Generating Three-Dimensional Neural Cells Based on Bayes Rules and Interpolation with Thin Plate Splines

Regina Célia Coelho[1,2] and Osvaldo Vargas Jaques[2,3,4]

[1]UNIMEP-Universidade Metodista de Piracicaba – Rodovia do Açucar, km 156 – Campus
Taquaral – Piracicaba – 13400 – 911, SP, Brazil
rccoelho@unimep.br
[2]UFPR – Universidade Federal do Paraná, Rua XV de Novembro, 1299, 80060-000, Curitiba,
PR, Brazil
[3]UEMS – Universidade Estadual de Mato Grosso do Sul, Caixa Postal 6176, 13083-970
Dourados, MS, Brazil
ojacques@uems.br
[4]UNIGRAN – Centro Universitário da Grande Dourados, R. Balbina de Matos, 2121, 79824-900 Dourados, MS, Brazil.
ojacques@dourados.br

Abstract. In this paper the use of Bayes rules and interpolation functions is proposed in order to generate three-dimensional artificial neural cells incorporating realistic biological neural shapes. A conditional vectorial stochastic grammar has been developed to control and facilitate the parallel growth of branching. A L-parser (parser to L-Systems) has also been developed to guarantee that the grammar is free from mistakes before its use. This parser has also the function to generate a group of points corresponding to the morphologic structure of a neural cell. These points are visualized in a three-dimensional viewer especially developed to show the neural cell generated.

1 Introduction

The conventional theory-experiment basis for scientific research has been expanded to include computer simulation, which is an increasingly important component of many research lines. Indeed, one of the most important uses of computers in neuroscience is the simulation of neural systems due to the large amount of data that is necessary to be analyzed [1]. Neural modeling has a fundamental role in both experimental and theoretical studies to determine the characteristics of nervous systems. There is an extraordinarily large number of neural cells in our brain as well as a great variety of them. Thus, it is extremely difficult to understand completely how the communications between them occur and how a specific area of the brain responds to a specific stimulus. In this context, there are several studies in the sense of trying to understand the functionality of some neuron types. It is well known that the shape of the cells is a fundamental factor to define the communication between them and to define their respective functions. Consequently, it has been proved that as more complex is the function of the neuron more complex will be its morphology. It is also shown that cells can change their shape redirecting their dendritic branching to

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 675–682, 2003.

respond some factors acting on them, such as the death of some neighborhood cells or the presence of some attraction/repulsion factor). Therefore, the simulations related to the neural growth should take into account the shape and plasticity of the cells.

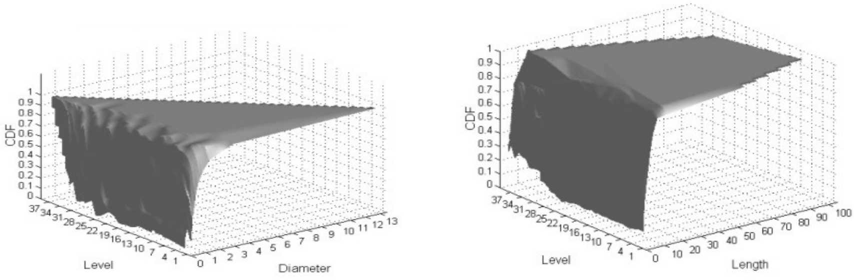
Hamilton [2], McCormick and Mulchandani [3], Ascolli and Krichmar [4] have published some works related to simulation of the neural growth. These authors suggest different methods for the growth of cells considering the shape of the natural cells. For instance, Hamilton [2] and Ascolli and Krichmar [3] proposed stochastic L-System and simulated the tropism. On the other hand, McCormick and Mulchandani [3] proposed the growth with stochastic behavior. However, any of these works take into account the historical of the growth or present a easy extension to the growth of different cells considering the shape of them in a neural structure.

The aim of this work is the synthesis of three-dimensional neural cells using interpolation functions and Bayes rules incorporated in stochastic graph grammars. Thus, the neural development considers the little historical about the growth and the shape characteristic that are incorporated by the interpolation functions. These functions are used as a way of generalizing several distribution functions obtained from extracted measures of natural cells, besides allowing the reduction of the amount of stored data. As will be showed, the generated cells present shape very similar to the natural cells. Two tools have been built to assist in this simulation: i) L-Parser, a parser of the graph grammar generated; ii) Neuron-Viewer, used to show the artificial cells.

2 The Use of Bayes Rule

The begging of the generation of a cell corresponds to the first level of growing. In this situation, in which there is not any previous level to consider, the measures to the artificial cells are obtained considering probability density function. However, very often the actual state of a cell depends on its previous state of growing (this is valid just after the cell ramified at least once). For instance, the diameter of the branch can be decreased with its growing. So, it is necessary to verify the previous diameter to determinate the actual, once that diameter has to be smaller than the previous one. The Bayes rule is used to allow this kind of implementation. First of all, it is necessary to obtain a set of shape measures of the natural cells. This work has concentrated on rat pyramidal cells (the files were acquired electronically in Canon [5]). The developed software to obtain the measures considers computer-acquired neuroanatomical files in Eutectic or SWC format [5]. Second, the measures considered in the generation of the artificial cells are calculated. The main shape measures considered are: number of primary branches, length and width of each dendritic segment and arc segment, and branching angles. These measures are organized according to the hierarchical level along the tree [6]. The next step is to estimate the probability density function and then the cumulative distribution functions (CDFs), based on the natural neural cells measures, which characterize the morphological properties of the neural cells. Except for the angles, each measure will generate a bivariated CDF whose random components correspond to the hierarchical level and the respective measure. The angles will generate a trivariated CDFs in which random components correspond to two measures (two angles), besides the hierarchical level. They are related to the torsion and curvature proposed in

McCormick and Mulchandani [3] work to generate three-dimensional cells. Two different groups of CDFs have been generated: one corresponding to the apical branches and another to the basal branches of the pyramidal cells. Fig. 1 presents two examples of CDFs created.



(a) (b)

Fig. 1. Examples of CDFs used in the generation of splines. (a) CDF of length dendritic segments by hierarchical level; (b) CDF corresponding of dendritic segment diameters by hierarchical level.

Now, it is necessary to use the Bayes rules to generate CDFs that considers the previous measure to calculate the actual measure. According to the Bayes rule, the probability of occurrence of a B specific event is affected by the fact of another event to have happened or not. Thus, it is necessary to calculate the occurrence of B conditioned to the previous occurrence of *A*, denoted by *P(B|A)* (probability of *B* given *A*) [7]. Its expression is given by:

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)} = \frac{P(A \cap B)}{P(A)} \tag{1}$$

The implementation was made considering the dependency of measures between two branching levels. Therefore, it was created a CDF of *B|A*, once that the occurrence of *B* depends on the *A*. After that, the calculated *B* values are used to find the *C* values, generating the CDF of *C|B*, and so on. In Fig. 2 is illustrated some examples of generated conditional CDFs. The conditional CDFs have been used to control the length, diameter and angles of dendritic segments. For example, the *n* segment length depends on the occurred length in the *n-1* segment. In the case of the angles, Bayes rules (or conditional CDFs) were used to control the two different angles considered in this work: torsion and curvature. Fig. 3 illustrates some examples of the graphic resulting of the application of Bayes rules to control these angles.

3 Obtaining the Interpolations

The interpolation functions were generated using the CDFs described in the previous section and they are applied in all CDFs. In this paper we have not used these functions to conditional CDFs, but the implementation can be also extended to include the interpolation function with conditional CDFs (this will be implemented in future works). There are several methods of interpolation functions. In this work we used Thin Plate Splines (TPS), which is a linear combination of radial basis functions.

In a TPS, the interpolating surface presents minimal deformation energy related to the known charge points, originating a smooth and continuous surface [8,9].

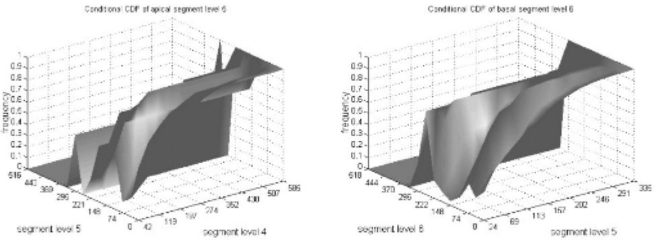


Fig. 2. Examples of conditional CDFs used in the generation of splines

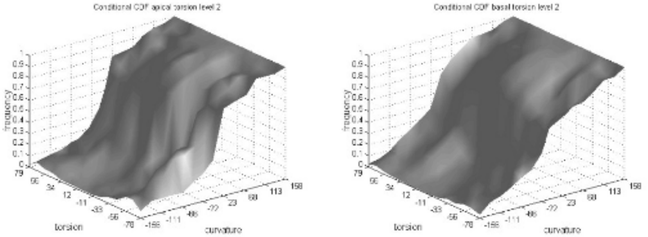


Fig. 3. Examples of conditional CDFs used to control the angles of dendritic segment.

Two groups of information are necessary for this interpolation: those known (in this case, the measures of the neural cells) and those we want to know (in this case, the measures that will be used in the generation of the cells in agreement with the probabilities and the level of each branch). The first group of information can be represented by a set of *n* vectors $V = \{\vec{v}_1, \vec{v}_2, ..., \vec{v}_n\}$, where $\vec{v}_i = (c_1, c_2, ..., c_d)$, $1 \leq i \leq n$ is a d-dimensional vector, and *n* is the number of samples of a given measure. Once chosen *V*, we have a set $F(V) = \{f(\vec{v}_1), f(\vec{v}_2), ..., f(\vec{v}_n)\}$, where $f(\vec{v})$ is a result of application on $\vec{v}$. We can describe the function interpolation as:

$$F(\vec{v}) = \vec{\alpha}^T \cdot \vec{v} + \beta + \sum_{i=1}^{n} w_i \Phi(\|\vec{v} - \vec{v}_i\|) \tag{2}$$

where $\Phi(x)$ corresponds to basic radial basis function, w_i is the contribution of the function $\Phi(x)$ in each distance *x*, β is a constant term corresponding to translation of the $\Phi(x)$ to fit it in the interpolation, and α is a vector used to the rotation of $\Phi(x)$ in each dimension. The function $\Phi(x)$ used in this work is defined by:

$$\Phi(x) = \begin{cases} 0 & if\ x = 0 \\ x^2 \log(x) & otherwise \end{cases} \tag{3}$$

By the equation (1) and using the known value of V and $F(V)$, we can calculate α, β, and *w*. After then, we use the equation (4) together with the calculated variables in (2) to estimate the unknown values of $F(\vec{x})$, being $\vec{x}$ the values of the measure for which we want to estimate $F(\vec{x})$.

$$F(\vec{x}) = \vec{\alpha}^{T} \cdot \vec{x} + \beta + \sum_{i=1}^{n} w_i \, \phi\left(\left\| \vec{x} - \vec{v}_i \right\|\right) \tag{4}$$

The axes were transposed to facilitate the implementation, such that, being provided the hierarchical level and a random number (between 0 and 1), the function returns a measure. Therefore, $F(\vec{v})$ in the equation (2) corresponds to a measure (for example, diameter), c_1 to the probability (CDF) and c_2 to the hierarchical level. In the equation (4), $\vec{x}$ is a vector with x_1 representing the probability and x_2, the hierarchical level.

4 Generating Neural Cells

To allow the parallel growth, it was created a stochastic graph grammar (a variation of L-Systems) that incorporates the interpolation functions and Bayes rules (or conditional CFDs). Some measures are controlled by interpolation functions (for example, length of dendritic segment and number of branches) and others, by Bayes rules (for example, angles and diameter). They are used to express variation on the growth pattern in terms of the current interaction of the generation process or external influences such as chemical markers or neurotrophic factors. The statistical functions incorporate the precise statistical characterization of the morphological features of the neurons, which can also take into account the stage of the generating process (previous and current stages).

A parser of L-system (L-parser) has also been developed in this work to compile and interpret the grammar. It allows the user to enter with the grammar (L-system) and the result is the object obtained by this grammar. It is very important to eliminate mistakes in the creation of the grammar before its use. This parser facilitates the definition of a specific grammar for each type of cell when is necessary to generate different types of cells in the same neural structure.

The probabilities of each grammar rule have been defined by the probability functions obtained from the measures extracted from biological cells. This means that there is not a fixed probability for each rule production, as commonly found in works that use stochastic L-systems. On the contrary, it is defined a probability function that allows varying a probability depending on the branching level of the branch. The statistical functions used were the CDFs described in section 2, which were incorporated in the grammar using polynomial approximation described in section 3.

It must be emphasized that the grammar results are strongly influenced by the interpolation functions that are related to the L-systems. These functions are used to avoid the necessity to work with several data files, once they generalize a specific measure for all the levels of branching of the cell. So, it is not necessary to have a file of measures for each level, but just the function that describes these measures as a whole.

The grammar below represents a summarized model of our proposal, which includes some control actions (lines 2, 3, 4, 6, 7, 7.1.1, 7.1.2 and 7.2.1). The execution of a graphic operation is conditioned to the logic value of the expression (lines 6.1, 7.1 and 7.2). In line 4 the ChooseSegmentLength (Level) chooses the dendritic segment length. This function uses the interpolation polynomial described previously referent to the dendritic segment length of each branching level.

```
Grammar:
1. Axiom : X;
2. Level := 1;
3 SumCp := 0;
4. SdLength := ChooseSegmentLength (Level);
5. X → $F_{LengthCp(Level)\ Curvature(Level)\ Torsion(Level)\ Diameter(Level)}$ Y
6. SumCp := SumCp+LengthCp(Level);
   6.1  [SumCp < LengthSd] : Y→X
7.  Bprob:=ChooseBranch(Level);
   7.1  [Bprob < BranchProb (Level)] :      Y → [-X][+X]
     7.1.1. Level := Level + 1;
     7.1.2.LengthSd =ChooseSegmentLength(Level);
   7.2  [Bprob > BranchProb (Level)] :      Y → NULL
     7.2.1 Level := Level -1;
```

Since the execution of its actions is not only depending on probabilistic functions, we call this model as *Conditional and Stochastic Grammar*.

In line 5, X is responsible for drawing a dendritic arc. The LenghtCp(Level), Torsion(Level), Curvature(Level) and Diameter(Level) functions are associated with the F symbol to give information about direction, length and thickness of the arc to be drawn.

The dendritic branching process starts in level 1 with the X axiom and the choice of the dendritic segment length (line 4). Note that there are several CDFs and thin plate splines associated with F in line 5, such as arc length, torsion angle, curvature angle and diameter. While the sum of arcs does not reach the chosen length segment (line 6.1), the branch continues to grow [10]. After this length is reached, it is verified if the branch will ramify or not (line 7). BranchProb(level), in line 7.1, verifies the corresponding value of branching probability in this level. If Bprob is smaller than the branching probability, then two new branches will start to grow starting, thus, a new dendritic segment level. Otherwise, no more new branching will appear in that branch and it will stop growing (line 7.2). The process continues recursively until all the conditions are satisfied (the entire branching stop growing).

Each generated neuron is different from any other previously produced because of the used statistical functions, but all cells present the same general characteristics related to neural shape.

Each neuronal dendrite is described as a series of small cylindrical compartments. Thus, these small compartments will express the tortuosity of the branches. Fig. 4 illustrates some examples of neural cells generated by the proposed method.

An environment of three-dimensional visualization has been generated using OpenGL tool. This environment allows the volumetric visualization, including illumination, transparency, texture, rotations, translations and changes of scale of the generated neural structures. It also includes a menu of options to the variation of visualization parameters such as color, light intensity, properties of the light and of

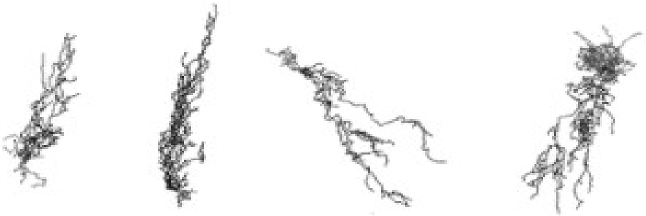


Fig. 4. Neural cells generated by the proposed methods.

the material of the object (in the case, of the neurons), transparency, wire frame visualization, etc, besides being able to include textures. One example of this environment can be seen in Fig. 5. It also allows defining how much percentage of a neuron will be drawn (100%, 90%, etc.). Fig. 6 shows some examples of this option. In those examples it is possible to see that the branches are growing in parallel.

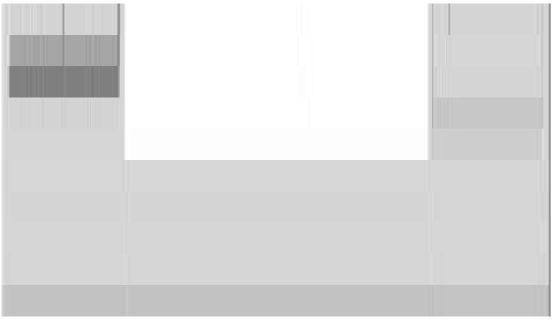


Fig. 5. Environment to visualize three-dimensional cells.

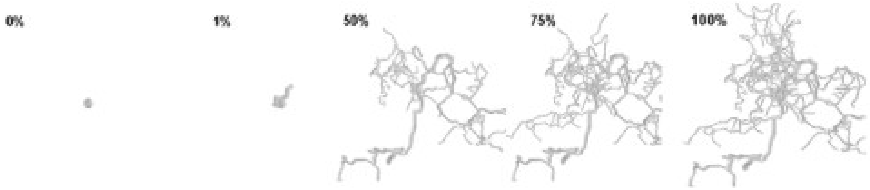


Fig. 6. Representation of three-dimensional cells drawn in different percentages**.**

5 Conclusion

The study of cells showed in this paper can provide important support to guide the simulation of more complex cells in the future, like Purkinjie cell.

It was shown in this work that the inclusion of Bayes rules in L-system allows that the generation of the current branch takes into account parameters of the previous stages. In addition, these rules allow relating two or more measures. For instance, we can consider that the diameter of the branch depends on its length. Therefore, it is possible to verify if there is some relationship among the measures or hierarchical levels. It is important to emphasize that this type of conditional statistics is already considered for the angles, once we have to work with two angles by the fact of we are generating three-dimensional cells. In this way, first of all we select a hierarchical level and a random number. Then, using the polynomial approximation of CDF, we can calculate an angle (curvature) and, given the curvature, we used Bayes to calculate the torsion. Additional studies have been initiated about the dynamical growth of the cell including the trophic fields corresponding to varying concentration of ions, chemoattractiors or electric fields.

Acknowledgments. Regina Célia Coelho is grateful to Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP-Brazil Proc. 98/05459-0) for financial help.

References

1. Seung, S. Neural Coding: Linear Models, 9.29 Lecture, 1, p. 1–6, 2002.
2. Hamilton, P. A language to describe the growth of neurites, Biological Cybernetics, 68, p. 559–565, 1992.
3. McCormick, B.H.; Mulchandani K. L-system modeling of neurons visualization. In: I Biomedical Computing Conference Proceedings, p. 693–705, 1994.
4. Ascoli, G.A.; Krichmar, J.L. L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology, Neurocomputing, 32-33, p. 1003–1011, 2000.
5. Cannon, R.C. http://www.cns.soton.ac.uk/~jchad/cellArchive/idex/toindex.
6. Coelho, R.C.; Costa, L.F. Realistic Neuromorphic Models and Their Application to Neural Reorganization Simulations, Neurocomputing, 48, 555–571, 2002.
7. Arango, H.G. Bioestatística teórica e computacional**,** Guanabara Koogan, 2001.
8. Mees, A.I.; Jackson, M.F.; Chua, L.O. Device Modeling by Radial Functions, IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, 39 (1), p. 19–27, 1992.
9. Costa, L.F.; César Jr, R.M.C. Shape Analysis and Classification, CRC Press, 2001.
10. Costa, L.F.; Cesar Jr., R.M.C.; Coelho, R.C.; Tanaka, J. S. Analysis and Synthesis of Morphologically Realistic Neural Networks, In R. Poznanski (Editor) Modeling in the Neurosciences: From Ionic Channels to Neural Networks, Gordon and Breach Science Publishers, 1999.