3D Cell Pattern Generation in Artificial Development

Arturo Chavoya, Irma R. Andalon-Garcia, Cuauhtemoc Lopez-Martin, and M.E. Meda-Campaña

Abstract. Cell pattern formation has an important role in both artificial and natural development. This paper presents an artificial development model for 3D cell pattern generation based on the cellular automata paradigm. Cell replication is controlled by a genome consisting of an artificial regulatory network and a series of structural genes. The genome was evolved by a genetic algorithm in order to generate 3D cell patterns through the selective activation and inhibition of genes. Morphogenetic gradients were used to provide cells with positional information that constrained cellular replication in space. The model was applied to the problem of growing a solid French flag pattern in a 3D virtual space.

1 Introduction

In biological systems, development is a fascinating and very complex process that involves following a sequence of genetically programmed events that ultimately produce the developed organism. One of the crucial stages in the development of an organism is that of pattern formation, where the fundamental body plans of the individual are outlined. Recent evidence has shown that gene regulatory networks play a central role in the development and metabolism of living organisms [13]. Researchers in biological sciences have confirmed that the diverse cell patterns created during the developmental stages are mainly due to the selective activation and inhibition of very specific regulatory genes.

On the other hand, artificial models of cellular development have been proposed over the years with the objective of understanding how complex structures and patterns can emerge from one or a small group of initial undifferentiated cells

Arturo Chavoya · Irma R. Andalon-Garcia · Cuauhtemoc Lopez-Martin · M.E. Meda-Campaña
Universidad de Guadalajara, Periférico Norte 799 - L308
Zapopan, Jal., Mexico CP 45000
e-mail: {achavoya,agi10073,cuauhtemoc,emeda}@cucea.udg.mx

J.R. González et al. (Eds.): NICSO 2010, SCI 284, pp. 127–139, 2010.
springerlink.com

[7, 21, 22, 24]. In this paper we propose an artificial cellular growth model that generates 3D patterns by means of the selective activation and inhibition of development genes under the constraints of morphogenetic gradients. Cellular growth is achieved through the expression of structural genes, which are in turn controlled by an Artificial Regulatory Network (ARN) evolved by a Genetic Algorithm (GA). The ARN determines at which time steps cells are allowed to grow and which gene to use for reproduction, whereas morphogenetic gradients constrain the position at which cells can replicate. Both the ARN and the structural genes make up the artificial cell's genome. In order to test the functionality of the development program found by the GA, the evolved genomes were applied to a cellular growth testbed based on the Cellular Automata (CA) paradigm that has been successfully used in the past to develop simple 2D and 3D geometrical shapes [8]. The model presented in this work was applied to a 3D version of what is known as the *French flag problem*. The 2D version of this problem has traditionally been used in biology —and more recently in computer science— to model the determination of cell patterns in tissues, usually through the use of morphogenetic gradients to help determine cell position.

The paper starts with a section describing the French flag problem with a brief description of models that have used it as a test case. The next section describes the cellular growth testbed used to evaluate the evolved genomes in their ability to form the desired patterns, followed by a section presenting the morphogenetic gradients that constrain cell replication. The artificial cell's genome is presented next, followed by a section describing the GA and how it was applied to evolve the genomes. Results are presented next, followed by a section of conclusions.

2 The French Flag Problem

The problem of generating a French flag pattern was first introduced by Wolpert in the late 1960s when trying to formulate the problem of cell pattern development and regulation in living organisms [30]. This formulation has been used since then by some authors to study the problem of artificial pattern development. More specifically, the problem deals with the creation of a pattern with three sharp bands of cells with the colors and order of the French flag stripes.

Lindenmayer and Rozenberg used the French flag problem to illustrate how a grammar-based L-System could be used to solve the generation of this particular pattern when enunciated as the production of a string of the type $a^n b^n c^n$ over the alphabet $\{a,b,c\}$ and with $n > 0$ [23]. On the other hand, Herman and Liu [18] developed an extension of a simulator called CELIA [1] and applied it to generate a French flag pattern in order to study synchronization and symmetry breaking in cellular development.

Miller and Banzhaf used what they called Cartesian genetic programming to evolve a cell program that would construct a French flag pattern [25]. They tested the robustness of their programs by manually removing parts of the developing pattern. They found that several of their evolved programs could repair to some extent the

damaged patterns. Bowers also used this problem to study the phenotypic robustness of his embryogeny model, which was based on cellular growth with diffusing chemicals as signaling molecules [4].

Gordon and Bentley proposed a development model based on a set of rules evolved by a GA that described how development should proceed to generate a French flag pattern [16]. The morphogenic model based on a multiagent system developed by Beurier et al. also used an evolved set of agent rules to grow French and Japanese flag patterns [3]. On the other hand, Dever et al. proposed a neural network model for multicellular development that grew French flag patterns [14] . Even models for developing evolvable hardware have benefited from the French flag problem as a test case [17, 28].

More recently, Knabe et al. [20] developed a model based on the CompuCell3D package [12] combined with a genetic regulatory network that controlled cell parameters such as size, shape, adhesion, morphogen secretion and orientation. They were able to obtain final 2D patterns with matches of over 75% with respect to a 60×40 pixel target French flag pattern.

3 Cellular Growth Testbed

Cellular automata were chosen as models of cellular growth, as they provide a simple mathematical model that can be used to study self-organizing features of complex systems [29]. CA are characterized by a regular lattice of N identical cells, an interaction neighborhood template η, a finite set of cell states Σ, and a space- and time-independent transition rule ϕ which is applied to every cell in the lattice at each time step.

In the cellular growth testbed used in this work, a $13 \times 13 \times 13$ regular lattice with non-periodic boundaries was used. The set of cell states was defined as $\Sigma = \{0,1\}$, where 0 can be interpreted as an empty cell and 1 as an occupied or active cell. The interaction neighborhood η considered was a 3D Margolus template (Fig. 1), which has previously been used with success to model 3D shapes [31]. In this template there is an alternation of the block of cells considered at each step of the CA algorithm. At odd steps, the seven cells shown to the left and the back in the figure constitute the interaction neighborhood, whereas at even steps the neighborhood is formed by the mirror cells of the previous block.

The CA rule ϕ was defined as a lookup table that determined, for each local neighborhood, the state (empty or occupied) of the objective cell at the next time step. For a binary-state CA, these update states are termed the rule table's "output bits". The lookup table input was defined by the binary state value of cells in the local interaction neighborhood, where 0 meant an empty cell and 1 meant an occupied cell and the parity bit p determined which of the two blocks of cells was being considered for evaluation [8]. The output bit values shown in Fig. 1 are only for illustration purposes; the actual values for a predefined shape, such as a cube, are found by a GA.

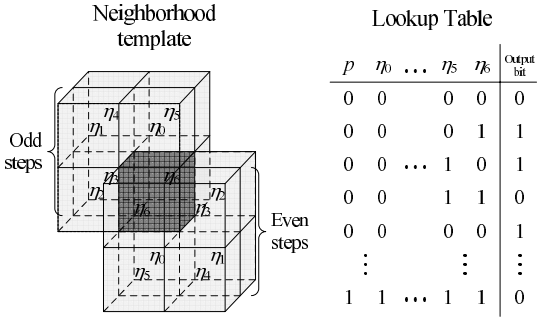

Fig. 1 Cellular automaton's 3D Margolus neighborhood template and the associated lookup table. The parity bit p in the lookup table determines which block of the neighborhood template is being considered for evaluation. The objective cell is depicted as a darker cube in the middle of the template

4 Morphogenetic Gradients

Ever since Turing's seminal article on the theoretical influence of diffusing chemical substances on an organism's pattern development [27], the role of these molecules has been confirmed in a number of biological systems. These organizing substances were termed *morphogens*, given their involvement in driving morphogenetic processes. In the present model, morphogenetic gradients were generated similar to those found in the eggs of the fruit fly *Drosophila*, where orthogonal gradients offer a sort of Cartesian coordinate system [5]. These gradients provide reproducing cells with positional information in order to facilitate the spatial generation of patterns. The artificial morphogenetic gradients were set up as suggested in [24], where morphogens diffuse from a source towards a sink, with uniform morphogen degradation throughout the gradient.

Before cells were allowed to reproduce in the cellular growth testbed, morphogenetic gradients were generated by diffusing the morphogens from one of the CA boundaries for 1000 time steps. Initial morphogen concentration level was set at 255 arbitrary units, and the source was replenished to the same level at the beginning of each cycle. The diffusion factor was 0.20, i.e. at each time step every grid position diffused 20% of its morphogen content and all neighboring positions received an equal amount of this percentage. This factor was introduced to avoid rapid morphogen depletion at cell positions and its value was experimentally determined to render a smooth descending gradient. The sink was set up at the opposite boundary of the lattice, where the morphogen level was always set to zero. At the end of each time step, morphogens were degraded at a rate of 0.005 throughout the CA lattice. Three orthogonal gradients were defined in the CA lattice, one for each of the main Cartesian axes (Fig. 2). In the figures presented in this work the following conventions are used: in the 3D insets the positive x axis extends to right, the positive y axis is towards the back of the page, the positive z axis points to the top, and the axes are rotated 45 degrees to the left to show a better perspective.

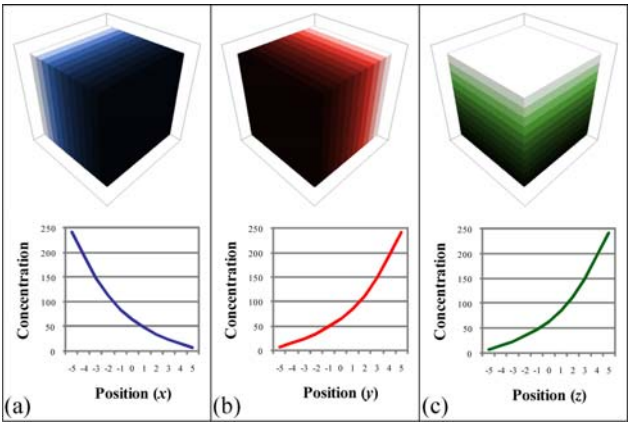

Fig. 2 Morphogenetic gradients. Positions with highest morphogen concentration are depicted in white; darker tones mean lower concentrations. **(a)** Left to right (*x axis*); **(b)** back to front (*y axis*); **(c)** top to bottom (*z axis*)

5 Genome

Genomes are the repository of genetic information in living organisms. They are encoded as one or more chains of DNA, and they regularly interact with other macromolecules, such as RNA and proteins. Artificial genomes are typically coded as strings of discrete data types. The genome used in this model was defined as a binary string starting with a series of ten regulatory genes, followed by a number of structural genes (Fig. 3).

5.1 Regulatory Genes

The series of regulatory genes at the beginning of the genome constitutes an Artificial Regulatory Network. ARNs are computer models whose objective is to emulate the gene regulatory networks found in nature. ARNs have previously been used to study differential gene expression either as a computational paradigm or to solve particular problems [2, 7, 15, 19, 26]. The gene regulatory network implemented in this work is an extension of the ARN presented in [9], which in turn is based on the model proposed by Banzhaf [2].

In the present model, each regulatory gene consists of a series of eight inhibitor/enhancer sites, a series of five regulatory protein coding regions, and three morphogen threshold activation sites that determine the allowed positions for cell reproduction (Fig. 3). Inhibitor/enhancer sites are composed of a 12-bit function defining region and a regulatory site. Regulatory sites can behave either as an enhancer or an inhibitor, depending on the configuration of the function defining bits associated with them. If there are more 1's than 0's in the defining bits region, then the regulatory site functions as an enhancer, but if there are more 0's than 1's, then

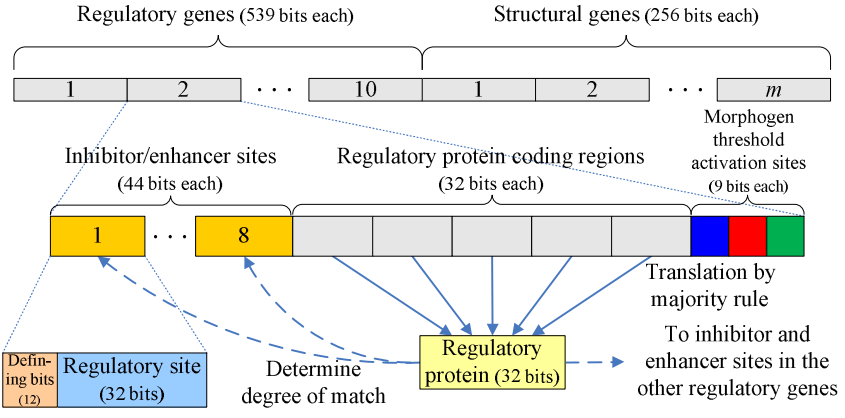

Fig. 3 Genome structure and regulatory gene detail. Regulatory genes make up an artificial regulatory network, whereas structural genes contain the lookup tables that control cell reproduction. The number of structural genes m depends on the pattern to be generated and whether or not structural genes are duplicated, as explained in Sect. 7. For the final simulations, $m = 6$

the site behaves as an inhibitor. Finally, if there is an equal number of 1's and 0's, then the regulatory site is turned off [10].

Regulatory protein coding regions "translate" a protein using the majority rule, i.e. for each bit position in these regions, the number of 1's and 0's is counted and the bit that is in majority is translated into the regulatory protein. The regulatory sites and the individual protein coding regions all have the same size of 32 bits. Thus the protein translated from the coding regions can be compared on a bit by bit basis with the regulatory site of the inhibitor and enhancer sites, and the degree of matching can be measured. As in [2], the comparison was implemented by an XOR operation, which results in a "1" if the corresponding bits are complementary. Each translated protein is compared with the inhibitor and enhancer sites of all the regulatory genes in order to determine the degree of interaction in the regulatory network. The influence of a protein on an enhancer or inhibitor site is exponential with the number of matching bits. The strength of enhancement *en* or inhibition *in* for gene i with $i = 1, \ldots, n$ is defined as

$$en_i = \frac{1}{v} \sum_{j=1}^{v} c_j e^{\beta \left(u_{ij}^{+} - u_{\max}^{+} \right)} \text{ and} \tag{1}$$

$$in_i = \frac{1}{w} \sum_{j=1}^{w} c_j e^{\beta \left(u_{ij}^{-} - u_{\max}^{-} \right)}, \tag{2}$$

where n is the total number of regulatory genes, v and w are the total number of active enhancer and inhibitor sites, respectively, c_j is the concentration of protein j, β is a constant that fine-tunes the strength of matching, u_{ij}^{+} and u_{ij}^{-} are the number

of matches between protein j and the enhancer and inhibitor sites of gene i, respectively, and $u^{+}_{\max}$ and $u^{-}_{\max}$ are the maximum matches achievable (32 bits) between a protein and an enhancer or inhibitor site, respectively [2].

Once the *en* and *in* values are obtained for all regulatory genes, the corresponding change in concentration c for protein i in one time step is calculated using

$$\frac{dc_i}{dt} = \delta\,(en_i - in_i)\,c_i\,, \tag{3}$$

where δ is a constant that regulates the degree of protein concentration change.

Protein concentrations are updated and if a new protein concentration results in a negative value, the protein concentration is set to zero. Protein concentrations are then normalized so that total protein concentration is always the unity. Parameters β and δ were set to 1.0 and 1.0×10^6, respectively, as previously reported [11].

The morphogen threshold activation sites provide reproducing cells with positional information as to where they are allowed to grow in the CA lattice. There is one site for each of the three orthogonal morphogenetic gradients described in Sect. 4. These sites are 9 bits in length, where the first bit defines the allowed direction (above or below the threshold) of cellular growth, and the next 8 bits code for the morphogen threshold activation level, which ranges from 0 to $2^8 - 1 = 255$. If the site's high order bit is 0, then cells are allowed to replicate below the morphogen threshold level coded in the lower order eight bits; if the value is 1, then cells are allowed to reproduce above the threshold level. Since in a regulatory gene there is one site for each of the orthogonal morphogenetic gradients, for each set of three morphogen threshold activation levels, the three high order bits define in which of the eight relative octants cells expressing the associated structural gene can reproduce.

5.2 Structural Genes

Structural genes code for the particular shape grown by the reproducing cells and were obtained using the methodology presented in [8]. Briefly, the CA rule table's output bits from the cellular growth model described in Sect. 3 were evolved by a GA in order to produce predefined 3D shapes. A structural gene is interpreted as a CA rule table by reading its bits as output bits of the CA rule. As mentioned in Sect. 3, at each time step of the CA run, an empty objective cell position can be occupied by an active cell (output bit $= 1$) depending on the configuration of the cells in the Margolus neighborhood block $(\eta_0, \ldots, \eta_6)$ and on the value of the parity bit p.

A structural gene is always associated with a corresponding regulatory gene, i.e. structural gene number 1 is associated with regulatory gene number 1 and its related translated protein, and so on. However, in a particular genome there can be less structural genes than regulatory genes; as a result, some regulatory genes are not associated with a structural gene and their role is only to participate in the activation or inhibition of other regulatory genes without directly activating a structural gene.

A structural gene was defined as being active if and only if the regulatory protein translated by the associated regulatory gene was above a certain concentration

threshold. The value chosen for the threshold was 0.5, since the sum of all protein concentrations is always 1.0, and there can only be a protein at a time with a concentration above 0.5. As a result, at most one structural gene can be expressed at a particular time step in a cell. If a structural gene is active, then the CA lookup table coded in it is used to control cell reproduction. Structural gene expression is visualized in the cellular growth model as a distinct external color for the cell.

6 Genetic Algorithm

Genetic algorithms are search and optimization methods based on ideas borrowed from natural genetics and evolution. A GA starts with a population of chromosomes representing vectors in search space. Each chromosome is evaluated according to a fitness function and the best individuals are selected. A new generation of chromosomes is then created by applying genetic operators on selected individuals from the previous generation. The process is repeated until the desired number of generations is reached or until the desired individual is found.

For the present work, chromosomes represent either the output bits from a CA rule table to be evolved to generate a simple form such a cube, or an ARN whose objective is to activate structural genes in a particular order to produce a multicolored shape such as a French flag pattern.

The GA in this paper uses tournament selection with single-point crossover and mutation as genetic operators. As in a previous report, we used the following parameter values [11]. The initial population consisted of 1000 binary chromosomes whose bit values were chosen at random. Tournaments were run with sets of 3 individuals randomly selected from the population. Crossover and mutation rates were 0.60 and 0.15, respectively. Finally, the number of generations was set at 50, as there was no significant improvement after this number of generations.

The fitness function used by the GA was defined as

$$Fitness = \frac{1}{k} \sum_{i=1}^{k} \frac{ins_i - \frac{1}{2} outs_i}{des_i} , \tag{4}$$

where k is the number of different colored shapes, each corresponding to an expressed structural gene, ins_i is the number of active cells inside the desired shape i with the correct color, $outs_i$ is the number of active cells outside the desired shape i, but with the correct color, and des_i is the total number of cells inside the desired shape i. The range of values for this function is $[0, 1]$ with a fitness value of 1 representing a perfect match.

7 Results

The GA described in Sect. 6 was used in all cases to obtain the CA's rule tables that made up the structural genes for specific simple patterns and to evolve the ARNs for the desired multicolored pattern. After an evolved genome was obtained, an initial

active cell containing it was placed in the center of the CA lattice and was allowed to reproduce for 60 time steps in the cellular growth testbed described in Sect. 3, controlled by the gene activation sequence found by the GA. In order to grow the desired structure with a predefined color and position for each cell, the regulatory genes in the ARN had to evolve to be activated in a precise sequence and for a specific number of iterations. Not all GA experiments produced a genome capable of generating the desired pattern.

In order to grow a solid 3D French flag pattern, three different structural genes were used. Expression of the first gene creates the white central cube, while the other two genes drive cells to extend the lateral walls to the left and to the right simultaneously, expressing the blue and the red color, respectively. These two last genes do not necessarily code for a cube, since they only extend a wall of cells to the left and to the right for as many time steps as they are activated and when unconstrained, they produce a symmetrical pattern along the x axis. The independent expression of these three genes is shown in Fig. 4. The two genes that extended the lateral walls were activated after a central white cube was first produced. In order to generate the desired French flag pattern, cells expressing one of these two genes should only be allowed to reproduce on each side of the white central cube (left for the blue cube and right for the red cube). This behavior was to be achieved through the use of genomes where the morphogen threshold activation sites evolved to allow growth only in the desired portions of the 3D CA lattice.

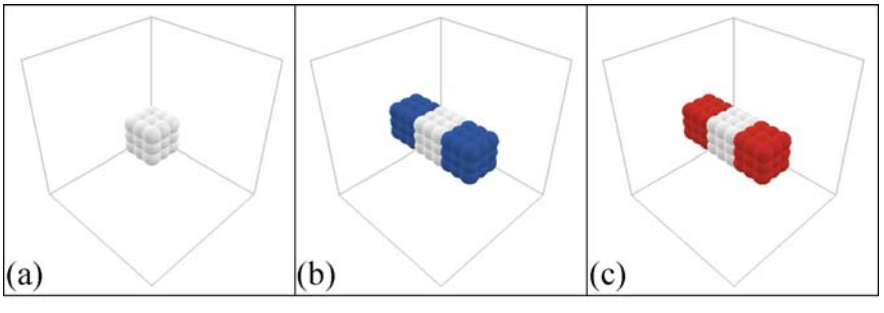

Fig. 4 Expression of the three genes used to create a 3D French flag pattern. **(a)** Create central white cube; **(b)** extend blue lateral walls; **(c)** extend red lateral walls. The last two genes were activated after the creation of a white central cube

However, when trying to evolve a genome to produce the 3D French flag pattern, it was found that the GA could not easily evolve an activation sequence that produced the desired pattern. Using the same approach as in [6], in order to increase the likelihood for the GA to find an appropriate genome, instead of using one series of three structural genes, a tandem of two identical series of three structural genes was used, for a total of six structural genes. In that manner, for creating the central

white cube, the genome could express either structural gene number 1 or gene number 4, and for the left blue and right red cubes, it could use genes 2 or 5, or genes 3 or 6, respectively. Thus, the probability of finding an ARN that could express a 3D French flag pattern was significantly increased.

Figure 5 shows a $9 \times 3 \times 3$ solid French flag pattern grown from the expression of the three different structural genes mentioned above. The graph of the corresponding ARN protein concentration change is shown in Fig. 5(e). Starting with an initial white cell (a), a white central cube is formed from the expression of gene number 4 (b), the left blue cube is then grown (c), followed by the right red cube (d). The evolved morphogenetic fields where cells are allowed to grow are depicted in the figure as a translucent volume for each of the three structural genes.

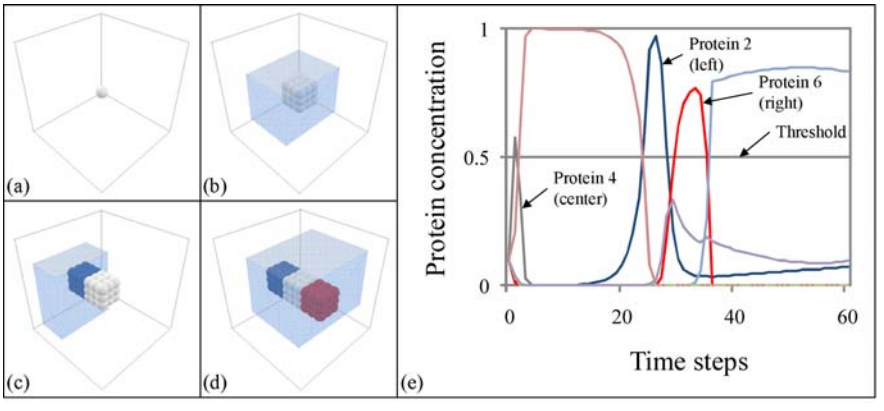

Fig. 5 Growth of a 3D French flag pattern. **(a)** Initial cell; **(b)** central white cube with morphogenetic field for gene 4 (cube); **(c)** central white cube and left blue cube with morphogenetic field for gene 2 (extend blue lateral walls); **(d)** finished flag pattern with morphogenetic field for gene 6 (extend red lateral walls); **(e)** graph of protein concentration change from the genome expressing the French flag pattern; the unlabeled lines correspond to proteins from regulatory genes that are not associated with structural genes

It is clear from the figure that for the genes that extend the wall of cells to the sides, the corresponding morphogenetic fields limited growth to the desired direction (left for blue cells and right for red cells). It should also be noted that the left blue cube is formed from the activation of the second gene from the first series of structural genes, while the other two genes are expressed from the second series of the tandem.

8 Conclusions

The results presented in this paper show that a GA can give reproducible results in evolving an ARN to grow predefined simple 3D cellular patterns starting with a

single cell. In particular, simulations showed that the combination of a GA and CA with a 3D Margolus interaction neighborhood was a feasible choice for modeling 3D pattern generation.

In general, the framework developed proved to be suitable for generating simple patterns, but more work is needed to explore generation of more complex structures. It is also desirable to study cellular structure formation allowing cell death and cell displacement, as in actual cellular growth. Furthermore, in order to build a more accurate model of the growth process, the use of a more realistic physical environment may be necessary. The long-term goal of this work is to study the emergent properties of the artificial development process. It is conceivable that highly complex structures will one day be built from the interaction of myriads of simpler entities controlled by a development program.

References

[1] Baker, R.W., Herman, G.T.: Celia - a cellular linear iterative array simulator. In: Proceedings of the fourth annual conference on Applications of simulation. Winter Simulation Conference, pp. 64–73 (1970)

[2] Banzhaf, W.: Artificial regulatory networks and genetic programming. In: Riolo, R.L., Worzel, B. (eds.) Genetic Programming Theory and Practice, ch. 4, pp. 43–62. Kluwer, Dordrecht (2003)

[3] Beurier, G., Michel, F., Ferber, J.: A morphogenesis model for multiagent embryogeny. In: Rocha, L.M., Yaeger, L.S., Bedau, M.A., Floreano, D., Goldstone, R.L., Vespignani, A. (eds.) Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems (ALife X), pp. 84–90 (2006)

[4] Bowers, C.: Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 149–158. Springer, Heidelberg (2005)

[5] Carroll, S.B., Grenier, J.K., Weatherbee, S.D.: From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design, 2nd edn. Blackwell Science, Malden (2004)

[6] Chavoya, A.: Cell pattern generation in artificial development. In: Rossi, C. (ed.) Brain, Vision and AI, In-Teh, Croatia, ch. 4, pp. 73–94 (2008)

[7] Chavoya, A.: Artificial development. In: Foundations of Computational Intelligence. Volume 1: Learning and Approximation (Studies in Computational Intelligence), vol. 8, pp. 185–215. Springer, Heidelberg (2009)

[8] Chavoya, A., Duthen, Y.: Using a genetic algorithm to evolve cellular automata for 2D/3D computational development. In: GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 231–232. ACM Press, New York (2006)

[9] Chavoya, A., Duthen, Y.: An artificial development model for cell pattern generation. In: Randall, M., Abbass, H.A., Wiles, J. (eds.) ACAL 2007. LNCS (LNAI), vol. 4828, pp. 61–71. Springer, Heidelberg (2007)

[10] Chavoya, A., Duthen, Y.: Use of a genetic algorithm to evolve an extended artificial regulatory network for cell pattern generation. In: GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, p. 1062. ACM Press, New York (2007)

[11] Chavoya, A., Duthen, Y.: A cell pattern generation model based on an extended artificial regulatory network. BioSystems 94(1), 95–101 (2008)
[12] Cickovski, T., Aras, K., Swat, M., Merks, R.M.H., Glimm, T., Hentschel, H.G.E., Alber, M.S., Glazier, J.A., Newman, S.A., Izaguirre, J.A.: From genes to organisms via the cell: A problem-solving environment for multicellular development. Computing in Science and Eng. 9(4), 50–60 (2007)
[13] Davidson, E.H.: The Regulatory Genome: Gene Regulatory Networks in Development And Evolution, 1st edn. Academic Press, London (2006)
[14] Devert, A., Bredeche, N., Schoenauer, M.: Robust multi-cellular developmental design. In: GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 982–989. ACM, New York (2007)
[15] Eggenberger, P.: Evolving morphologies of simulated 3D organisms based on differential gene expression. In: Harvey, I., Husbands, P. (eds.) Proceedings of the 4th European Conference on Artificial Life, pp. 205–213. Springer, Heidelberg (1997)
[16] Gordon, T.G.W., Bentley, P.J.: Bias and scalability in evolutionary development. In: GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 83–90. ACM, New York (2005)
[17] Harding, S.L., Miller, J.F., Banzhaf, W.: Self-modifying cartesian genetic programming. In: GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 1021–1028. ACM, New York (2007)
[18] Herman, G.T., Liu, W.H.: The daughter of Celia, the French flag and the firing squad. In: WSC 1973: Proceedings of the 6th conference on Winter simulation, p. 870. ACM, New York (1973)
[19] Joachimczak, M., Wróbel, B.: Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In: Bullock, S., Noble, J., Watson, R., Bedau, M.A. (eds.) Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 297–304. MIT Press, Cambridge (2008)
[20] Knabe, J.F., Nehaniv, C.L., Schilstra, M.J.: Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 321–328. MIT Press, Cambridge (2008)
[21] Kumar, S., Bentley, P.J.: On Growth, Form and Computers. Academic Press, London (2003)
[22] Lindenmayer, A.: Mathematical models for cellular interaction in development, Parts I and II. Journal of Theoretical Biology 18, 280–315 (1968)
[23] Lindenmayer, A., Rozenberg, G.: Developmental systems and languages. In: STOC 1972: Proceedings of the fourth annual ACM symposium on Theory of computing, pp. 214–221. ACM, New York (1972)
[24] Meinhardt, H.: Models of Biological Pattern Formation. Academic Press, London (1982)
[25] Miller, J.F., Banzhaf, W.: Evolving the program for a cell: from French flags to Boolean circuits. In: Kumar, S., Bentley, P.J. (eds.) On Growth, Form and Computers, pp. 278–301. Academic Press, London (2003)
[26] Reil, T.: Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In: Floreano, D., Mondada, F. (eds.) ECAL 1999. LNCS, vol. 1674, pp. 457–466. Springer, Heidelberg (1999)

[27] Turing, A.M.: The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 237(641), 37–72 (1952)
[28] Tyrrell, A.M., Greensted, A.J.: Evolving dependability. J. Emerg. Technol. Comput. Syst. 3(2), 7 (2007)
[29] Wolfram, S.: Statistical mechanics of cellular automata. Reviews of Modern Physics 55, 601–644 (1983)
[30] Wolpert, L.: The French flag problem: a contribution to the discussion on pattern development and regulation. In: Waddington, C. (ed.) Towards a Theoretical Biology, pp. 125–133. Edinburgh University Press, New York (1968)
[31] Wu, P., Wu, X., Wainer, G.A.: Applying cell-devs in 3D free-form shape modeling. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 81–90. Springer, Heidelberg (2004)