Tissue-like P System for Segmentation of 2D Hexagonal Images

Rafaa I. Yahya[1,2,3], Siti Mariyam Shamsuddin[1,2], Shafaatunnur Hasan[1,2] and Salah I. Yahya[1,4]

[1]UTM Big Data Center, Ibnu Sina Institute for Scientific and Industrial Research
Universiti Teknologi Malaysia, UTM Skudai, 81310 Johor, Malaysia

[2]Faculty of Computing, Universiti Teknologi Malaysia
UTM Skudai, 81310 Johor, Malaysia

[3]Department of Computer, Collage of science, University of Al-Mustansiriyah
Baghdad – F.R. Iraq

[4]Department of Software Engineering, Faculty of Engineering, Koya University
University Park, Danielle Mitterrand Boulevard, Koya KOY45, Kurdistan Region - F.R. Iraq

Abstract—**Membrane computing, which is a new computational model inspired by the structure and functioning of biological cells and by the way the cells are organized in tissues. MC has been adopted in many real world applications including image segmentation. In contrast to the traditional square grid for representing and sampling digital images, hexagonal grid is an alternative efficient mechanism which can better represents and visualizes the curved objects. In this paper, a tissue-like P system with region-based and edge-based segmentations is used to segment two dimensional hexagonal images, wherein P-Lingua programming language is used to implement and validate the proposed system. The achieved experimental results clearly demonstrated the effectiveness of using hexagonal connectivity to segment two dimensional images in a less number of rules and computational steps. Moreover, the results reveal that this approach has the potential of segmenting large images in little number of steps.**

Index Terms—**Membrane computing, edge-based image segmentation, P-Lingua, region-based image segmentation, Tissue-like P system.**

I. Introduction

Membrane Computing (MC) is a recent and fast growing branch of natural computing which is inspired by the structure and functioning of the living cell aiming to build computational models (Dang, et al., 2005). It takes its inspiration from the way that the living biological cells employ to process chemical compounds in their compartmental structures.

Hence, in P systems (the computational models of MC), regions which are described by a membrane structure consist of objects evolving which based on specific evolution rules. Those objects can be represented by symbols or alternatively by strings of symbols, wherein multisets of objects are placed inside regions delimited by membranes.

Venn diagram and a tree-like structure are two different organizations of membranes where one membrane may contain other membranes inside it. The transaction from one configuration to another configuration can be accomplished by using the evolution rules in a non-deterministic and maximally parallel way. A series of transitions reveals how the P system is evolving. Different procedures of how to control the communication of objects from a region to another and how to apply the rules as well as membrane desolation, division or creation have been investigated widely in the literature (Păun, 2002).

As stated previously, MC gets the ideas of the cell aiming to learn mechanisms that could be useful from the perspective of computer science, but the research also takes into consideration how the cells are organized in tissues, as well as, the arrangement of neurons in the human brain. Based on this formalization, there are three basic variants of P systems, namely; (i) cell-like P system, (ii) tissue-like P system, and (iii) neural-like P system (Ibarra and Paun, 2006).

Different types of computing models known as P systems were proposed, motivated by biological related facts or inspired from mathematical or computer science perspectives. A vast number of applications were proposed in the past few decades in different domains – biology, bio-medicine, linguistics, computer graphics, economics, approximate optimization, cryptography and image processing (Paun, 2007).

In image processing, segmentation is the process of partitioning a digital image into separate regions, with the aim

of simplifying or modifying its representation to be more meaningful and easier to understand. Basically, image segmentation is used to find objects and boundaries (lines, curves and etc.) in digital images. Moreover, image segmentation is the method of giving labels to every pixel in an image,; in a way that pixels having same label have similar visual features (Shapiro and Stockman, 2001). The main objective of any image segmentation method is to perform segmentations in a sense that there is a high correlation between the entities of the real world objects and the regions of the segmentation (Kohler, 1981).

This paper is organized as follows: Section II provides a comprehensive summary of related works pertaining to image segmentation using P systems. A basic definition of tissue-like P systems is presented in Section III. Section IV describes the use of tissue-like P system for segmenting hexagonal images, region-based and edge-based segmentation. Section V presents the methodology of segmentation with membrane computing using P-Lingua programming language. The experimental results and comparison are presented in Section VI. Finally, Section VII concludes the paper and suggests some directions for the future work.

II. Related Works

MC possesses several interesting features including the encapsulation of data, a trivial way of representing information as well as the maximal parallelism, all of which are most proper when handling digital images.

Most recently, a number of research trends have been using MC approaches for solving problems pertaining to digital image processing. In a work relating to segmentation problems, Christinal, et al. (2009) have designed a family of tissue-like P system using communication and evolution rules to get edge-based segmentation of 2D image using 4-adjacency and 3D image using 26-adjacency. Their results have been obtained in a constant number of steps where the system has been implemented by tissue simulator software to check its validity and effectiveness. Christinal, et al. (2010) proposed a membrane computing approach to solve the threshold problem by using cell-like P system rules where the massive parallelism feature of MC has helped the solution to be reached in linear time which depends on the size of the input image.

Carnero, et al. (2010) have proposed a new hardware tool including a Field-Programmable Gate Array unit (FPGA) to perform segmentation of digital images for solving edge-based detection and noise removal problem. Their system uses MC method as well as a hardware programming (VHDL) language to propose an ad–hoc processor. In the same context, Reina-Molina, et al. (2010) have developed a new version of tissue-like P system by replacing the concept of one cell with the use of multiple auxiliary cells to deal with segmentation problem to get all the potential parallelization. In another work, Díaz-Pernil, et al. (2010) have proposed a new software tool for segmenting 2D digital images on the basis of tissue-like P system, wherein the object oriented C++ programming language has been used in the implementation part. However, they did not provide a clear explanation regarding the technical aspects of developing the proposed tool.

Christinal, et al. (2011) have proposed a tissue-like P system (using communication rules) to perform a region-based segmentation in nine computational steps. In their work, 4-adjacency pixels neighborhood has been used to segment 2D digital images, whereas 26-adjacency has been employed for 3D digital images. A bio-inspired MC software has been proposed by Peña-Cantillana, et al. (2011) to solve the threshold problem and it has been implemented by an innovative device architecture called Compute Unified Device Architecture (CUDA™). Carnero, et al. (2011) have presented the use of the FPGA to implement tissue-like P system rules for solving segmentation problems. Sheeba, et al. (2011) have proposed tissue-like P system to segment medical image, nuclei of the white blood cells (WBCs) of the peripheral blood smear images in morphology segmentation technique. Their algorithm has been implemented using MATLAB software.

Similarly, Zhang and Peng (2012) have proposed novel infrared object segmentation based on thresholding method using cell-like P system to get the best set of parameters quickly. Christinal, et al. (2012) have proposed a variant of P system (tissue-like P system) using the rules to perform a parallel color segmentation of 2d images based on a threshold method. Peng, et al. (2012) have developed a novel threshold segmentation method based on cell-like P system to improve the performance of the threshold segmentation. Similarly, Yang, et al. (2013) have proposed an image segmentation approach using tissue-like P system to develop conventional region-based image segmentation.

In the work of Díaz-Pernil, et al. (2013), a CUDA™ has been presented to implement tissue-like P system rules for segmenting images by the use of gradient-based edge detection to enhance the traditional methods of segmenting digital images. Peng, et al. (2014) have proposed a novel segmentation by improving the traditional region-based colour image segmentation method using tissue-like P systems. Isawasan, et al. (2014) used tissue-like P system rules to perform region-based segmentation of 2D hexagonal images where the segmentation steps has been done in 7 steps, but they did not illustrate how they used P-Lingua to perform the segmentation. Peng, et al. (2015) have proposed a new method using cell-like P system to solve the optimal multi-level thresholding problem. Yahya, et al. (2015) have presented a traditional region-based segmentation with tissue-like P system rules. In their proposed work, a simple artificial image has been used to give a more detailed illustration of the basic idea of how P system works. Furthermore, various colour relations have been investigated to illustrate the effect of colors on the segmentation results.

As can be depicted from the literature review presented above, only one work has implemented the region-based segmentation in 6-adjacency of hexagonal image (Isawasan, et al., 2014). However, Isawasan, et al., did not clearly illustrate how their approach has been implemented in P-Lingua programming language. Hence, in this paper tissue-like P

system has been used to segment two dimensional hexagonal artificial images by employing both of edge-based segmentation and region-based segmentation, simultaneously, and illustrate the implementation of hexagonal segmentation in P-Lingua with more details.

Hexagonal grid is an efficient pixel tessellation scheme different from the traditional square grid for modelling and representing digital images. In contrast to square images, the pixels of the hexagonal images are much closer to each other which make the edges more clear and sharp. The main reasons to use hexagonal image processing are:

1) To improve the performance of the image recognition algorithms.
2) To minimize the computational complexity of processing the image and make it much faster.
3) By using this approach, image features can be detected more precisely (He and Jia, 2005).

In this paper, we comprehensively illustrate the idea of 2D image segmentation using hexagonal connectivity based on tissue-like P system. P-Lingua programming language has been used to perform segmentation and to check the validity of the proposed approach. For illustration purpose, a large artificial image with three colors is used to test the effectiveness of the system, as will be shown in Section V.

III. Definition of Tissue-like P Systems

Tissue-like P system, a variant of membrane computing models, was proposed by Martine vide (2003). The structure of tissue-like P systems is arranged as a graph having two biological inspirations, namely intercellular communications that represent all communication channels available between the cells as well as the communication between cells and environment. The second motivation pertaining to this model is the cooperation between neurons. Tissue-like P system has a distinguishing feature from the computational point of view which is cells do not have electrical charges (Păun, 2010).

Formally, a family of a tissue-like P system with input of degree q¸ 1 is a tuple (Pan and Pérez-Jiménez, 2010):

$$\Pi(n,m) = (\Gamma, \Sigma, w1 \ldots, wq, R, i\Pi, o\Pi)$$

Where the components are defined as follows:
1) $\Gamma = \Sigma \cup \mathcal{E}$ is a finite alphabet where each symbol working is called object that is placed in a cell or surrounding the cell (in the environment).
2) $\Sigma \subseteq \Gamma$ is the input alphabet (the objects inside the cell)
3) $\mathcal{E} \subseteq \Gamma$ is an infinite set of objects that are available in the environment with arbitrary large amount of copies.
4) The multi-sets $w_1, \ldots, w_q$ represent the objects placed in the cells at the beginning of the computation.
5) R is the finite set of communication rules in the following form
   $(i, u / v, j) \text{ for } I, j \in \{0,1,2,\ldots,q\}, i \neq j, u, v \in \Gamma$
6) $i\Pi \in \{1 \ldots q\}$ refers to the input cell.
7) $o\Pi \in \{0,1 \ldots q\}$ refers to the output of the cell.

In the typical framework of MC, each cell is viewed as a computing unit working in a maximally parallel and non-deterministic manner. The configuration is an instantaneous description of the P system at a particular time, where a sequence of computation steps can be applied in a parallel way to get a new configuration. A computation is said to be successful if it halts, reaching a specific configuration where no more rules can be further applied to the current objects. With a halting computation, the associated output can be codified by the content of the output membrane.

IV. Segmentation-Based on Tissue-like P System of 2D Hexagonal Image

As can be depicted from the literature review presented previously, only one work has implemented the segmentation of hexagonal (6-adjacent) image (Isawasan, et al., 2014). The majority of the work reviewed was implemented using 4-adjacent. Isawasan, et al. (2014) have proposed the 6 adjacent in P-Lingua programming platform to obtain region-based segmentation, but they have not provide a comprehensive illustration of how to use P-Lingua or refer to the computational time of segmentation. Hence, this paper adopts the work of (Isawasan, et al., 2014) and obtains edge-based segmentation with region-based segmentation and presents in more details the implementation of P-Lingua as long as showing the time of execution by using a large artificial apple image.

We define a family of tissue-like P system to perform region-based segmentation for 2HD image as follows:

$$\Pi(n,m) = (\Gamma, \Sigma, w1, w2, R, i\Pi, o\Pi)$$

Where the components are defined as follows:

1) Finite all alphabets that working in the system is $\Gamma = \Sigma \cup \mathcal{E}$
2) The input is $\Sigma \subseteq \Gamma$ $\Sigma = \{aij : a \in C \wedge 1 \leq i \leq n \wedge 1 \leq j\}$
3) The object of environment is $\mathcal{E} \subseteq \Gamma$, where
   $\mathcal{E} = \{\bar{a}_{ij} : a \in C \wedge 1 \leq i \leq n \wedge 1 \leq j \leq m\} \cup \{Z_i : 1 \leq i \leq 7\}$
4) The mulisets (objects placed in the cell1 and cell2) $w1, \ldots, w2$ is
   $w1 = z_1^{\left\lceil r_1^{1/2^5} \right\rceil}, 2 = \emptyset$, where $r1 = max(n,m)$
5) R represents the set of communication rules:

Type 1 rules:
$(1, Z_i/Z_{i+1}^2)$ for $i = 1, \ldots, 7$.
These communication rules make the counter Z_i be duplicated in each step.

Type 2 rules:
$(1, a_{ij} b_{kl}/\bar{a}_{ij} b_{kl}, 0)$ for $a, b \in C$, $a < b$, $1 \leq i, k \leq n$ and $1 \leq j, l \leq m$.

The type 2 of the rules is basically applied when two adjacent pixels having different colors, those pixels are called border pixels. For instance, the two adjacent pixels with different colours are shown in Fig. 1. The pixel which has small associated color will be marked, for example, if the system learned that Red colour is greater than Green then the green pixel (G) will be marked and brought from the environment the marked green pixel (Gx) instead of Green pixel for any of the possible directions as shown in Fig. 2.
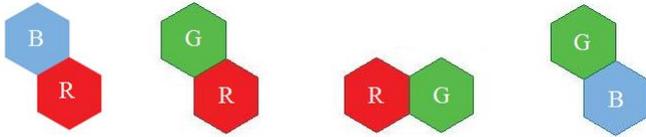
Fig. 1. Two adjacent pixels with different colors.

Fig. 2. Marked adjacent pixels using type 2 rules.

Type 3 rules:
$(1, Z_7 \bar{a}_{ij}/\lambda, 2)$ for $1 \leq i \leq n$, $1 \leq j \leq m$.
These communication rules will take the marked pixel and send them to cell 2, where the marked pixel will be disappeared
1) The input of the cell is $i\Pi = 1$.
2) The output of the cell
   $o\Pi = 1$ if we need to obtain the region based segmentation.
   $o\Pi = 2$ if we need to obtain the edge- based segmentation.

V. P-Lingua for Segmentation of 2HD Images

P-Lingua is considered as the standard and official programming language of MC (Díaz-Pernil, et al., 2009). This language is created to include all P systems variants (cell-like P systems (García-Quismondo, et al., 2009), tissue-like P system (Perez–Hurtado, et al., 2014) and neural-like P system (Macías–Ramos, et al., 2012)). The principal component in P-Lingua is a PLinguaCore Java library. PLinguaCore Java library accepts as input all text files, XML or P-Lingua format (.PLI), which describe the P system model. Furthermore, P-Lingua includes parsers to handle input files and built-in simulators to produce P system computations. It can export several output file formats to represent P systems. It is an open source product since expert developers with good knowledge

of Java can contribute and add new components to enlarge the library (García-Quismondo, et al., 2010).

In this paper, the tissue-like P system is used along with the PLinguacore4 java library to implement the segmentation of the hexagonal artificial image as shown below in Fig. 3 which describes the methodology of segmentation implemented in P-Lingua platform.
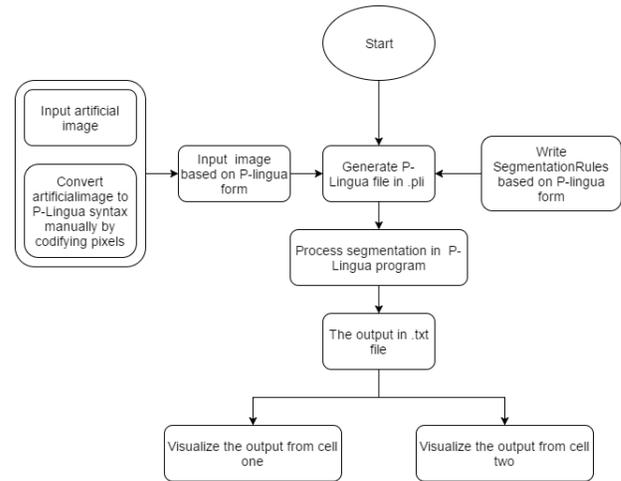

Fig. 3. Methodology of 2DH image segmentation.

A. Generating P-Lingua File in pli Format

The pixels of the artificial image along with the rules of segmentation of hexagonal 2D image (2DH) is written and placed together in one text file (.txt) and saved as P-Lingua file.Pli. Thus, the basic steps of generating P-Lingua file are explained in detail as shown below.

Input image based on P–Lingua form
The artificial apple image will be analyzed and codified manually as an input in a text file according to the standard syntax of P-Lingua.

1) Input artificial image
The image has hexagonal pixels that are manually codified and inserted into the system as an input image according to the standard syntax of P-Lingua. In tissue-like P system, the image is considered as a cell and the pixels of the image are represented as objects in the cell. Specifically, every pixel (object) has the color and coordinates (x, y). In this paper, the artificial image used is a simple apple image with three colors (Red, Blue and Green) which has been drawn manually with size (13×13) as shown in Fig. 4.

2) Convert Image to P-Lingua Syntax
The pixels should have color and coordinates as mentioned earlier to match the syntax of P-Lingua. Basically, the color and the coordinates of each pixel will be read and converted to P-Lingua syntax. The standard syntax is to put the color of the

pixel followed by curly bracket that contain the coordinate of the associated pixel which in turn will be written in text files. For example, $R\{x, y\}$, where x and y are the coordinates of the red pixel in the image as shown below in Fig. 5. All pixels are converted to the syntax of P-Lingua by following this procedure.
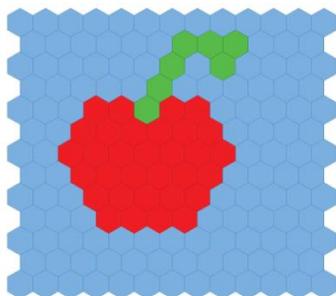

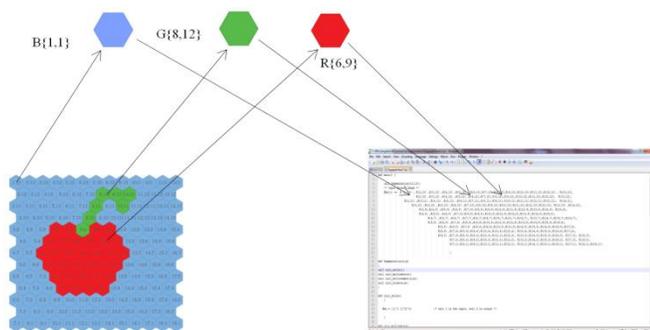Fig. 4. Simple artificial apple image (size of 13×13).

Fig. 5. Artificial apple image of (size 13×13) in P-Lingua syntax.

Writing the rules of segmentation based on P–Lingua form

In this paper, the segmentation rules is based on the rules of membrane computing for 2HD image segmentation method employed in (Isawasan, et al., 2014) . The rules are written in the P-Lingua format with input image and saved in a text file, but with (.Pli) extension. This file is now ready to be executed as shown in Appendix I.

Technically speaking, a 2D digital image can be represented by a matrix where each pixel in the image is an element of the matrix. Basically, in this paper, the 6-adjacent connectivity of neighborhoods surrounding a pixel is considered. The hexagonal pixels are:

$$\{(x, y + 1), (x, y - 1), (x + 1, y), (x - 1, y), (x - 1, y + 1), (x + 1, y - 1)\}$$

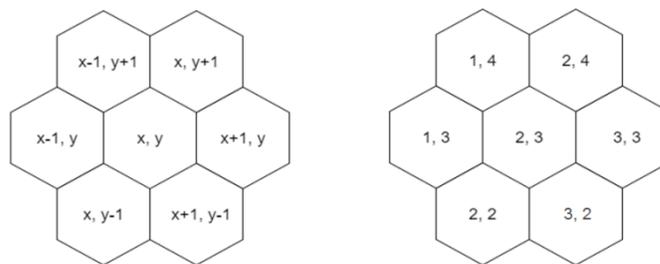which contains 6 directions from the central pixel as demonstrated in Fig. 6.

Fig. 6. Illustration of the 6-adjacent (hexagonal) neighborhoods.

When the image contains two different colors, namely Red and Blue, then if the six- adjacency (hexagonal) is considered, the segmentation rules will be as the following:

$$[B\{i,j\}, R\{i, j + 1\}]'1 < --> [Bx\{i,j\}, R\{i, j + 1\}]'0;$$
$$[B\{i,j\}, R\{i, j - 1\}]'1 < --> [Bx\{i,j\}, R\{i, j - 1\}]'0;$$
$$[B\{i,j\}, R\{i + 1, j\}]'1 < --> [Bx\{i,j\}, R\{i + 1, j\}]'0;$$
$$[B\{i,j\}, R\{i - 1, j\}]'1 < --> [Bx\{i,j\}, R\{i - 1, j\}]'0;$$
$$[B\{i,j\}, R\{i - 1, j + 1\}]'1 < --$$
$$> [Bx\{i,j\}, R\{i - 1, j + 1\}]'0;$$
$$[B\{i,j\}, R\{i + 1, j - 1\}]'1 < --$$
$$> [Bx\{i,j\}, R\{i + 1, j - 1\}]'0;$$

B. Process segmentation in P-Lingua program

The PLinguaCore4 java library will simulate the P-Lingua rules to implement the segmentation of the artificial image. All the steps above are to prepare the P-Lingua file (image and rules of segmentation) for simulation by PLinguaCore4 Java library.

C. Output of segmentation in file.txt

The final result of the segmentation process is available in the file that is generated after segmentation (.txt). This file contains the details of every step of segmentation; it contains information like the configuration, input of the cell, the output of cell, the environment, the time of every step, memory used and final time of execution.

Once the image has been segmented using MC, bearing in mind that our model contains two different cells, the output can be read from any of these cells as needed. In the first case, if the output is being read from cell two, then edge-based segmentation will be obtained and visualized. It is worth mentioning that cell two contains the border and edge pixels, that is why edge-based segmentation can be obtained from this particular cell as shown in Fig. 7. Otherwise, in the second case if the output is being read from cell one, then, region-based segmentation will be obtained as shown in Fig. 8. More specifically, there are two types of segmentation strategies which can be obtained regarding the philosophy of reading the output from the chosen cell.
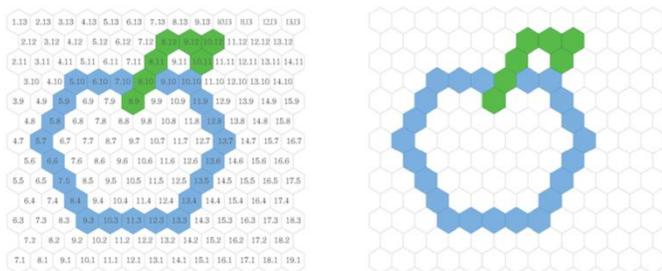
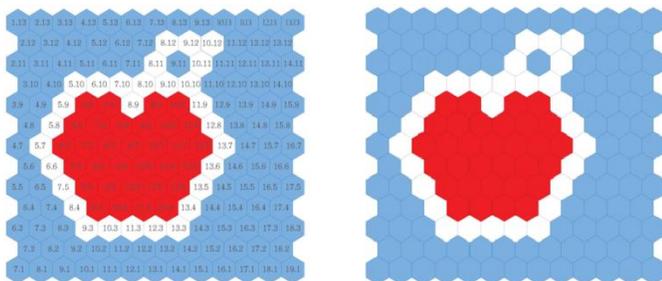Fig. 7. Output of segmentation from cell 2 (edge-based).

Fig. 8. Output of segmentation from cell 1 (region-based).

VI. Experimental Results

According to the literature review, the majority of the works on image segmentation have used 4-adjacency to perform segmentation. Yahya, et al., (2015) presented the segmentation of an artificial apple image with size of 13×13 and three colours (Red, Blue and green) using 4-adjacency connectivity. The time of segmentation was not considered because the segmentation was achieved by implementing a tissue simulator. The same artificial apple image of (Yahya, et al., 2015) was used in this paper, and the total time needed for segmentation using 4 and 6-adjacency connectivity was achieved for the sake of comparison. The results are presented in Table I.

TABLE I
COMPARISON BETWEEN DIFFERENT TYPES OF CONNECTIVITY

Type of adjacency	Input image	Size of image	Time	Memory usage	No. of Configuration steps
4-adjacency	Apple	13×13	2.136 s	310272 kb	9
6-adjacency	Apple	13×13	1.327 s	178688 kb	7

It is obvious that using 6-adjacency is more efficient and the processing time is much faster than the 4- adjacency. Moreover, the number of computational steps has been reduced from 9 to 7 steps. The reason of the better performance of 6-adjacency connectivity, as compared with the 4-adjacency, is attributed to the fact that the computational

requirement for processing a hexagonally connected image is less than that for square sampled images due to the compact and circular nature of the hexagonal connectivity. There are many conclusions can be drawn from the results in Table I. However, the main conclusion is; the number of edge pixels using both 4 and 6-adjacency were roughly the same but the edge detected of hexagonal-image appears to be qualitatively better. This stems from the consistent connectivity of the pixels in hexagonal-images which aids image segmentation. The hardware platform that used to achieve the results of this paper is a PC with an Intel® Core™ i5-M430 processor running at 2.27 GHz and 4 GB of memory.

VII. Conclusion

In this paper, tissue-like P system, a variant of MC, is used to segment two dimensional hexagonal artificial images by employing edge-based segmentation and region-based segmentation. P-Lingua programming language is used to implement and validate the proposed P system. The experimental results have shown that the use of the hexagonal connectivity is more efficient than the four connectivities where the number of rules and computational steps have been reduced from 9 steps to 7 steps. Finally, we can conclude that the use of hexagonal-images rather than the classical square-images for image segmentation has several advantages as demonstrated in this paper. These are due mainly to the connectivity of the individual hexagonal pixels generating more consistent contours. The proposed framework of segmentation offers an added advantage in using hexagonal-images for image segmentation, namely, great computational savings. This makes a strong case for hexagonal-based image segmentation and seems to reinforce the point that hexagonal-image processing has the potential to be a viable alternative to square-image processing in the future. However, the major limitation of the proposed approach is that the images are codified manually in the system. The reason for that is the hard processing of hexagonal images compared with the traditional square image. In our future work, we aim to find an automatic solution to enter the hexagonal-image in the system.

References

Carnero, J., Diaz-Pernil, D. and Gutierrez-Naranjo, M.A., 2011. Designing tissue-like P systems for image segmentation on parallel architectures. *Ninth Brainstorming Week on Membrane Computing*, pp.43-62.

Carnero, J., Diaz-Pernil, D., Molina-Abril, H. and Real, P., 2010. Image segmentation inspired by cellular models using hardware programming. *3rd International Workshop on Computational Topology in Image Context, 2010.* pp.143-150.

Christinal, H.A., Diaz-Pernil, D., Gutierrez-Naranjo, M.A. and Perez-Jimenez, M.J., 2010. Thresholding of 2D images with cell-like P systems. *Romanian Journal of Information Science and Technology (ROMJIST),* 13, pp.131-140.

Christinal, H.A., Diaz-Pernil, D. and Jurado, P.R., 2009. Segmentation in 2D and 3D image using tissue-like P system. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications,* 5856, pp.169-176.

Christinal, H.A., Diaz-Pernil, D. and Jurado, P.R. and Selvan, S.E., 2012. Color Segmentation of 2D Images with Thresholding. *Eco-friendly Computing and Communication Systems,* 305, pp.162-169.

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

Dang, Z., Ibarra, O.H., LI, C. and Xie, G., 2005. On model-checking of P systems. *Unconventional Computation,* 8340, pp.151-172.

Diaz-Pernil, D., Berciano, A., Pena-Cantillana, F. and Gutierrez-Naranjo, M.A., 2013. Segmenting images with gradient-based edge detection using Membrane Computing. *Pattern Recognition Letters,* 34(8)**,** pp.846-855.

Diaz-Pernil, D., Molina-Abril, H., Real, P. and Gutierrez-Naranjo, M.A, 2010. bio-inspired software for segmenting digital images.   Bio-Inspired Computing: Theories and Applications (BIC-TA). In: *IEEE Fifth International Conference, 2010*, IEEE, pp.1377-1381.

Diaz-Pernil, D., Perez-Hurtado, I., Perez-Jimenez, M.J. and Riscos-Nunez, A., 2009. A P-lingua programming environment for membrane computing. *Membrane* Computing, 5391, pp.187-203.

Garcia-Quismondo, M., Gutierrez-Escudero, R., Martinez-Del-Amor, M.A., Orejuela-Pinedo, E. and Perez-Hurtado, I., 2009. P-Lingua 2.0: A software framework for cell-like P systems. *Int J Comput Commun Control,* 4(3), pp.234-43.

Garcia-Quismondo, M., Gutierrez-Escudero, R., Perez-Hurtado, I., Perez-Jimenez, M.J. and Riscos-Nunez, A. 2010. An overview of P-Lingua 2.0. *Membrane Computing,* 5957, pp.264-288.

HE, X. and Jia, W., 2005. Hexagonal structure for intelligent vision. Information and Communication Technologies. In:, *First International* Conference on ICICT 2005, IEEE, pp.52-64.

Ibarra, O.H. and Paun, G., 2006. Membrane computing: A general view. *Ann Eur Acad Sci. EAS Publishing House, Liege*, pp.83-101.

Isawasan, P., Venkat, I., Subramanian, K., Khader, A., Osman, O. and Christinal, H., 2014. Region-based segmentation of Hexagonal digital images using membrane computing. In: *Asian Conference on Membrane Computing (ACMC), 2014*. IEEE, pp.1-4.

Kohler, R. 1981. A segmentation system based on thresholding. *Computer Graphics and Image Processing,* 1594)**,** pp.319-338.

Macias-Ramos, L.F., Perez–Hurtado, I, Garcia–Quismondo, M., Valencia–Cabrera, L., Perez-Jimenez, M. J. and Riscos–Nunez, A., 2012. A P–Lingua Based Simulator for Spiking Neural P Systems. *Membrane Computing,* 7184, pp.257-281.

Martin-Vide, C., Pazos, J., Paun, G. and Rodriguez-Paton, A., 2002. A new class of symbolic abstract neural nets: Tissue P systems. *Computing and Combinatorics.* 2387, pp.290-299.

Pan, L. and Perez-Jimenez, M.J., 2010. Computational complexity of tissue-like P systems. *Journal of Complexity,* 26(3)**,** pp.296-315.

Paun, G., 2007. Tracing some open problems in membrane computing. *Romanian Journal of Information Science and Technology,* 10(4)**,** pp.303-314.

Paun, G., 2002. Introduction: Membrane Computing—What It Is and What It Is Not. *Membrane Computing.* pp.1-6.

Paun, G., 2010. A quick introduction to membrane computing. *The Journal of Logic and Algebraic Programming,* 79(6)**,** pp.291-294.

Pena-Cantillana, F., Diaz-Pernil, D., Berciano, A. and Gutierrez-Naranjo, M. A., 2011. A parallel implementation of the thresholding problem by using tissue-like P systems. Computer Analysis of Images and Patterns, *Computer Analysis of Images and Patterns - 14th International Conference, CAIP 2011*, August pp.29-31, 2011, Seville, Spain.

Peng, H., Shao, J., LI, B., Wang, J., Perez-Jimenez, M. J., Jiang, Y. and Yang, Y., 2012. Image thresholding with cell-like P systems. In: *Proceedings of the Tenth Brainstorming Week on Membrane Computing,* 2.

Peng, H., Wang, J. and Perez-Jimenez, M.J., 2015. Optimal multi-level thresholding with membrane computing. *Digital Signal Processing,* 37, pp.53-64.

Peng, H., Yang, Y., Zhang, J., Huang, X. and Wang, J., 2014. A Region-based Color Image Segmentation Method Based on P Systems. *Sceince and Technology,* 17(1), 63-75.

Perez-Hurtado, I., Valencia–Cabrera, L., Chacon, J.M., Riscos–Nunez, A. and Perez–Jimenez, M. J., 2014. A P–Lingua based Simulator for Tissue P Systems with Cell Separation. *Sceince and Technology,* 17(1), pp.89-102.

Reina-Molina, R., Carnero, J. and Diaz-Pernil, D., 2010. Image segmentation using tissue-like P systems with multiple auxiliary cells. *Image-A,* 1, pp.143-150.

Shapiro, L. and Stockman, G.C., 2001. *Computer Vision.* 1st ed. Prentice Hall.

Sheeba, F., Thaburaj, R., Nagar, A.K. and Mammen, J.J., 2011. Segmentation of peripheral blood smear images using tissue-like P systems. Bio-Inspired Computing: Theories and Applications (BIC-TA), *2011 Sixth International Conference on*, IEEE, pp.257-261.

Yahya, R.I., Hasan, S., George, L.E. and Alsalibi, B., 2015. Membrane Computing for 2D Image Segmentation. *Int. J. Advance Soft Compu. Appl,* 7(1), pp.35-50.

Yang, Y., Peng, H., Jiang, Y., Huang, X. and Zhang, J., 2013. A region-based image segmentation method under P systems. *J. Inf. Comput. Sci,* 10(10)**,** pp.2943-2950.

Zhang, Z. and Peng, H., 2012. Object segmentation with membrane computing. *Journal of Information & Computational Science,* 9**,** pp.5417-5424.

APPENDIX I

Rules of Segmentation Using six Adjacency in P-Lingua Syntax

```
def init_cells()
   {
   @mu = [[]'1 []'2]'0;          /* cell 1 is the input, cell 2 is output */
   }
def init_multisets(n)            /* define the initial multisets*/
      {
      @ms(1) += z{1} ;
      }
 def init_environment(n,m)      /*define the environment */
   {
   @ms(0) += z{i}: 1<=i<=7;      /*define the Z counter */
   @ms(0) += Gx {i,j},Bx {i,j},R {i,j} ,B{i,j},G{i,j}: 1<=i<=n ,1<=j<=m;
   }
      /* ***************    Rules    ****************   */
def init_rules(n,m)
   {
              /********** Type 1 rules ****** */
/* Type one the rule is used  to increment the counter z after each
configuration duplicating the number of copies in each step */
/*R1*/ [z{i}]'1 <--> [z{i+1}*2]'0 : 1 <= i <= 7;
          {
              /********** Type 2 rules ****** */

           /* ********* G<B ********* */
/*Type two the rules are used when the image has two adjacent pixels with
different colours, so it will mark the pixel with the less associated colour.
Pixel Gx will be marked because it is less than B. */
/*R2,1*/ [G {i,j} , B {i,j+1}]'1 <--> [Gx {i,j} , B{i,j+1}]'0;
```

/*R2,2*/ [G {i,j} , B {i,j-1}]'1 <--> [Gx {i,j} , B{i,j-1}]'0;
/*R2,3*/ [G {i,j} , B {i+1,j}]'1 <--> [Gx {i,j} , B{i+1,j}]'0;
/*R2,4*/ [G {i,j} , B {i-1,j}]'1 <--> [Gx {i,j} , B{i-1,j}]'0;
/*R2,5*/ [G {i,j} , B {i-1,j+1}]'1 <--> [Gx {i,j} , B{i-1,j+1}]'0;
/*R2,6*/ [G {i,j} , B {i+1,j-1}]'1 <--> [Gx {i,j} , B{i+1,j-1}]'0;
             }: 1 <= i <= n , 1 <= j <= m;
        {

/* ********* B<R ********* */

/*Type two the rules are used when the image has two adjacent pixels with different colours, so it will mark the pixel with the less associated colour. Pixel Bx will be marked because it is less than R. */
/*R2,1*/ [B {i,j} , R {i,j+1}]'1 <--> [Bx {i,j} , R{i,j+1}]'0;
/*R2,2*/ [B {i,j} , R {i,j-1}]'1 <--> [Bx {i,j} , R{i,j-1}]'0;
/*R2,3*/ [B {i,j} , R {i+1,j}]'1 <--> [Bx {i,j} , R{i+1,j}]'0;
/*R2,4*/ [B {i,j} , R {i-1,j}]'1 <--> [Bx {i,j} , R{i-1,j}]'0;
/*R2,5*/ [B {i,j} , R {i-1,j+1}]'1 <--> [Bx {i,j} , R{i-1,j+1}]'0;
/*R2,6*/ [B {i,j} , R {i+1,j-1}]'1 <--> [Bx {i,j} , R{i+1,j-1}]'0;
             }: 1 <= i <= n , 1 <= j <= m;
        {

/* ********* G<R ********* */

/*Type two the rules are used when the image has two adjacent pixels with different colours, so it will mark the pixel with the less associated colour. Pixel Gx will be marked because it is less than R. */
/*R2,1*/ [G {i,j} , R {i,j+1}]'1 <--> [Gx {i,j} , R{i,j+1}]'0;
/*R2,2*/ [G {i,j} , R {i,j-1}]'1 <--> [Gx {i,j} , R{i,j-1}]'0;
/*R2,3*/ [G {i,j} , R {i+1,j}]'1 <--> [Gx {i,j} , R{i+1,j}]'0;
/*R2,4*/ [G {i,j} , R {i-1,j}]'1 <--> [Gx {i,j} , R{i-1,j}]'0;
/*R2,5*/ [G {i,j} , R {i-1,j+1}]'1 <--> [Gx {i,j} , R{i-1,j+1}]'0;
/*R2,6*/ [G {i,j} , R {i+1,j-1}]'1 <--> [Gx {i,j} , R{i+1,j-1}]'0;
             }: 1 <= i <= n , 1 <= j <= m;

/* ********* Type 3 ********* */

 /* Type three the rules are used to send the marked pixels to the environment using counter z*/
/*R3*/ [z{7},Gx {i,j}]'1 <--> [#]'2 : 1 <= i <= n , 1 <= j <= m;
/*R3*/ [z{7},Bx {i,j}]'1 <--> [#]'2 : 1 <= i <= n , 1 <= j <= m;
        }