Epigenetic Tracking, a Method to Generate Arbitrary Shapes By Using Evolutionary-Developmental Techniques

Alessandro Fontana
IEEE
alessandro.fontana@ieee.org

Abstract

This paper describes an Artificial Embryology method (called "Epigenetic Tracking") to generate predefined arbitrarily shaped 2-dimensional arrays of cells by means of evolutionary-developmental techniques. It is based on a model of development, whose key features are: i) the distinction bewteen "normal" and "driver" cells, being the latter able to receive guidance from the genome, ii) the implementation of the proliferation/apoptosis events in such a way that many cells are created/deleted at once, in order to speed up the morphogenetic process. iii) the presence in driver cells of an epigenetic memory, that holds the position of the cell in the driver cell lineage tree and represents the source of differentiation during the development. The computer simulations performed with a number of 100x100 black and white and colour target shapes (the dolphin, the hand, the horse, the frog, the baby, the stomach, the french flag, the head, etc.) bring to the conclusion that the method described is able to generate any target shape, outperforming any other known method in terms of size and variety of the generated shapes. The interpretation of the proposed method as a model of embryogenesis and its biological implications are discussed.

1. Introduction and Related Work

This paper belongs to the field of Artificial Embryology and more specifically addresses the problem of morphogenesis. It describes a method, called "Epigenetic Tracking", to generate arbitrarily shaped 2-dimensional arrays of cells by means of evolutionary-developmental techniques, i.e. by evolving genomes that guide the development of the shape starting from a single cell. The paper is organised as follows: the rest of this section surveys the related work, section 2 describes the model of development, section 3 describes the simulations performed, section 4 discusses the biological implications, section 5 draws the conclusions and outlines future work.

The previous work in the field of Artificial Embryology (see (Kumar and Bentley, 2003; Stanley and Miikkulainen, 2003) for a comprehensive review) can be divided into two broad categories: the grammatical approach and the cell chemistry approach. The grammatical approach, originated by Lindenmayer (Lindenmayer, 1968), evolves sets of rules in the form of grammatical rewrite systems; the grammar can be context-free or context-sensitive and can utilise parameters; variations on this theme include using instruction trees or directed graphs in place of actual grammars. L-systems were employed as a means of describing the complex fractal patterns observed in nature and particularly the architecture of plants. The cell chemistry approach draws inspiration from the early work of Turing (Turing, 1952), who introduced a mathematical model of diffusion and reaction within a physical substrate. This approach attempts to mimic more closely how physical structures emerge in biology; cells are arranged in a physical space where simulated proteins can be sent as signals from one cell to another, as in nature.

Within the grammatical approach, Sims (Sims, 1994) used directed graphs to evolve the body morphologies and neural networks of artificial creatures in a simulated 3d physical world; in these graphs, a node represents a body part and an edge specifies how body parts are connected. Using a domain similar to Sims', Hornby and Pollack (Hornby and Pollack, 2002) applied L-systems to the simultaneous evolution of the body morphologies and neural networks of artificial creatures in a simulated 3d physical environment. Cangelosi, Nolfi and Parisi (Cangelosi et al., 1994) devised a model of neural development which includes cell division and cell migration in addition to axonal growth and branching. Gruau's Cellular Encoding (Gruau, 1994) uses grammar trees to encode steps in the development of a neural network starting from a single ancestor cell; the grammar tree

contains developmental instructions at each node.

Within the cell chemistry approach, Random Boolean Networks (RBN's) were originally developed by Kaufmann as a model of genetic regulatory networks (Kauffman, 1969); in the context of the development of multi-cellular organisms, the attractors of RBN's are interpreted as the different "cell types" of the organism. Dellaert and Beer (Dellaert and Beer, 1996) presented models of development to evolve functional autonomous agents, complete with bodies and neural control systems. De Garis (De Garis, 1999) developed a model for evolving shapes in 2d reproductive cellular automata; the model was successful in evolving convex shapes but non-convex shapes (e.g. the L-shape) presented a problem. Bongard and Pfeifer (Bongard and Pfeifer, 2001) proposed a minimal model of ontogenetic development to evolve both the morphology and neural control of agents that perform a block-pushing task in a physically-realistic, virtual environment. Miller and Banzhaf (Miller and Banzhaf, 2003) developed artificial organisms (the french flag) based on a method called Cartesian Genetic Programming, which evolves a developmental program inside cells. Eggenberger (Eggenberger-Hotz, 2004) used artificial cells endowed with genetic regulatory networks to evolve and develop simulated creatures; by using developmental mechanisms such as asymmetric cell division, genetic regulation, cell adhesion and physical interactions between cells, he achieved to shape multicellular (moving) 3d organisms.

The task of evolving predefined arbitrary shapes has as yet proved to be a difficult one: the evolved shapes reported in the literature are often very simple and of small size (e.g. the french flag). The method described in this paper seeks to overcome these limitations and provide a general solution to the problem of generating arbitrary shapes with arbitrary size.

2. The Model of Development

In our model of development the phenotype of the organism is represented as a 2-dimensional array of square-shaped cells, being each cell associated to a position on a grid. The development starts with a single cell placed in the middle of the grid, and unfolds in n development phases, counted by the variable "global development phase" (GDP) that runs from 0 to n-1 (n is a parameter). The term "global" refers to the fact that the variable GDP is shared by all cells (and therefore it can be considered the global "clock" of the organism). To each cell four variables are associated:

- a flag indicating whether the cell is "driver" or "normal";
- the "genome", organised as an array of "development operators", which is identical in all cells;
- the "cell epigenetic type" (CET), organised as an array of n integers (n is the number of development phases), which is not identical in all cells; the CET is present only in driver cells;
- an integer representing the cell's colour. In the current implementation four colour values are foreseen (0,1,2,3), an extra value (-1) indicates cell absence;

Cells belong to two categories: "driver" cells (coloured in yellow in the figures) and "normal" cells (coloured in orange or blue). The basic difference between driver and normal cells is that the first can be instructed by the genome (by means of an operator whose left part matches the CET value of the cell) to proliferate or induce apoptosis in the surrounding area. Figure 1 shows an example: a driver cell associated to a CET value labelled with "A"(called "mother cell") proliferates in an area around it (called "development area", delimited by a dotted line in the figure). While proliferating, it mostly generates normal cells (which fill the development area) and other driver cells, which are much fewer in number and "dot" the development area.

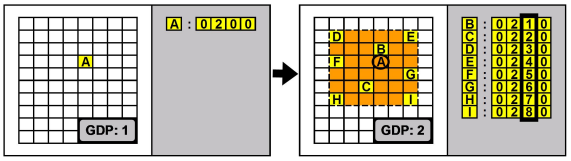

Figure 1: Proliferation. A driver cell proliferates in an area around it, generating normal and driver cells.

A key point is the assignment of the CET values on the newly created driver cells. To each new driver cell a new CET value is assigned, starting from the mother cell's CET value (the array [0200] in the figure, labelled with "A") and adding 1 to the value of the i-th position of the array at each new assignment, where i is the current GDP value (2 in the figure, the column corresponding to the GDP value is highlighted with a thicker border); with reference to the figure, the new driver cells are assigned the values [0210],[0220],... , labelled with "B","C", etc. In practise the variable CET holds the position of the driver cell in the driver cell lineage tree: this ensures that the new CET values are all different from the mother's value and from each other. Whether one of these new CET values will become the centre of another proliferation event depends on the presence in the genome of an operator whose left part matches such value.

The genome as we said is organised as an array of development operators (see figure 2). Excluding mutations (not dealth with in this work), the genome

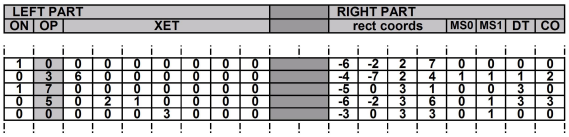

Figure 2: The genome.

is not modified during the development and is identical in all cells. Each development operator has a left part and a right part. The left part consists of a variable called XET, having the same structure of the variables CET: if the XET value is equal to the CET value of a given driver cell, the operator is activated and the relevant code specified in the right part is executed for that cell; the XET value is preceded by two parameters, one (OP, "order of precedence") indicating which operator takes precedence in case of multiple matches (XET values are not guaranteed to be unique) and one (ON) indicating whether the operator is "structurally" inactive or not. The right part of the operator has:

- a field with the coordinates of the rectangle which delimits the development area (row and column values of the north-west and south-east corners of the rectangle)
- a field holding a "master switch" (MS0) that defines the shape of the development area ("rectangular" -value = 0, "diagonal left" -value = 1, "diagonal right" -value = 2)
- a field holding a second "master switch" (MS1) that defines the type of "development event" that is going to occur ("proliferation" -value=0, "apoptosis" -value=1).
- a field with a parameter (DT) that specifies the thickness of the diagonal (valid only if MS0=1 or 2)
- a field with a parameter (CO) that specifies the colour of the newly created cells (both normal and driver)

In case of proliferation the development area is filled with newly created cells: most of the cells generated are normal cells, some are driver cells. The driver cells are much fewer in number (usually a "linear normal to driver ratio" of 5 has been used, corresponding to a 2-dimensional ratio of $5 \cdot 5 = 25$) and are deployed evenly on the development area (the precise algorithm to place the driver cells in not important as long as it ensures a uniform distribution). In case of apoptosis, all the cells contained in the development area "die", i.e. are deleted from the grid. The different types of development events (proliferation in different shapes -rectangular, diagonal left and right- and apoptosis) can be regarded as "painting primitives", i.e. basic painting actions that can be combined together to yield more complicated shapes. At the end of all development events the mother cell is always removed from the grid.

A special procedure is required if the development area is not empty. In this case the cells present must be either moved to other locations in the grid or removed altogether (overwritten). The solution chosen consists of first removing the cells present in the development area, carrying out the proliferation and finally redeploying the cells removed to the first empty positions available, starting from the position of the mother cell and going outwards (this procedure is carried out for each quadrant of the development area: first NW, then NE, SE and finally SW).

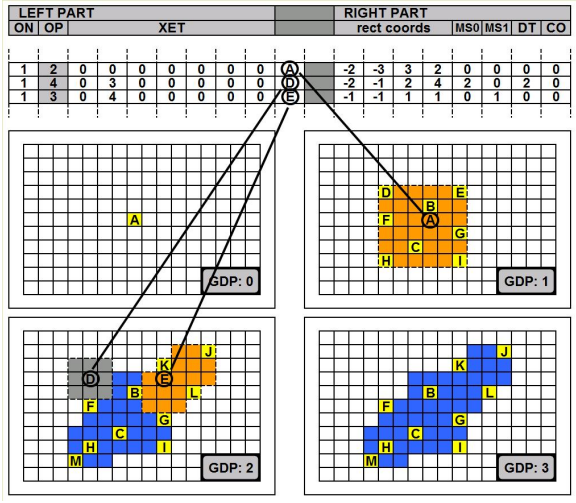

Figure 3: Example of development in four phases, driven by three operators (normal newly generated cells in orange, normal old cells in blue, driver cells in yellow). All snapshots are taken at the end of the relevant phase.

Figure 3 shows an example of development in four development phases (GDP=0,1,2,3) steered by three development operators, the first (a rectangular proliferation) triggered by the CET value labelled "A" in phase 1, the second (a rectangular apoptosis) triggered by the CET value labelled "D" in phase 2, the third (a diagonal right proliferation) triggered by the CET value labelled "E" also in phase 2. The CET value "A" was present at the end of phase 0, The CET values "D" and "E" have been created in phase 1. This example illustrates the "core" of the machine: a CET value produces a development event, which in turn produces other CET values, some of which produce other development events and so on, in an indefinitely sustainable way.

Before presenting the outcomes of the experiments performed, we wish to highlight the key features of

the model of development described. The first key feature of our model is the distinction between normal cells and driver cells. Only driver cells have a CET value and can therefore be instructed to develop (proliferate or die) by the genome; they represent the scaffolding, the backbone of the developing shape and make it possible to steer the development of the whole shape by acting on a small subset of cells. If all cells (both driver and normal) had an associated CET value, the space the GA would have to search would be unmanageable.

The second feature is the implementation of the development events of proliferation and apoptosis in such a way that they create/delete many cells at once (instead of one). This increases the power of the single development event, allows a reduction of the number of development operators needed to generate a given shape and has the ultimate effect of considerably speeding up the morphogenetic process. Together with the previous one, this feature serves the purpose of reducing the number of cells the genome has to steer, at the same time endowing such cells with a capacity to more profoundly influence the course of the development.

The third feature of the model is the explicit presence of an epigenetic memory, i.e. a cell variable (the CET, only present in driver cells) that takes different values in different cells and represents the source of differentiation during the development, leading different cells at different times to reading out and executing different portions of the genome. It is by means of the cell epigenetic type that driver cells know what type of cells they are and what their behaviour has to be like.

The fourth feature is the mechanism of assignment of the CET values on the newly generated driver cells during a proliferation event. Just as a mother gives each of her newborn babies a distinct "name", such mechanism ensures that each new driver cell is assigned a new, previously unseen CET value; the CET value, corresponding to the position of the driver cell in the driver cell lineage tree, represents the link by which these driver cells in subsequent phases can be picked up by the genome and given other instructions to be executed. If new driver cells were not guaranteed to have a distinct name, the genome would not be able to pick them individually: as a result, the developmental trajectories would be biased towards certain regions of the search space, making the development of arbitrary shapes hard.

The fifth feature, that descends from the latter two features, is the "looseness" of the driver cell lineage tree. It is worth pointing out that, when new CET values are generated in the course of a proliferation event, the mother cell does not "know" in advance if these CET values will activate a development operator in the genome: such an operator

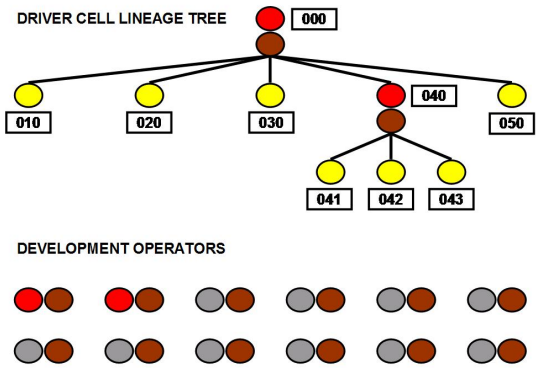

Figure 4: The driver cell lineage tree. the yellow circles represent CET values that do not match any XET value; the grey circles represent XET values that do not match any CET value; the red circles represent both CET values and XET values that match; the brown circles represent right parts of operators; the square boxes hold the CET values.

does not need to be there. Lacking this information, the most reasonable thing to do is to assign the daughter cells distinct names, thus creating the potential for a match to occur. The byproduct of this approach is that at any given moment in the course of evolution there will be many CET values that do not match any XET values during the development and, at the same time, many XET values that do not match any CET values: the presence of inactive driver cells and inactive operators is an unavoidable characteristic of the model proposed (figure 4 illustrates this concept) which, we think, contributes to the model's evolvability. We argue that this set of features is unique to our model and allows to reach a high level of performance in terms of both size and variety of the evolved shapes.

Our model has some similarities with L-systems. Both models have productions that replace existing symbols with other symbols: the key difference lies in the mechanism to generate new symbols. In an L-system the new symbols have to be listed explicitly, e.g.:

$$\mathrm{a} \rightarrow \mathrm{abbc}, \mathrm{b} \rightarrow \mathrm{cc} \tag{1}$$

In our model the production specifies only the number of new symbols (proportional to the dimension of the development area), while the symbols themselves (the CET values) are generated through a fixed procedure (i.e. such procedure never changes and therefore does not need to be encoded in the genome). This characteristic is particularly important in an evolutionary perspective, because it allows a more compact representation of the productions in the genome, able to generate many new symbols

nevertheless. Another important difference is that L-systems draw the symbols from a finite alphabet, while in our case the alphabet is virtually unbounded (an array of size 8 of 10-valued scalars has 10^8 possible values). We believe that this "unboundedness" paves the way for an open-ended evolution.

Gruau's Cellular Encoding has a cell variable similar to the CET, but lacks the distinction between driver and normal cells. Due to this lack, each cell needs to be individually guided by the genome to develop, leaving the genetic algorithm with the impossible task to evolve as many operators as are the organism's cells (in the case of the human body, the genome would contain 10^{14} genes, instead of the $3 \cdot 10^4$ it seems to have): for this reason, a CE-based system would not scale well to organisms with many cells. Moreover, in Gruau's model the cell state variable (the reading-head) is not a variable with an independent existence: it is a pointer on the tree of developmental instructions; such tree constitutes the main structure and a development operator cannot exist unless it coincides with a node of the tree. Our model on the contrary has a more flexible structure: there can be operators that are activated during the development and operators that are never activated, drivers cells that activate an operator and drivers cells that never activate any operator. Both the drivers cells and the operators can move freely (by means of mutations) between the "class of active elements" and the "class of inactive elements".

CA-based models of development also have a cell state variable and again the key difference lies in the mechanism of assignment of such value. While in CA-based models the value of the cell state is determined by the states (values of the same variable) of the neighbouring cells, in our model it is assigned to cells when they are created, during a proliferation event. Of course this is not the only difference: in CA models there is no distinction between driver and normal cells either, etc. Actually, the main idea captured by CA-based models, which is intercellular communication, is not incompatible with our model. We can foresee an extension of it to include the influence of surrounding cells as determinants of cell behaviour, together with the CET value.

In cell chemistry-based models of development the analog of the cell epigenetic type in our model is represented by the set of concentrations of certain chemicals which vary from cell to cell and trigger the activation of different genes in different cells: in this case the difference is again in the mechanism to generate new values. In cell chemistries no mechanism is present that guarantees that such "memories" are assigned different values in different cells; in addition, as the concentrations of chemicals can be influenced by many sources and are therefore quite "volatile" as memories, they are likely to change in the course of development, whereas the CET value is conceived as a non-volatile memory that, unless a mutation occurs, maintains its value during the entire development. A biologically plausible feature that many cell chemistry-based approaches possess is the presence of a genetic regulatory network, that allows cell behaviour to be determined by the interaction of many genes: also this idea is not incompatible with our model and it is in our plans to include it in future work.

3. Simulations

The model of development described in the previous section has been tested on the problem of morphogenesis achieved by means of evolutionary techniques, i.e the task is to generate predefined 2-dimensional shapes by evolving genomes that guide the development of the shape starting from a single cell; the implicit assumption behind this choice is that a model of development is good if it is evolvable. The experimental procedure consists in evolving a population of genomes, at each generation letting the development unfold for each memory (starting from a single cell with CET = [0,...,0] placed in the middle of the grid and running GDP from 0 up to a maximum value), and then using the adherence of the shape at the end of the development to the target shape as fitness measure. The genetic population is composed of 600 individuals (represented as strings of quaternary digits), undergoing elitism selection for up to 20000 generations. GA parameters are 50% single point crossover, mutation rate of 0.1% per digit. The fitness function formula is the same adopted by H. de Garis (De Garis, 1999):

$$F = (ins - outs)/des \tag{2}$$

where ins is the number of cells of the evolved shape falling inside the target shape, outs is the number of cells of the evolved shape falling outside the target shape, des is total number of cells of the target shape. Other important parameters are (relevant reference values in parentheses):

- number of development phases (8)
- max no. of operators active in each phase (8)
- normal to driver cells ratio in proliferation (25)
- max linear dim. of development area (10-18 cells)

As the number of possible epigenetic types (and hence the search space of the genetic algorithm) can be very large, the evolutionary process can become very slow. In order to overcome this problem, we introduced a mechanism called "germline penetration"

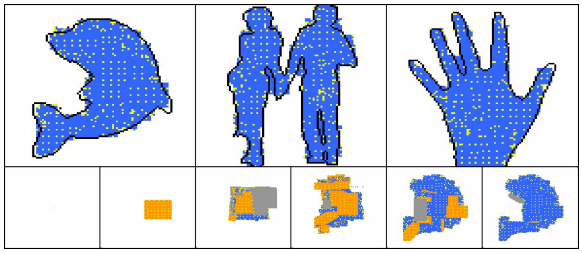

Figure 5: The dolphin, the couple and the hand (dynamical view, target contour superimposed). In the lower part the development sequence of the dolphin.

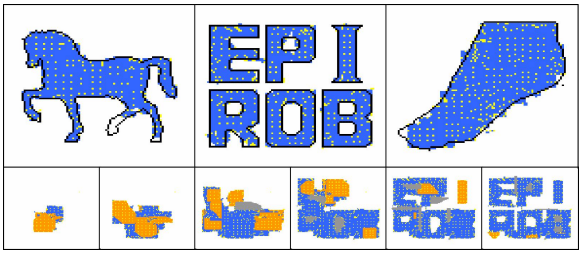

Figure 6: The horse, "EPIROB" and the foot (dynamical view, target contour superimposed). In the lower part the development sequence of "EPIROB".

statistics	dolphin	couple	hand
image size	100x100	100x71	100x100
generations	1580	5700	7280
fitness value	90.84	88.05	92.15
dev. operators	64	64	64
operators used	27	26	36
CET's created	471	497	520
genome size	5635	5635	5635

statistics	horse	EPIROB	foot
image size	84x100	100x100	100x100
generations	10720	11720	1560
fitness value	88.93	93.04	94.00
dev. operators	64	128	64
operators used	34	37	28
CET's created	332	701	467
genome size	5635	11395	5635

by which (a subset of) the CET values that have occurred during the development of any given individual are copied into the XET variables of the operators in the genome as a "suggestion" for the genetic algorithm. The rationale behind this measure is the consideration that the only way to change the course of development of an individual is to introduce in the genome development operators acting on CET values that have occurred in (and hence have determined) the current development; otherwise most XET values in the left parts of the development operators would be wasted on CET values that never occurred. To avoid disrupting the current developmental path, the operators with the newly introduced epigenetic types are set as structurally inactive.

For each development phase two views are possible for the developing shape: one that shows the "dynamics" of the development operators, and one that shows the colours of cells. In the "dynamical view" driver cells are coloured in yellow, normal cells are coloured in orange if they have been just (i.e. in the current development phase) created, in blue if they have been created in one of the previous phases; areas where cells have been deleted by an apoptosis event are coloured in grey. In the "colour view" (which of course makes sense only for colour targets) cells are shown with their actual colours. In addition the contour of the target shape has sometimes been superimposed.

Simulations have been conducted with a number of different target shapes; the targets have been chosen with the objective of testing the method on shapes as diverse as possible, to prove its effectiveness in generating any kind of shape. All targets are 100x100 multi-cellular arrays: the limited computational resources available prevented us from putting to a test larger shapes. Figures 5-9 show the results of simulations conducted with nine black-and-white targets (the dolphin, the couple, the hand, the horse, "EPIROB", the foot, the frog, the baby, the stomach) and two colour targets (the french flag and the head).

As we can see, all target shapes have been approximated to a good degree, with the exception perhaps of the frog. This is due to the fact that the frog's forelimbs are very thin (they contain a small number of cells), which raises the chances that a proliferation event creates cells falling outside the target shape, fact that is penalised by the fitness function; colour targets have proved more difficult to evolve, as one may have expected. To our knowledge, no other method is able, by means of evolutionary techniques, to generate target shapes with this size and variety; the french flag presented in (Miller and Banzhaf, 2003), for example, has a size of 16x16. Furthermore, the proposed method can be easily extended to the generation of 3d shapes.

4. Biological Interpretation and Possible Applications

The method presented can be interpreted as a model of embryogenesis. In this view the driver cells play the role of Spemann's organisers; if a driver cell destined to give rise to a certain shape part is moved to a different position of the growing shape, that shape part will grow in the new, ectopic position, fact which is consistent with the experimental obser-

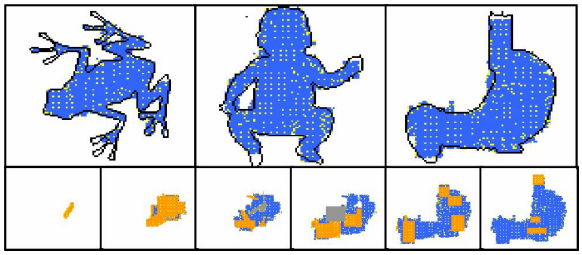

Figure 7: The frog, the baby and the stomach (dynamical view, target contour superimposed). In the lower part the development sequence of the stomach.

statistics	frog	baby	stomach
image size	100x84	100x100	100x88
generations	10080	2320	1040
fitness value	82.36	84.44	90.06
dev. operators	128	64	64
operators used	26	27	25
CET's created	411	444	420
genome size	11395	5635	5635

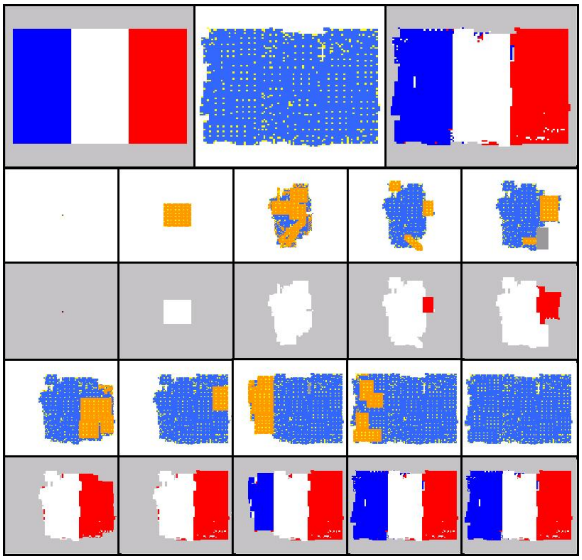

Figure 8: The french flag. In the upper part, on the left the target shape, in the middle the best evolved shape in dynamical view and on the right the best evolved shape in colour view. In the lower part some development phases.

vations. The biological counterpart of the cell epigenetic type could be implemented in real cells by means of either epigenetic (methylation patterns) or genetic (nucleotide sequences) mechanisms. In either case it would allow the biological equivalent of a driver cell to activate a gene without resorting to an external source, by providing the "first transcription factor", which gives origin to the whole cascade of gene activations that generate the transcriptome; all other cells must be given the first transcription factor from the outside (through inter-cell signalling).

The mechanism of germline penetration, which is very useful in reducing the search space of the GA, may well be present in nature and may be linked to the mobile DNA elements, whose presence is well documented and whose function is still largely unknown. This mechanism implements a sort of Lamarckian evolution, in that the developmental history of the organism influences the genome and therefore is passed on to the next generation. Speaking about features of the model proposed that lack biological plausibility, at present inter-cellular signals are not modeled and hence the local environment of a cell has no influence on cell fate determination, which is in contrast with the biological evidence. Moreover, in our model any development event is carried out by a single operator activated by a single CET value; in other words, we have a genetic regulatory network in which the outputs are connected directly to the inputs, with no "hidden layer". We know by contrast that in real cellular systems all events are determined by the interplay of many genes.

In molecular biology, junk DNA is a collective label for the portions of the DNA sequence of a genome for which no function has yet been identified (non protein- or RNA-coding DNA). According to the model of development proposed, operators having XET values that never show up during the normal embryonic development can be defined "junk" operators ("junk" XET values -the grey circles in figure 4). On the other hand driver cells (i.e. their CET values) that do not activate any operator during the development can be defined "junk" driver cells ("junk" CET values, -the yellow circles in figure 4); in other words the presence of junk XET values mirrors the presence of junk CET values. Following this line of thought, the presence of junk DNA could be motivated by the necessity to search for a match in the biological equivalent of the set of all the CET values generated in the course of development, a necessity that, as we have seen, is effectively supported by the mechanism of germline penetration.

If, in a successive generation, a mutation hits a junk XET value in such a way that it is now activated by a junk CET value, both the junk XET value and the junk CET value cease to be junk and give a contribution to the developmental process, that can go on and generate new lifeforms. On the other hand, if the embryonic development for whatever reason departs from normality and as a result new, unexpected CET values are created, development operators that would normally be junk can now become active. These considerations suggest a more blurred scenario for the junk DNA, in which a given development operator can be junk or not depending on the actual conditions encountered by the organism during its development.

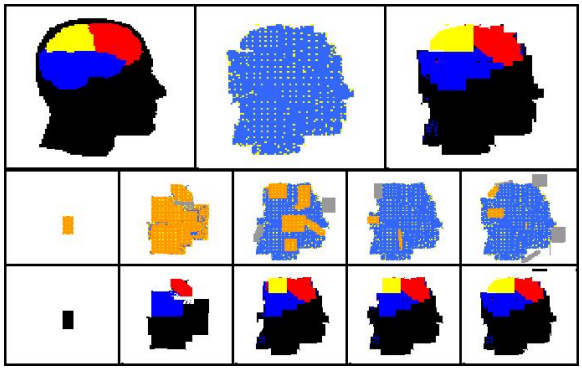

Figure 9: The head. In the upper part, on the left the target shape, in the middle the best evolved shape in dynamical view and on the right the best evolved shape in colour view. In the lower part some development phases.

In this work the method called "Epigenetic Tracking" has been applied to the problem of morphogenesis. The method has indeed quite general validity and is potentially suitable for any kind of developmental application. Potential foreseeable applications include evolvable hardware and developmental neural networks; in the field of developmental robotics it could be used to grow the bodies and/or the brains of the robots. More generally it can be extended by replacing the arrays of cells embedded in a physical 2 or 3-dimensional space through sets of objects embedded in an n-dimensional abstract feature space. Finally, it could find application in the field of nanotechnology, to guide the self-assembly of the basic components into functional structures.

5. Conclusion

We have presented a method to generate arbitrary shapes by using evolutionary-developmental techniques, that can also be interpreted as a model of embryogenesis. The method has been successfully tested on many black and white and colour target shapes, very different from each other, which brings us to conclude that the method has general validity and is capable of generating any kind of shape. Future work includes modelling of inter-cellular signalling and replacing the current direct input-output connections with a multi-layered genetic regulatory network.

Acknowledgements

I am much grateful to my friend Perry Petrella for helping me review this paper.

References

Bongard, G. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *Proceedings of The Genetic and Evolutionary Computation Conference, GECCO-2001*, pages 829–836. Morgan Kaufmann.

Cangelosi, A., Nolfi, S., and Parisi, D. (1994). A gene network model for developing cell lineages. *Artificial Life*, 5:497–515.

De Garis, H. (1999). *Artificial Embryology and Cellular Differentiation*. Academic Press.

Dellaert, F. and Beer, R. D. (1996). A developmental model for the evolution of complete autonomous agents. In (Eds.), P. M., (Ed.), *From animals to animats 4: Proc. of the 4th International Conference on Simulation of Adaptive Behavior*, pages 393–401. MIT Press.

Eggenberger-Hotz, P. (2004). Asymmetric cell division and its integration with other developmental processes for artificial evolutionary systems. In J. Pollack, M. Bedau, P. H. T. I. and (Eds.), R. W., (Eds.), *ALife9: Proceedings of the ninth international conference on artificial life*. MIT Press.

Gruau, F. (1994). *Neural network synthesis using cellular encoding and the genetic algorithm*. PhD thesis, Ecole Normale Superieure de Lyon, France.

Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8(3).

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.

Kumar, S. and Bentley, P. J. (2003). *On Growth, Form and Computers*. Academic Press.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–289.

Miller, J. F. and Banzhaf, W. (2003). Evolving the program for a cell: from french flags to boolean circuits. In Press, A., (Ed.), *On Growth, Form and Computers*.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. In Brooks, R. A. and (Eds.), P. M., (Eds.), *Proceedings of Artificial Life IV*, pages 28–39.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, v.9 n.2:93–130.

Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*, 237:37–72.