4

Step 9

Fig. 6: An example of execution (Step 0 to 9)

In the first experiment, we have used the color relations where the red color is smaller than Green and both of them are smaller than the blue (R<G<B). The result of this experiment is shown in Figure 5. In the second experiment, the color relations have been used as in [14] where (R>G>B) as shown in Figure 7.

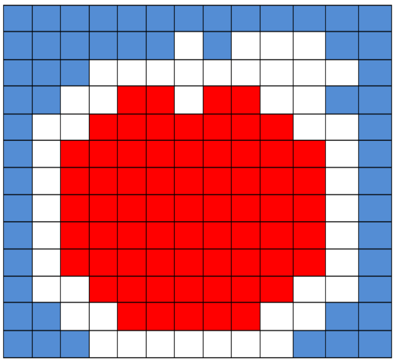


Fig. 7: The output of tissue simulator when (R>G>B).

6 Conclusion

Region based segmentation method under the framework of P systems is proposed in this paper, where a simple artificial 2D image (apple) is used to illustrate the steps of the algorithm.The tissue-like P systems is a variant of P systems which takes advantage of communication rules to achieve adaptive region based for image segmentation, Therefore, the proposed image segmentation method based on tissue-like P systems have the advantage of fast segmentation and the segmentation is done in a constant number of steps (9 steps) regardless of the size of the image. The tissue simulator is used to check the validity of the system and to show the effect of color on the segmentation results. The future work is to use the real image by using P Lingua or CUDA^{TM}.

ACKNOWLEDGEMENTS

Authors is grateful to the Ministry of Higher Education & Scientific Research / University of Al –Mustansiriyah in Iraq for providing sponsorship to continue her PhD. This work is partially supported by The Flagship Research Grant Scheme (Q.J130000.2428.02G50 and Q.J130000.2428.03G17). The authors would like to thanks Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the support in R & D.

References

[1] Neumann, J.v. and A.W. Burks, *Theory of self-reproducing automata.* 1966.

[2] John, H., *Holland, Adaptation in natural and artificial systems*, 1992, MIT Press, Cambridge, MA.

[3] McCulloch, W.S. and W. Pitts. *A logical calculus of the ideas immanent in nervous activity.* in *Neurocomputing: foundations of research.* 1988. MIT Press.

[4] Abelson, H., et al., *Amorphous computing*. Communications of the ACM, 2000. 43(5): p. 74-82.

[5] Adleman, L.M., *Molecular computation of solutions to combinatorial problems.* Science-AAAS-Weekly Paper Edition, 1994. 266(5187): p. 1021-1023.

[6] de Castro, L.N., *Fundamentals of natural computing: an overview.* Physics of Life Reviews, 2007. 4(1): p. 1-36.

[7] Ciobanu, G., G. Păun, and M.J. Pérez-Jiménez, *Applications of membrane computing*. Vol. 17. 2006: Springer.

[8] Xue, J. and X. Liu. *Applied membrane computation in creative design*. in *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*. 2012. IEEE.

[9] Shapiro, L. and G. Stockman, *Computer Vision. chap. 12*, 2001, New Jersey: Prentice Hall.

[10]H.A., D. Díaz-Pernil, and P.R. Jurado, *Segmentation in 2D and 3D image using tissue-like P system*, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. 2009, Springer. p. 169-176.

[11]Christinal, H.A., et al., *Thresholding of 2D images with cell-like P systems.* Romanian Journal of Information Science and Technology (ROMJIST), 2010. 13(2): p. 131-140.

[12]Carnero, J., et al. *Image segmentation inspired by cellular models using hardware programming*. in *3rd International Workshop on Computational Topology in Image Context*. 2010.

[13]Reina-Molina, R., J. Carnero, and D. Diaz-Pernil, *Image segmentation using tissue-like P systems with multiple auxiliary cells.* Image-A, 2010. 1(3): p. 143-150.

[14]Christinal, H.A., D. Díaz-Pernil, and P. Real, *Region-based segmentation of 2D and 3D images with tissue-like P systems.* Pattern Recognition Letters, 2011. 32(16): p. 2206-2212.

[15]Peña-Cantillana, F., et al. *A parallel implementation of the thresholding problem by using tissue-like P systems*. in *Computer Analysis of Images and Patterns*. 2011. Springer.

[16]Carnero, J., D. Dıaz-Pernil, and M.A. Gutiérrez-Naranjo, *Designing tissue-like P systems for image segmentation on parallel architectures.* Ninth Brainstorming Week on Membrane Computing: p. 43-62.

[17]Sheeba, F., et al. *Segmentation of peripheral blood smear images using tissue-like P systems*. in *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on*. 2011. IEEE.

[18]Christinal, H.A., et al., *Color Segmentation of 2D Images with Thresholding*, in *Eco-friendly Computing and Communication Systems*. 2012, Springer. p. 162-169.

[19]Yang, Y., et al., *A Region-based Image Segmentation Method Under P Systems⋆.*

[20]Peng, H., et al., *Image Thresholding with Cell-like P Systems.* Proceedings of the Tenth Brainstorming Week on Membrane Computing, 2012. 2: p. 03.

[21]Díaz-Pernil, D., et al., *Segmenting images with gradient-based edge detection using Membrane Computing.* Pattern Recognition Letters, 2013. 34(8): p. 846-855.

[22]PENG, H., et al., *A Region-based Color Image Segmentation Method Based on P Systems.* SCIENCE AND TECHNOLOGY, 2014. 17(1): p. 63-75.

[23]Păun, G., *Membrane computing: an introduction*. 2002: Springer.

[24]Păun, G. and M.J. Pérez-Jiménez, *Membrane computing: Brief introduction, recent results and applications.* BioSystems, 2006. 85(1): p. 11-22.

[25]Ibarra, O.H. and G. Paun, *Membrane computing: a general view.* Ann Eur Acad Sci. EAS Publishing House, Liege, 2006: p. 83-101.

[26]Păun, G., *A quick introduction to membrane computing.* The Journal of Logic and Algebraic Programming, 2010. 79(6): p. 291-294.

[27]Martın-Vide, C., et al., *Tissue P systems.* Theoretical Computer Science, 2003. 296(2): p. 295-326.

[28]forja.rediris.es/frs/download.php/374/index_ing.html.