Counting Cells with Tissue-like P Systems

Ioan Ardelean[1,2], Daniel Díaz-Pernil[3], Miguel A. Gutiérrez-Naranjo[4], Francisco Peña-Cantillana[4], Raúl Reina-Molina[3], Iris Sarchizian[1]

[1]Ovidius University, Constanţa 90052, Romania
[2]Institute of Biology Bucharest, Romanian Academy, 060031 Bucharest, Romania
ioan.ardelean57@yahoo.com, irissarchizian@yahoo.com

[3]Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics - University of Sevilla, 41012, Spain
sbdani@us.es, raulrm75@gmail.com

[4]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, 41012, Spain
magutier@us.es, frapencan@gmail.com

Summary. Counting the number of cells obtained in an experiment is crucial in many areas in Biology. Nonetheless, this is usually performed by hand by the researcher due the intrinsic difficulty of the task. In this paper, we present a set of techniques for counting cells inspired in the treatment of Digital Images via tissue-like P systems with promoters.

1 Introduction

Due to the increasing amount of information stored as visual data, the development of new software for dealing efficiently with digital images becomes a necessity. The number of application areas is growing and the progress of new technology needs the design of new software for handling such information. Among the classical areas, we can cite biometrics [1], surveillance [15] or medical imaging [3], but there are many others.

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, [13, 14, 16, 17, 39], *thresholding* [12] or *smoothing* [33]. Special attention deserves [20], where the *symmetric dynamic programming stereo* (SDPS) algorithm [21] for stereo matching was implemented by using simple P modules with duplex channels.

We focus here on a problem from Microbiology. Automated image analysis is increasingly used in Microbiology to quantify important parameters for research and application. The most studied so far are the following: cell numbers, cell volumes, frequencies of dividing cells, in situ classification of bacteria, enumeration

of actively respiring bacteria, characterization of bacterial growth on solid medium, viability and physiological activity in biofilms (e.g. [4, 5, 19, 23, 26, 30, 11, 34, 35, 37, 38, 40, 41, 43]). In this paper we report our study of the application of Membrane Computing techniques to the problem of counting cells and show some preliminary results. The whole process is a combination of different techniques of processing images (binarization, segmentation, noise reduction ...) which can be performed by different families of P systems. The final algorithm is a sequence of partial processes which can be performed by Membrane Computing techniques, and the application of such processes can be seen as a global machine which takes as input a digital image showing a biological entity (usually, a photograph taken with a microscopy in a wet lab) and the output is the number of cells in the picture.

The different families of P systems used in the stages of the process have inspired parallel software programs which have been developed by using a device architecture called CUDA™, (Compute Unified Device Architecture). CUDA™ is a general purpose parallel computing architecture that allows the parallel NVIDIA[1] Graphics Processors Units (GPUs) to solve many complex computational problems in a more efficient way than on a CPU. GPUs constitute nowadays a solid alternative for high performance computing, and the advent of CUDA allows programmers a friendly model to accelerate a broad range of applications. This novel architecture has been previously used to implement parallel software that simulates the behavior of P systems [6, 8, 9, 10, 32, 33], and, in a similar way than in other implementations, the obtained results in the problem of counting cells are quite promising.

The paper is organized as follows: Next, we recall the computational model used to design the different families of P systems that performs the stages of the algorithm. In section 3, we outline the steps of the process that takes as an input a digital image taken in a wet lab and outputs the number of cells in the image. Section 4 shows an illustrative example and some details of the CUDA implementation. The paper finishes with some conclusions and open lines for a future research.

2 Formal Framework

Next, we recall some basics on the P system model chosen for implementing the solution described below. The model is tissue-like P systems with promoters. Promoters are usually defined on cell-like models [24] and its extension to tissue-like is quite natural. Next, we recall the formal definition.

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

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

where

[1] `http://www.nvidia.com.`

1. Γ *is a finite alphabet, whose symbols will be called objects;*
2. $\Sigma \subseteq \Gamma$ *is the input alphabet;*
3. $\mathcal{E} \subseteq \Gamma$ *is a finite alphabet representing the set of the objects in the environment available in an arbitrary large amount of copies;*
4. $w_1, \ldots, w_q$ *are strings over* Γ *representing the multisets of objects associated with the cells in the initial configuration;*
5. $\mathcal{R}$ *is a finite set of rules of the following form:*

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

In these rules, the labels $1, \ldots, q$ *correspond to the* q *cells and the label* 0 *corresponds to the environment;*

6. $i_{in} \in \{1, 2, \ldots, q\}$ *denotes the input region;*
7. $i_{out} \in \{1, 2, \ldots, q\}$ *denotes the output region.*

The rule $(pro \,|\, i, u/v, j)$ can be applied over two cells (or a cell and the environment) i and j such that u (contained in cell i) is traded against v (contained in cell j). The rule is applied if in i the objects of the promoter *pro* are present. The promoter is not modified by the application of the rule. If the promoter is empty, we will write $(i, u/v, j)$ instead of $(\emptyset \,|\, i, u/v, j)$.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). In one step, each object in a membrane can only be used for one rule (non-deterministically chosen when there are several possibilities), but any object which can participate in a rule of any form must do it, i.e. in each step we apply a maximal multiset of rules. A *configuration* is an instantaneous description of the system Π, and it is represented as a tuple $(w_0, w_1, \ldots, w_q)$, where $w_1, \ldots, w_q$, where represent the multiset of objects contained in the q cells and w_0 represent 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 sequence of computation steps is called a *computation*. A configuration is *halting* when no rules can be applied to it.

3 Counting Cells

Counting cells in a picture taken by a microscopy in a wet lab is a hard task. The study of the cells is usually made in such conditions (light filters, noise data, aqueous media, ...) where it is difficult for the human expert to decide whether a spot in the image correspond to a cell or not. In such conditions, the development of a software that provides the exact number of spots in the image that correspond to cells is impossible, since two different experts hardly agree in this issue.

From this starting point, and bearing in mind that the research in Microbiology needs computer aid, we propose a Membrane Computing protocol for obtaining

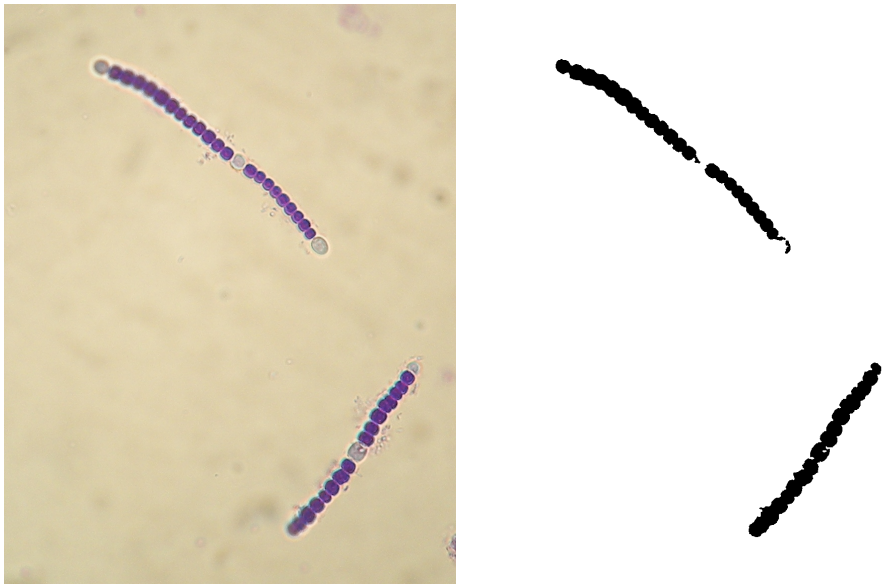

Fig. 1. (Left) Image with two filaments of heterocystous cyanobacterium labeled with 0.02% crystal violet (the arrows indicates the presence of heterocysts, specialized cells, not labeled by crystal violet in our experimental labeling conditions: 0.02% crystal violet concentration and 5 minutes of coloration). (Right) Binarization of the image.

the number of cells in a image, or more exactly, the number of spots in the image that probably correspond to a cell. Since this is a empirical problem, instead of a formal definition of the problem, we will base our description in a case study. We start by considering the image of Fig. 1. Such image correspond to filamentous cyanobacterium and has been taken by using the number of cells within each filament. It is very important to calculate the ratio of the number of heterocysts to non-differentiate cells, also called vegetative cell, as well as to calculate the distribution of filaments lengths (correlated to the total number of cells within each filament) within a population (composed of hundreds of filaments per each milliliter) of this cyanobacterium. In microbiological, studies for the determination of number of dividing cells related to the number of total cells are also very useful to calculate the growth rate in natural population of bacteria/cyanobacteria, heterocystous or not ([2, 7, 22, 25, 27, 29, 42]).

The target is to take the image as an input and provide a the number of cells as output and performing the different stages of the process by using Membrane Computing techniques. The stages are the following:

- **Stage 1: Binarization**. This fist stage consist on getting a new image from the original with only two colors (a binary image). In this stage we use a family

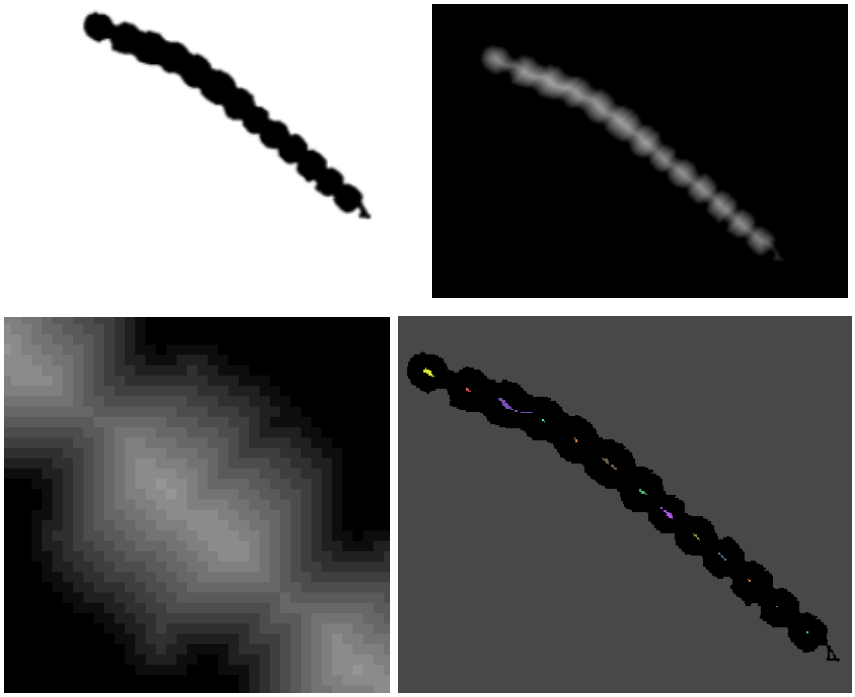

Fig. 2. The first image (top-left) is a detail of Figure 1 (left). The second and third ones (top-right and bottom-left) show details of the thinning process. Finally, the last image (bottom-right) shows a set of pixels inside each individual cells. Different cells are marked with different colors. The number of such different colors provides the number of cells.

of tissue-like P systems similar to the described one in [12, 32] for this aim. A result of this process can be seen in Figure 1 (right).

- **Stage 2: Segmentation**. This is the process that split the image in several *meaningful* regions. It is basic for the treatment of digital images and it is widely used in medical images for identifying the region of interest. We also perform this stage by using Membrane Computing techniques, namely, the algorithm described in [13].
- **Stage 3: Noise Reduction**. The images taken in a wet lab are usually far from having homogeneous regions. Due to the intrinsic nature of the biological research, in the image it is common tho find little spots that in a mechanic process of counting cells, can be easily taken as little cells when they are merely spots due to the noise. Obviously, removing such noise imply to take difficult decision and should be done carefully. We propose three different steps to eliminate noise from an image:

  - **Stage 3.1: Labeling black connected components**. The first step consists on determining connected components in the image. The set of pixels to be considered as a cell is a connected set of pixels. The connectivity problem among pixels and the study of connected components is a problem linking Digital Imagery and Algebraic Topology and it has been recently studied in the framework of Membrane Computing. For this step, we propose a partial application of the Membrane Computing algorithm proposed in [18].
  - **Stage 3.2: Calculating Areas**. This step starts after the identification of the connected components of the image. Our proposal is to use Membrane Computing techniques in order to determinate the *size* of the connected component. In our approach, the size of the spot is its number of pixels. This process is also performed by using the symport/antiport rules used in tissue like P systems and basically consists on counting the number of objects in the P system which represent the pixels of a concrete spot.
  - **Stage 3.3: Eliminating small components**. From the previous step, we have a number associated to each spot, representing its *size*. In this stage we will consider a *threshold* in order to decide if a spot is large enough to be considered *meaningful* or if the spot should be considered as noise, and hence, to be ignored. Obviously, the threshold depends on the experiment and it must be provided by the experts.
- **Stage 4: Counting Cells**. The three previous stages can be considered as preprocessing of the image. The algorithm for counting cells starts now. The key point in this stage is to consider the geometry that a cell usually shows in an image taken form a microscopy in a wet lab. Such images usually show the cells as convex spots and the image of multicellular beings usually shows a wavy border. An appropriate process of thinning, inspired in other thinning process[2] with Membrane Computing techniques, produces little isolated set of point for each cell (see Figure 2 (bottom-right)).
- **Stage 5: Output**. In the last stage, the isolated set of pixels that represent the cells are codified as an appropriate set of objects of the alphabet in the output membrane of the corresponding P system. Counting these sets of objects provides the number of cells in the image

3.1 Implementation

Inspired in the families of tissue-like P systems that perform the stages of the process of counting cells, a software tool has been implemented by using CUDA™, (Compute Unified Device Architecture) [28, 31]. CUDA™ is a general purpose parallel computing architecture that allows the parallel NVIDIA Graphics Processors Units (GPUs) to solve many complex computational problems in a more efficient way than on a CPU.

[2] See these proceedings

The experiments have been performed on a computer with a CPU AMD Athlon II x4 645, which allows to work with four cores of 64 bits to 3.1 GHz. The computer has four blocks of 512KB of L2 cache memory and 4 GB DDR3 to 1600 MHz of main memory. The used graphical card (GPU) is an NVIDIA Geforce GT240 composed by 12 *Stream Processors* with a total of 96 cores to 1340 MHz. It has 1 GB DDR3 main memory in a 128 bits bus to 700 MHz. So, the transfer rate obtained is by 54.4 Gbps. The used Constant Memory is 64 KB and the Shared Memory is 16 KB. Its Compute Capability level is 1.2 (from 1.0 to 2.1). The implementation deals with N blocks of threads for the complete image in our GPU of 96 cores.

3.2 Example

Figures 1 and 2 shows several details of an illustrative example of the process. The original image in Fig. 1 shows a filamentous cyanobacterium able to differentiate heterocysts (the arrows indicates the presence of heterocysts, not labeled by crystal violet in our experimental conditions) when grown on mineral medium (so called BG0) in the absence of combined nitrogen (biological identification at the level of genus is under work). In these conditions the biological specimen (the filamentous cyanobacterium) is able to utilize atmospheric nitrogen as a source of nitrogen to synthetizise its own biochemical components (amino acids, proteins etc); the first major steps in this utilization of atmospheric nitrogen occurs in heteocysts, differentiate cells within the filament [36]. The number of cells found in the image from Fig. 2 is 13.

4 Conclusions

The development of experimental sciences where a big amount of data is stored as digital images needs of powerful software which helps the researcher to understand the studied processes. In particular, Microbiology significantly benefits if automated image analysis is used by microbiologists in their professional activities; furthermore the interplay between microbiologists, mathematicians and engineers in this field could be helpful in developing new opportunities within *old* software, or ,even, to generate new software more appropriate for different microbiological task.

In this respect, Membrane Computing devices have features that make them suitable for the study of digital images, as the encapsulation of the information and its treatment in parallel. This paper reports how an appropriate combination of families of tissue-like P systems can solve the problem of counting cells. The study of how these or other families can be combined for solving more problems from Microbiology (or from other experimental sciences) is a challenge for the next years.

Acknowledgements

DDP and MAGN acknowledge the support of the project 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. IS and IA are grateful to Dr. Tech. Jyrki Selinummi (Department of Signal Processing, Tampere University of Technology, Finland) for very useful and kind discussion concerning digital image analysis.

References

1. Adeoye, O.S.: A survey of emerging biometric technologies. International Journal of Computer Applications 9(10), 1–5 (2010)
2. Agawin, N.S., Agusti, S.: Abundance, frequency of dividing cells and growth rates of synechococcus sp. (cyanobacteria) in the stratified northwest mediterranean sea. Journal of Plankton Research 19(11), 1599–1615 (1997)
3. 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)
4. Belyaev, N., Paavilainen, S., Korpela, T.: Characterization of bacterial growth on solid medium with image analysis. Journal of Biochemical and Biophysical Methods 25(2-3), 125–32 (1992)
5. Bloem, J., Veninga, M., Shepherd, J.: Fully automatic determination of soil bacterium numbers, cell volumes, and frequencies of dividing cells by confocal laser scanning microscopy and image analysis. Applied and Environmental Microbiology 61(3), 926–936 (1995)
6. Carnero, J., Díaz-Pernil, D., Gutiérrez-Naranjo, M.A.: Designing tissue-like P systems for image segmentation on parallel architectures. In: Martínez-del-Amor, M.A. *al.* (eds.) Ninth Brainstorming Week on Membrane Computing. pp. 43–62. Fénix Editora, Sevilla, Spain (2011)
7. Carpenter, E., Campbell, L.: Diel patterns of cell division and growth rates of synechococcus spp. in long island sound. Marine Ecology Progress Series 47, 179–183 (1988)
8. Cecilia, J.M., García, J.M., Guerrero, G.D., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: Implementing P systems parallelism by means of GPUs. In: Păun, Gh. *et al.* (eds.) Workshop on Membrane Computing. Lecture Notes in Computer Science, vol. 5957, pp. 227–241. Springer, Berlin Heidelberg (2009)
9. Cecilia, J.M., García, J.M., Guerrero, G.D., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: Simulating a P system based efficient solution to SAT by using GPUs. Journal of Logic and Algebraic Programming 79(6), 317–325 (2010)
10. Cecilia, J.M., García, J.M., Guerrero, G.D., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: Simulation of P systems with active membranes on CUDA. Briefings in Bioinformatics 11(3), 313–322 (2010)
11. Chávez de Paz, L.E.: Image analysis software based on color segmentation for characterization of viability and physiological activity of biofilms. Applied and Environmental Microbiology 75(6), 1734–9 (2009)

12. 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)
13. 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, 169–176 (2009)
14. 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)
15. 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 (aug 2000)
16. 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)
17. 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)
18. 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)
19. Fero, M., Pogliano, K.: Automated quantitative live cell fluorescence microscopy. Cold Spring Harb Perspectives in Biology 2(8), (2010)
20. Gimel'farb, G., Nicolescu, R., Ragavan, S.: P systems in stereo matching. In: Real, P. *et al.* (eds.) Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 6855, pp. 285–292. Springer Berlin / Heidelberg (2011)
21. Gimel'farb, G.L.: Probabilistic regularisation and symmetry in binocular dynamic programming stereo. Pattern Recognition Letters 23(4), 431–442 (2002)
22. Hagström, A., Larsson, U., Hörstedt, P., Normark, S.: Frequency of dividing cells, a new approach to the determination of bacterial growth rates in aquatic environments. Applied and Environmental Microbiology 37(5), 805–12 (1979)
23. Heydorn, A., Nielsen, A.T., Hentzer, M., Sternberg, C., Givskov, M., Ersbøll, B.K., Molin, S.: Quantification of biofilm structures by the novel computer program COMSTAT. Microbiology 146(10), 2395–2407 (2000)
24. Ionescu, M., Sburlan, D.: On p systems with promoters/inhibitors. Journal of Universal Computer Science 10(5), 581–599 (2004)
25. Joung, S.H., Kim, C.J., Ahn, C.Y., Jang, K.Y., Boo, S.M., Oh, H.M.: Simple method for a cell count of the colonial cyanobacterium, microcystis sp. Journal of Microbiology 44(5), 562–5 (2006)
26. Lehmussola, A., Ruusuvuori, P., Selinummi, J., Huttunen, H., Yli-Harja, O.: Computational framework for simulating fluorescence microscope images with cell populations. IEEE Transactions on Medical Imaging pp. 1010–1016 (2007)
27. Li, W.K.W., Dickie, P.M.: Relationship between the number of dividing and nondividing cells of cyanobacteria in north atlantic picoplankton. Journal of Phycology 27(5), 559–565 (1991)

28. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable parallel programming with CUDA. Queue 6, 40–53 (2008)
29. Nielsen, S.L.: Size-dependent growth rates in eukaryotic and prokaryotic algae exemplified by green algae and cyanobacteria: comparisons between unicells and colonial growth forms. Journal of Plankton Research 28(5), 489–498 (2006)
30. Ogawa, M., Tani, K., Yamaguchi, N., Nasu, M.: Development of multicolour digital image analysis system to enumerate actively respiring bacteria in natural river water. Journal of Applied Microbiology 95(1), 120–128 (2003)
31. Owens, J.D., Houston, M., Luebke, D., Green, S., Stone, J.E., Phillips, J.C.: GPU Computing. Proceedings of the IEEE 96(5), 879–899 (2008)
32. 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. *et al.* (eds.) CAIP (2). Lecture Notes in Computer Science, vol. 6855, pp. 277–284. Springer (2011)
33. 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)
34. Pernthaler, J., Alfreider, A., Posch, T., Andreatta, S., Psenner, R.: In situ classification and image cytometry of pelagic bacteria from a high mountain lake (Gossenköllesee, Austria). Applied and Environmental Microbiology 63(12), 4778–83 (1997)
35. Pettipher, G., Rodrigues, U.M.: Semi-automated counting of bacteria and somatic cells in milk using epifluorescence microscopy and television image analysis. Journal of Applied Microbiology 53(3), 323–329 (1982)
36. Sarchizian, I., Ardelean, I.I.: Axenic culture of a diazotrophic filamentous cyanobacterium isolated from mesothermal sulphurous springs (Obanul Mare - Mangalia). Romanian Journal of Biology - Plant Biology 55(1), 47–53 (2010)
37. Selinummi, J., Ruusuvuori, P., Podolsky, I., Ozinsky, A., Gold, E., Yli-Harja, O., Aderem, A., Shmulevich, I.: Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images. PLoS ONE 4(10), 9 (2009)
38. Selinummi, J., Seppälä, J., Yli-Harja, O., Puhakka, J.A.: Software for quantification of labeled bacteria from digital microscope images by automated image analysis. Biotechniques 39(6), 859–863 (2005)
39. Sheeba, F., Thamburaj, R., Nagar, A.K., Mammen, J.J.: Segmentation of peripheral blood smear images using tissue-like P systems. Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on 0, 257–261 (2011)
40. Shopov, A., Williams, S.C., Verity, P.G.: Improvements in image analysis and fluorescence microscopy to discriminate and enumerate bacteria and viruses in aquatic samples. Aquatic Microbial Ecology 22(2), 103–110 (2000)
41. Van Wambeke, F.: Enumeration and size of planktonic bacteria determined by image analysis coupled with epifluorescence. Annales de l'Institut Pasteur Microbiology 139(2), 261–72 (1988)
42. Yamamoto, Y., Shiah, F.K.: Relationship between cell growth and frequency of dividing cells of microcystis aeruginosa. Plankton Benthos Res 5(4), 131–135 (2010)
43. Yang, X., Beyenal, H., Harkin, G., Lewandowski, Z.: Quantifying biofilm structure using image analysis. Journal of Microbiological Methods 39(2), 109–119 (2000)