A Framework For Modeling Neuron Morphology

B. H. McCormick

Scientific Visualization Laboratory
Department of Computer Science
Texas A&M University
email: mccormick@cs.tamu.edu

K. Mulchandani

Scientific Visualization Laboratory
Department of Computer Science
Texas A&M University
email: kishorem@cs.tamu.edu

January 21, 1995

ABSTRACT

A representational framework for neuron morphology is presented that is adequate both for the quantitative description and stochastic generation of neuron populations. All models are presently limited to modeling neuron morphology as viewed at the limit of optical resolution. This object-oriented description of cells(their dendritic and axonal arbors, spines, and soma) forms the basis for the Data Repository. Drawing from the Data Repository the Data Visualizer allows a 3D interactive display of neurons, either sigly or in juxtaposition.

1 INTRODUCTION

A formal representation of neuron morphology, adequate for the geometric modeling of manually-traced neurons, is presented. *

The concept of a stochastic L-system is introduced and the critical distribution functions governing the stochastic generation of dendritic and axonal trees are defined. Experiments with various stochastic L-system models for pyramidal, motoneuron, and Purkinje cells are reported which generate synthetic neurons with promising proximity to neurons in the neurobiology literature. Work is in progress to improve this degree of proximity, but more importantly to validate

*Presentation Preference: Oral
Presentation Category: Modeling and Simulation
Theme: Cell Biology

the derived stochastic models against available databases of manually-traced neurons. To this end a neuron morphology modeler is described which provides a methodology for iterative refinement of the stochastic L-system model.

2 REPRESENTATION OF NEURON MORPHOLOGY

2.1 Terminology for neuron morphology

The constituents of a neuron are its soma, one or more dendritic trees (also called dendritic arbors), and its axon or axonal tree.(Figure 1)

Given a dendritic tree, three types of nodes are identified: (1) the *soma node* representing the point of attachment of the tree to the soma; (2) *branch points* at which the dendrite branches; and (3) *terminal nodes*, the endpoints of the tree. Starting at the soma node a unique path exists to any other node of the dendritic arbor.

A *dendritic segment* is the portion between two consecutive nodes in the dendritic tree. The *stem segment* has its origin at the soma node and is counted as the first order segment. Daughter segments arising from the first order branch point are called second order segments, and so on. A segment ending in a terminal node is called a *terminal segment*; all others are called *non-terminal segments*.

The *junction* at a branch point is called a *bifurcation* if the parent segment branches into two daughter segments, and a *multifurcation* if the junction gives rise to more than two daughter segments.

Axonal processes, including axonal arbors if present, are described by nodes, segments, and junctions in like manner, as discussed above for dendritic trees.

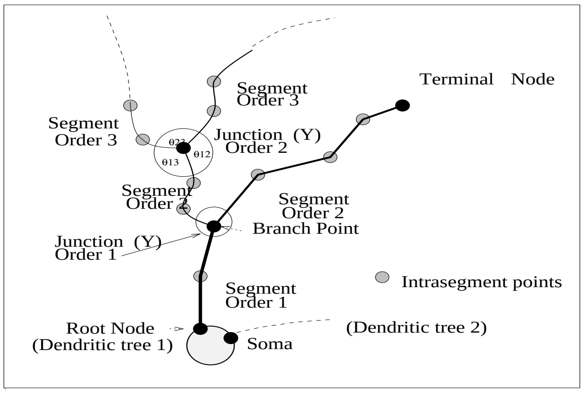


Figure 1: Representation of neuron morphology

2.2 OBJECT DESIGN FOR MORPHOLOGICAL DESCRIPTION

Our object-oriented description of dendritic or axonal arbors defines two basic classes: segments and junctions, whose instances are interlinked at nodes(a third class) to form the overall neuron morphology. Each segment is thus representable as a generalized cylinder with collection of points along its trajectory and diameters at those points. A segment can be a terminal segment or it could bifurcate(multifurcate) to produce a junction and two or more daughter segments. The junction owes its existence to three(or more) segments which share the branch node. These segments can be identified as the base segment, the primary, secondary (and possibly tertiary) segments. The intermediate angles between the segments are also stored for the junction.

Some important characteristics of the soma for modeling at the level of optical resolution are its shape, its volume and the number of first order segments (axonal or dendritic) that arise from it. By measuring the points along the soma cross-section borders and stacking up the resulting polygons, one can coarsely approximate the soma shape and also compute the volume. The accuracy and precision of these soma attributes are a function of the number of stacks and slices used. Though the soma may also be thought of as a special node or a special multifuraction junction, due to the special significance it has in the neuron function we have kept it as a separate class of its own.

The spines are also known to have different shapes depending on the different synaptic properties. The shape of a spine is too small to distinguish at the current limit of optical resolution and thus our spine class is presently a virtually empty class. We have provided it as a spaceholder class.

The Cell class has its physiological type, its parent organism, its position in the organ, its sources and other text fields carrying information about the experiment and the researcher that carried out the study, as the current attributes.

This description of neuron morphology is intuitive and brings all the benefits of the object-oriented paradigm, which include portability, extensibility, and reusability.

Building on these three basic classes, we have devised a six class model (Segment, Junction, Node, Soma, Spine, Cell) which can be used as a canonical model for morphological measurements. A segment, as defined above, is the piece of dendrite or axon between two node points, where the nodes are either bifurcation (multifurcation) points or origin points. With additional classes for the Soma, the Spine and the Cell, the complete morphological information of a neuron is captured as the attributes and relationships of the object instances of these six classes.

3 DATA ACQUISITION

As shown in the Figure 3) the neuron morphology data is captured through two different sources namely the actual measurements of the manually traced neurons and the synthetically generated neuron coordinates.

3.1 Manually-Traced Neurons

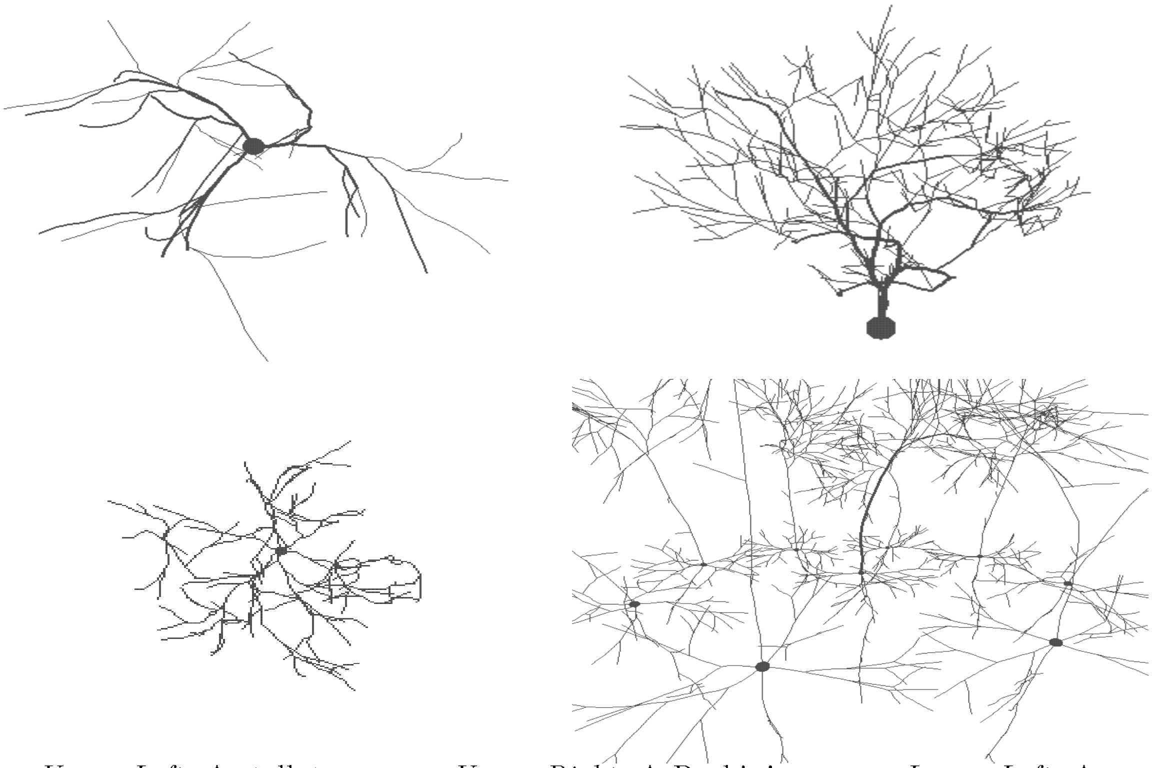
Different neuron-tracing equipments and software are available that allow the neuroscientists to trace the neurons. These typically output a series of points along the segment trajectory with the segment width at these points. The output files from these systems can be read into the data repository as discussed later.

3.2 Synthetically-generated Neurons

Artificial neuron morphologies are generated using the L-system grammars.[1] An L-system grammer generates a parenthesized string of terminal symbols. Each terminal symbol serves to orient or generate a geometric model of the constituent part of the neuron. For modeling dendritic or axonal trees, the principal constituent parts are segments, which are realized as generalized cylinders and junctions. A neuron generator is utilized for this purpose.

4 DATA REPOSITORY AND VISUALIZER

The Data Repository is a collection of a relational database, converter programs that convert to and from different formats to the object model format. A visualizer that can read the object



Upper Left: A stellate neuron *Upper Right:* A Purkinje neuron *Lower Left:* A motoneuron *Lower Right:* A Pyramidal forest

Figure 2: Synthetically-Generated Neuron as taken from [1]

model format of a neuron morphology and displays it in 3D is also provided. The relational database table format, the manually-traced neuron data format and the L-system synthesized data format can all be converted into the object-model format and vice versa.

4.1 Relational Database

The object model, a volatile memory model, is made persistent by converting it into a nearly identical relational model for widespread distribution amongst researchers. The links between segments and junctions are replaced by reference fields or Segment and Junction IDs. Objects are downloaded into tables, which instead of volatile pointer links, now carry the IDs of the Segments or Junctions the links were attached to. By assigning IDs, it is also possible to refer to unique pieces of morphology. For example, a researcher refering to an unusual spine density of a segment can identify the exact segment or segment range by specifying the appropriate segment ID. Similarly junctions, neurons, somas and spines are assigned IDs and form the keys of their respective relations (or tables).

Seven table formats have been designed: two storing segment-related information, one for junctions and one each for nodes, somas, spines, and cells. The design is flexible and can be easily extended.

4.2 Spreadsheet Package

The tables in the relational database can be exported into a spreadsheet package or imported from it. The Spreadsheet provides a

4.3 Visualization

Display of neurons (manually-traced or synthetically generated) in 3D uses the principal application programming interface in graphics (OpenGL, Silicon Graphics, Inc.) running on either a Windows NT based personal computer or an Iris or Onyx graphical workstation.

The input format for the Visualizer is the object model. All other formats can be converted into the object model file format. This file format is then loaded into the visualizer program, which creates the same volatile in-memory object-oriented model as referred to above. The visualizer allows a 3-D interactive display of the neuron. A cut-and-paste feature is available that allows the user to cut and paste subtrees creating new morphologies for experimental analysis. The modified neuron can then be dumped back into the object-model file format. It may then be converted into the relational database format for permanent storage.

Additional visualization software developed by Brent Burton, Texas A&M University, changes the stick figures of *lsys* into 3D volumes. Burton is also developing a navigator to traverse graphical models of brain nuclei and other neural tissue as viewed in animation. This visualization facility requires the Onyx workstation.

5 NEURON MORPHOLOGY MODELER

Experiments with various stochastic L-system models of neuron morphology have given us neurons with promising proximity in shape to neurons illustrated in the neurobiology literature. Work is in progress to improve this degree of proximity, but more importantly to validate the L-system neuron models against databases of manually-traced neurons. To this end the Neuron Morphology Modeler (NMM), described below, has been devised to provide for iterative refinement of L-system models (Figure 3).

5.1 Data level of the modeler

Central to the scheme at the data level is an *object-oriented neuron morphology data repository* (Paradox for Windows, Version 5.0, Borland International, Scotts Valley, CA). This DBMS stores for each neuron a representation of its soma and for each dendritic or axonal tree a representation of its segments and junctions, as described above.

This DBMS can receive input (at the data level) from the *neuron tracer* (Neuron Tracing System, Eutectic Electronics, Inc., Raleigh, NC) or (at the knowledge level) from the L-system *neuron generator* (a derivative of public-domain software known as *lsys*, copyright (c) 1990, 1991 by Jonathan Leech, University of North Carolina). In this sense the DBMS is indifferent as to whether it is storing data from real manually-traced neurons or L-system-generated synthetic neurons.

Upon command, the DBMS can prepare database files for *the spreadsheet* program (Excel, Version 5.0, Microsoft Corporation). The spreadsheet program allows data editing and graphical summarization.

Spreadsheets, in turn, provide the conventional input to the *statistical analyzer* (Systat, Version 5.03, Systat Inc., Evanston, IL). In particular the statistical analyzer estimates the distribution functions for segment non-termination, segment length and trajectory generation, and the rotations characterizing bifurcations. At present only ad hoc techniques are available for using Systat for this purpose.

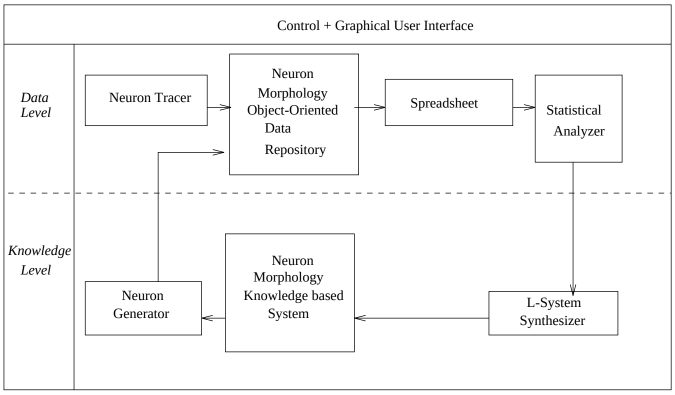


Figure 3: Neuron morphology modeler

5.2 Knowledge level of the modeler

We now enter the knowledge level (Figure 3) where the parametric statistical dependencies and distribution functions ascertained by the statistical analyzer are used to refine the L-system models of the neuron population. The *L-system synthesizer* is presently little more than a convenient name for the collection of ad hoc rules which effect the L-system modifications.

Next, paralleling the neuron morphology DBMS, we have at the knowledge level the *neuron morphology knowledge base*, which stores the L-system grammars and their associated distribution functions for stochastic generation of synthetic neurons.

And finally, closing the loop, we return to the *neuron generator*, described above.

With experience we anticipate that the system components at the knowledge level will merge into an expert-system-driven simulation system. The data level NMM provides the external environment for this knowledge-level subsystem. The long term goal, of course, is for the knowledge level subsystem to converge upon a stochastic L-system model such that the input from real manually-measured neurons and synthesized neurons can not be statistically distinguished.

6 SIGNIFICANCE

The applications of these stochastic L-system models are several. The L-system models bring coherence to existing neuron databases; they allow us to estimate the new information contributed by the additional neuron morphological measurements.

A paradigm shift to a new mode of analyzing neuron morphology is suggested. The developing bifurcations and multifurcations of growing dendritic and axonal arbors bear striking resemblance, in ultra-slow motion, to nuclear scattering events. We recognize this resemblance, and statistically describe these junction "events" by a variant of the S-matrix theory used in the formal theory of nuclear reactions. Conceptually, the block of embedded histological tissue replaces the bubble chamber of high energy physics. The methodology is now largely in place for the measurement and morphological analysis of large neuron populations.

7 REFERENCES

[1] McCormick B. H., Mulchandani K., "L-system Modeling of Neurons", Proceedings of the Visualization in Biomedical Computing, 2359:693-705,1994.

[2] Rozenberg G., Salomaa A. (Eds), "Lindenmayer Systems", Springer-Verlag, 1992.

[3] Prusinkiewicz P., Lindenmayer A., "The Algorithmic Beauty of Plants", Springer-Verlag, New York, 1990.

[4] Kreowski Hans-Jörg, "Parallel Hyperedge Replacement", "Lindenmayer Systems", Rozenberg G., Salomaa A.(Eds.), Springer-Verlag, Berlin, 1992.

[5] Lindenmayer A., Jürgensen, "Grammars of Development: Discrete-State Models for Growth, Differentiation, and Gene Expression in Modular Organisms", "Lindenmayer Systems", Springer-Verlag, New York, 1992.

[6] Böers E., Kuiper H., "Biological Metaphors and the Design of Modular Artificial Neural Networks", Master's thesis, Department of Computer Science and Experimental and Theoretical Psychology at Leiden University, Netherlands, 1992.

[7] Lindenmayer A., "Mathematical Models for Cellular Interaction in Development, Parts I and II.", Journal of Theoretical Biology, 18:280-315, 1968.

[8] Chen X.Y.,Wolpaw J. R., "Triceps Surae Motoneuron Morphology in the Rat: A Quantitative Light Microscopic Study.", J. Comp. Neurol., 343:143-157, 1994.

[9] Ohara P.T., Havton L.A., "Dendritic Architecture of Rat Somatosensory Thalamocortical Projection Neurons", J. Comp. Neurol., 341:159-171, 1994.

[10] Farin G., "Curves and Surfaces for Computer Aided Geometric Design", Academic Press, San Diego, 1993.

[11] Rall, W., "Core Conductor Theory and Cable Properties of Neurons". In E.R. Kandel (ed): The Nervous System: Cell Biology of Neurons. Section 1: Handbook of Physiology, Part 1, Vol. 1. Bethesda: American Physiological Society, pp. 39-97.

[12] Blatt, J.M., Weisskopf, V.F.,"Theoretical Nuclear Physics", John Wiley & Sons, New York, 1952, Chapter X.

[13] Knupp, P., Steinberg, S.,"Fundamentals of Grid Generation", CRC Press, Boca Raton, 1994.